Ministero dell'Istruzione, dell'Università e della Ricerca Servizio Automazione Informatica e Innovazione Tecnologica

Modulo 8

Reti di reti

ForTIC

Piano Nazionale di Formazione degli Insegnanti sulle Tecnologie dell'Informazione e della Comunicazione

Percorso Formativo C

Materiali didattici a supporto delle attività formative

2002-2004

Promosso da:
- Ministero dell'Istruzione, dell'Università e della Ricerca, Servizio Automazione Informatica e Innovazione Tecnologica
- Ministero dell'Istruzione, dell'Università e della Ricerca, Ufficio Scolastico Regionale della Basilicata

Materiale a cura di:
- Università degli Studi di Bologna, Dipartimento di Scienze dell'Informazione
- Università degli Studi di Bologna, Dipartimento di Elettronica Informatica e Sistemistica

Editing:
- CRIAD - Centro di Ricerche e studi per l'Informatica Applicata alla Didattica

Progetto grafico:
- Campagna Pubblicitaria - Comunicazione creativa

In questa sezione verrà data una breve descrizione del modulo.

Gli scopi del modulo consistono nel mettere in grado di:

- Distinguere tra topologie WAN e MAN.
- Distinguere tra opzioni basate su *router*, *switch*e *bridge*.
- Conoscere i passi necessari per connettere una rete ad Internet.
- Distinguere le differenze tra una connessione *dial-up* e una connessione dedicata.
- Definire le componenti *software* fondamentali di una WAN.
- Spiegare le funzioni e gli scopi di un *firewall*.
- Configurare liste di accesso per limitare il traffico ed aumentare la sicurezza.

Il modulo è strutturato nei seguenti argomenti:

- **Topologie WAN**
  - Descrivere topologie WAN e topologie MAN.
  - Distinguere tra topologie WAN e topologie LAN.
- **Opzioni di interconnessione**
  - Distinguere tra opzioni basate su *router*, su *switch* e su *bridge*.
  - Spiegare i passi necessari per connettere una rete ad Internet.
  - Spiegare le differenze tra una connessione *dial-up* e una connessione dedicata.
- ***Software* di interconnessione**
  - Definire le componenti *software* fondamentali di una WAN.
  - Spiegare le funzioni e gli scopi di un *firewall*.
  - Configurare liste di accesso per limitare il traffico ed aumentare la sicurezza.
- **Sicurezza**
  - Spiegare i principali aspetti della sicurezza connessi alla trasmissione dei dati.
  - Descrivere gli attuali *standard* di crittografazione: chiavi pubbliche e private, NSA, DES, PGP.
  - Descrivere le funzioni e le caratteristiche di un *firewall*.

Introduzione

Topologie WAN

Franco Callegati
Paolo Zaffoni

Reti di reti

Nel modulo 5 si è discusso delle topologie di rete e si è affermato che una rete di telecomunicazioni può essere rappresentata con un grafo, ossia una struttura logica, composta da nodi e da archi, dove i nodi rappresentano gli elementi che svolgono funzioni di commutazione ed i rami gli elementi (collegamenti) che svolgono funzioni di trasmissione. La struttura del grafo è anche topologia della rete.

Sempre nel modulo 5 si sono discusse varie alternative per le topologie di rete evidenziando che, quando la rete assume una certa complessità possono risultare convenienti topologie di tipo gerarchico.

Inoltre si è detto che le reti vengono tradizionalmente classificate in base alla distanza:

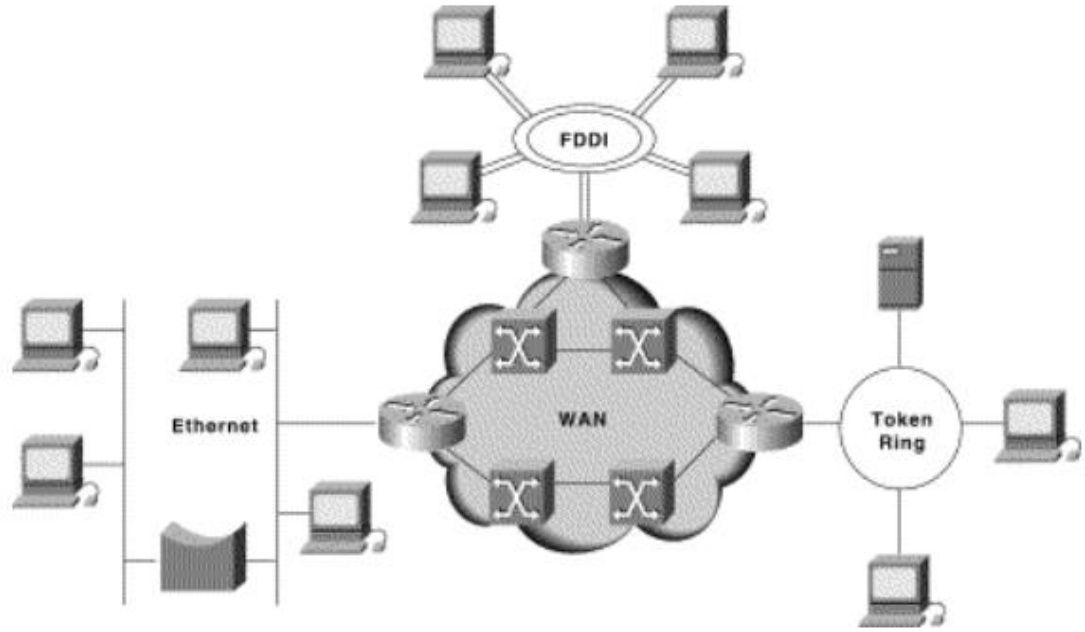
- **LAN** - *Local Area Network* o **reti locali**: tipicamente sono reti private per l'interconnessione di *computer* ed altri apparati appartenenti ad un unico ente o azienda.
- **MAN** - *Metropolitan Area Network* o **reti metropolitane**: possono essere reti private o pubbliche e fornire servizi di vario tipo in ambito urbano, dall'interconnessione di *computer*, alla telefonia, alla TV via cavo.
- **WAN** - *Wide Area Network* o **reti geografiche**: in passato erano le reti dei grandi gestori tipicamente pubblici che fornivano servizi e connettività a livello nazionale; oggi, dopo la *deregulation*, possono anche appartenere a privati ed offrire connettività a livello mondiale.

Le reti WAN, a causa della complessità e dimensione che le caratterizza, sono solitamente realizzate utilizzando una topologia di tipo gerarchico che permetta di distinguere diversi livelli di rete e, come tali, risultano dall'interconnessione di reti di minori dimensioni, che possono utilizzare anche tecnologie di tipo diverso.

Ci si trova quindi di fronte ad una tipologia di reti diverse da quelle studiate finora, fatto di reti fra loro interconnesse che danno origine ad un panorama di reti di reti

Solitamente una rete di reti si dice *internetwork*, facendo riferimento ad un insieme di reti individuali, collegate tra loro attraverso dispositivi di rete, che agisce come una singola grande rete. *Internetworking* si riferisce all'industria, ai prodotti, e alle procedure che concorrono al raggiungimento dello scopo della creazione e dell'amministrazione di una rete.

La figura illustra un esempio di realizzazione di una *internetwork* tramite collegamento di alcune reti realizzate con varie tecnologie.



Esempio di interwork

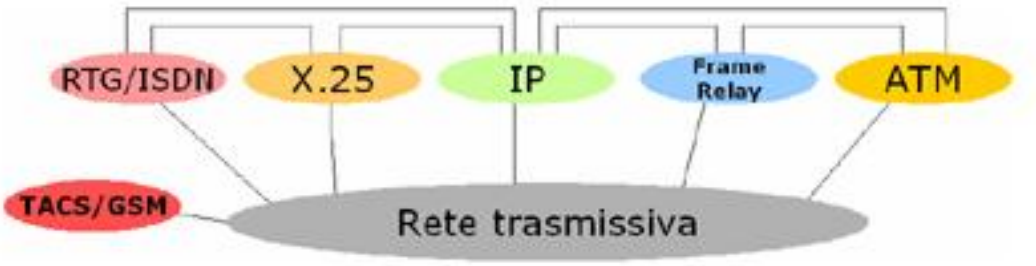
Reti geografiche e reti pubbliche

Le reti WAN sono quindi caratterizzate dal fatto che si estendono su di un ampia area geografica che solitamente comprende l'attraversamento del suolo pubblico. In generale, sia per i notevoli costi di realizzazione, sia per la complessità di gestione, una rete WAN appartiene ad un gestore che è un soggetto avente come compito primario quello di realizzare e mantenere la rete e venderne i servizi a terzi, realizzandone un profitto.

Fino a qualche anno fa, per ragioni di tipo strategico, la normativa prevedeva l'esistenza di un solo gestore nazionale di reti di questo tipo, parzialmente sotto il controllo statale, che operava in regime di monopolio. La deregolamentazione del settore delle telecomunicazioni, avvenuta in Italia negli anni '90, ha posto fine al monopolio ed oggi esiste una pluralità di soggetti possono svolgere la funzione di gestore realizzando il mercato multi-gestore che conosciamo.

Le reti WAN sono anche dette reti geografiche e, quando appartenenti ad un gestore pubblico, reti pubbliche.

Le interazioni fra i terminali (telefoni, calcolatori, eccetera) delle reti d'utente (centralino telefonico locale, rete LAN, eccetera) in una grande organizzazione avviene su distanze locali, regionali, nazionali ed a volte anche internazionali. Per realizzare questa interconnessione si ricorre ai servizi di trasmissione forniti dalle infrastrutture pubbliche.



Infrastruttura di rete pubblica

Un'infrastruttura pubblica è quindi preposta a fornire una qualche forma di connettività (più o meno trasparente) fra reti private. Tradizionalmente le reti geografiche di telecomunicazioni sono state progettate, realizzate ed esercitate nell'ottica di reti dedicate. Una rete dedicata è un'infrastruttura pensata per la fornitura di una ristretta classe di servizi di telecomunicazioni e quindi adatta a determinate fasce di applicazioni di utente.

Reti WAN per la trasmissione dati

In una rete WAN per la trasmissione dati, gli elementi di commutazione sono elaboratori specializzati utilizzati per connettere fra loro due o più linee di trasmissione. Gli elementi di commutazione sono tipicamente identificati da dispositivi denominati *router*. Una tipica WAN è utilizzata per connettere più LAN fra loro.

In generale una WAN contiene numerose linee, spesso telefoniche, che congiungono coppie di *router*. I compiti dei *router* sono:

- ricezione dei pacchetti dalle linee di ingresso;
- memorizzazione dei pacchetti in un *buffer* interno;
- instradamento dei pacchetti sulle linee se queste non sono già occupate da altre trasmissioni di pacchetti.

Per connettere i *router* tra di loro vengono utilizzate delle linee di trasmissione. Queste differiscono per tipologia dei circuiti, che possono essere analogici (obsoleti) o digitali, per modalità trasmissiva, plesiocrona o sincrona, infine per modalità di commutazione, ossia di circuito, di pacchetto, di trama o di cella. In base a queste qualità, le tecnologie per la trasmissione dati vengono suddivise in queste categorie:

- tecnologie trasmissive a collegamento diretto; i collegamenti diretti possono essere sia analogici (CDA) che numerici (CDN), in questa categoria rientra il PDH (*Plesiochronous Digital Hierarchy*) e SDH (*Synchronous Digital Hierarchy*);
- tecnologie commutate a circuito; sono tecnologie commutate a circuito sia la rete telefonica nazionale (RTN), analogica, che ISDN, digitale;
- tecnologie di strato data link; fanno parte di questa categoria sia l'ATM (*asynchronous transfer mode*) che il *Frame relay*;
- tecnologia a commutazione di etichetta o MPLS (*Multi Protocol Label Switching*); una nuova tecnologia sviluppata per migliorare le prestazioni

della funzione di instradamento dei *router* nonché permettere la gestione qualità di servizio.

Plesiochronous Digital Hierarchy (PDH)

Plesiocrono vuol dire che gli orologi (*clock*) degli apparati di una stessa rete lavorano a frequenze simili ma non identiche. Più canali numerici, detti tributari possono essere raggruppati mediante tecniche TDM (multiplazione temporale) per formare canali più veloci.

La rete telefonica tradizionale utilizza una rete di trasporto di tipo PDH con i canali tributari multiplati secondo una ben precisa gerarchia. La prima trama della gerarchia plesiocrona PDH è negli USA la trama T1, mentre in Europa è la trama E1.
La trama T1 permette l'invio di 24 canali tributari a 64 Kbps su un canale multiplo a 1.544 Mbps, di cui 1.536 Mbps sono utilizzati per la trasmissione dei dati e 8 Kbps per le informazioni di sincronismo. I canali tributari della trama T1 possono essere utilizzati a 56 Kbps (soluzione idonea nel caso di canali telefonici digitali) lasciando libero l'ottavo bit che può essere dedicato a funzioni di segnalazione, oppure a 64 Kbps (soluzione migliore nel caso di trasmissione dati) dedicando un intero canale tributario alle funzioni di segnalazione.
La trama E1 prevede invece la trasmissione di 32 canali a 64 Kbps, di cui uno riservato al sincronismo e uno alle informazioni di controllo.

La trasmissione dati su flussi della gerarchia plesiocrona può avvenire in due modalità: non strutturata o strutturata. Nel collegamento non strutturato, l'apparecchiatura di interfaccia, fornisce un flusso a 2.048 Mbps (1.544 Mbps negli USA) senza imporre alcuna struttura di trama. L'unico vincolo è la necessità di sincronizzarsi al *clock* di trasmissione fornito dall'interfaccia.

Nel collegamento strutturato, invece, è necessaria la conformità alla struttura di trama E1 (T1). È anche possibile non utilizzare tutti i 30 (23) canali tributari disponibili, ottenendo canali a velocità pari a un multiplo di 64 Kbps, ed in tal caso si parla di collegamento strutturato partizionato. L'apparecchiatura di interfaccia è programmabile per estrarre dalla trama E1 (T1) i canali destinati all'utente.

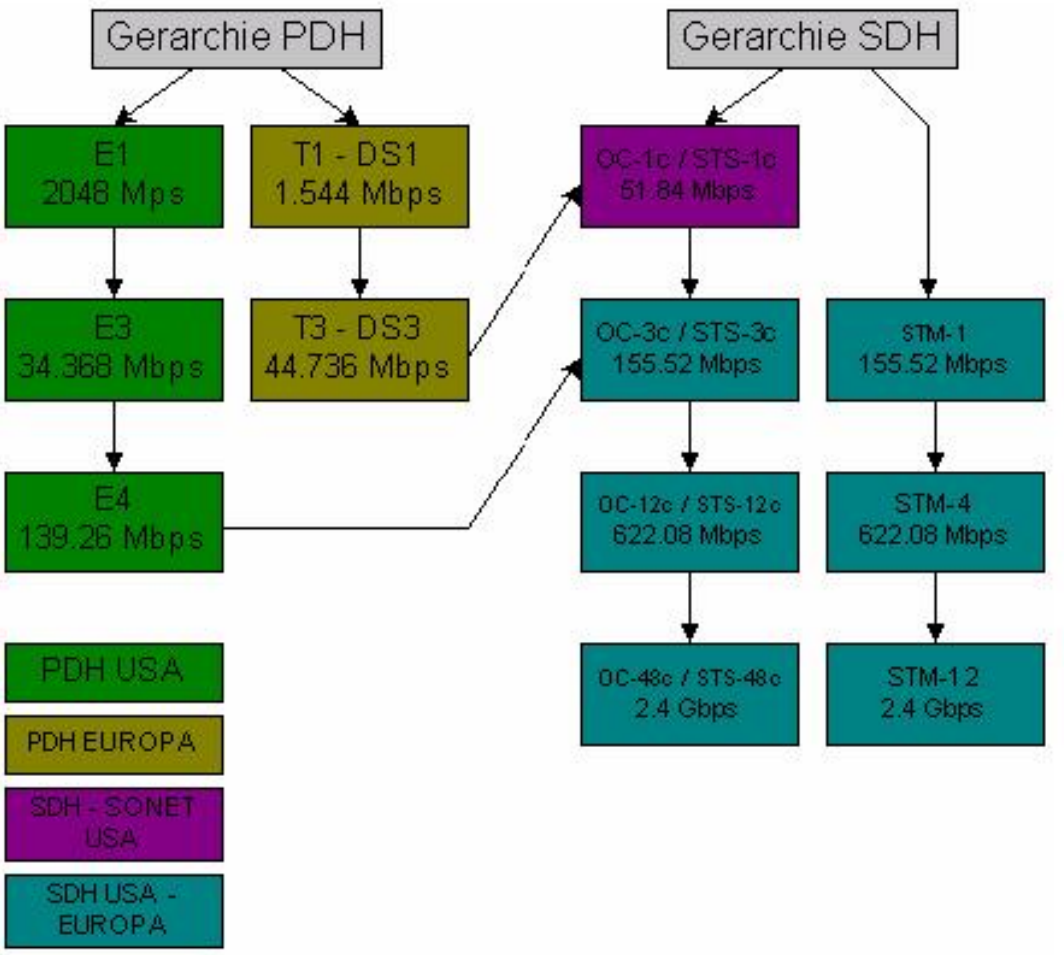
I limiti principali delle gerarchie PDH possono essere così riassunti:

- mancata unificazione a livello mondiale: esistono tre gerarchie (europea, nord-americana e giapponese) tra loro incompatibili;
- la relazione di fase tra i diversi canali tributari in una trama multipla è casuale e variabile nel tempo; questo richiede complesse operazioni di riallineamento (*pulse stuffing*) ed impedisce di inserire od estrarre facilmente un canale tributario se non ricorrendo ad un'operazione di demultiplazione completa della trama multipla seguita da una multiplazione dei nuovi tributari;
- formati di trame diversi per i diversi ordini gerarchici; non esiste un modo standard per ottenere a partire da una trama quella di ordine superiore;
- il formato dei dati non prevede esplicitamente informazioni per il controllo e la gestione della rete.

Synchronous Digital Hierarchy (SDH)

Per superare i limiti presenti in PDH è stata introdotta la gerarchia SDH che è unificata a livello mondiale anche se negli USA si chiama SONET e ha anche una velocità di 51.84 Mbps non presente altrove. In questo caso il synchronous significa che gli apparati di una stessa rete utilizzano un unico *clock* di riferimento. Le rete SDH prevede anche la possibilità di trasportare trame di diversi formati, fra cui il PDH, all'interno delle sue trame.
I vantaggi principali del SDH sono:

- l'utilizzo di una multiplazione sincrona che permette di inserire flussi a bassa velocità in flussi ad elevata velocità senza dover effettuare una demultiplazione e una multiplazione completa; analogamente è possibile l'estrazione diretta di un flusso a bassa velocità da un flusso ad alta;
- una topologia di rete ad anello che offre una maggior resistenza ai guasti;
- l'integrazione di vari canali ausiliari nelle trame che permettono un controllo continuo del tasso di errore e contengono le informazioni per le procedure di gestione, amministrazione, manutenzione e configurazione, che sono a loro volta standardizzate.



Gerarchie SDH e PDH a confronto

Vale la pena notare che entrambi gli standard, ed in particolare SDH, sono

caratterizzati da un ambiente multivendor.

Opzioni di interconnessione

Franco Callegati
Paolo Zaffoni

Apparati per Internetworking

I requisiti di velocità e di affidabilità che si hanno nelle LAN possono venire a mancare quando il numero degli elaboratori o la dimensione della LAN stessa inizia ad essere troppo grande. Diventa a questo punto indispensabile suddividere la LAN in più parti e interconnetterla con i dispositivi appositamente progettati.

Qualora le LAN da interconnettere sono tutte localizzate nella medesima area (azienda, campus, eccetera), amministrata da un medesimo soggetto (il possessore delle LAN), la loro interconnessione può essere realizzata semplicemente interponendo fra loro apparati per l'interconnessione. Esistono varie alternative al riguardo che verranno discusse nel seguito.

Quando invece le LAN o i calcolatori si trovano distribuiti su vaste aree geografiche è necessario ricorrere per l'interconnessione all'uso delle infrastrutture e dei servizi offerti dalle reti pubbliche. Anche in questo caso esistono numerose alternative dovute alla coesistenza di diverse tecnologie, frutto dello sviluppo tecnologico, così come delle diverse scelte tecnologiche effettuate dai gestori.

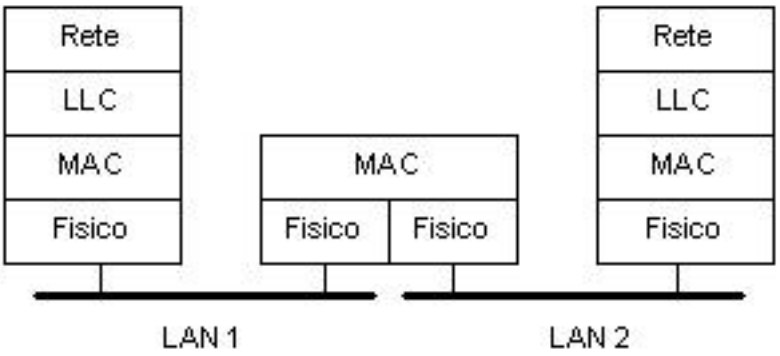
Bridge

La coesistenza di tecnologie diverse, le prestazione limitate in caso sia di molti utenti sia di elevato traffico, la ridotta estensione geografica specialmente nel caso di LAN ad alte velocità, ha dato origine a degli apparati per l'interconnessione di LAN a livello MAC, che prendono il nome di *bridge*. L'interconnessione di LAN tramite *bridge* non ha quindi il solo obiettivo di far comunicare elaboratori posti su LAN differenti ma anche quello di permettere la creazione di LAN estese composte da più LAN realizzate con la stessa tecnologia e fra loro interconnesse. Inizialmente i *bridge* si limitavano a interconnettere due LAN, successivamente l'evoluzione della topologia da *bus* a stella ha favorito la nascita di *bridge* multiporta come centro stella, che diventano dei veri e propri commutatori (*switch*).

I *bridge* operano al sottolivello MAC del livello *data link* e uniscono segmenti di LAN. Le LAN da unire possono essere omogenee, ossia hanno lo stesso MAC, o eterogenee, MAC differenti, ad esempio *token ring* e *ethernet*. Gli algoritmi di instradamento che utilizzano sono molto semplici, e vengono usati principalmente in ambito locale. L'utilizzo di *bridge* porta ad un aumento della banda complessiva, grazie alla segmentazione della LAN, inoltre aumenta anche la portata geografica della LAN stessa.
Il *bridge* ha quattro funzioni principali:

- la ricezione dei pacchetti;
- l'eventuale traduzione da un formato di sottolivello MAC ad un altro;
- il filtraggio dei pacchetti sulla base dell'indirizzo tenendo conto sia della posizione della destinazione sia delle indicazione del gestore, che possono anche riguardare l'indirizzo della sorgente ed il tipo di protocollo;

- la ritrasmissione dei pacchetti.



Schema dell'architettura di interconnessione tramite bridge

Esistono due tipi di *bridge* che si differenziano all'atto di inoltro dei pacchetti: *transparent bridge* (standard IEEE 802.1D) e *source routing bridge* (deriva dal *token ring*).

Transparent bridge

Lo standard IEEE 802.1D deriva architetturalmente dai primi *bridge ethernet*. Il *transparent bridge* gestisce direttamente le tabelle di instradamento, che risiedono in una sua memoria locale. Questi *bridge* sono del tutto invisibili alle stazioni appartenenti alle LAN interconnesse, e non necessitano quindi di alcuna riconfigurazione quando la rete modifica la sua topologia.
Un *transparent bridge* opera in questo modo:

- monitora tutto il traffico;
- verifica gli indirizzi di origine e destinazione di ciascun pacchetto;
- costruisce una tabella di instradamento man mano che le informazioni sono disponibili.

Se la destinazione del pacchetto non è elencata nella tabella di instradamento, il *bridge* inoltra i pacchetti a tutti i segmenti. Se la destinazione è elencata nella tabella di instradamento il *bridge* inoltra i pacchetti a quel segmento (a meno che non si tratti del segmento stesso di origine).

Source routing bridge

Deriva dai *bridge* della rete *token ring*. Non presenta tabelle di *routing* locali come nel caso del *transparent bridge*, bensì richiede che siano le stazioni a mantenere le tabelle di *routing* e scrivano nel pacchetto la sequenza delle LAN da attraversare. Quando una stazione deve imparare l'instradamento verso un'altra stazione invia un pacchetto di *route location*.

Translating Bridge

Translating bridge vengono utilizzati quando si devono interconnettere due segmenti di LAN di tipo differente, ad esempio *token ring* e *ethernet*. Questi *bridge* hanno le due interfacce di tipo differente, inoltre hanno il compito sia di tradurre il formato della trama, sia di adattarsi al *data rate* delle due architetture.

Switch

Gli *switch* sono dei dispositivi di rete ad alte prestazioni con funzionalità di inoltro dei pacchetti realizzata al livello *hardware*. Questi dispositivi si sono evoluti dai *router* ad alte prestazioni. Il principio di funzionamento degli *switch* è stato applicato ai *router*, è possibile pertanto trovare degli *switch* che operano sia a livello MAC sia a livello di rete.

Se ogni porta dello *switch* ha una sola stazione connessa, fra le stazioni direttamente connesse allo *switch* non esiste più la condivisione del mezzo, le collisioni avvengo solo per ricezioni e trasmissioni contemporanee e lo *switch* si comporta come un commutatore tra stazione sorgente e stazione ricevente.
Le tecniche di attraversamento di uno *switch* sono:

- *store and forward* (immagazzinamento e rilancio); è quella utilizzata dai *bridge* (prevista da IEEE 802.1d), il pacchetto viene ricevuto interamente e poi ritrasmesso;
- *cut through* o *on the fly switching*; la decisione di inoltro viene presa durante il transito del pacchetto nello *switch*; i tempi di latenza sono molto bassi (40-60 microsecondi) perché quando lo *switch* legge l'indirizzo di destinazione decide la porta di uscita;
- *fragment free*; prima di iniziare a ritrasmettere il pacchetto si aspetta comunque un tempo pari alla *collision window* (51.2 millisecondi).

Le tecniche *cut through* e *fragment free* possono essere utilizzate solo se su tutte le porte è presente lo stesso tipo di MAC, se tutte le porte hanno la stessa velocità trasmissiva, se la porta di destinazione è libera infine se il pacchetto non è *broadcast* o *multicast*, altrimenti si ricorre allo *store and forward*. Per i pacchetti corti i tre metodi sono equivalenti, in più con velocità elevate (100 Mb/s o 1 Gb/s), la latenza dello *store and forward* è comunque molto piccola. C'è da notare che il *cut through* inoltra anche i frammenti di collisione.

Lo *switch* di tipo *store and forward* opera come se fosse un *bridge* multiporta ad alte prestazioni. Può interconnettere MAC diversi e può operare a velocità diverse, non inoltra pacchetti contenenti errori poiché controlla il CRC infine non inoltra i frammenti di collisione.

Router

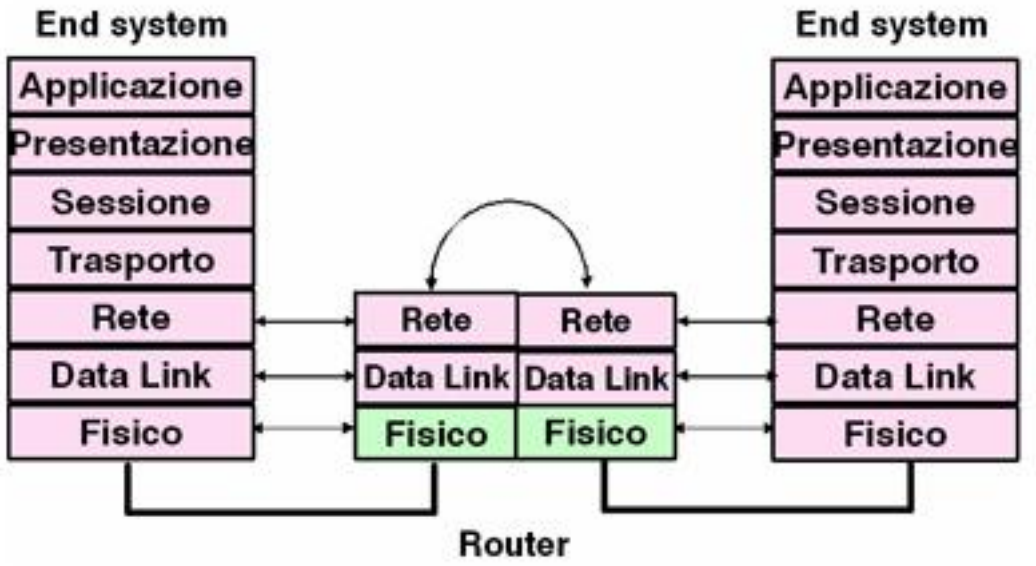
I *router* sono i dispositivi di interconenssione di LAN che lavorano a livello di rete. Sono adeguati a gestire topologie anche molto complesse. Non propagano incondizionatamente traffico *broadcast* o *multicast*. Permettono un *routing* gerarchico suddividendo le reti in aree.

Le differenze tra *router* e *bridge* stanno nei seguenti punti:

- indirizzamento; i *router* sono indirizzati esplicitamente, i *bridge* sono trasparenti ai nodi;
- calcolo instradamento; i *router* ricevono ed usano molte informazioni, mentre i *bridge* usano solo gli indirizzi di mittente e destinatario di livello *data link*;
- pacchetti/*frame*; i *router* operano su pacchetti di livello di rete e possono

dividere o riunire i messaggi per adattarli a reti con lunghezze massime diverse. I *bridge* non possono modificare i campi dati;
- *feedback*; i *router* possono fornire informazioni sullo stato della rete all'utente finale;
- *forwarding*; i *router* ritrasmettono i messaggi cambiando gli indirizzi di livello 2;
- *priority*; i *router* possono utilizzare priorità;
- *security*; i *router* possono realizzare tecniche di *firewall*, ossia di protezione.

Un *router* è un apparato per commutazione di pacchetto in grado di operare interconnessione di reti locali e geografiche che adottano diversi standard di comunicazione (*router* multiprotocollo). In particolare i *router* IP realizzano l'instradamento di pacchetti IP fra sottoreti di natura diversa ed implementano algoritmi di *routing* (statico o dinamico) necessari a realizzare comunicazioni *any_IP_address-to-any_IP_address*.
Un *router* con funzioni di interconnessioni fra due LAN riceve un pacchetto dalla sua scheda LAN o WAN. La scheda verifica se il pacchetto sia destinato al *router* (condizione vera nei collegamenti punto-punto, ma da verificare con gli indirizzi MAC nelle LAN) e in caso affermativo lo passa al processo *software* di *forwarding*.
Il processo di *forwarding*, consultando la tabella di instradamento determina la linea fisica di uscita sulla quale dovrà proseguire il pacchetto.



Schema dell'architettura di interconnessione tramite bridge

Molto spesso nei *router* le funzionalità non comprendono soltanto l'instradamento dei pacchetti di utente, ma anche l'implementazione di uno o più protocolli ausiliari legati al problema del *neighbour greetings* (legato alla conoscenza che un *router* dovrebbe avere sullo stato delle macchine collegate alla stessa LAN).

L'accesso alle reti pubbliche

Il modo più immediato per ottenere un servizio di trasmissione dati in ambito geografico è quello di sfruttare un comune canale telefonico tramite un dispositivo denominato

modem che consente di convertire l'informazione in forma digitale trasmessa da un *computer* o da un terminale in un formato tale da poterla inviare attraverso un canale telefonico progettato per la trasmissione della voce. Ovviamente in ricezione un dispositivo analogo effettuerà la conversione inversa.

La trasmissione dati attraverso linee telefoniche, detto anche accesso *dial-up*, è ancora oggi ampiamente utilizzata, in virtù della sua flessibilità e di continui sviluppi tecnologici degli apparati, quali appunto i modem. Le linee telefoniche sono linee commutate che, grazie alla enorme diffusione della rete telefonica, permettono di raggiungere qualsiasi altro utente raggiungibile con il telefono.

In realtà, non sempre una connessione *dial-up* offre un servizio di comunicazione soddisfacente. Esistono quindi numerose alternative frutto dello sviluppo delle tecnologie di trasmissione di commutazione. Nel seguito verranno descritte alcune delle soluzioni più significative.

Accesso Dial-Up

La connessione *dial-up* è realizzata tramite una normale chiamata telefonica. Il costo corrisponde a quello di una normale telefonata a voce. Tale tipo di connessione è anche indicato con RTC (Rete telefonica Commutata) o come connessione su linea commutata. Di fatto, il numero di telefono che si compone consente di connettersi a qualunque calcolatore connesso anch'esso alla rete telefonica. Generalmente si utilizza l'accesso *dial-up* per collegarsi al *server* di un *Internet service provider* che vende il servizio di accesso ad Internet.

Ovviamente sul terminale remoto da cui si vuole effettuare l'accesso si dovrà specificare, oltre al numero telefonico suddetto, anche il protocollo di comunicazione di livello due che verrà utilizzato durante la connessione. Tipicamente le scelte possibili sono fra il protocollo SLIP (*Serial Link Internet Protocol*), ormai di vecchia concezione e fra il più recente PPP (*Point to Point Protocol*).

Sarà necessario poi specificare gli indirizzi IP e del *server* DNS che identificano il *service provider* attraverso il quale è possibile realizzare la connessione. Tipicamente per le connessioni *dial-up* effettuate al fine di connettersi alla rete Internet tali informazioni d'indirizzo sono fornite automaticamente, in fase di connessione, dal *service provider*.

Le ultime informazioni da specificare per completare l'interconnessione riguardano la procedura di autenticazione nella quale è necessario precisare il nome d'utente e la relativa *password* registrati dal *service provider* e che permettono di accedere a tutti i servizi da forniti sulla base del contratto stipulato fra utente e fornitore stesso.

I servizi di *dial-up* offrono un metodo *cost-effective* per la connettività attraverso le WAN. Due metodi abbastanza popolari di implementazioni *dial-up* sono:

- *Dial-on-Demand Routing* (DDR);
- *dial backup*.

La tecnica DDR fa sì che un *router* possa dinamicamente iniziare e chiudere una sessione di circuito commutato alla richiesta di trasmissione di una stazione. Un *router* viene configurato stimando un determinato traffico interessante (come ad esempio quello relativo ad un particolare protocollo) ed altro determinato traffico non

interessante.

Ogni qualvolta il *router* riceve traffico interessante destinato ad una rete remota, verrà stabilito un circuito ed il traffico sarà trasmesso normalmente. Se il *router* riceve traffico non interessato ed il circuito è già stabilito, anche questo traffico sarà normalmente trasmesso. Il *router* gestisce un *idle timer* che verrà azzerato solo quando sarà ricevuto traffico interessato. Se il *router* non riceve traffico interessato prima che scada l'*idle timer*, il circuito sarà terminato. Se riceve solo traffico non interessato e non c'è un circuito già stabilito, il *router* scarterà questo traffico.

Solo al ricevimento di traffico interessato, il *router* inizia un nuovo circuito. Il sistema DDR può essere utilizzato per sostituire i collegamenti *point-to-point* ed i servizio *switched multiaccess* WAN.

Il *dial backup* è un servizio che attiva una linea seriale di *backup* in certe specifiche condizioni. La linea seriale secondaria può agire da collegamento di *backup* utilizzato quando il collegamento primario fallisce o come fonte di larghezza di banda aggiuntiva quando il carico sul collegamento principale risulta eccessivo. Il *dial backup* fornisce protezione contro la degradazione delle performance WAN.

Public switched telephone network (PSTN) - Rete telefonica pubblica

La rete telefonica PSTN nasce con l'obiettivo di trasferire informazioni di natura vocale. Viene utilizzato un canale con banda passante inferiore ai 4 KHz, in particolare si usa la banda che va dai 400 Hz ai 3400Hz.

Per trasferire dati sulla linea PSTN si utilizzano dei dispositivi che operano una modulazione, chiamati modem (modulatori-demodulatori). Per permettere una comunicazione *full duplex* si è suddivisa la banda passante in due parti. In fase di trasmissione il modem converte il flusso numerico proveniente dall'elaboratore in un segnale analogico adatto per transitare sulle linee PSTN e lo trasmette, mentre in fase di ricezione campiona il segnale ricevuto e ricostruisce il flusso informativo, passandolo all'elaboratore. Nel corso degli anni si sono imposti alcuni standard per la trasmissione via, e sono:

- V.21, trasmette 300 bps in modalità *full duplex*;
- V.22, trasmette 1200 bps in modalità *full duplex*;
- V.22 bis, come la V.22 ma trasmette 2400 bps in modalità *full duplex*;
- V.23, trasmette 1200 bps in modalità *half duplex* con canale di segnalazione a 75 bps, questo standard è stato utilizzato principalmente per il Videotel;
- V.29, trasmette a 9600 bps in modalità *full duplex*, viene utilizzato per i fax;
- V.32, trasmette a 9600 bps in modalità *full duplex*;
- V.32 bis, trasmette sia a 14400 bps sia a 12000 in modalità *full duplex*;
- V.34, trasmette a 28800 bps in modalità *full duplex*, utilizza però 9 bit per carattere invece degli 8 utilizzati normalmente, perciò trasferisce 3200 caratteri al secondo;
- V.34+ trasmette a 33600 bps in modalità *full duplex*;
- V.90, opera fino a 56000 bps in trasmissione e 32000 bps in ricezione, però

richiede che i canali fino alle centrali telefoniche siano completamente digitali (in pratica solo dalla centrale all'utente è analogico);
- V.42, standard per la correzione degli errori;
- V.42 bis, standard per la compressione dei dati da trasmettere e ricevere.

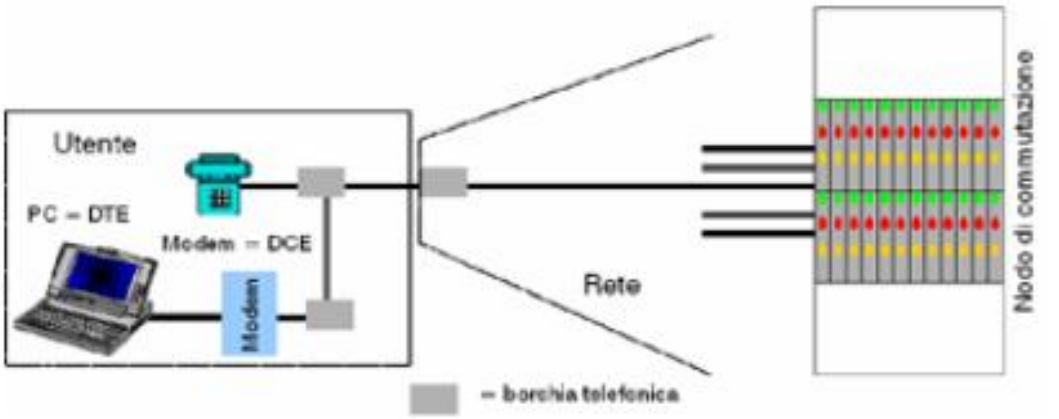
I modem utilizzano per il controllo il set di comandi *Hayes*, che è di tipo unificato, ossia vale per tutti i produttori di modem e per qualsiasi standard implementato dal modem stesso.

L'accesso alla rete telefonica

L'accesso alla rete telefonica è il più diffuso tipo di accesso, dal momento che il servizio telefonico è stato il primo ad essere ingegnerizzato (1892: automatizzazione della commutazione telefonica negli USA). Ad oggi, l'utente telefonico viene connesso con il nodo di rete pubblica attraverso una coppia in rame dalla borchia di impianto d'abbonato fino al nodo di centrale, in maniera dedicata.
L'accesso telefonico tradizionale avviene con:

- autocommutatore analogico (superato);
- autocommutatore numerico (attuale).

e impiega una coppia in rame non schermata (doppino telefonico non schermato, UTP, *Unshielded Twisted Pair*), consentendo la trasmissione *full duplex* di segnali analogici telefonici e criteri di controllo della chiamata.
La condivisione del doppino con tecniche *duplex* è ormai superata e per nuove installazioni il gestore dispone di coppie sufficienti per collegare l'utenza individualmente. I vecchi impianti *duplex* per normativa devono continuare ad esistere almeno per ora, a meno che non ci sia una richiesta specifica dell'utente.



Schema d'accesso alla rete telefonica

In relazione alla distanza (non superiore a 3 km) fra l'impianto di utente e il nodo telefonico può rendersi necessaria l'interposizione di apparati di rete, aventi funzioni trasmissive di raccolta, che riducono la distanza fra la sede di utente ed il punto di alimentazione della linea telefonica. Tali apparati (*Multiplex*) realizzano una remotizzazione di alcune funzioni del nodo di commutazione (le funzioni di attacco di utente) e vengono collocati in esterno, in cabinet o all'interno di grandi edifici residenziali e pubblici.

L'accesso commutato analogico

L'accesso commutato analogico è realizzato con un collegamento telefonico tradizionale alla Rete Telefonica Nazionale (RTN).
La linea di utente è in rame ed impiega tipicamente una coppia di conduttori da 4,5 o 6/10 mm di diametro.
La linea termina su una scheda dell'autocommutatore urbano, su cui si attestano tipicamente 8 o 16 linee utente.
Le funzioni svolte dalla scheda sono riassunte dall'acronimo BORSCHT:

- *Battery* (alimentazione microfonica da batteria di centrale);
- *Overvoltage* (protezione da sovratensioni estranee);
- *Ring* (invio della tensione di chiamata);
- *Supervision* (sensori di rilevazione di linea aperta/chiusa);
- *Coding* (conversione A/D e D/A del segnale fonico);
- *Hybrid* (passaggio da 2 a 4 fili e separazione delle vie foniche);
- *Testing* (funzioni di prova e verifica dei componenti di impianto dal posto operatore di rete).

Una delle funzioni che, per l'accesso telefonico analogico, caratterizza e limita la velocità trasmissiva è legata alla conversione analogico/numerica del segnale telefonico.
Il processo di conversione del segnale a qualità telefonica presuppone le operazioni di:

- filtraggio nella banda 0-4000 Hz;
- campionamento del segnale filtrato ad una frequenza pari a 2x4000 Hz;
- quantizzazione di ciascuno degli 8000 campioni/s ottenuti tramite una griglia di 256 quanti;
- codifica di tipo logaritmico (un numero maggiore di bit per i livelli piccoli, un numero minore di bit per i livelli grandi);

Il filtraggio limita la velocità di modulazione per un modem che sia chiamato a trasformare il segnale dati digitale in un segnale adatto ad essere trasmesso su un collegamento telefonico.

Tecnologie trasmissive a collegamento diretto

Il servizio offerto dalla rete telefonica è caratterizzato dal fatto che agli utenti viene messo a disposizione un collegamento fisico per tutta la durata del servizio. In tale periodo, il collegamento è indisponibile per tutti gli altri utenti della rete.
A seconda delle necessità, il collegamento fisico può essere:

- commutato: significa che esso viene richiesto dagli utenti a seguito di una chiamata e rimane a loro disposizione per l'intera durata della chiamata, come nell'accesso *dial-up* appena descritto;
- dedicato: significa che esso è predisposto in fase di configurazione degli utenti e permane a loro disposizione 24 ore su 24 (CDN, Canali Diretti Numerici).

I collegamenti dedicati possono essere distinti in due categorie:

- CDA - Collegamento Diretto Analogico;

- CDN - Collegamento Diretto Numerico.

Il CDA può essere definito come il servizio di interconnessione fornito dal gestore di una rete pubblica di telecomunicazioni. Consiste nel realizzare fra due sedi dell'utente una connettività analogica permanente adatta al trasferimento di un segnale caratterizzato da una dinamica in ampiezza e frequenza prestabilite.

Le velocità di trasferimento dati vanno da 2400bps a 64000 bps. Questa tecnologia è obsoleta e viene sostituita da soluzioni più recenti. Il CDA veniva utilizzato nei centri di calcolo che necessitavano di un collegamento dedicato, quando non erano soddisfacenti le linee commutate. Con l'introduzione delle centrali numeriche i collegamenti analogici tra le centrali stesse sono stati rimpiazzati da dorsali digitali ad alta velocità.

Il CDN è un collegamento fisico permanente *full duplex*, realizzato tra due sedi del cliente mediante apparati che rappresentano la terminazione del collegamento, installati in entrambe le sedi e mediante apparati della rete trasmissiva pubblica. Equivale ad una linea diretta dedicata tra le due sedi. Viene fornito con diverse opzioni di velocità: da 2400 bit/s fino a 2048 kbit/s (volgarmente detto 2Mbit/s), per passi discreti (2.4, 4.8, 9.6, 14.4, 19.2, 48 , 64, Nx64 kbit/s, con N variabile da 2 a 32). La rete trasmissiva che consente di fornire tale servizio è caratterizzata da bassi tassi di errore, alta disponibilità e possibilità di gestione da remoto; la trasmissione dati tra gli apparati disposti agli estremi del CDN è trasparente temporalmente (ritardo costante) e la rete è trasparente ai codici utilizzati.

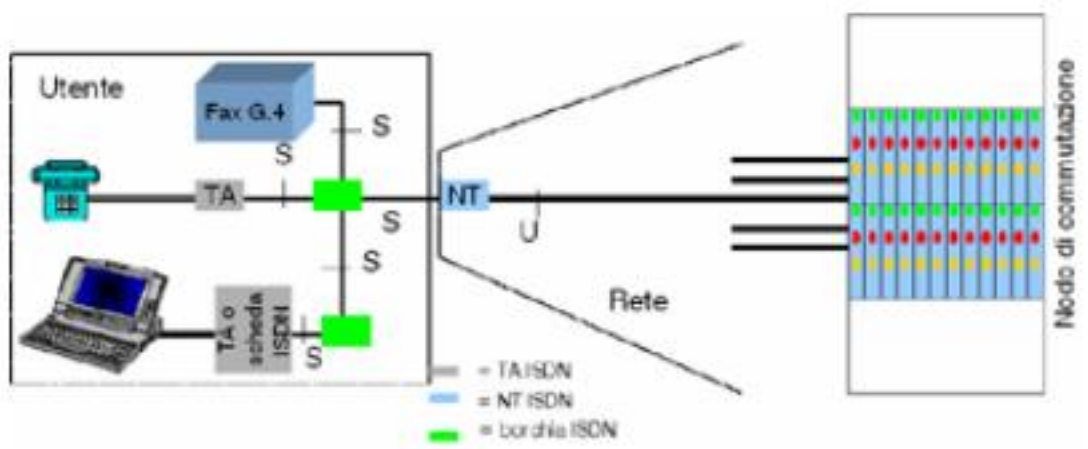
Prolungando un collegamento digitale dall'interno della centrale fino alla presa dell'utente, è stato possibile fornire un servizio completamente digitale, a velocità più elevata e minor tasso d'errore. Il CDN sfrutta la commutazione di pacchetto, pertanto con i CDN è possibile sfruttare tutto, o in parte, il flusso di dati delle dorsali inserendo all'interno delle trame utilizzate per i canali telefonici digitali il traffico degli utenti.

Integrated Services Digital Network - ISDN

ISDN rappresenta l'evoluzione delle reti commutate pubbliche analogiche. Basata sulla tecnologia digitale, offre l'integrazione di servizi di elevata qualità, quali telefonia digitale, trasmissione dati, telecontrolli e teleallarmi, fax G4, videotelefonia, attraverso un ridotto numero di interfacce standard.

La prestazione ISDN è stata sperimentata in Italia dalla fine degli anni '80 al 1994, anno in cui il servizio è stato diffuso commercialmente su scala nazionale. Ad oggi è disponibile in pratica in tutte le città italiane.

L'ISDN prevede una serie di servizi portanti (servizi di rete); uno dei più utilizzati attualmente è la connettività numerica.



Esempio di schema d'interconnessione ISDN

Tale servizio consente ad un terminale di utente (può essere un PC dotato di opportuna scheda ISDN o di adattatore esterno) di realizzare, mediante un'opportuna procedura di chiamata e con protocolli standard (realizzati in parte dall'*hardware* della scheda e in parte dal *software* del PC) connessioni numeriche. Tali connessioni equivalgono, durante la fase di trasmissione dati (a risposta avvenuta) ad un collegamento diretto numerico tra i due *computer*.

La linea di utente è costituita da un doppino identico a quello usato per l'accesso telefonico analogico, salvo il fatto che, sul nodo la scheda su cui termina la linea ed il *software* preposto al trattamento degli impegni hanno una maggiore complessità.

Trattandosi di uno standard internazionale per rete digitale commutata, è possibile collegarsi e usufruire di questi servizi con qualsiasi utente della rete. La rete ISDN prevede due tipi di accesso: l'accesso base BRA (*Basic Rate Access*), principalmente concepito per l'utente finale, e l'accesso primario PRA (*Primary Rate Access*), destinato a centri a loro volta erogatori di servizi, quale un centralino telefonico privato.

L'accesso base consiste in due canali a 64 Kbps (detti canali B) e in un canale dati di servizio a 16 Kbps (detto canale D).
L'accesso base prevede una velocità di trasmissione di 192 Kbps, di cui 144 utilizzati per i 2 canali B e il canale D, e i restanti 48 per informazioni di controllo e di sincronismo.

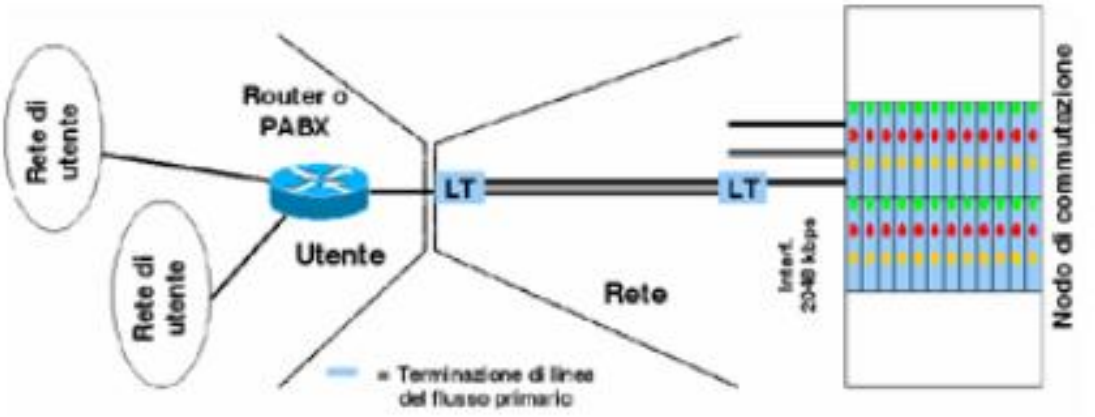
L'accesso numerico BRA (*Basic Rate Access*) ISDN su autocommutatore numerico, impiega una coppia in rame non schermata (doppino telefonico) e consente la trasmissione *full duplex* di segnali numerici (voce, video, fax, dati) e della segnalazione necessaria al controllo dei servizi commutati.
Inoltre consente di collegare al *bus* S di utente fino ad 8 terminali (ciascuno con il suo numero di rete pubblica).

Per il collegamento ad utenze particolari, è previsto un altro tipo di accesso, detto accesso primario. Si tratta di un accesso a 1.544 Mbps negli Stati Uniti (23 canali B più un canale D) e a 2 Mbps in Europa (30 canali B più un canale D).
Oltre ai canali di tipo B e D, esistono anche canali di tipo H, formati dall'aggregazione di più canali B:

- H0, 384 Kbps formato da 6 canali B;
- H11, 1536 Kbps formato da 24 canali B;
- H12, 1920 Kbps formato da 30 canali B.

L'accesso numerico PRA (*Primary Rate Access*) ISDN su autocommutatore numerico, impiega due coppie simmetriche in rame, oppure due coassiali. Consente la trasmissione *full duplex* di segnali numerici (voce, video, fax, dati) e della segnalazione necessaria al controllo dei servizi commutati relativamente ai canali numerici disponibili tra sede di utente e centrale.

I canali sono numerati da 0 a 31; quelli impiegati per l'accesso ISDN sono compresi da 1 a 31, essendo il canale 0 riservato per le funzioni di allineamento di trama e di sincronizzazione fra gli apparati trasmissivi attestati agli estremi della linea fisica (lato utente e lato centrale).
Il canale 16 viene utilizzato per trasportare la segnalazione relativamente a tutti i canali assegnati (fino a 30 canali).

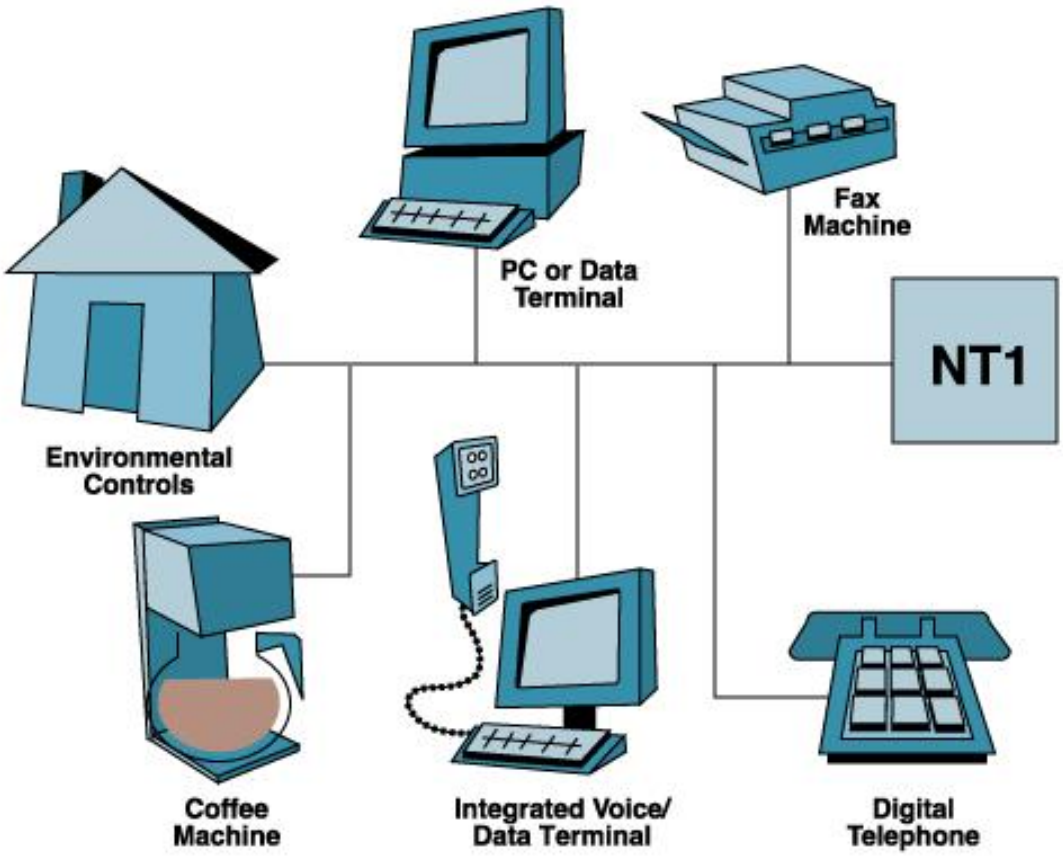


Accesso primario ISDN

L'accesso primario generalmente termina su un centralino privato di commutazione che offre servizi telefonici fra i derivati interni e consente a tali derivati di effettuare chiamate verso l'esterno (e accettare chiamate provenienti dall'esterno) utilizzando i canali B dell'accesso primario; l'accesso primario può terminare su un apparato dati (esempio: *router* con interfaccia primaria per un *Internet Service Provider*).

Architettura ISDN

Si può affermare che la rete ISDN (*Integrated Services Digital Network*) è una rete che, evolvendo dalla rete telefonica di tipo numerico IDN (*Integrated Digital Network*), fornisce connettività numerica da estremo ad estremo per supportare un insieme ampio di servizi applicativi, includendo servizi di fonia, dati, videocomunicazione, fax, eccetera, e alla quale gli utenti hanno accesso da un limitato numero di interfacce standardizzate (definizione tratta dalla Raccomandazione ITU-T I.110, 1998).



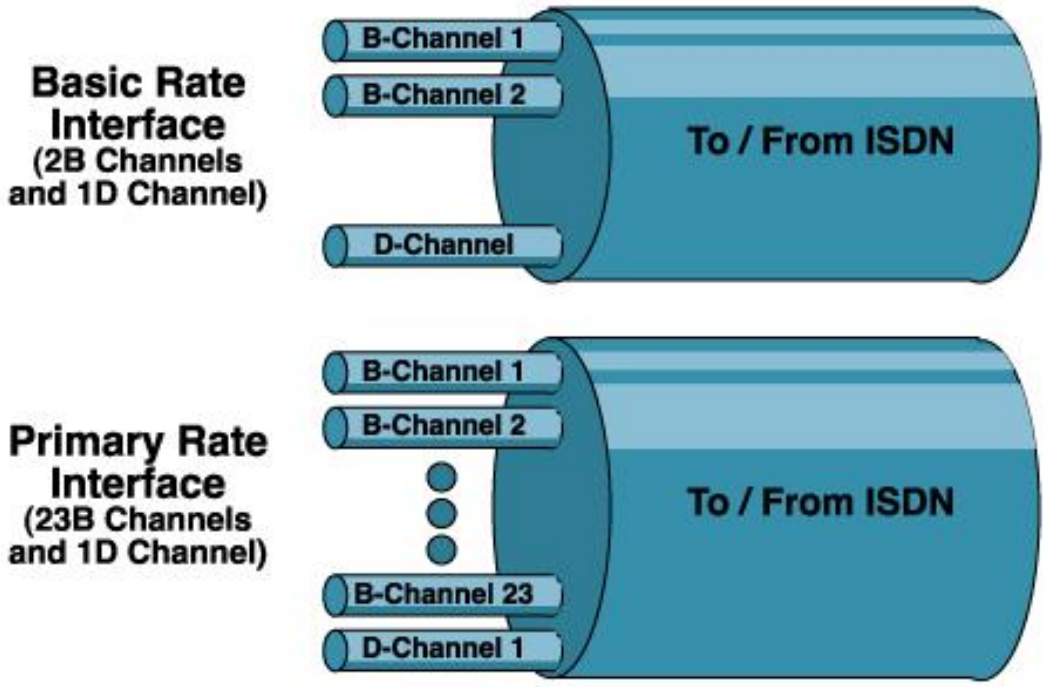
Esempio di connettività di tipo ISDN

Le raccomandazioni che riguardano la rete ISDN specificano l'interfaccia dell'utente verso la rete, e non descrivono come deve essere fatta la rete. **L'interfaccia è unica e indipendente dal tipo di servizio che l'utente utilizza**.
La rete ISDN permette in generale ad un utente di accedere in modo integrato ad una molteplicità di servizi offerti da **infrastrutture di rete distinte**; si ha solo perciò un'integrazione dell'accesso a servizi offerti da varie reti, consentendo un'ottimizzazione della rete di accesso.

Interfacce d'accesso ISDN

Le interfacce utente-rete ISDN definite dall'ITU-T possono essere di due tipi:

- Interfaccia o Accesso Base, rispettivamente *Basic Rate Interface* (BRI) e *Basic Rate Access* (BRA).
- Interfaccia o Accesso Primario, rispettivamente *Primary Rate Interface* (PRI) e *Primary Rate Access* (PRA).



Interfacce d'utente ISDN

Una delle principali differenze tra le due interfacce riguarda il numero di canali numerici disponibili all'utente: rispettivamente 2 canali B a 64 kbit/s e 1 canale D a 16 kbit/s nel caso dell'interfaccia base e 30 canali B (23 in Nord America e in Giappone) più un canale D. Tutti i canali dell'accesso primario sono caratterizzati da una capacità trasmissiva di 64 kbit/s, anche il canale D.

Configurazioni per l'accesso base

L'accesso base è indicato per utilizzatori che necessitano di uno o più terminali ma non utilizzano grosse strutture di multiplazione proprie. Scopo di tale accesso è di fornire i vantaggi dell'ISDN senza richiedere investimenti eccessivi, per questo nella sua installazione si cerca di avere la compatibilità con le installazioni non ISDN preesistenti e quindi l'utilizzo del doppino telefonico. Si possono avere diverse configurazioni:

- La configurazione punto-punto prevede un solo terminale d'utente connesso all'interfaccia; si raggiungono distanze dell'ordine del Km e comunque fino ad una attenuazione massima di linea tra NT1 e LT di 6 dB a 96 kHz. Per il corretto funzionamento dell'interfaccia è necessario definire il ritardo massimo di propagazione di andata e ritorno del segnale da TE a NT1 e viceversa che deve essere compreso tra 10 e 42 microsecondi.
- Nella configurazione a *bus* passivo corto i TE possono essere connessi in modo casuale lungo il *bus* (fino ad 8). In questo caso il ricevitore dell'NT1 deve essere in grado di riconoscere impulsi dal TE in arrivo con ritardi differenti. Ne deriva che la lunghezza massima del *bus* è funzione del tempo di ritardo e non dell'attenuazione del cavo.

- Nella configurazione a *bus* esteso l'estensione del cavo può essere compresa tra i 100 e i 1000 metri, ma i TE devono essere raggruppati all'estremità remota del *bus* con distanza reciproca compresa tra i 25 e i 50 metri. Il rispetto di questi parametri consente di soddisfare le esigenze di installazione di utente in diverse parti del mondo.

Nelle configurazioni descritte sono sempre presenti resistenze r di terminazione per realizzare una chiusura adattata dell'interfaccia (valore 100 +/- 5% Ohm).

xDSL

Il termine xDSL (*Digital Subscriber Line*) si riferisce a diversi tipi di modem che consentono accessi fino a 300 volte superiori a quelli dei normali modem analogici. Poiché xDSL funziona su linee telefoniche tradizionali e poiché le compagnie telefoniche cercano modi vantaggiosi per fornire velocità più elevate ai propri clienti, i sistemi xDSL sono considerati come basilari per risolvere il collo di bottiglia rappresentato dall'ultimo miglio dell'infrastruttura della rete telefonica.

xDSL è principalmente una tecnologia ad elevata velocità che può essere utilizzata per trasmettere i dati di qualunque applicazione ad elevata velocità, come la videoconferenza, l'accesso veloce ad Internet, eccetera.

Per installare xDSL è necessario accedere direttamente all'infrastruttura di rete, i modem xDSL debbono essere installati ad entrambe le estremità della linea di rame (un modem deve essere piazzato presso il CPE dell'utente, mentre l'altro deve essere installato presso la centrale telefonica).

A differenza delle precedenti tecnologie di linee di rame, il sistema xDSL una volta installato non necessita di un aggiustamento manuale. Il modem xDSL analizza automaticamente la linea e adatta se stesso al fine di inizializzare il link in pochi secondi. Il processo di adattamento continua anche dopo l'inizializzazione del link, in quanto il modem compensa tutti i cambiamenti futuri. I modem contengono sofisticati algoritmi di elaborazione dei segnali digitali (DSP, *Digital Signaling Processing*) che elaborano modelli matematici delle distorsioni causate dalla linea e producono correzioni automatiche.

Per ottenere tassi di trasferimento fino a 300 volte superiori a quelli dei modem analogici, le tecnologie xDSL usano una banda di frequenze più larga. Inoltre, poiché xDSL usa un segnale digitale, a differenza dei modem analogici, le trasmissioni xDSL non passano sulla rete telefonica analogica tradizionale. Questa caratteristica di xDSL può eliminare la congestione causata dal traffico Internet.

Sono stati sviluppati diversi tipi di tecnologie xDSL, la tabella sottostante ne descrive i principali benefici:

xDSL	*Downstream*	*Upstream*	Distanza	Numero di linee telefoniche
HDSL	2 Mbit/s	2 Mbit/s	Fino a 5 Km	2
HDSL2	2 Mbit/s	2 Mbit/s		1
ADSL	Fino a 8 Mbit/s	Fino a 768 kbit/s	3.6 Km	1
ADSL II	Fino a 8 Mbit/s	Fino a 768 kbit/s	Circa 4 Km	1
RADSL	Fino a 8 Mbit/s	Fino a 768 kbit/s	Fino a 6 Km	1
SDSL	Fino a 768 kbit/s	Fino a 768 kbit/s	4 Km	1
VDSL	13, 26 o 52 Mbit/s	6 o 13 Mbit/s	Fino a 1.5 Km	1

A seguito dell'introduzione della tecnologia ISDN (che fornisce connessioni a 128 kbit/s), l'attenzione è stata rivolta allo sviluppo di tecnologie xDSL più veloci per la realizzazione di linee dati ad alta velocità (HDSL è frutto di tale sforzo), affittate dalla compagnie telefoniche ad utenze di tipo *business*. Queste tradizionali linee date in affitto operano a tassi T1 (1.54 Mbit/s) negli Stati Uniti e a tassi E1 (2 Mbit/s) in Europa.

Frame relay

Lo standard *Frame Relay*, è stato definito per fornire un accesso per la trasmissione dati di tipo commutato e di media capacità. Il *Frame Relay* prevede la separazione delle informazioni di segnalazione da quelle di utente in modo da eliminare la necessità di dover mantenere nei nodi intermedi le tabelle di stato e di dover gestire le informazioni di controllo di chiamata a livello di singola connessione. Inoltre, la multiplazione e la commutazione delle connessioni logiche vengono effettuate a livello due non a livello tre, permettendo di semplificare l'architettura di rete.

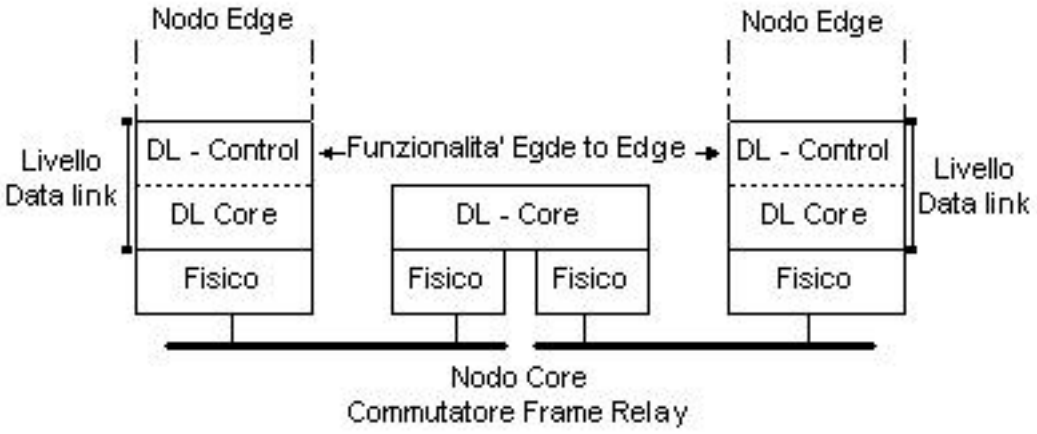
Infine, il controllo di flusso e di errore non sono applicate fra nodi successivi ma solamente, se utilizzate, a livello *end-to-end*. Proprio in quest'ultimo aspetto ha sede lo svantaggio principale dello standard *Frame Relay* in quanto rispetto all'X.25 non prevede la possibilità di effettuare un controllo di errore e di flusso orientato al singolo collegamento. In realtà questo limite può essere efficientemente superato grazie alla ormai elevata affidabilità dei dispositivi di trasmissione e di commutazione.

Il vantaggio introdotto dal *Frame Relay*, sta invece nel fatto che semplificando le funzioni protocollari sia all'interfaccia utente-rete sia all'interno della rete, si ha la possibilità di definire collegamenti caratterizzati da un minore ritardo e da un maggiore *throughput*.

Le informazioni nel *Frame Relay* sono organizzate in trame il cui formato è molto simile a quello definito per il protocollo LAPF o *core protocol*.

L'architettura protocollare necessaria per realizzare il trasporto di tali trame prevede la presenza di due piani di funzionamento separati: un piano di controllo (C) che si occupa dell'instaurazione e del rilascio delle connessioni logiche ed un piano d'utente che gestisce il trasferimento dei dati.

In particolare i protocolli del piano di controllo si collocano tra l'utente e la rete mentre quelli del piano d'utente operano a livello *end-to-end*.



Architettura protocollare nello standard Frame Relay

Le connessioni instaurate da *frame relay* sono di tipo circuito virtuale permanente e la commutazione viene effettuata al livello *data link* (nelle reti a pacchetto la commutazione avviene al livello di rete). Funziona con un approccio *core-edge*, ossia vengono differenziati i compiti che spettano ai nodi all'interno (*core*) della rete *frame relay*, con i compiti dei nodi che appartengono al bordo (*edge*).

A livello fisico utilizza canali con velocità che vanno da 64 kbps a 2 Mbps (E1). La variabilità dei ritardi introdotti da ogni nodo della rete non rendono *frame relay* idoneo per trasmettere comunicazioni vocali.

L'accesso nella rete frame relay

Anche l'accesso alla rete *frame relay* pubblica avviene realizzando un collegamento dedicato fra l'apparato di utente ed il nodo di rete.
Tale collegamento è caratterizzato da velocità superiori rispetto ad X.25, essendo il *frame relay* una tecnica a commutazione veloce di pacchetto.
L'accesso realizzato con collegamento diretto numerico fra apparato di utente e nodo di rete, *full duplex*, e velocità di accesso possibili da 64 a 2048 kbit/s fornisce un *throughput* garantito fino a 1024 kbit/s per un singolo canale virtuale assegnato (la somma dei *throughput* di tutti i canali virtuali non supera 1920 kbit/s). La rete non opera controllo di flusso né recupero di errori; effettua il *policing* e il controllo di congestione.

Un collegamento virtuale *frame relay* (DLC, *Data Link Connection*) di tipo permanente può interessare ai suoi estremi apparati del tipo: *host*, *bridge*, *router*, *switch*, *frad*. Il caso più frequente riguarda l'accesso alla rete *frame relay* impiegando un *router* con interfaccia *frame relay* (V.35) per realizzare l'interconnessione di reti locali di calcolatori a distanza.
Il servizio fornito dalla rete consiste in una o più connessioni virtuali permanenti fra gli apparati di accesso di utente, ciascuna da un CIR definito all'atto dell'installazione.

Il CIR (*Committed Information Rate*) fornito dalla rete *frame relay*, analogamente alla classe di *throughput* delle reti X.25, rappresenta la banda netta che la rete deve

garantire alla connessione virtuale. Con il *frame relay* il DTE (per esempio un *router*) può utilizzare una banda in eccesso, (detta EIR - *Excess Information Rate*) se la capacità trasmissiva della linea è tale da consentirlo, che la rete concede e trasporta in assenza di congestione se può. Il traffico in eccesso viene marcato dalla rete e reso quindi riconoscibile rispetto a quello compreso nel CIR, al fine di scartarlo in caso di congestione, privilegiando quello non marcato.

ATM - Asynchronous Transfer Mode

ATM è un modo di trasferimento che è stato sviluppato come parte integrante della rete ISDN a banda larga (B-ISDN), ma che è stato ed è tuttora utilizzato anche in altri contesti di rete, data la sua elevata efficienza. ATM sfrutta l'affidabilità offerta dai sistemi numerici ed offre una tecnica di commutazione a pacchetto caratterizzata da servizi che possono essere sia tempo reale che *store&forward*.

Le informazioni sono organizzate in celle di lunghezza costante (5 *byte* di intestazione più 48 *byte* di campo informativo), dove le informazioni relative all'identificazione della comunicazione sono contenute nell'intestazione. Le celle sono assegnate su domanda, in dipendenza delle caratteristiche del traffico della connessione.

Per il trasporto di traffico ATM, è necessario definire una tecnologia trasmissiva che sia compatibile col formato dei dati previsto da questo protocollo. In tal senso, una possibile alternativa consiste nell'utilizzo di una tecnica di sincronizzazione orientata alla cella in base alla quale l'interfaccia trasmissiva invia un flusso continuo di celle senza ricorrere ad una multiplazione dei dati orientata alla trama.

Il modello di riferimento del protocollo ATM suddivide la struttura in tre strati:

- strato fisico (*physical layer*); in questo strato sono contenute le funzioni relative all'adattamento del flusso informativo alle caratteristiche del mezzo trasmissivo e alla trasmissione delle informazioni;
- strato ATM (*ATM layer*); contiene sia le funzioni comuni a tutti i tipi di informazioni, che le funzioni riguardanti il trattamento dell'intestazione delle celle;
- strato di adattamento (*adaptation layer*); in questo strato sono presenti le funzioni dipendenti dal particolare tipo di informazione da trasferire e riguardanti l'adattamento tra sezioni di rete ATM e non-ATM, ed anche le funzioni riguardanti il trattamento del campo infomativo delle celle.

Il livello di adattamento presente in ATM è necessario per garantire l'utilizzo di protocolli per il trasferimento dei dati che non sono necessariamente basati su ATM. Tale strato ha, infatti, il compito di inserire le informazioni degli utenti in unità dati di 48 *byte* che vengono poi inseriti nelle celle ATM. Si può intuitivamente comprendere come tale operazione possa prevedere operazioni di aggregazione e di segmentazione di bit al fine di rispettare il formato dell'unità dati ATM

Le velocità di trasmissione dell'ATM sono di 155.520 Mbps, 622.080 Mbps e 2488.080 Mbps. Sono definite due tipologie di interfacce tra l'utente e la rete, una è orientata alla trama, dove le celle ATM vengono mappate all'interno di trame SDH, l'altra è orientata alle celle, ed il flusso è formato da celle ATM una di seguito all'altra. I dati viaggiano lungo dei canali virtuali, che possono essere permanenti (*Permanent Virtual Connection*) o commutati (*Switched Virtual Connection*).

La rete ATM elabora le celle solo per la parte di etichetta, risultando completamente trasparente al *payload* (salvo il caso di celle che trasportino informazioni di segnalazione o di gestione), errori compresi.
I servizi di rete che ATM può fornire sono raggruppati in categorie (in ambito ITU sono chiamate *Transfer Capabilities*; in ambito ATM Forum sono denominate *Service Categories*); ciascuna categoria è stata definita per classi di applicazioni. Le categorie di servizi sono:

- CBR (*Constant Bit Rate*, adatta ad applicazioni che richiedono alla rete ATM ritardi praticamente costanti, come la voce, il video, e l'*internetworking* di centralini telefonici privati/pubblici);
- VBR (*Variable Bit Rate*, adatta alla trasmissione dati, sensibile particolarmente alla perdita di celle ed alla qualità del trasporto, come tasso di errore del *payload* di celle, ma non sensibile particolarmente al ritardo;
- ABR (*Available Bit Rate*, una categoria abbastanza complessa che consente al dispositivo ATM di utente di richiedere più banda, in relazione alle esigenze delle applicazioni dati, relativamente alla connessione virtuale attiva.

Le connessioni virtuali che fornisce una rete ATM sono, nella maggior parte delle reti pubbliche nel mondo (Italia compresa), di tipo permanente, quindi vengono predisposte su base contrattuale senza procedure di segnalazione fra elementi della rete.

L'accesso realizzato con collegamento diretto numerico fra apparato di utente e nodo di rete, *full duplex*, e velocità di accesso possibili 2048 kbit/s, Nx2048 kbit/s, 34 Mbit/s, garantisce un molteplicità di classi di servizio (per dati, voce, video) e *throughput* adeguati alle esigenze di banda dell'utente.
La trasmissione, come in *Frame Relay*, è di tipo non riscontrata: la rete non opera controllo di flusso né recupero di errori; effettua il *policing* e il controllo di congestione.

Connessioni ATM

Le connessioni ATM sono denominate *VIRTUAL CHANNEL CONNECTION*, VCC e rappresentano l'elemento fondamentale della commutazione ATM.
Si tratta essenzialmente di un circuito virtuale analogo a quello della rete X.25 che viene stabilito fra due utenti terminali della rete e che permette lo scambio di dati a flusso variabile *full duplex* sottoforma di celle di dimensione fissa. Le VCC sono utilizzate per la gestione delle informazioni di segnalazione all'interfaccia utente-rete, per la gestione effettiva della rete e per l'instradamento.

ATM prevede un ulteriore livello di astrazione applicato al concetto di canale virtuale che si concretizza nella definizione del concetto di connessione a percorso virtuale, *VIRTUAL PATH CONNECTION*, VPC. DI fatto una VPC, può essere intesa come un fascio di VCC, caratterizzate dagli stessi punti terminali e quindi tali per cui tutte le celle di tale fascio vengono commutate assieme.

L'introduzione del concetto di percorso virtuale è legata al fatto che essa permette di ridurre i costi legati alle operazioni di controllo nelle reti ad alta velocità, dato che raggruppando le connessioni che condividono lo stesso percorso all'interno della rete, si ha la possibilità di ridurre sensibilmente le operazioni di gestione che di fatto saranno definite solo per un sottoinsieme di gruppi di connessioni, invece che su un elevato numero di connessioni singole.

Come conseguenza di tale organizzazione si ha innanzi tutto una notevole semplificazione delle funzioni di trasporto che possono essere suddivise in due gruppi distinti, uno associato ai canali virtuali e l'altro ai percorsi virtuali.
Inoltre, l'aggregazione che caratterizza tale modo di trasferimento consente un aumento delle prestazioni e dell'affidabilità della rete dato che la rete deve occuparsi delle gestione di un numero minore di entità.

L'attivazione di nuove connessioni è realizzata grazie a semplici operazioni di gestione esclusivamente nei punti terminali della rete, senza richiedere di fatto alcuna elaborazione in fase di instaurazione ai nodini transito. Ciò si ripercuote positivamente sui tempi di elaborazione e di instaurazione della connessione che sono in tal modo estremamente ridotti.

Infine l'organizzazione in percorsi virtuali permette all'utente terminale di avere una diretta visibilità del percorso stesso che può essere gestito in modo da migliorare i servizi offerti dalla rete in relazione alle esigenze di qualità richieste dall'utente stesso.

La definizione di una connessione a canale virtuale è in prima istanza subordinata all'esistenza di una connessione a percorso virtuale fra i terminali di rete cui si fa riferimento. In secondo luogo, pur ammettendo la presenza di una connessione VPC, per attivare la connessione a canale virtuale VCC, deve poter essere soddisfatta la qualità di servizio richiesta; in caso contrario la richiesta di attivazione non viene soddisfatta.

Software di interconnessione

Franco Callegati
Paolo Zaffoni

Point-to-Point Protocol

Il protocollo PPP (*Point-to-Point Protocol*) definisce un metodo standard per trasportare datagrammi generati da protocolli di livello *network* su link punto-punto.
Il protocollo PPP è costituito da 3 componenti principali:

- Un modo per incapsulare datagrammi di livello superiore.
- Un protocollo per il controllo del link LCP (*Link Control Protocol*) per instaurare, configurare e testare connessioni a livello *Data-Link*.
- Un insieme di protocolli di controllo della rete NCP (*Network Control Protocol*) per selezionare e configurare diversi protocolli di livello *Network*.

Il protocollo PPP è stato progettato per semplici link che consentono il trasporto di pacchetti tra due *peer*. Questi link sono *full-duplex* e si assume che consegnino i pacchetti nello stesso ordine con cui sono stati spediti.
L'incapsulamento del protocollo PPP consente simultaneamente il *multiplexing* di differenti protocolli di livello *Network* sullo stesso link. Tale meccanismo è stato appositamente progettato per essere compatibile con la totalità dell'*hardware* esistente.
Per essere sufficientemente versatile e portabile rispetto ad una varietà di ambienti di esecuzione, il protocollo PPP fornisce un meccanismo LCP. Il protocollo LCP viene utilizzato per accordarsi sulle opzioni di incapsulamento, per definire il formato e la lunghezza dei pacchetti, per individuare errori e/o cattive configurazioni e per terminare un link.
L'incapsulamento PPP è utilizzato per distinguere i datagrammi di protocollo superiore che possono essere trasferiti attraverso il protocollo PPP. I pacchetti del protocollo PPP sono delimitati da due *flag*, uno iniziale ed uno finale e da tre campi (*Protocol*, *Information* e *Padding*). Il significato dei campi è il seguente:

- *Protocol* - Il campo *Protocol* è costituito da uno o due *byte* ed indica il protocollo di livello superiore a cui è destinato il datagramma contenuto nel campo *Information*. Il *byte* più significativo di questo campo viene trasmesso per primo.
- *Information* - Il campo *Information* è costituito da zero o più *byte*. Tale campo contiene il datagramma di livello superiore che deve essere trasmesso attraverso il protocollo PPP. La lunghezza massima per il campo *Information*, incluso il campo *Padding* ma escluso il campo *Protocol*, viene definita *Maximum Receive Unit* (MRU), il cui valore di *default* è 1500 *byte*. Tale valore può essere modificato in base a degli accordi tra i due *host* che si trovano agli estremi del link PPP.
- *Padding* - Il campo *Padding* serve per completare il campo *Information* in modo tale che si arrivi al valore MRU che è stato stabilito dai due *host* che comunicano attraverso il protocollo PPP.

L'instradamento o routing

La funzione fondamentale dell'instradamento (*routing*) consiste nell'inoltro (*forwarding*) di pacchetti ed avviene generalmente in modalità *store-and-forward* (memorizza ed

inoltra). La necessità di ricevere completamente il pacchetto prima di ritrasmetterlo introduce un tempo di latenza pari al tempo di trasmissione.
Le tecniche fondamentali di inoltro, che differiscono per il metodo di analisi del problema instradamento, sono le seguenti:

- ***Routing by network address***. L'indirizzo di un sistema, che deve essere univoco sulla rete, è scritto direttamente nel pacchetto. Gli IS (*Intermediate System*) usano tale indirizzo come chiave di ricerca nella loro tabella di instradamento e determinano lungo quale cammino il pacchetto debba essere ritrasmesso. Tale tecnica è usata nei *transparent-bridge* (livello OSI 2), e in IP. È in generale adottata dai protocolli non connessi.
- ***Label swapping***. È generalmente usata nei protocolli connessi e trova applicazioni in ATM. Ogni pacchetto è marcato con una *label* che serve come chiave in una tabella di instradamento sull'IS. L'IS, prima di ritrasmettere il pacchetto, sostituisce la *label* con una nuova *label*. Le *label* devono quindi essere univoche solo all'interno di un dato link. Se il protocollo è connesso, le *label* altro non sono che gli identificativi delle connessioni.
- ***Source routing***. È una tecnica usata tramite una opzione del protocollo IP (per esempio, dai *bridge Token Ring*). Nel *source routing* la lista degli IS da attraversare, è scritta nel pacchetto dal nodo mittente, che lo chiede ad un IS o lo scopre con meccanismi di *route location*.

La tecnica presa in esame in questa trattazione sarà la prima, poiché è quella adottata negli schemi di instradamento IP, e quindi integrata nei protocolli e nei *router* IP.

Routing - Definizioni

Con la dicitura rete fisica si indica un insieme di calcolatori aventi le interfacce di rete attestate su una stessa sottorete, in cui una particolare tecnologia di trasporto assicura la connessione.

Una rete logica è l'insieme delle interfacce, a cui è stato assegnato lo stesso indirizzo di *subnet*, che possono comunicare senza dover passare attraverso un *router* (instradatore). Tale condizione viene detta di *routing* implicito. IP assumeva originariamente una corrispondenza biunivoca tra reti fisiche e logiche; realizzazioni più moderne ammettono anche più reti logiche nella stessa rete fisica.

Il *routing* tra reti logiche diverse è esplicito ed è gestito dai *router* tramite tabelle di instradamento.

IP adotta i concetti di destinazioni dirette e indirette nella sua logica di *routing*.

Un *host* diretto è una stazione collegata direttamente alla rete ed al *router* della rete, mentre un *host* indiretto è un *host* di destinazione situato su una rete diversa da quella dell'*host* di origine; questo significa che il datagramma deve essere inviato ad un *router* intermedio prima di essere consegnato all'*host* di destinazione.

Il modo in cui IP gestisce gli indirizzi e decide i percorsi di *routing*, richiede che una macchina esamini solo la parte di indirizzo di rete dedicata all'indirizzo di destinazione, per determinare se l'*host* di destinazione è collegato direttamente o indirettamente alla

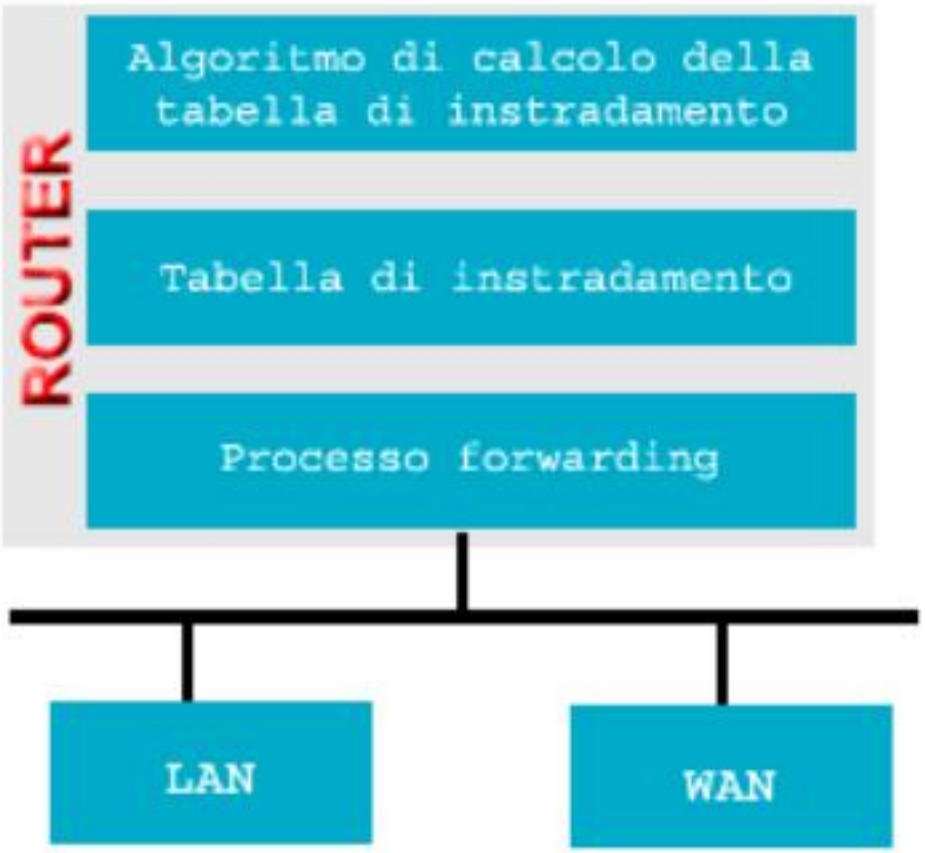
rete dell'*host* di origine: in altri termini, la macchina verifica la corrispondenza della parte rete dell'indirizzo di destinazione e sceglie se effettuare un *forwarding* diretto o *forwarding* indiretto.

- *Forwarding* diretto: la trasmissione di un datagramma IP tra due *host* connessi su una singola rete logica IP (stesso *netid*): non coinvolge i *router*. Il trasmettitore incapsula il datagramma nel *frame* fisico e lo invia direttamente all'*host* destinatario.
- *Forwarding* indiretto: i datagrammi passano da un *router* all'altro finché non raggiungono un *router* che può trasmetterli direttamente. I *router* realizzano l'interconnessione tra le diverse reti.

Tabella di instradamento: ogni *router* contiene una tabella di instradamento, visto che se un pacchetto viene destinato al *router* questo dev'essere instradato. Ogni riga nella tabella deve contenere almeno i seguenti tre elementi:

- un indirizzo di destinazione: il *router* può avere più di un percorso per la stessa destinazione.
- L'interfaccia su cui inoltrare i pacchetti.
- Il costo per raggiungere la destinazione sul percorso, che inizia con l'interfaccia indicata nella riga.

Il costo consentirà all'IS di scegliere tra eventuali percorsi alternativi; l'unità di misura di questo costo dipende dal protocollo utilizzato. Si indicano genericamente con il termine *route* le informazioni presettate su una riga della tabella di *routing*.

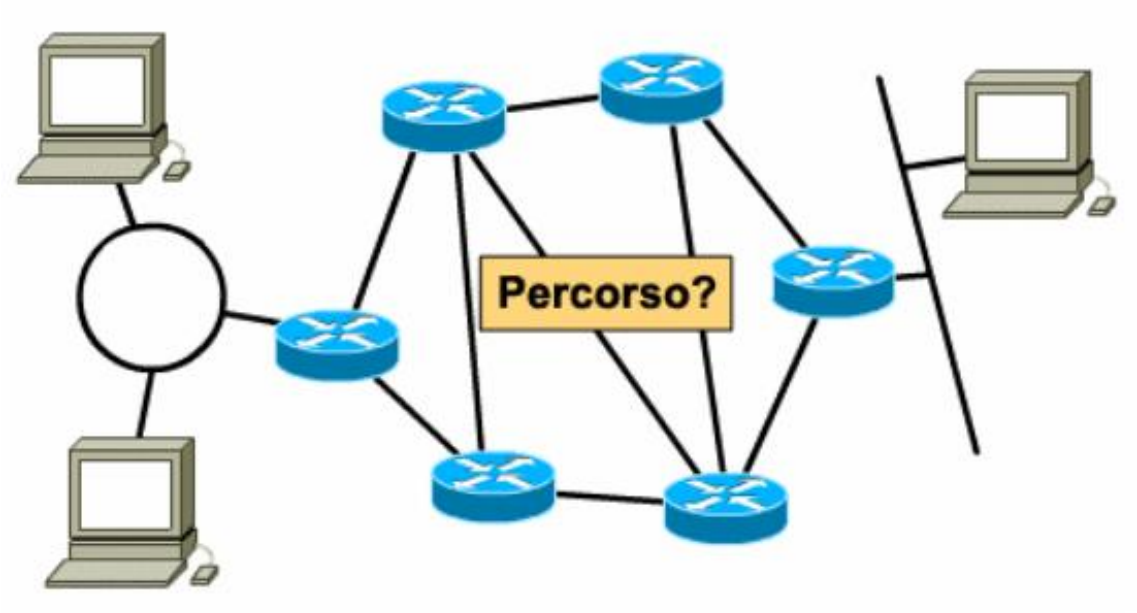


Schema dei blocchi funzionali di un router

Quando il *router* deve inoltrare un pacchetto, scorre la tabella per individuare la riga corrispondente al destinatario del pacchetto stesso. Mediamente il tempo di ricerca (*table lookup*) è pari alla metà del numero di righe. Considerando che tale operazione viene eseguita ogni volta che si deve inoltrare un pacchetto, diventa molto critica la complessità della tabella ai fini delle prestazioni dell'apparato.

Routing by network address

Nel *routing by network address*, la ricerca non verrà basata sull'intero indirizzo del destinatario, ma su un prefisso, molto spesso di lunghezza variabile. La ricerca dovrà essere eseguita nei confronti di quella riga che specifica il *route* con più lungo prefisso comune all'indirizzo del destinatario (*longest prefix matching*).
Affinché i pacchetti arrivino a destinazione è indispensabile che le tabelle nei vari IS siano coerenti tra di loro, al fine di evitare l'invio di pacchetti in percorsi ciclici (*routing loop*). In tal caso i pacchetti girerebbero a vuoto, consumando inutilmente risorse computazionali e trasmissive dei vari *router*.



Il problema del routing

Dal percorso lungo il quale un pacchetto viene inoltrato, dipendono il ritardo che esso subirà, la probabilità che venga scartato a causa di eventuali congestioni del IS e il fatto che esso raggiunga o no la destinazione. Inoltre se la rete contiene maglie, una destinazione potrà essere raggiunta attraverso una o più percorsi alternativi; in presenza di guasti, la scelta di un percorso che eviti nodi o collegamenti non funzionanti consentirà alla rete di continuare a recapitare dati.
Dunque la scelta del percorso, in altre parole il *routing*, sarà un fattore chiave per il buon funzionamento della rete e per la sua robustezza (*fault-tolerance*).

Classificazione degli algoritmi di routing

Gli algoritmi di *routing* possono essere classificati per tipo:

- Statico: negli algoritmi statici, le tabelle di *routing* che vengono memorizzate sono compilate da una persona (amministratore di rete) e i valori di tali tabelle non cambiano per nessun motivo fino a quando l'amministratore di rete non li cambia, mentre negli algoritmi dinamici le tabelle vengono continuamente aggiornate e cambiate secondo i cambiamenti della rete (caduta di una rete, inserimento di una rete).
- Gerarchici: i *router* gerarchici hanno funzioni diverse da quelli che non lo sono, poiché vengono suddivisi più nodi in gruppi logici chiamati domini di *routing*, *autonomous system* o aree. Solo alcuni di questi *router* possono interagire con ulteriori *router* di altri domini di *routing*, mentre altri possono interagire con *router* appartenenti allo stesso dominio.
- *Link-State*: *link-state* (conosciuto anche come *shortest path first*) trasferisce tutte le informazioni di *routing* a tutti i nodi: ogni *router* invia solo la porzione di tabella che descrive lo stato dei suoi link. Gli algoritmi del tipo *distance-vector* inviano tutta o parte della tabella ai soli *router* vicini. Quindi

link-state spedisce piccoli aggiornamenti a tutti, *distance-vector* spedisce grossi aggiornamenti ma solo ai *router* vicini: i *link-state* richiedono più risorse *hardware* (*CPU* e memoria) rispetto ai *distance-vector*, ma sono meno propensi ai *routing loop*.

Sicurezza

Franco Callegati
Paolo Zaffoni

La sicurezza delle reti

Le problematiche di sicurezza di una rete sono distinte da quelle legate alla sicurezza dei sistemi informativi o dei singoli calcolatori e richiedono strumenti ad hoc. Ciononostante per garantire la sicurezza globale delle informazioni è necessario che gli strumenti per la garanzia della sicurezza nei calcolatori, nei sistemi operativi e nelle reti siano in grado di interagire in modo sinergico al fine di permettere la più elevata possibile garanzia.

Quando si fa riferimento ad un evento atto a violare la sicurezza delle informazioni trasmesse all'interno della rete, si parla di procedura di attacco.

Gli attacchi alla sicurezza della rete si possono classificare secondo due grandi categorie:

- MINACCE PASSIVE: definite anche intercettazioni che rappresentano i tentativi da parte di terzi di accedere alle informazioni trasmesse durante una comunicazione.
- MINACCE ATTIVE: in cui l'accesso alle informazioni trasmesse da parte di un'entità non autorizzata è seguito dall'alterazione delle informazioni stesse e dalla trasformazione delle stesse in modo da trasmettere informazioni false.

Sulla base di tali considerazioni si può intuitivamente comprendere come l'implementazione di tecniche di protezione e la definizione dei servizi da loro offerti coinvolga in modalità diverse l'architettura di rete.

È dunque necessario, innanzi tutto, comprendere come agiscono i diversi protocolli rispetto alle molteplici problematiche di protezione dell'informazione trasmessa tenendo presente che la *suite* protocollare *TCP/IP*, sulla quale si basa la rete Internet, come noto, risulta essere, universalmente la più diffusa.

Una volta apprese le caratteristiche architetturali, la progettazione di sistemi di sicurezza ad hoc per lo scenario considerato deve prevedere un'analisi dei rischi al fine di poter individuare la tecnica migliore sia dal punto di vista dell'efficienza, sia dal punto di vista strettamente economico.

Inoltre, la conoscenza delle tecniche d'attacco consente all'amministratore di sistema di proteggere i propri sistemi prevenendo gli attacchi, ovvero adottando le misure necessarie a ridurre i fattori di rischio di esposizione.

Una fra le più comuni tecniche di protezione della rete in termini di sicurezza è rappresentata dal *firewall*. Un *firewall* si può definire come un oggetto che consente l'implementazione di una politica di sicurezza. Ovviamente per poter implementare una adeguata politica di sicurezza mediante l'aiuto di un *firewall* è necessario comprendere quali strumenti e quali tecniche vengono comunemente adottati dagli attaccanti per penetrare all'interno delle reti e per meglio comprendere la tecnologia dei *firewall* e le operazioni che svolge è necessario conoscere gli oggetti con cui un *firewall* interagisce.

Particolare importanza in tema di sicurezza è ricoperta dalle tecniche di crittografia e cifratura che costituiscono uno strumento potente ed estremamente importante per garantire la protezione delle informazioni trasmesse all'interno della rete.

In un contesto di crittografia convenzionale l'elemento fondamentale è la chiave, condivisa fra due entità, che consente di cifrare e decifrare le informazioni, ma come si può intuitivamente comprendere la distribuzione e la protezione delle chiavi costituisce a sua volta uno degli elementi di debolezza del sistema e quindi che a sua necessita di una particolare attenzione e cura in termini di sicurezza.

Nella crittografia a chiave pubblica si dispone di una coppia di chiavi, una per la cifratura e l'altra per la decifratura. Una delle due chiavi è di dominio pubblico mentre l'altra è mantenuta segreta da parte del soggetto che ha generato la coppia.

Spesso nella gestione della sicurezza di rete le due alternative vengono combinate per garantire maggiori funzionalità o una maggiore efficienza ad un sistema per la protezione delle informazioni. In particolare la chiave pubblica è utilizzata per la gestione di applicazioni di firma digitale che danno la possibilità di autenticare la sorgente delle informazioni inviate.

La presente sezione di tale modulo formativo ha dunque lo scopo di affrontare le problematiche ora descritte, al fine di garantire una conoscenza esaustiva dei vari aspetti legati alla sicurezza delle reti di telecomunicazioni, fornendo poi ulteriori spunti su alcuni degli aspetti descritti nella relativa sezione presente negli approfondimenti.

La sicurezza telematica

La presenza di molti servizi Internet standard sempre più richiesti ed utilizzati dagli utenti fa intuitivamente comprendere come tenda esponenzialmente a crescere la probabilità che, in alcuni casi, fornire un determinato servizio possa rendere la nostra rete vulnerabile rispetto ad alcune tecniche di attacco che mirano alla violazione o addirittura alla distruzione delle informazioni trasmesse all'interno della rete cui l'utente accede.

In questa unità didattica ci occuperemo dei principali servizi forniti in Internet e cercheremo di comprendere quali sono i loro principali problemi di sicurezza.

Parlando di servizi sicuri, tipicamente ci si riferisce a servizi che forniscono due tipi di garanzie:

- il servizio non può essere utilizzato in nessun modo se non per le operazioni previste.
- non è possibile leggere e/o falsificare le transazioni che avvengo attraverso il servizio.

Tali garanzie non implicano che si possano eseguire transazioni con il servizio continuando ad essere al sicuro. Per esempio, si potrebbe utilizzare un HTTP (*HyperText Transfer Protocol*) sicuro per effettuare il *download* di un *file*, ed essere sicuri che si stia effettivamente effettuando il *download* del *file* a cui si è interessati, e che nessuno lo stia modificando nel transito. Ma non si possono avere garanzie che il *file* non contenga dei virus o programmi dannosi.

È possibile anche utilizzare servizi insicuri in modo sicuro, ma ciò richiede maggiore

cautela. Ad esempio, la posta elettronica attraverso il protocollo SMTP (*Simple Mail Transfer Protocol*) è un classico esempio di un servizio insicuro.

Tutte le volte che si valuta la sicurezza di un servizio, bisogna contestualizzare le valutazioni al proprio ambiente e tener conto delle proprie configurazioni; non è interessante la sicurezza in astratto di un servizio.

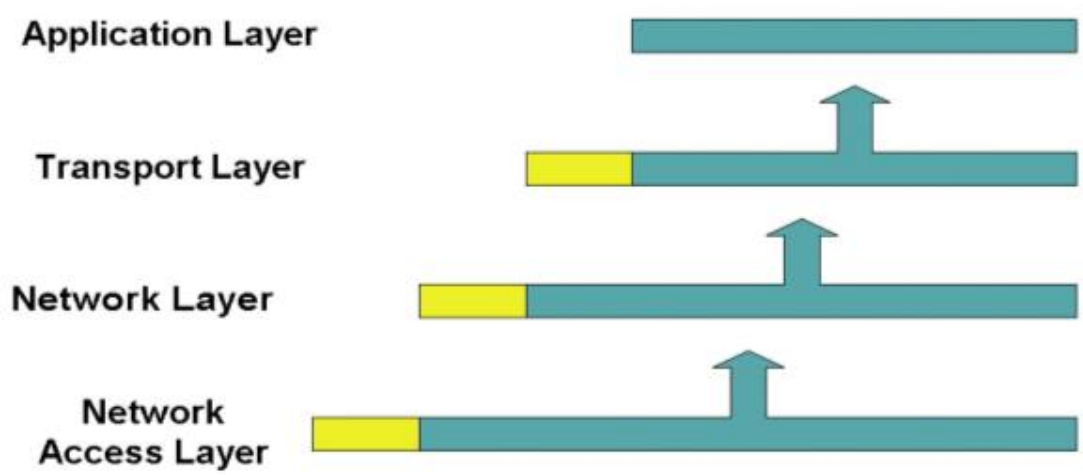
Sicurezza nei protocolli TCP/IP

Per comprendere le tecniche di *packet filtering*, una delle possibili tecniche adottate dai *firewall* per difendere la propria rete, è necessario comprendere come siano composti i pacchetti da ciascuno strato *software* che costituisce l'architettura *TCP/IP*:

- *Application Layer* (FTP, HTTP, eccetera);
- *Transport Layer* (*TCP* o UDP);
- *Internet Layer* (IP).

Ad ogni livello un pacchetto si compone di due parti: l'intestazione (*header*) e i dati (*payload*). L'intestazione contiene informazioni rilevanti per il protocollo, mentre il *payload* contiene i dati.

La costruzione del pacchetto avviene in base al meccanismo che prevede che ciascuno strato aggiunga proprie informazioni di controllo al campo dati ricevuto dalo strato soprastante. Questo procedimento che consiste nel ricevere un pacchetto da un protocollo di livello superiore e nell'aggiungere a tale pacchetto una propria intestazione viene detto incapsulamento.



Costruzione del pacchetto

Il protocollo IP

Il protocollo IP fornisce il servizio di *internetworking* in maniera non connessa e non riscontrata. Il trasporto dei pacchetti IP può avvenire con l'impiego di differenti reti fisiche basate su diverse tecnologie (locali, geografiche).

La maggior parte dei pacchetti IP sono di tipo *unicast* (sono spediti verso un unico *host* di destinazione).IP prevede anche la trasmissione e l'indirizzamento *multicast* (spediti ad un gruppo di *host*) oppure di tipo *broadcast* (indirizzati a tutti gli *host* che possono riceverli nell'ambito della rete logica di appartenenza del mittente).

Lo scopo del *multicasting* è quello di migliorare l'efficienza. Un pacchetto di tipo *multicast* è un singolo oggetto. Se diversi *host* desiderano la stessa informazione, un pacchetto di tipo *multicast* consente di spedire loro diverse informazioni trasmettendo una sola copia del pacchetto, anzichè spedire un pacchetto ciascuno.

Si noti che gli indirizzi di *multicast* e di *broadcast* debbono essere intesi come indirizzi di destinazione, e non come indirizzi di origine. Altrimenti, gli indirizzi di origine di tipo *multicast* e di tipo *broadcast* potrebbero essere utilizzati da un attaccante che sta utilizzando una macchina di destinazione per amplificare l'attacco.

L'attaccante probabilmente non sarebbe in grado di raggiungere un grande numero di *host* senza usare questo genere di scorrettezza. Non c'é interesse ad ottenere informazioni di tipo *broadcast* da altre reti, poiché non sono rilevanti per la propria organizzazione: potrebbero essere altresì potenzialmente dannosi. Un *firewall* quindi deve rifiutare i pacchetti destinati ad un indirizzo di *broadcast* e i pacchetti il cui indirizzo di origine sia un *multicast* o un *broadcast*.

L'intestazione del pacchetto IP include un campo *Options* che solitamente non viene utilizzato. Il campo opzioni IP è stato progettato per utilizzare informazioni speciali o per gestire istruzioni che non avevano un proprio campo specifico nell'intestazione. In pratica, le opzioni IP sono usate raramente eccezion fatta per i tentativi di attacco.

La più comune opzione IP che un *firewall* è costretto a controllare è l'opzione di *source routing*. Il *source routing* consente al mittente del pacchetto di specificare il percorso che il pacchetto dovrebbe seguire per giungere a destinazione, piuttosto che consentire ad ogni *router* lungo il cammino di usare la propria *routing table* per decidere a quale *router* successivo consegnare il pacchetto. Il *source routing* è stato progettato per sovrascrivere le istruzioni presenti nelle *routing table*. Lo scopo del *source routing* è di aggirare i *router* che possiedono *routing table* guaste o non corrette. In pratica, il *source routing* viene comunemente utilizzato solamente dagli attaccanti che tentano di aggirare le misure di sicurezza costringendo i pacchetti a seguire cammini inaspettati.

Alcuni sistemi di protezione seguono l'approccio di scartare tutti quei pacchetti che hanno le opzioni IP impostate, senza analizzarle; tale approccio solitamente non causa grossi problemi.

Una delle caratteristiche del protocollo IP è la sua capacità di dividere un pacchetto di grandi dimensioni, che altrimenti non potrebbe attraversare una rete (a causa delle limitazioni imposta dalle diverse porzioni di reti fisiche attraversate) in pacchetti più piccoli chiamati frammenti, che possono attraversare la rete. I frammenti vengono quindi riassemblati nell'*host* di destinazione.

Qualunque *router* può decidere di frammentare un pacchetto. Un *flag* nell'intestazione IP può essere utilizzato per evitare che un *router* frammenti un pacchetto. In passato tale *flag* non era molto utilizzato, perchè un *router* che necessità di frammentare un pacchetto ma è impossibilitato a farlo è costretto a scartare il pacchetto, cosa peraltro meno desiderabile della frammentazione stessa. Per apprendere la MTU (*Maximun Transmission Unit*) che può essere utilizzata lungo un cammino viene utilizzato un

sistema che fa uso del *flag* suddetto.

La tecnica per l'individuazione della massima MTU è un modo che consente di determinare qual'è il più grande pacchetto che può essere spedito ad una macchina senza subire frammentazione. Pacchetti grandi non frammentati consentono di avere un efficienza maggiore rispetto a pacchetti piccoli. Perciò, la massima efficienza dipende dalla conoscenza di quanto possono essere grandi i pacchetti. Al fine di scoprire tale limite massimo, i sistemi spediscono pacchetti impostando il *flag* che vieta la frammentazione e attendono messaggi di errore. Se si verifica un errore, la macchina riduce la dimensione dei pacchetti, altrimenti la aumenta.

Dal punto di vista della sicurezza il problema che si incontra con la frammentazione sta nel fatto che solo il primo frammento contiene le informazioni relative ai protocolli di più alto livello che i sistemi di *firewalling* devono controllare per decidere se far passare o meno un pacchetto. In principio, un approccio comune era quello di consentire il passaggio a tutti i frammenti facendo il controllo solamente sul primo. Questo approccio era considerato sicuro perché se il *firewall* decideva di scartare il primo frammento, l'*host* di destinazione non poteva essere in grado di riassemblare tutto il contenuto. Se il pacchetto originale non può essere ricostruito, il pacchetto parzialmente riassemblato non può essere accettato.

I problemi con i pacchetti frammentati ancora oggi persistono. Se si consente il passaggio a tutti i pacchetti eccetto il primo, l'*host* di destinazione mantiene tali frammenti in memoria per un certo periodo, in attesa di ricevere il pezzo mancante; questo consente ad un attaccante di usare i pacchetti frammentati in un attacco di tipo DoS. Quando l'*host* di destinazione rinuncia ad assemblare un pacchetto, spedisce un messaggio ICMP di tipo *packet reassembly time expired* in risposta al mittente, tale messaggio informa l'attaccante dell'esistenza dell'*host* e del motivo per cui la connessione non può essere stabilita (presenza del *firewall*).

Inoltre, gli attaccanti possono usare pacchetti frammentati in modo speciale per nascondere delle informazioni. Ogni frammento contiene i riferimenti che indicano dove i dati iniziano e finiscono. Normalmente, ogni frammento inizia dopo la fine di quello precedente. Comunque un attaccante può costruire pacchetti nelle parti in cui i frammenti si sovrappongono. Questo ovviamente non accade in condizioni normali; può accadere solamente nel caso di errori o di attacchi.

I sistemi operativi differiscono nelle modalità con cui gestiscono i frammenti che si sovrappongono. Poichè tali frammenti non sono normali, molti sistemi operativi li gestiscono male e possono riassemblarli in pacchetti non validi. Tre tecniche di attacco sono possibili grazie ai frammenti che si sovrappongono:

- Semplici attacchi di tipo DoS contro sistemi che gestiscono male i frammenti che si sovrappongono.
- Attacchi di tipo *Information-hiding*. Se un attaccante sa che sono stati installati sistemi che rilevano virus, individuano le intrusioni, o altri sistemi che sono attenti al contenuto dei pacchetti allora può costruire frammenti che nascondono il reale contenuto del pacchetto.
- Attacchi che prelevano informazioni da servizi che non dovrebbero essere accessibili. Un attaccante può costruire un pacchetto con un'intestazione valida nel primo frammento e quindi sovrapporla con il prossimo frammento. Poichè un *firewall* non si aspetta intestazioni nei frammenti

successivi al primo, non analizza tali frammenti.

Se non è possibile eseguire il riassemblaggio dei pacchetti nel *firewall* la cosa migliore da fare è scartare tutti i frammenti. Tale approccio potrebbe distruggere connessioni che avrebbero potuto andare a buon fine ma, in ogni caso, tra i due mali questo è certamente il minore.

Il protocollo TCP

Il protocollo *TCP* è il protocollo di strato trasporto più comunemente usato in Internet. Ad esempio, i servizi Telnet, FTP, SMTP, NNTP e HTTP sono tutti servizi basati sul protocollo *TCP*. Il protocollo *TCP* fornisce alle applicazioni connessioni affidabili e bidirezionali tra due *host*.
Il protocollo *TCP* è affidabile nel senso che garantisce lo strato applicativo, cioè:

- La destinazione riceve i dati applicativi nello stesso ordine in cui sono stati spediti.
- La destinazione riceve tutti i dati applicativi.
- La destinazione non riceve dati duplicati.

Il protocollo *TCP* è anche in grado di chiudere una connessione nel caso in cui non riesca a garantire le tre precedenti proprietà. Ad esempio, se vengono persi i pacchetti *TCP* nell'ambito di una sessione, il protocollo *TCP* proverà a ritrasmettere i pacchetti prima di chiudere la connessione definitivamente.

Queste garanzie implicano ritardi sui tempi di *setup* (i due lati di una connessione debbono scambiarsi delle informazioni prima che possano realmente spedire dei dati) ed influenzano le prestazioni (i due lati di una connessione debbono tenere traccia dello stato della connessione).

Il protocollo *TCP* è bidirezionale nel senso che dal momento in cui viene stabilita una connessione tra un *client* ed un *server*, il *server* ha la possibilità di rispondere al *client* sulla stessa connessione.

Per bloccare una connessione *TCP* è sufficiente bloccare il primo pacchetto di tale connessione (quello che contiene la *flag* SYN=1). Senza il primo pacchetto, qualunque altro pacchetto successivo al primo non può essere riassemblato in uno *stream* sul lato ricevente. Qualunque altro pacchetto, successivo al SYN iniziale, indifferentemente dalla direzione in cui viaggia, è contraddistinto dal bit di ACK impostato ad 1.

Il riconoscimento dei pacchetti di apertura della connessione consente il rafforzamento delle politiche di sicurezza, dal momento che, ad esempio, permette ai *client* interni di connettersi ai *server* esterni, e può vietare ai *client* esterni di connettersi ai *server* interni.

Le opzioni presenti in un pacchetto *TCP* sono:

- URG (*URGent*).
- ACK (*ACKnowledgement*).
- PSH (*PuSH*).
- RST (*ReSeT*).
- SYN (*SYNchronize*).
- FIN (*FINish*).

I *flag* URG e PSH vengono utilizzati per identificare dati particolarmente critici; PSH comunica al ricevente di interrompere il *buffering* e consegnare i dati allo strato applicativo, mentre URG identifica i dati che il mittente considera genericamente importanti. In pratica entrambi non sono implementati in maniera affidabile, quindi i *firewall* possono tranquillamente trascurarli. Potrebbe essere utile scartare i pacchetti con i bit URG e PSH impostati nei casi in cui fossero indirizzati a delle destinazioni che non li gestiscono.

I *flag* ACK e SYN vengono utilizzati per implementare il protocollo *Three-way Handshake*. Il *flag* SYN è impostato ad 1 nei primi due pacchetti che vengono utilizzati per stabilire una connessione.

I *flag* RST e FIN vengono utilizzati per chiudere le connessioni. Il *flag* RST viene utilizzato per una chiusura brutale, mentre il *flag* FIN viene utilizzato per una chiusura concordata tra i due lati della connessione.

Si deduce quindi che gli unici due *flag* interessanti per un *firewall* sono ACK e RST:

- ACK perché consente di rilevare in maniera affidabile il primo pacchetto della connessione.
- RST perché fornisce un modo utile per chiudere una connessione senza dover spedire messaggi di errore.

Si possono pensare diversi attacchi che coinvolgono l'impostazione ad 1 di alcuni *flag* che normalmente non vengono impostati. Molte implementazioni *TCP/IP* rispondono erroneamente a strane combinazioni di *flag*, bloccando ad esempio la macchina. Altre implementazioni rispondo a tali pacchetti ma non effettuano il *logging* del pacchetto, consentendo agli attaccanti di non essere rilevati, eccetera.

Il protocollo *TCP* garantisce alle applicazioni che riceveranno i dati nell'ordine corretto, ma nulla garantisce al protocollo *TCP* che i pacchetti arriveranno nell'ordine corretto. Al fine di poter ricostruire correttamente i pacchetti ricevuti, il protocollo *TCP* identifica i pacchetti attraverso un numero, chiamato numero di sequenza. All'inizio di una connessione tra due *host*, ciascun *host* seleziona un numero da cui iniziare, tali numeri vengono scambiati attraverso il protocollo *Three-way Handshake*.

Un attaccante, per poter dirottare una connessione, deve indovinare i corretti numeri di sequenza. Poiché tali numeri vengono semplicemente incrementati durante una connessione, è facile per un attaccante prevedere i numeri di sequenza futuri. D'altra parte, tale operazione è molto difficile se non si ha la possibilità di osservare i numeri di sequenza iniziali stabiliti durante l'apertura della connessione; i numeri di sequenza iniziali dovrebbero essere scelti in modo casuale. In alcune implementazioni *TCP/IP* i numeri di sequenza sono predicibili.

Per poter dirottare una connessione predicendo i numeri di sequenza, un attaccante deve:

- avere la possibilità di costruire i pacchetti *TCP/IP*;
- conoscere il numero di sequenza iniziale di una connessione;
- conoscere l'esistenza di una connessione interessante;
- avere informazioni precise sull'istante di tempo in cui una connessione è iniziata;
- avere la possibilità di rispondere in modo che nessuno possa rilevare la

sua presenza.

Per anni tale attacco è stato considerato un attacco puramente teorico, che non comporta rischi reali. Attualmente è molto diffuso ed esistono programmi di libero dominio che ne rendono possibile l'attuazione.

Analisi dei rischi

Una volta che sono ben chiare le caratteristiche di gestione della sicurezza nell'architettura protocollare che caratterizza la nostra rete è opportuno, prima di affrontare l'effettiva implementazione di strumenti e servizi per la protezione dell'informazione, effettuare un'analisi dei rischi. In effetti, la sicurezza in ciascun sistema deve essere valutata rispetto ai rischi. Il processo che consente di determinare quali controlli siano appropriati e realizzabili dal punto di vista economico è molto spesso complesso ed a volte soggettivo.

Ci sono diversi approcci per quel che riguarda la fase di analisi dei rischi, ma essi possono essere facilmente ricondotti ad uno dei due seguenti tipi:

- Analisi dei rischi di tipo quantitativo. Questo approccio impiega due elementi fondamentali: la probabilità che si verifichi un evento disastroso e le perdite stimate che possono essere associate a tale evento.
- Analisi dei rischi di tipo qualitativo. Questo approccio è quello più largamente utilizzato. Ci si basa sull'individuazione delle risorse da proteggere, dei possibili attaccanti e delle possibile tecniche di attacco.

COSA PROTEGGERE

I dati

I dati di una organizzazione possiedono tre caratteristiche che necessitano di essere protette:

- Segretezza. Si desidera che altri non possano leggerne il contenuto.
- Integrità. Si desidera che altri non possano modificarli.
- Disponibilità. Si desidera che siano sempre accessibili.

Anche se i propri dati non sono particolarmente segreti, bisogna sempre preoccuparsi delle conseguenze che si verificherebbero in seguito alla loro modifica. In questo caso si genererebbe la perdita di fiducia da parte di utenti e/o dei clienti rispetto alle tecnologie e alle politiche di amministrazione e quindi una perdita di fiducia nell'organizzazione.

Le risorse

Le risorse elaborative di un'organizzazione sono considerate pregiate e, come tali, l'organizzazione deve tutelarle evitando che gli attacchi dall'esterno ne sfruttino le capacità per scopi maliziosi.

La reputazione

Un attaccante che riesce a penetrare all'interno di un sistema si presenta in Internet con l'identità dell'organizzazione che è riuscito ad attaccare.
È noto come sia possibile comporre e spedire messaggi di posta elettronica senza ottenere l'accesso ad un certo *server*, ma è molto più facile farlo dopo essere penetrati

all'interno del sistema stesso. I messaggi che provengono dal sito attaccato ed inviati dall'attaccante non sono distinguibili da quelli inviati dalle persone realmente autorizzate.
Attacchi di tale genere riducono la fiducia verso l'organizzazione.

DA COSA PROTEGGERE

Le prerogative di un *hacker* sono:

- evitare l'individuazione e la cattura;
- nascondere la propria identità mediante un *nickname*;
- nascondere la propria collocazione geografica.

Se ottengono l'accesso su un sistema, tentano certamente di conservarlo.

Joyrider

I *joyrider* sono persone annoiate che cercano dei divertimenti. Essi violano i sistemi perché pensano di trovarvi cose interessanti o perché trovato eccitante la possibilità di usare le risorse di altri.

Vandali

I vandali sono coloro che cancellano i dati nei siti. Sono invisi anche alle persone che fanno parte dell'*underground*.

Scorekeeper

A questa categoria appartengono i collezionisti di successi. Non sono interessati solo alla qualità dei sistemi violati, ma anche alla quantità.

Spia

Appartengono a questa classe le persone cha praticano il furto di informazioni, direttamente o indirettamente convertibili in valore economico (informazioni relative a carte di credito, schede/ricariche telefoniche, eccetera).
Le precauzioni che governi e organizzazioni attuano per proteggere informazioni sensibili sono complesse e costose (schermi elettromagnetici, controllo degli accessi ossessivo, eccetera).

Strategie per la sicurezza della rete

L'approccio più semplice possibile per quel che riguarda la sicurezza è quello di considerare quali strategie adottare per ridurre al minimo i rischi.
È bene partire dal presupposto che non esiste un approccio o una strategia che possa risolvere tutti i problemi. Non esiste nulla che possa fornire una protezione perfetta e non esiste neanche una strategia che sia in grado di risolvere tutti i problemi di gestione.

Sicurezza attraverso l'*obscurity*

Un'altra semplice strategia di sicurezza è quella a cui comunemente ci si riferisce con il termine *security through obscurity*. Con questa strategia, un sistema si considera sicuro semplicemente perché si suppone che nessuno sia a conoscenza della sua

esistenza. Tuttavia, esistono molti modi per venire a conoscenza dell'esistenza di un *host*.

Ci sono diversi modi per ottenere informazioni sensibili da una macchina. Ad esempio, conoscendo l'*hardware*, il *software* e la versione del sistema operativo di un *host*, è possibile individuare le tecniche da utilizzare per accedervi. In molti casi la versione del proprio sistema operativo viene rivelata al *server* al momento del *login*.

Si inviano informazioni sensibili anche quando ci si connette con macchine esterne alla propria rete. Ad esempio, quando si effettua una connessione ad un *server* HTTP, il *client* comunica la versione del browser e del sistema operativo utilizzati.

A lungo termine, quindi, la scelta della tecnica di *obscurity* non si rivela molto efficace.

Host security

Una strategia molto utilizzata è quella che si basa sulla sicurezza a livello di *host*. Con questa strategia, si rafforza la sicurezza di ciascun *host* separatamente, e ci si sforza di evitare o alleviare tutti i problemi di sicurezza sui singoli *host*. Tale soluzione presenta difficoltà di scalabilità, al crescere del numero degli *host* e al crescere della varietà degli *host* stessi (*hardware* diverso, sistemi operativi diversi, applicazioni diverse, configurazioni eterogenee, eccetera).

La sicurezza a livello di *host* dipende fortemente dalle competenze di chiunque abbia un accesso privilegiato ad ogni macchina.

Una sicurezza a livello di *host* può essere molto appropriata per piccoli siti oppure per siti con elevati requisiti di sicurezza.

Network security

Al crescere della consistenza degli ambienti di elaborazione la soluzione basata sulla sicurezza a livello di *host* diviene sempre meno attuabile e gestibile; per questo motivo molte organizzazioni e reti di *computer* adottano una strategia a livello di rete.

In tal caso si concentra l'attenzione sul controllo degli accessi alla rete ed ai servizi offerti. Gli approcci a livello di rete includono la realizzazione di *firewall* per la protezione delle reti e dei sistemi interni, l'uso di meccanismi di autenticazione forte e l'uso della cifratura per proteggere i dati particolarmente sensibili.

Un sito può ottenere importanti vantaggi utilizzando un approccio basato sulla rete. Infatti, un singolo *firewall* può proteggere molte macchine da attacchi che provengono da reti esterne, senza preoccuparsi del tipo di sicurezza a livello dei singoli *host*.

Least privilege

Molto probabilmente, il principio fondamentale per la sicurezza è quello dei privilegi minimi. Tale principio prevede che utenti, amministratori, programmi, sistemi, dovrebbero possedere solamente i privilegi necessari per eseguire uno specifico *task*. Il principio dei privilegi minimi è un importante principio per limitare l'esposizione agli attacchi e per limitare i danni.

Tutti gli utenti non necessitano in generale di accedere ad ogni servizio Internet. Ogni utente probabilmente non necessita di modificare o leggere ogni *file* in un sistema.

Ogni utente probabilmente non necessita di conoscere la *password* di amministratore di una macchina. Ogni amministratore di sistema probabilmente non necessita di conoscere le *password* di amministrazione di tutti i sistemi. Molti sistemi operativi non sono configurati con privilegi minimi, anche per semplificare l'avviamento all'uso della macchina da parte dell'utente.

Ci possono essere due problemi nel momento in cui si decide di applicare il principio dei privilegi minimi. Innanzitutto, può essere difficile applicarlo a causa dell'esistenza di programmi e/o di protocolli non progettati per supportare tale schema. In secondo luogo, si può correre il rischio di impostare in un sistema un numero di privilegi inferiore a quelli minimi.

Il tentativo di applicare il principio dei privilegi minimi sulle persone piuttosto che sui programmi potrebbe rivelarsi controproducente. Si può predire abbastanza facilmente quali permessi siano necessari per un *mail server* (il paradosso a cui si può andare incontro è di trasformare involontariamente i propri utenti in potenziali nemici della propria rete).

Defense in depth

Un secondo principio di sicurezza è basato sulla difesa in profondità. In altri termini ci si affida a diversi meccanismi, anche per motivi di *fault tolerance*.
Si possono prevedere diversi meccanismi che forniscono *backup* e ridondanza:

- meccanismi di *network security* (*firewall*);
- meccanismi di *host security*;
- meccanismi di sicurezza per gli utenti.

Tutti questi meccanismi sono importanti e possono essere molto efficienti, ma è bene utilizzarli in maniera combinata.

Choke point

Un *choke point* obbliga un attaccante ad utilizzare un canale o un accesso obbligato. Per quanto riguarda la sicurezza nelle reti, il *firewall* tra una rete privata ed Internet (assumendo che esista un unico cammino che le interconnetta) costituisce un *choke point*; tutti coloro che desiderano attaccare la rete privata debbono passare per il *firewall*.

Un *choke point* è inutile se ci sono modi per aggirarlo; un attaccante proverà ad entrare dalla via di accesso meno sicura e meno sorvegliata.

Un *choke point* costituisce un meccanismo di sicurezza centralizzato e come tale sarà più agevole l'esercizio, la manutenzione, la configurazione.

Link più debole

Un'assunzione fondamentale per quanto riguarda la sicurezza è che una catena di sicurezza è tanto forte quanto il suo anello più debole. Gli attaccanti più scaltri cercano di individuare il punto più debole in una rete e si concentrano solo ed esclusivamente su di esso.

È necessario che l'amministratore sia a conoscenza dei punti deboli delle proprie difese per poterli eliminare quanto più possibile e per controllare attentamente quelli non

eliminabili.

Strategie di configurazione

Un altro fondamentale principio di sicurezza è quello delle configurazioni *fail safe*, cioè quelle configurazioni che continuano ad essere sicure anche a seguito di un errore e/o di un *failure*. Se, infatti, falliscono, dovrebbero lasciare gli accessi completamente bloccati.
Ci sono due approcci principali che si possono seguire rispetto alle politiche ed alle strategie da adottare per quel che riguarda la sicurezza:

- *Default deny*. Si specifica solamente ciò che è consentito e si vieta qualunque altra cosa.
- *Default permit*. Si specifica solamente ciò che è proibito e si abilita qualunque altra cosa.

Dal punto di vista della sicurezza, l'approccio migliore è quella relativo al *default deny*, mentre, probabilmente, dal punto di vista degli utenti l'approccio migliore è quello del *default permit*.

TUTTO CIÒ CHE NON È ESPRESSAMENTE CONSENTITO DEVE ESSERE PROIBITO

Tale approccio ha senso da un punto di vista della sicurezza in quanto è *fail safe*.
Con questo approccio, si proibisce di *default* qualunque cosa; per poter individuare cosa è consentito occorre:

- esaminare i servizi necessari agli utenti;
- considerare le implicazioni relative alla sicurezza con l'erogazione di tali servizi;
- permettere solamente i servizi che si conoscono, che possono essere forniti in maniera sicura e che sono strettamente necessari.

I servizi in questo modo vengono abilitati in maniera controllata.

TUTTO CIÒ CHE NON È ESPRESSAMENTE PROIBITO DEVE ESSERE PERMESSO

In questo caso, possono rilevarsi alcune malfunzioni in alcuni servizi:

- NFS non è permesso attraverso il *firewall*;
- l'accesso al WWW è consentito solamente agli utenti che sono stati adeguatamente istruiti sulle possibilità di attacco che provengono dal *Web*;
- gli utenti non possono installare servizi non autorizzati.

Tale approccio richiede di specificare cosa sia ritenibile pericoloso. Ciò che si considera pericoloso viene vietato, mentre si permette qualunque altra operazioni che non si considera pericolosa.
Ipotizzare quali siano tutti i rischi in un sistema o in Internet è un'impresa impossibile. Finché non si hanno notizie del rischio derivante dall'utilizzo di determinati servizi, tali servizi non verranno inseriti nell'elenco.

L'utente spesso reagisce a questo approccio limitativo, cercando nuovi modi per accedere ai servizi chiusi.

Architetture per reti sicure

Non esiste una terminologia completa e consistente per le architetture e componenti di *firewall*. Per quanto riguarda i *firewall* sicuramente si può schematizzare quanto segue:

- *Firewall*: un componente o un insieme di componenti che limitano l'accesso tra una rete protetta ed Internet.
- *Host*: un *computer* connesso ad una rete.
- *Bastion host*: un *computer* che deve essere reso molto sicuro in quanto potrebbe essere oggetto di attacchi.
- *Dual-homed host*: un *computer* che ha almeno due interfacce di rete.
- *Network address translation*: una procedura mediante la quale un *router* modifica i pacchetti che lo attraverso cambiando gli indirizzi di rete in base ad una opportuna politica.
- Pacchetto: l'unità fondamentale di comunicazione in Internet.
- *Packet filtering*: l'azione intrapresa da un dispositivo per controllare in maniera selettiva il flusso dei dati proveniente e/o diretto verso la rete.
- Rete perimetrale: una rete aggiunta (interposta) tra una rete protetta ed una rete esterna (Internet) al fine di fornire un ulteriore livello di sicurezza. Una rete perimetrale viene qualche volta chiamata DMZ, *De-Militarized Zone* (Zona DeMilitarizzata, riferimento alla zona che separa le due Coree).
- *Proxy*: un'applicazione *software* che dialoga con *server* esterni per conto dei *client* interni.
- *Virtual Private Network* o VPN: una rete che trasporta pacchetti, appartenenti ad una rete privata implementata sull'infrastruttura pubblica, che non possono essere decifrati dagli attaccanti.

Firewall

Per meglio comprendere la tecnologia dei *firewall* e le operazioni che svolge è necessario conoscere gli oggetti con cui un *firewall* interagisce: i pacchetti e i protocolli che sono stati utilizzati per assemblare tali pacchetti. In questa unità didattica vengono illustrate le principali problematiche relative alla sicurezza dei protocolli comunemente adottati in Internet.

Un *firewall* è un oggeto che consente l'implementazione di una politica di sicurezza. Per poter definire un'adeguata politica di sicurezza, e per poterla successivamente implementare mediante l'aiuto di un *firewall* è necessario comprendere quali strumenti e quali tecniche vengono comunemente adottati dagli attaccanti per penetrare all'interno delle reti.

La conoscenza delle tecniche di attacco consente all'amministratore di sistema di proteggere i propri sistemi prevenendo gli attacchi, ovvero adottando le misure necessarie a ridurre i fattori di rischio di esposizione.

Un buon amministratore di reti e sistemi deve essere un po' *hacker*.

Personal firewall

Si tratta di un programma progettato per proteggere adeguatamente un *computer* quando questo è collegato ad una rete. Un *personal firewall* analizza i canali di comunicazione, negando l'elaborazione del traffico ritenuto rischioso sia in ingresso sia in uscita. Di seguito si analizzano le caratteristiche di alcuni prodotti molto diffusi e si riassumono le caratteristiche comparate, in una tabella.

Tiny Personal Firewall

Tiny Personal Firewall è un prodotto facile da configurare ed utilizzare e protegge completamente un *computer* dagli attacchi. *Tiny Personal Firewall* include dei *wizard* semplici per il rilevamento delle intrusioni che individuano attività sconosciute e chiedono all'utente di impostare i parametri del *firewall*.

Per proteggere il *computer* da cavalli di Troia o applicazioni non autorizzate, *Tiny Personal Firewall* include degli *application filter*. Appositi *wizard* rilevano i tentativi di connessione alle porte di comunicazione e creano delle regole di *filtering* in base all'indicazioni dell'utente.

Per garantire che dei cavalli di Troia non si nascondano all'interno di applicazioni viene utilizzata la firma digitale con algoritmo MD5.

I *log file* generati possono essere salvati localmente oppure trasmessi ad un *server Syslog*.

Norton Personal Firewall

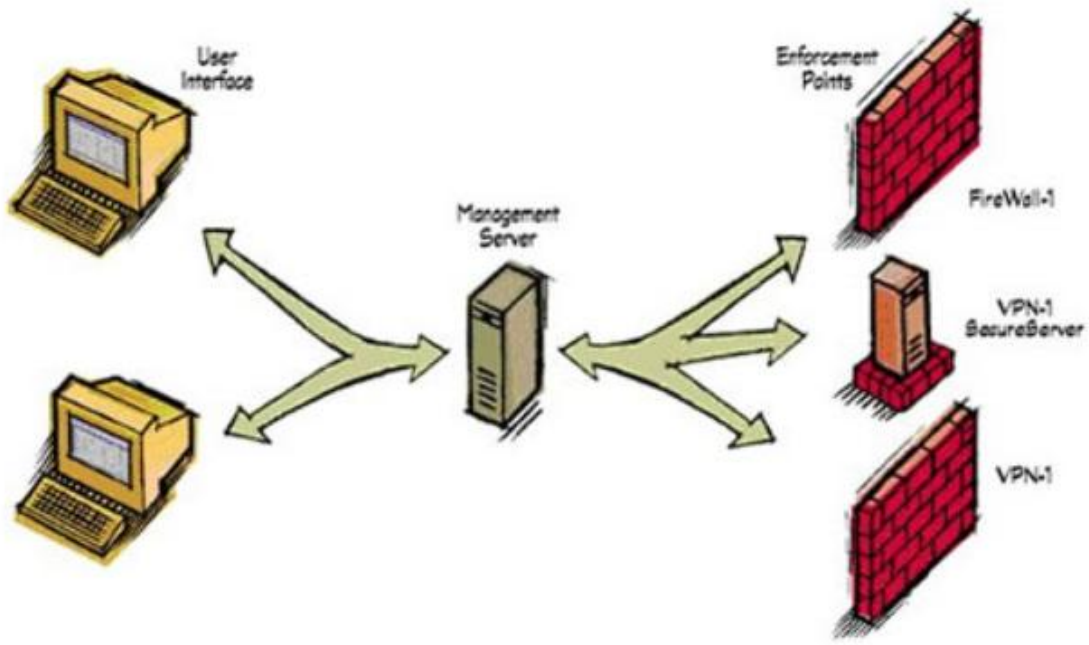
Norton Personal Firewall controlla tutte le connessioni tra il *computer* e la rete. Fornisce dei *tool* e dei *wizard* per la configurazione automatica delle regole di *filtering*.

Zone Alarm

Personal Firewall simile ai precedenti per quel che riguarda protezione e *tool* di configurazione.

Esempio di firewall commerciali

Una soluzione per la sicurezza di un'organizzazione deve essere in grado di dichiarare una politica a livello di organizzazione, distribuirla e ricevere i *log*. Deve inoltre consentire all'organizzazione di controllare l'intera infrastruttura di sicurezza (i *firewall* dell'organizzazione, le reti private virtuali) da un unico punto di amministrazione.



Il firewall: esempio d'utilizzo

Esistono diversi prodotti che soddisfano i requisiti di sicurezza e che forniscono i *tool* per la protezione delle reti private delle organizzazioni. Si analizzano a titolo di esempio le caratteristiche di due prodotti commerciali molto diffusi: *Cisco PIX* e *Checkpoint FIREWALL* 1

Firewall Cisco

Le principali caratteristiche sono:

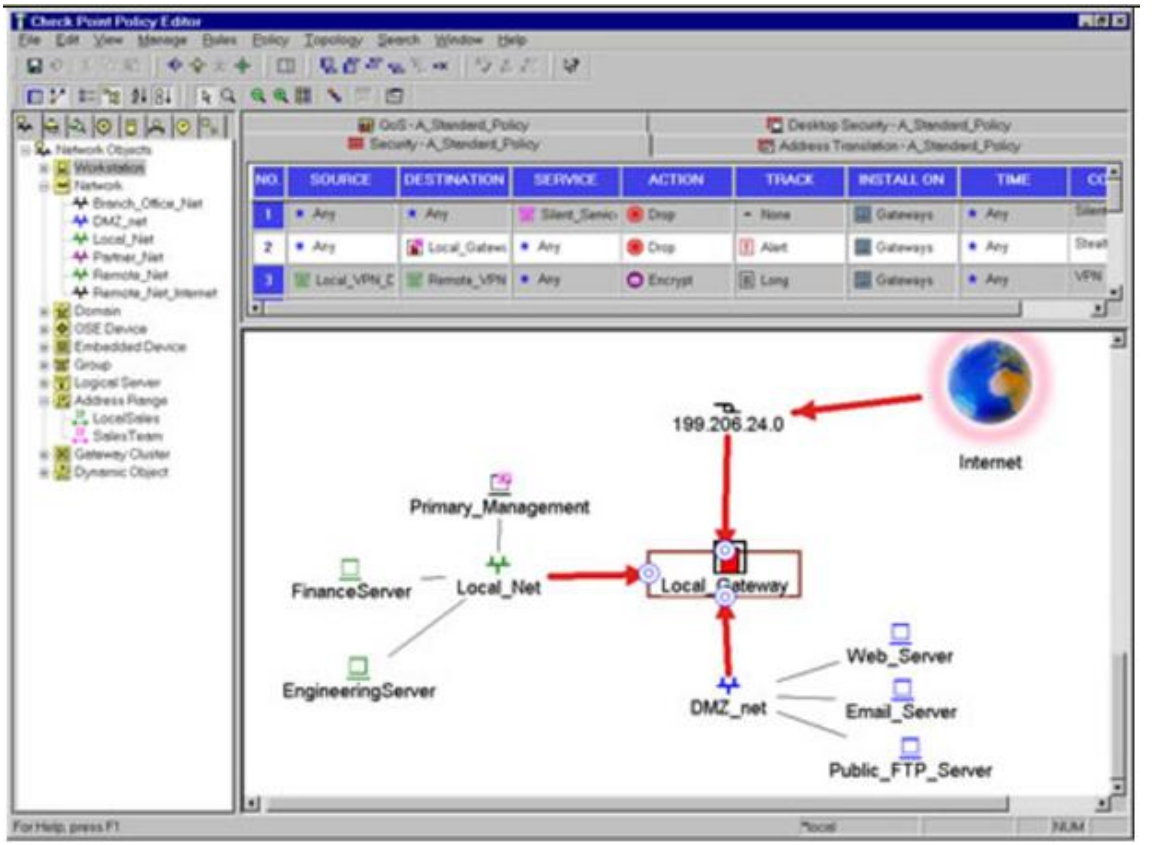
- *Context-Based Access Control*: fornisce agli utenti interni un controllo di accesso sicuro per tutto il traffico attraverso il *firewall*.
- Rilevamento delle intrusioni: fornisce il monitoraggio, l'intercettazione e la risposta in tempo reale agli abusi nella rete rilevando un vasto insieme di attacchi comuni.
- *Proxy* di autenticazione: fornisce meccanismi di autenticazione e autorizzazione degli utenti per quel che riguarda le comunicazioni di rete e/o *dial-up*.
- Rilevamento e prevenzione di attacchi di tipo DoS: difende e protegge le risorse del *router* da attacchi comuni.
- Assegnazione dinamica delle porte.
- Blocco degli *applet Java*.
- Supporto per reti VPN, cifratura IPSec e qualità del servizio.
- *Alert* in tempo reale.
- Funzionalità di *auditing* dettagliati: memorizza la data, l'*host* di origine, l'*host* di destinazione, le porte, la durata e il numero totale di *byte* trasmessi.
- *Logging* degli eventi: consente agli amministratori di rilevare in tempo reale, potenziali buchi di sicurezza o altre attività non standard effettuando il

logging dei messaggi di errore di sistema su un *Syslog server*.

- Funzionalità di gestione del *firewall*: *tool* di configurazione che offre la possibilità di definire passo passo le azioni necessarie per la protezione della rete.
- Strategie di *filtering* del traffico base ed avanzate.
- Ridondanza/*fileover*: dirotta automaticamente il traffico ad un *router* di *backup* nell'eventualità in cui il *firewall* vada in errore.
- Funzionalità NAT.
- Regole per il *filtering* temporizzato.

***Checkpoint Firewall*-1**

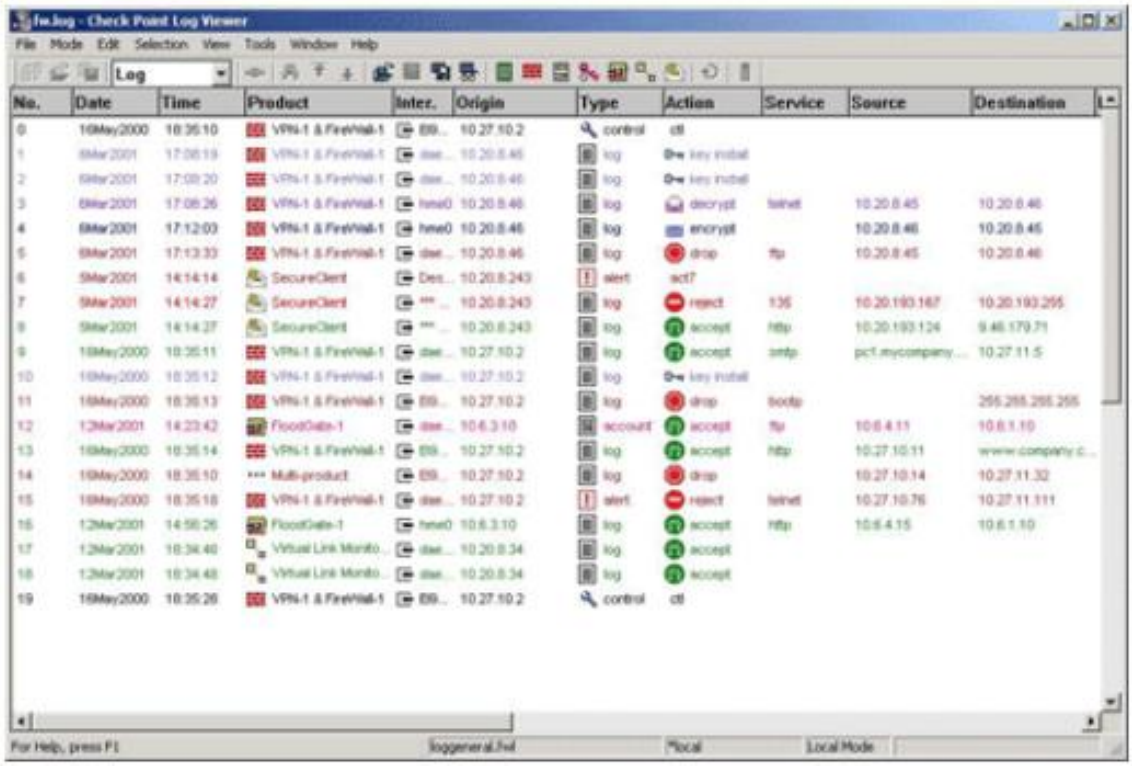
Un *firewall Cisco* è un dispositivo *hardware* per la protezione di una rete, *Checkpoint Firewall*-1 è invece un'applicazione *software*. La *console* di *management* di *Checkpoint Firewall*-1 fornisce una singola interfaccia grafica per definire e gestire molti elementi di una rete. Tutte le definizioni degli oggetti sono condivise tra tutte le applicazioni.



Interfaccia grafica della console di management di Checkpoint Firewall-1

Gli amministratori della sicurezza possono selezionare la locazione degli oggetti oppure modificarne le caratteristiche utilizzando l'*editor* visuale per la definizione delle politiche di sicurezza.
Checkpoint Firewall-1 fornisce anche un *editor* visuale per i *log* che consente un'analisi in tempo reale delle informazioni relative al *tracking*, al monitoraggio e all'*accounting* di tutte le connessioni. Il modulo per la generazione dei *report* permette agli amministratori di trasformare i dettagliati *log* del *firewall* in *report* di gestione che

rappresentano le informazioni mediante tabelle e grafici.



Editor visuale di Checkpoint Firewall-1 per i log

I *firewall* proteggono le organizzazioni in Internet fornendo accessi sicuri: garantendo che utenti validi possano accedere alle risorse di rete di cui hanno bisogno. Determinare chi sia un utente valido è compito del sistema di autenticazione; mentre determinare quali risorse un utente possa accedere è compito del sistema di autorizzazione (*Access Control*).

NO.	SOURCE	DESTINATION	SERVICE	ACTION
1	Sales@Any	Public_FTP_Serv	TCP ftp	User Auth

Esempio di sistema di autenticazione

Le regole per l'*Access Control* determinano quali utenti possono accedere alle risorse. Per fornire meccanismi di *Access Control*, un *firewall* richiede una comprensione profonda dei servizi e delle applicazioni utilizzati in rete. *Checkpoint Firewall*-1 fornisce tecnologie di ispezione di tipo *statefull*.

Le funzionalità di *content security* di *Checkpoint Firewall*-1 estendono le ispezioni dei dati fino al livello applicativo per proteggere gli utenti nei confronti di:

- Virus.

- Oggetti *ActiveX*.
- *Applet Java*.

Checkpoint Firewall-1 ha inoltre la possibilità di mascherare la visibilità interna di una rete attraverso meccanismi di NAT.

Checkpoint Firewall-1 cura inoltre tutti gli aspetti relativi alle prestazioni per non introdurre ritardi penalizzanti nell'instradamento dei pacchetti che sono abilitati a transitare.

PGP - Pretty Good Privacy

Le problematiche legate alla sicurezza della rete coinvolgono molti aspetti che sono, di fatto, diversi tra loro e condizionano in vario modo le strategie attuate per garantire un'implementazione di rete sicura. Si parla, infatti, di sicurezza telematica, sicurezza dei protocolli, sicurezza dei sistemi informativi. In tale scenario, un aspetto di particolare importanza è rappresentato dalla necessità di proteggere le informazioni sensibili durante la trasmissione delle stesse all'interno della rete. Al fine di perseguire quest'obiettivo sono stati progettati strumenti opportuni che si basano su algoritmi di crittografia di consolidata efficienza. In particolare, in questi ultimi anni, il sistema di protezione denominato PGP, *Pretty Good Privacy*, ha conseguito un enorme successo di mercato in virtù delle sue caratteristiche di particolare efficienza e di facile reperibilità a livello mondiale.

È dunque necessario per completezza della trattazione relativa agli aspetti di sicurezza delle reti, descrivere quali siano le principali caratteristiche e le relative potenzialità di tale servizio.

PGP può essere definito come un servizio d'autenticazione e d'amministrazione confidenziale delle informazioni utilizzato per la gestione sia di applicazioni di memorizzazione di *file*, sia della posta elettronica.

In effetti, come rilevato nella precedente sessione si tratta sostanzialmente dello sforzo di una singola persona, *Phil Zimmermann*, il quale, nell'implementazione di questo nuovo servizio di crittografia, ha seguito alcuni fondamentali aspetti, che si sono poi rilevati l'elemento chiave del successo del PGP a livello mondiale.

Gli elementi in questione possono essere brevemente riassunti attraverso il seguente elenco:

- Selezione dei migliori algoritmi di crittografia esistenti da utilizzare come elementi basilari per la definizione del nuovo servizio.
- Definire un'efficiente integrazione di tali algoritmi in un'applicazione di tipo *general-purpose* ossia in grado di operare in modo indipendente rispetto dall'*hardware* ed al sistema operativo utilizzato dall'utente.
- Una volta conclusa l'implementazione del servizio, garantire la libera fruizione dell'intero pacchetto applicativo via Internet, corredato da un contributo di documentazione che sia il più completo ed il più aggiornato possibile.
- Definire accordi commerciali con le più importanti compagnie al fine di fornire piena compatibilità al prodotto, garantendo anche un suo costo commerciale quanto più possibile ridotto.

In effetti, il perseguimento di tali obiettivi ha consentito al PGP di riscuotere un immediato successo pochi anni dopo la sua nascita con ritmi di crescita impressionati. Le ragioni che hanno consentito quest'ampia diffusione a livello mondiale possono essere così riassunti:

- Il PGP è disponibile in modo gratuito e facilmente reperibile in Internet (si rimanda al sito citato nella bibliografia del presente modulo dove viene offerta la possibilità di fruire liberamente di un esaustiva documentazione sul PGP, oltre che della possibilità di scaricare il relativo *software* applicativo) sia per ambiente *Window* sia per ambiente *Unix/linux*.
- Si basa su algoritmi di crittografia di provata e consolidata affidabilità: in particolare si utilizza RSA per l'operazione di crittografia delle chiavi pubbliche, IDEA (*International Data Encryption Algorithm*) per le operazioni di crittografia convenzionale ed MD5 per le codifiche *hash*.
- È caratterizzato da un elevato livello di applicabilità in relazione alle più diverse aree di utlizzo.
- Non essendo controllato da nessun ente governativo o di standardizzazione lo rende inconsciamente enormemente appetibile.

Dal punto di vista dei servizi il PGP è in grado di fornirne cinque fondamentali:

- Autenticazione.
- Trattamento confidenziale.
- Compressione.
- Compatibilità col servizio di posta elettronica.
- Segmentazione.

Tali servizi possono essere spiegati in modo semplice e schematico attraverso la seguente tabella che ne rappresenta le funzioni, i relativi algoritmi utilizzati, descrivendone le principali caratteristiche.

Funzione	Algoritmo utilizzato	Descrizione
Message Encryption	IDEA, RSA	Il messaggio è criptato utilizzando IDEA con una singola chiave di sessione generata dal trasmettitore. La chiave di sessione è poi sottoposta ad operazione di crittografia attraverso RSA con la chiave pubblica del beneficiario e inclusa nel messaggio.
Firma digitale	RSA, MD5	Utilizzando MD5 si crea un codice *hash* del messaggio. Il messaggio viene dunque sottoposto ad operazione di crittografia con RSA con la chiave privata del trasmettitore ed incluso nel messaggio.
Compressione	ZIP	Il messaggio viene compresso utilizzando ZIP al fine di ottenere una trasmissione ed una memorizzazione più efficiente.
Compatibilità con e-mail	Conversione Radix 64	Al fine di garantire trasparenza in caso di applicazioni *e-mail* il messaggio criptato può essere convertito in formato ASCII utilizzando Radix 64.
Segmentazione		Al fine di garantire il rispetto della massima dimensione ammessa per il messaggio il PGP realizza funzioni di segmentazione e riassemblaggio.

Approfondimento

Collegamenti telefonici analogici via modem

Franco Callegati
Paolo Zaffoni

8.2.1 (Distinguere tra opzioni basate su router, su switch e su bridge), 8.2.2 (Spiegare i passi necessari per connettere una rete ad Internet), 8.2.3 (Spiegare le differenze tra una connessione dial-up e una connessione dedicata)

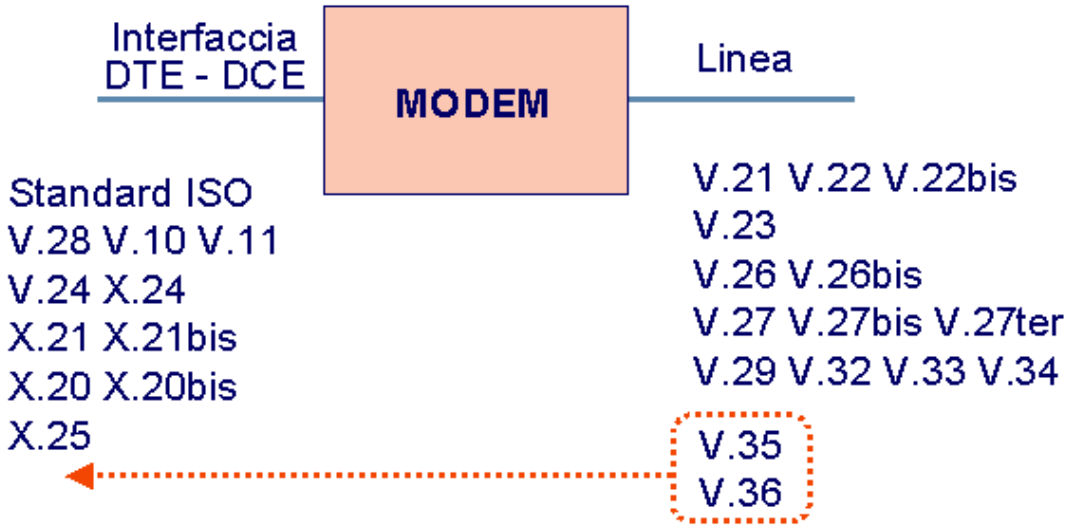
Modem per collegamenti telefonici analogici

La modulazione consiste nel modificare lo spettro di frequenze del segnale informativo digitale (codifica di linea) e di traslazione dello spettro nel dominio delle frequenze. La codifica interessa tutti i modem o **DCE** in banda base e quelli in banda fonica, la modulazione interessa soltanto i modem in banda fonica. Infatti i modem in banda base vengono utilizzati per accedere a reti specializzate che non trattano fonia e quindi non devono imporre ai segnali l'appartenenza al *range* 300 - 3400 Hz. Nella terminologia dei modem, si parla di velocità di trasmissione esprimendola in bit/s oppure in baud = simboli/s. I simboli che un modem può trasmettere sono quelli di un alfabeto predefinito e stabilito dallo standard impiegato e tipico di un determinato schema di modulazione. Per esempio se gli stati di modulazione sono 2n, ossia il modulatore emette simboli appartenenti ad un alfabeto di 2n simboli, allora la velocità trasmissiva in bit/s è n volte la velocità di segnalazione. Ossia bit/s = n x baud. Per standardizzare le caratteristiche dei modem il **CCITT** (ITU) ha emesso una serie di raccomandazioni. Le Raccomandazioni CCITT che regolano la trasmissione dati sono suddivise in due gruppi:

- Serie V inerenti alla trasmissione dati su rete telefonica commutata o su linee telefoniche dedicate.
- Serie X riguardanti i collegamenti dei terminali dati di utente con reti pubbliche per trasmissione dati (es.: reti a commutazione di pacchetto X.25).

Queste raccomandazioni definiscono le specifiche dei modem, delle interfacce, delle apparecchiature di test e la qualità delle linee.

Principali standard per le interfacce dei modem



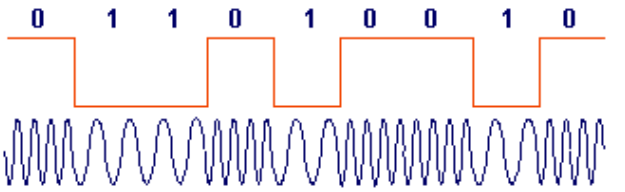
Standard per trasmissione dati

Rac. ITU	Velocità di trasmissione (bit/s)	Tipo di rete	Num. fili di linea	Modalità di trasmissione	Tipo di trasmissione	Tipo e livelli di modulazione	Baud	Frequenza portante (Hz)
V.21	300	RC	2	*Full duplex*	Asincrono	FSK2	300	1080 chiamante 1750 chiamato
V.22	1200(600)	RC	2	*Full duplex*	Asincrono o Sincrono	DPSK4(2)	600	1200 chiamante 2400 chiamato
V.22bis	2400(1200)	RC	2	*Full duplex*	Asincrono o Sincrono	QAM16(DPSK4)	600	1200 chiamante 2400 chiamato
V.23	1200(600)	RC	2	*Half duplex*	Asincrono o Sincrono	FSK2	1200(600)	1700(1500)
V.26	2400	M1020	4	*Full duplex*	Sincrono	DPSK4	1200	1800
V.26bis	2400(1200)	RC	2	*Half duplex*	Sincrono	DPSK4(2)	1200	1800
V.27	4800	M1020	4	*Full duplex*	Sincrono	DPSK8	1600	1800
V.27bis	4800(2400)	M1020	2,4	*Half/Full duplex*		DPSK8(4)		
V.27ter	4800(2400)	RC	2	*Half duplex*		DPSK8(4)		
V.29	9600(7200-4800)	M1020 M1025	4	*Full duplex*	Sincrono	QUAM16(8-4)	2400	1700
V.32	9600	RC	2	*Full duplex*	Sincrona	QUAM32e16(4)	2400	1800
V.32bis	14400	RC	2	*Full duplex*	Sincrona	QUAM128(64)	2400	
V.33	14400(12000)	M1020 M1025	4	*Full duplex*	Sincrono	QUAM128(64)	2400	1800
V.34	28800	RC	2	*Full duplex*	Sincrona			
V.34bis	31200	RC	2	*Full duplex*	Sincrona			
V.34+	33600	RC	2	*Full duplex*	Sincrona			
V.35	48 k (48.8 k)	Gruppo primario da 60 a 108 kHz	4	*Full duplex*	Sincrono / non sincrono	AM		100 kHz

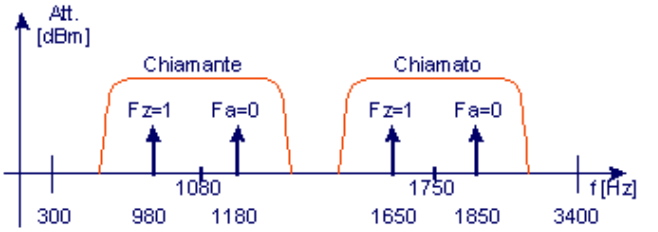
V.36	72 k (56-64-72 k)	Gruppo primario da 60 a 108 kHz	4	*Full duplex*	Sincrono	AM	100 kHz
V.37	144 k (96-112-128 k)	Gruppo primario da 60 a 108 kHz	4	*Full duplex*	Sincrono	AM	100 kHz

ITU-T V.21

- 300 bit/s asincrono (start/stop)
- rete commutata o linea dedicata a 2 fili
- modulazione FSK2



- Full duplex con ripartizione della banda del canale telefonico fra modem chiamante e modem chiamato



Modulazione V.21

L'**FSK** è una modulazione di frequenza particolare caratterizzata dall'impiego di due segnali portanti sinusoidali a frequenza f0 e f1, entrambe contenute nel range telefonico.

Il modulatore trasforma il bit 0 (1) in un treno (di durata pari alla durata del tempo di bit) di impulsi sinusoidali a frequenza f0 (f1).

Lo spettro del canale telefonico è suddiviso in due parti, utilizzate rispettivamente dal modem chiamante e dal modem chiamato. Ciascun dispositivo quindi deve poter funzionare come chiamante/chiamato e adattarsi ad utilizzare una qualsiasi delle due parti del canale, in relazione al verso della chiamata.

La normativa per l'interfaccia **DTE** - Modem prevede quanto descritto di seguito (valido per l'avviso **V.21** e per gli altri).

L'interfaccia lato DTE segue la Racc. **V24/V28** ed utilizza i seguenti circuiti:

C102	Terra di segnale	C107	Modem pronto
C103	Dati trasmessi	108/1 /2	Richiesta connessione
C104	Dati ricevuti	C109	Rivelatore portante dati
C105	Richiesta TD	C125	Rivelatore di chiamata
C106	Pronto a trasmettere	C126	Selezione canale

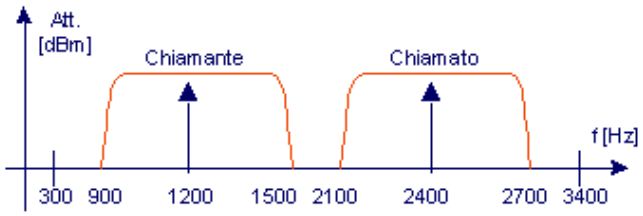
il livello segnale linea in trasmissione è regolabile da 0 a -15 dBm a passi di 2 dBm (-13dBm0); la soglia del rivelatore di portante in ricezione:

-43 dBm C109 ON,

-48 dBm C109 OFF.

ITU-T V.22

- 1200 (600) bit/s, 600 baud
- rete commutata o linea dedicata a 2 fili
- punto-punto o multipunto
- full duplex a divisione di banda (1200 canale basso e 2400 Hz canale alto)



- DP SK4 (2)

Variazione di fase DP SK4		
Dibit	Modulazione (modi I, II, III e IV)	Modulazione (modo V)
00	+90°	+270°
01	0°	+180°
10	+180°	0°
11	+270°	+90°

Variazione di fase DP SK2		
bit	Modulazione (modi I, II, III e IV)	Modulazione (modo V)
0	+90°	+270°
1	+270°	+90°

Modulazione a norma V.22

Questo tipo di modem utilizza una modulazione di fase differenziale che consente di ottenere trasmissione sincrona o asincrona su linee telefoniche a velocità 600 o 1200 bit/s.

Quando la velocità di trasmissione è 600 bit/s, ogni bit informativo viene codificato con un salto di fase rispetto alla fase del bit precedente.

Quando la velocità di trasmissione è 1200 bit/s, i bit informativi vengono raggruppati due a due (dibit) e a ciascuna coppia di bit viene associato un salto di fase della portante rispetto al valore della coppia di bit precedente.

Per entrambe le velocità trasmissive il modem trasmette lo stesso numero di simboli al secondo.

Le tabelle di codifica illustrano le regole di associazione dei salti di fase a ciascuna configurazione di bit informativi.

ITU-T V.22

- lato DTE interfaccia seriale asincrona e sincrona conforme alle Racc. V.24/V.28; circuiti utilizzati:

C102	Terra di segnale	C111	Selezione velocità trasmissione
C103	Dati trasmessi	C113	Clock Tx fornito dal DTE
C104	Dati ricevuti	C114	Clock Tx fornito dal DCE
C105	Richiesta di trasmettere	C115	Clock Rx
C106	Pronto a trasmettere	C125	Indicatore di chiamata
C107	Modem pronto	C140	Comando LOOP 2 remoto
C108/1 C108/2	Richiesta connessione Terminale dati pronto	C141	Comando LOOP 3
C109	Rivelatore portante	C142	Indicatore di test

- livello segnale linea in trasmissione da 0 a -15 dBm
- soglia del rivelatore di portante in ricezione:
  -43 dBm C109 ON, / -48 dBm C109 OFF
- scrambler/descrambler
- prevede l'inclusione di circuiti di test
- equalizzatore

Criteri all'interfaccia DTE-DCE a norma V.22

A differenza del precedente modem, il **V.22** utilizza una sola portante (chiamante: trasmissione 1200 Hz, ricezione 2400 Hz), essendo la modulazione di tipo a fase.

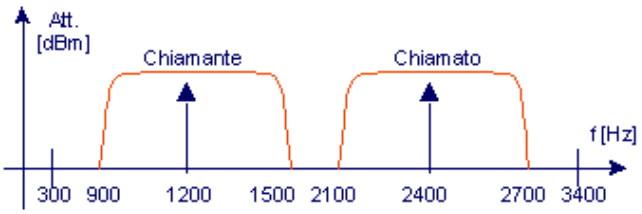
È possibile trasmettere sul canale alto (2400 Hz) ed in tal caso è prevista la trasmissione simultanea di un tono di guardia a 1800 Hz o a 550 Hz.

Questo tipo di modem utilizza anche un dispositivo interno denominato *scrambler*, avente la funzione di manipolare i bit in trasmissione prima della modulazione (e conseguentemente in ricezione dopo la demodulazione) al fine di rendere pseudocasuale la sequenza di cifre binarie trasmesse ed evitare lunghe sequenze di 0 o di 1. Lo *scrambler* è basato su un circuito digitale che genera una sequenza periodica di cifre binarie (di opportuno periodo) che viene combinata con la sequenza dei bit informativi con la regola della somma modulo 2 (1+0=0+1=1; 0+0=1+1=0).

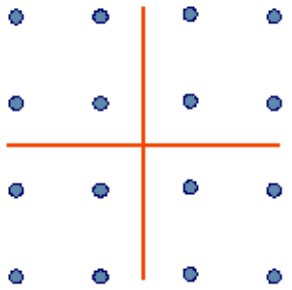
Per quanto riguarda le funzioni di diagnosi, il modem supporta il test in *loop* locale e remoto.

ITU-T V.22bis

- sincrono e asincrono a 2400 (1200), 600 baud
- rete commutata o linea dedicata a 2 fili
- punto-punto o multipunto
- full duplex a divisione di banda (1200 canale basso e 2400 Hz canale alto)



- QUAM16 (DPSK4)



Caratteristiche del modem V.22bis

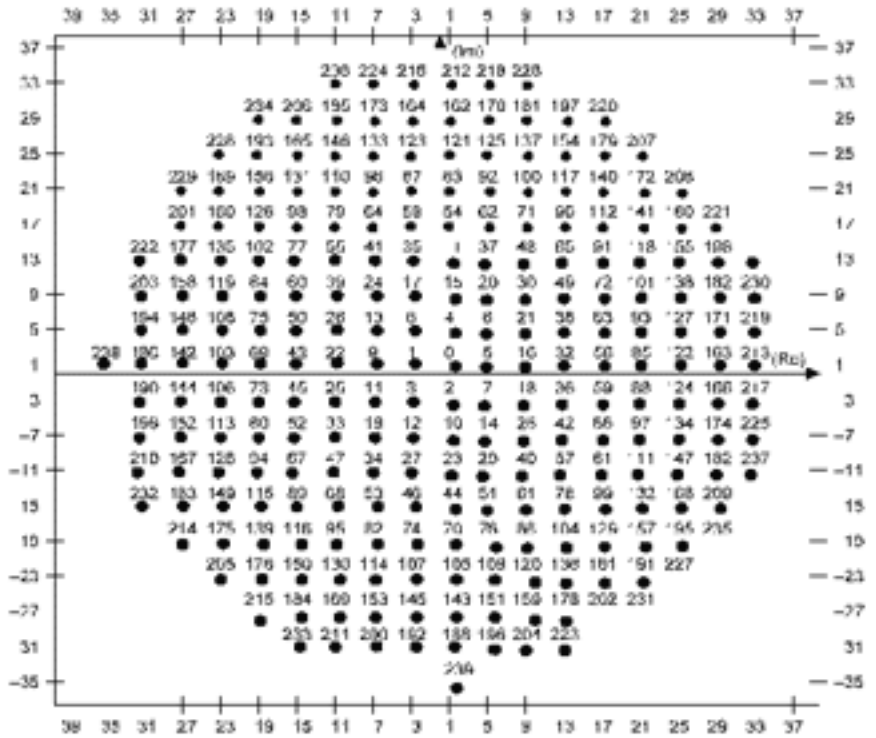
Lo schema di modulazione utilizzato da questo modem consente velocità di trasmissione di 1200 e 2400 bit/s, impiegando una portante 1200 Hz (ricezione) o 2400 Hz (trasmissione).

La modulazione è di tipo **Quadratura *Amplitude Modulation*** ed equivale ad una combinazione di modulazione in ampiezza e fase. In figura è riportata la costellazione che caratterizza questa modulazione, per la quale, ciascun simbolo di un alfabeto di 16 valori codifica 4 bit. La velocità di segnalazione è di 600 baud e a seconda del modo di funzionamento è possibile associare 2 o 4 bit a ciascun simbolo, ottenendo velocità trasmissive di 1200 o 2400 bit/s.

Nel caso di velocità 2400 bit/s, i primi due bit di ciascun gruppo di 4 determinano il cambiamento del quadrante e gli altri due identificano uno dei 4 punti di modulazione associati a ciascun quadrante.

Alla velocità di 1200 bit/s ciascuna coppia di bit determina il cambiamento di fase rispetto al quadrante occupato dalla coppia di bit precedente. In tal modo si garantisce la compatibilità con il precedente modem V.22.

ITU-T V.34



Costellazione V.34

Questo modem opera alla velocità di 28.800 bps su rete commutata o diretta a due fili (1994).

Il modem consente la trasmissione *full-duplex* impiegando la tecnica della cancellazione di eco per la separazione dei versi trasmissivi sul circuito a 2 fili.

La modulazione utilizzata da questo apparato è di tipo **QAM** (***Q****uadrature* ***A****mplitude* ***M****odulation*) per ogni canale con trasmissione sincrona a frequenza di simbolo selezionabile tra i valori obbligatori 2400, 3000 e 3200 simboli/s e quelli opzionali, 2743, 2800 e 3429 simboli/s.

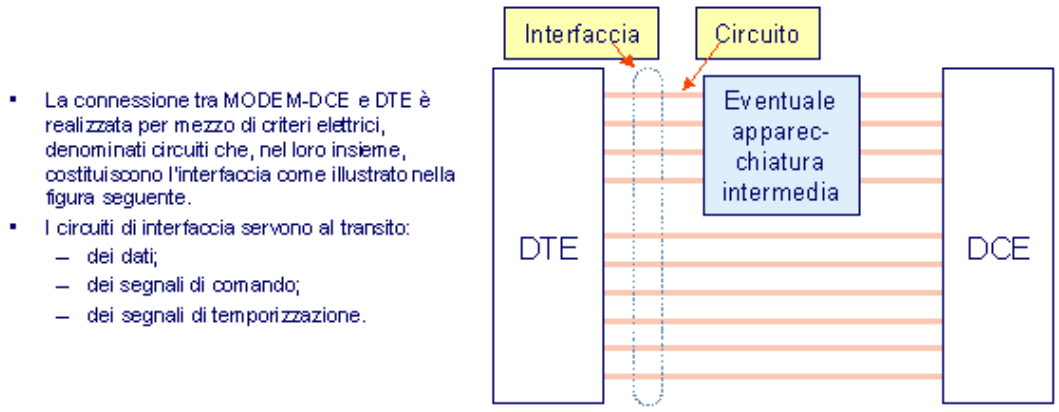
Il bit *rate* del canale sincrono primario in bit/s corrispondente è 28800, 26400, 24000, 21600, 19200, 16800, 14400, 12000, 9600, 7200, 4800, 2400; è disponibile un canale ausiliario opzionale a 200 bit/s.

Prima della modulazione viene attuata sulla sequenza di cifre binarie una codifica *Trellis* (traliccio). Tale codifica prevede una protezione dagli errori e un contenimento dello spettro di frequenze del segnale modulato. In pratica l'emissione dei segnali non è combinatoria ma correlata attraverso una memoria del sistema, secondo una regola precisa e reversibile. Il modulatore è considerabile come una macchina a stati finiti che evolve nel tempo e associa ad ogni stato un insieme di segnali possibili. Ogni stato corrisponde ad una scelta del sottoinsieme di segnali e la scelta dei segnali appartenenti alla sottocostellazione è legata ai bit informativi che non contribuiscono all'evoluzione dello stato.

La velocità di segnalazione è negoziata tra i due modem in relazione alle caratteristiche del collegamento telefonico con tecniche adattative che consentono di ottimizzare la velocità in rapporto al segnale/rumore.

La costellazione è costituita da 960 punti, un quarto dei quali è mostrato nella figura. La costellazione completa è ottenibile sovrapponendo le quattro ottenute ruotando la costellazione di 0, 90, 180 e 270 gradi.

Interfaccia DTE-modem



Costellazione V.34

Il livello più basso di un'architettura di comunicazione è lo strato fisico che costituisce la frontiera tra rete di comunicazione e apparati di elaborazione. Il suo compito è di rendere indipendenti i livelli superiori dal particolare mezzo fisico di trasmissione (linea bifilare, coppie simmetriche, cavi coassiali, fibre ottiche, ponti radio). Questo obiettivo è realizzato demandando il controllo e la gestione del portante fisico alle unità denominate ***D**ata **C**ircuit-terminating **E**quipment* (**DCE**) e definendo attraverso delle normative i segnali che permettono al *Data Terminal Equipment* (**DTE**) di attivare, mantenere e disattivare le connessioni fisiche, oltre che per trasmettere cifre binarie.

I servizi forniti dallo strato fisico a quello di collegamento (livello 2) sono:

- gestione delle connessioni fisiche: una connessione fisica può essere costituita da una o più connessioni in cascata;
- identificazione delle connessioni fisiche;
- trasmissione delle unità dati: la trasmissione può essere seriale o parallela ed eseguita in modalità *simplex, **half-duplex**, **full-duplex***;
- notifica di malfunzionamenti.

Interfaccia DTE-DCE

	V.24/V.28	V.35	V36
meccaniche	ISO 2110 o 4902	ISO 2593	ISO 4902
elettriche	V.28	V.10 eV.11.	V.10 e V.11
funzionali	V.24	V.24	V.24
procedurali	V.24, V.25 o V.25 bis		

	X.20	X.20bis	X.21	X.21 bis
meccaniche	ISO 4903	ISO 2110	ISO 4903	ISO 2110
elettriche	X.26 o X.27	V.28	X.26 o X.27	V.28
funzionali	X.24	V.24	X.24	V.24
procedurali	X.20	X.20bis	X.21	X.21 bis

Interfacce DTE-DCE

Le raccomandazioni inerenti lo strato fisico definiscono l'interfaccia specificando le caratteristiche:

- meccaniche: tipo di connettore, numero di pin ed assegnazione di ogni circuito ad un pin;
- elettriche: polarità e valori massimi e minimi delle tensioni e delle correnti; bilanciamento dei circuiti elettrici;
- funzionali: funzione di ogni circuito e numero di segnali necessari al funzionamento dell'interfaccia;
- procedurali: temporizzazione dei segnali di controllo.

La tabella riporta le caratteristiche fondamentali relative ai diversi standard utilizzati per l'interfacciamento di terminali/*host* alle reti di telecomunicazioni pubbliche a circuito e a pacchetto.

La comunicazione tra due computer su collegamento telefonico commutato avviene seguendo alcuni passi:

- si sceglie un programma (*terminal* o *hyperterminal* di *windows*, o altro) di comunicazione per PC;
- si inizializza il modem (inviando da programma, attraverso la porta seriale del PC una stringa di inizializzazione del modem che è fornita dal costruttore, reperibile dal manuale, o disponibile sul sito internet del costruttore);
- si programma il protocollo di comunicazione (sincrono, asincrono, bit di start, bit di stop, parità, ecc) che deve essere compatibile con il computer corrispondente;
- si imposta il numero telefonico del computer da chiamare, la modalità di selezione (decadica o multifrequenza), eccetera;
- si avvia il programma (nel caso più semplice è un semplice emulatore di terminale) con le impostazioni fornite;
- eventualmente è possibile programmare direttamente il modem

(difficilmente se ne ravvede la necessità), attraverso la porta seriale inviando comandi diretti con l'emulatore di terminale, per impostare la configurazione del modem (comandi AT).

Modem con caratteristiche avanzate

Limiti alle prestazioni dei modem sono imposti dalle caratteristiche del collegamento telefonico (rumore, attenuazione, diafonia, ecc.) per le tratte analogiche (code di accesso di utente).

Il modem a standard **V.34** supporta anche la velocità 33.6 kbit/s nella versione **V.34+**. Originariamente la raccomandazione prevedeva 28800 bit/s come massima velocità; successivamente alcuni costruttori hanno prodotto modem in grado di elevare tale limite e la norma V.34 è stata quindi revisionata dall'**ITU** per prevedere anche la velocità 33600 bit/s.

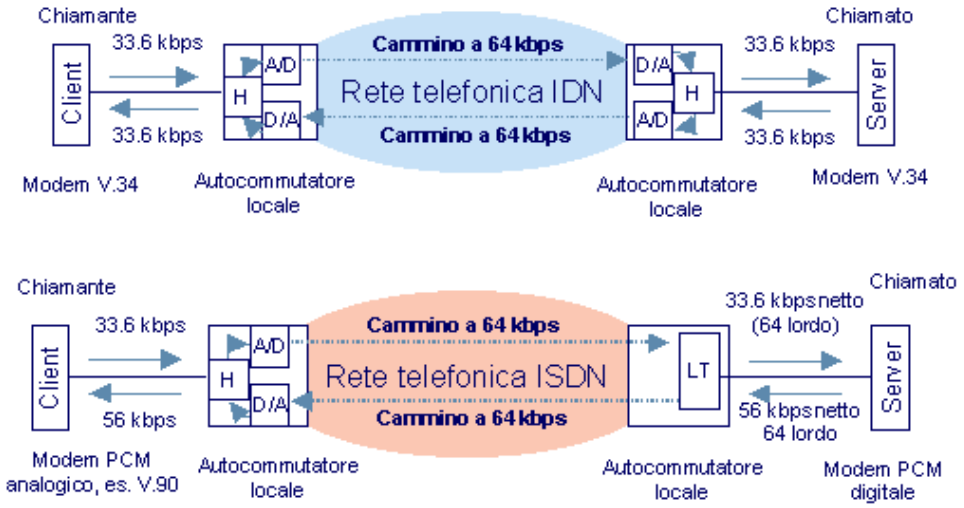
Il modem V.34 attualmente è compatibile con le seguenti velocità:

28800, 26400, 24000, 21600, 19200, 16800, 14400, 12000, 9600, 7200, 4800, 2400 e 33,6 kbps, con 1664 simboli al secondo, 10,7 bit per simbolo.

Di più recente definizione è lo standard **V.90** per modem a 56 kbit/s. È caratterizzato da un bit *rate* asimmetrico:

- 56 kbit/s in ricezione (una terminazione del collegamento deve essere numerica, eg. ISDN);
- 33.6 kbit/s in trasmissione.

Trasmissione dati su rete telefonica



Connessioni dati su rete telefonica

Un'infrastruttura IDN (*Integrated Digital Network*) è una rete in cui le funzioni di trasmissione fra nodi e di commutazione nei nodi, operano su segnali digitali. In particolare i nodi telefonici effettuano il trattamento delle chiamate utilizzando un

sistema di elaborazione basato su *software* e trattano il segnale telefonico in commutazione in maniera numerica. In tale struttura le parti interessate da segnali analogici sono i rilegamenti di utente. La principale prestazione offerta da una tale rete, consiste nel rendere la qualità della conversazione (in ambiti geografici nazionali) indipendente dalla distanza tra i nodi a cui gli utenti risultano attestati. La qualità della comunicazione è fortemente influenzata dalle caratteristiche delle linee analogiche di accesso che collegano gli utilizzatori ai nodi di competenza.

Un'infrastruttura ISDN, rispetto alla precedente, utilizzando opportuni apparati (sia lato utente che nel nodo) utilizza la linea di accesso per trasmettere le informazioni in maniera numerica, per cui la qualità della comunicazione non dipende più neanche dalla distanza tra l'utente ed il nodo di appartenenza. I vantaggi della ISDN sono anche altri, e vengono descritti successivamente.

Raccomandazione V.90

A differenza delle altre raccomandazioni sui modem, la **V.90** (rilasciata in forma definitiva in settembre 1998) specifica un metodo per trasmettere dati fra un modem analogico connesso ad una linea telefonica analogica ed un modem digitale collegato ad una linea numerica (**ISDN: BRA o PRA**).

Utilizza modalità di trasmissione *full duplex* su doppino telefonico e tecniche di cancellazione di eco.

Lo schema di modulazione utilizza la banda di frequenze del canale telefonico (0 - 4000 Hz). Le prestazioni ottenibili in senso *downstream* sono possibili per effetto di elaborazione numerica del segnale e per la mancanza di una conversione A/D (lato modem digitale). La conversione A/D introduce come è noto un rumore di quantizzazione che limita il bit *rate* della trasmissione, a parità di banda del canale telefonico.

Il bit *rate downstream* varia da 28000 bit/s a 56000 bit/s con incrementi di 8000/6 bit/s.

In senso *upstream* la trasmissione avviene con tecniche identiche a quelle specificate dalla norma V.34.

Il bit *rate upstream* varia da 4800 bit/s a 28800 bit/s con incrementi di 2400 bit/s; opzionalmente è previsto il supporto delle velocità 31200 e 33600 bit/s.

In senso *downstream* la velocità massima di 56 kbit/s è condizionata alla qualità del *local loop* cioè della linea di accesso telefonica di utente.

Fornisce un controllo adattativo delle caratteristiche del collegamento per ottimizzare la velocità trasmissiva.

In caso di impossibilità a supportare la modalità V.90, si adatta automaticamente alla modalità V.34.

ISDN

Franco Callegati
Paolo Zaffoni

8.2.1 (Distinguere tra opzioni basate su router, su switch e su bridge)

ISDN

ISDN

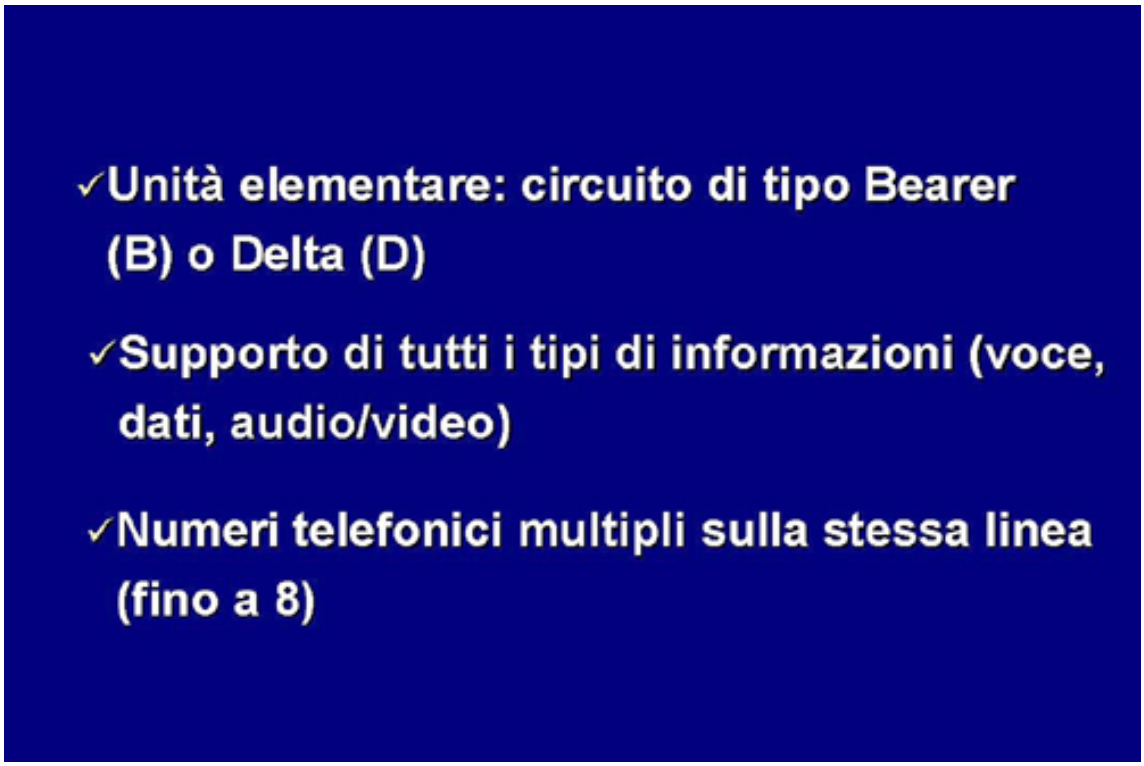
Bene, veniamo adesso alla terza lezione di questa terna dedicata alla introduzione alle reti; lezione in cui parliamo di una particolare rete, che ben si presta ad esprimere ed applicare i concetti che abbiamo visto nelle scorse due lezioni. Ci riferiamo dunque alla rete ISDN, che significa *Integrated Services Digital Network*, cioè rete digitale integrata di servizi (o di servizi integrati). Allora, la rete ISDN, prima di tutto, va collocata nel momento storico in cui nasce e cioè parecchi anni fa. Viene standardizzata nel 1988 dal CCITT (oggi ITU), quindi da un organismo di standardizzazione delle reti telefoniche e telegrafiche, ed era considerata una evoluzione della rete telefonica. In Europa è stata poi ripresa questa standardizzazione dall'ETSI, che è l'organismo europeo di standardizzazione delle telecomunicazioni, per cui gli standard che seguiamo in Europa per l'ISDN non sono perfettamente compatibili con quelli che vengono seguiti nel resto del mondo. Di conseguenza ci sono problemi di interoperabilità tra apparati ISDN europei, e quindi compatibili con gli standard ETSI, e apparati invece americani o internazionali, compatibili con gli standard ITU (o CCITT).

Introduzione

Introduzione

Vediamo un attimino cos'è ISDN: può essere vista come un insieme di protocolli di trasmissione numerica definiti da ITU (a suo tempo da CCITT). In realtà però, oltre ad essere un insieme di protocolli, e quindi un'iniziativa, un insieme di tecnologie, ISDN è anche, e forse più specificatamente, una rete, sempre nel senso che abbiamo utilizzato nelle scorse due lezioni. ISDN è una rete perché c'è qualcuno che amministra un piano di numerazione, quindi posso ordinare un accesso, in particolare a Telecom, che mi fornisce un numero di telefono ISDN. Nel momento in cui io ho questo numero sono interconnesso a ISDN e sono parte di quella rete. Ovviamente, i miei apparati, in casa mia (in casa dell'utente), e quelli di Telecom, dovranno rispettare questi protocolli e standard per la trasmissione numerica. Quindi, la grande innovazione rispetto al servizio telefonico tradizionale (quindi ISDN si contrappone, vuol rappresentare, una evoluzione della rete telefonica), è che la trasmissione è numerica. Numerica vuol dire digitale: questo significa che mentre nella rete telefonica passano soltanto segnali analogici (e quindi eventuali dati digitali devono essere prima resi analogici attraverso i modem e poi trasmessi), viceversa, sulla rete ISDN passano i dati digitali e quindi accade il contrario, cioè, se devo trasmettere la voce su ISDN devo prima digitalizzarla e poi la posso trasmettere. Quindi, la prima caratteristica è quella di essere numerica.

Caratteristiche (1)



Caratteristiche (1)

Abbiamo questa prima distinzione: l'unità elementare è un circuito di tipo B o D. Prima di tutto c'è questa terminologia tipicamente telefonica, che nel caso delle reti di calcolatori un pochino disturba, ma qui siamo nel dominio delle reti di telecomunicazione e di conseguenza bisogna utilizzarla. Abbiamo un circuito di tipo B o di tipo D, che sono quelli che trasportano qualunque cosa ci sia da trasportare, quindi: voce, informazioni, segnalazioni. Dopodiché possiamo dire che la rete ISDN, così viene presentata, supporta tutti i tipi di informazioni (voce, dati, audio/video), però se uno va a guardare il perché, si accorge che qualunque rete supporta qualunque tipo di informazione. Nel senso che io posso trasmettere voce, dati, audio/video anche su una rete telefonica tradizionale, ma ovviamente la qualità che avrò su una rete telefonica tradizionale sarà più bassa. Invece su ISDN, siccome, come vedremo, abbiamo una banda un pochino più alta, riusciamo ad avere una migliore qualità del servizio. Poi ci sono anche altri aspetti, per esempio possiamo avere numeri telefonici multipli sulla stessa linea. Ricordate il canale, abbiamo visto prima l'interconnessione a *bus*, è previsto in ISDN che io possa avere un cablaggio a *bus* che unisce fino a otto terminali: cioè fino ad otto telefoni, o fino ad otto *computer*, fino ad otto nodi. Potrebbero essere un fax, un telefono, un calcolatore, per esempio, o due calcolatori. Comunque fino ad otto, sullo stesso *bus*, quindi una interconnessione punto-multipunto.

Caratteristiche (2)

Caratteristiche (2)

Possiamo avere chiamate multiple sulla stessa linea, nel senso che la linea supporta più chiamate. Possiamo avere velocità di trasmissione variabile, nel senso che, attraverso il *bonding*, possono essere unite insieme più linee. E poi, di fatto, il vero vantaggio è che noi abbiamo delle connessioni digitali *switched*, che vuol dire commutate, in sostituzione delle connessioni dedicate. Cosa vuol dire questo? Vuol dire che in Europa, e in particolare in Italia, le connessioni dedicate, cioè le linee dedicate, hanno un costo molto elevato. Questo accade a causa delle politiche di tariffazione eseguite nel nostro paese, per cui diciamo che usare i CDN (circuiti diretti numerici), cioè le linee dedicate, è impraticabile per molti utilizzatori. Mentre invece, utilizzare la connessione digitale commutata, che ha un costo pari a quello del telefono normale, è di fatto un fattore che consente e stimola l'uso delle reti di calcolatori e quindi lo sviluppo. I servizi che ISDN fornisce sono servizi essenzialmente di trasporto digitale di dati. In aggiunta a questi, siccome è stata pensata come evoluzione della rete telefonica, consente di avere dei servizi di tipo telefonico, quindi legati alla telefonia. Per fare un esempio, i servizi di tipo supplementare che fornisce riguardano la possibilità di identificare il chiamante o il chiamato: quando qualcuno mi chiama, posso avere su un *display* la identificazione del numero chiamante, nel caso in cui il numero che io ho chiamato abbia dirottato la telefonata, è possibile che io venga a sapere chi effettivamente mi risponde. Quindi, questi sono servizi supplementari che si trovano su ISDN, ma il servizio di base è quello di trasporto dati sui canali B e D, come vedremo meglio adesso.

Fondamenti (1)

Fondamenti (1)

Come viene fornito ISDN all'utente? ISDN viene fornito attraverso due tipi di connessioni verso a centrale pubblica. Immaginiamoci la situazione in cui si trova l'utente, che è quella che accade normalmente: se ricordate, nelle reti geografiche abbiamo i nodi di accesso e l'utente che è connesso. Forse vale la pena di fare un piccolo schema, abbiamo detto che noi abbiamo una rete di qualche tipo, questa per esempio è una ISDN, e abbiamo un nodo terminale connesso attraverso la *User-Network Interface*. Quindi questo è il nostro utilizzatore, questa è la nostra rete e noi interconnettiamo l'uno all'altro attraverso la *User-Network Interface*. Questa *User-Network Interface* è quella che è standardizzata da ISDN. Volendo entrare un po' più nel dettaglio, dobbiamo pensare di avere, nella terminologia ISDN, quella che viene chiamata la SAN (*Subscriber Access Network*), e qui abbiamo il nostro utente che comunica con la *Inter Exchange Network*, poi un'altra SAN e un altro utente. Quindi, sostanzialmente, la SAN è quella rete che realizza la *User-Network Interface*, attraverso la quale il nostro utente è collegato alla nostra (quindi questo è il nostro utente e questo è l'altro utente) *Inter Exchange Network*, che è la rete di tutti gli apparati ISDN, quindi di fatto è la nostra rete ISDN. Questa è la simbologia e la terminologia che viene usata nei manuali, negli standard, ISDN.

Fondamenti (2)

Fondamenti (2)

Ovviamente, la nostra *Subscriber Access Network* avrà una parte che è fisicamente localizzata nei locali dell'utente. Quindi, se io ho un attacco ISDN, la SAN (*Subscriber Access Network*) inizia con un oggetto che è fisicamente localizzato in casa mia e finisce (su quest'altro lato) con un oggetto che è fisicamente localizzato nella centrale telefonica a cui io faccio riferimento, quindi nella centrale telefonica a me più vicina. È chiaro che quegli apparati che stanno al di qua e al di là, più il filo che interconnette tali apparati, rappresentano la *Subscriber Access Network*. Di fatto, questi apparati vengono chiamati sostanzialmente terminatori di linea. Viceversa, la *Inter Exchange Network* è la rete ISDN vera e propria, che per noi utilizzatori, noi come progettisti o utilizzatori di reti di calcolatori ci collochiamo tra coloro che usano ISDN e non tra coloro che la progettano, di conseguenza a noi non interessa come è realizzata la *Inter Exchange Network*, cioè come è realizzata la ISDN. A noi interessa che si presenti all'interfaccia, e quindi alla *User-Network Interface*, in modo standard e ci interessa che i nostri apparati siano conformi a questi standard.

Canali B e D (1)

Canali B e D (1)

Dicevamo che abbiamo due tipi di connessione: la *Basic Rate Interface* e la *Primary Rate Interface*. La *Basic Rate Interface* è quella che viene normalmente utilizzata nelle interconnessioni più semplici. Nella *Basic Rate Interface*, come vedremo, abbiamo (in tutte e due) un certo numero di canali fonici e un canale per la segnalazione. In particolare i canali fonici sono i canali di tipo B, mentre il canale per la segnalazione è un canale di tipo D. Questa è una terminologia che viene usata, non c'è alcuna ragione, ma comunque dobbiamo associare alla lettera B il canale di tipo fonico. In realtà non è un canale di tipo fonico, è un canale digitale, perché ricordiamo che l'ISDN è digitale, che ha una banda sufficiente ad allocare un canale fonico. Qui vale la pena fare una piccola digressione, che eventualmente riprenderemo alla fine di questa lezione. L'ampiezza di banda prevista per il canale B era sufficiente, o più precisamente necessaria, per allocare un canale fonico nel 1988, o forse qualche anno dopo, ma recentemente effettivamente ci sono state delle evoluzioni per cui basterebbe una banda molto inferiore. È per questa ragione che oggi si sta parlando, fra l'altro, di trasportare la voce su reti di tipo IP, senza la necessità di allocare questo tipo di canale che presenta invece una sovraccarico notevole.

Canali B e D (2)

Canali B e D (2)

I canali B e D. Il canale B ha una velocità di trasmissione di 64Kbps, vediamo perché, da dove esce fuori questo 64Kbps. Viene fuori dal fatto che la voce di qualità telefonica ha una ampiezza di banda di 4 KHz, che per essere campionata e ricostruita deve essere campionata ad una frequenza di 8 KHZ, cioè doppia, e poiché ogni campione è di 8 bit, è chiaro che deriva da questo la necessità di 64Kbps. Quindi io voglio prendere la voce, campionarla ad una frequenza doppia di quella massima e ad ogni campione associare 8 bit, da cui viene fuori 64Kbps.

Canali B e D (3)

Canali B e D (3)

Questo 64Kbps è un taglio che ci si porta dietro in tutte le telecomunicazioni, perché si assume che la voce richieda 64Kbps. Sia nelle linee dedicate, sia in ISDN, sia nei *Multiplexer*, sia negli apparati di telecomunicazione, bene o male si parla sempre di multipli di 64Kbps, come se questo fosse un valore di riferimento assoluto. Di fatto, per un certo periodo lo è stato, quando appunto la voce richiedeva 64Kbps. Oggi le tecniche di compressione della voce consentono di raggiungere delle richieste di banda molto inferiori. Lo stesso GSM viaggia sui 13Kkbps, su IP si riesce ad andare anche più in basso (10Kbps) e ci sono anche degli apparati in grado di adattarsi alle caratteristiche dei parlatori che riescono ad avere richieste di banda ancora inferiore, qualche migliaio di bps. È chiaro che, in questo contesto, avere un riferimento fisso di 64Kbps per allocare una conversazione telefonica è un po' fuori luogo, perché non c'è alcuna ragione per allocare così tanta banda per un servizio che ne richiede invece molta meno. Quindi 64Kbps comunque lo prendiamo come un dato di fatto e da questo non si può scappare.

Canali B e D (4)

Canali B e D (4)

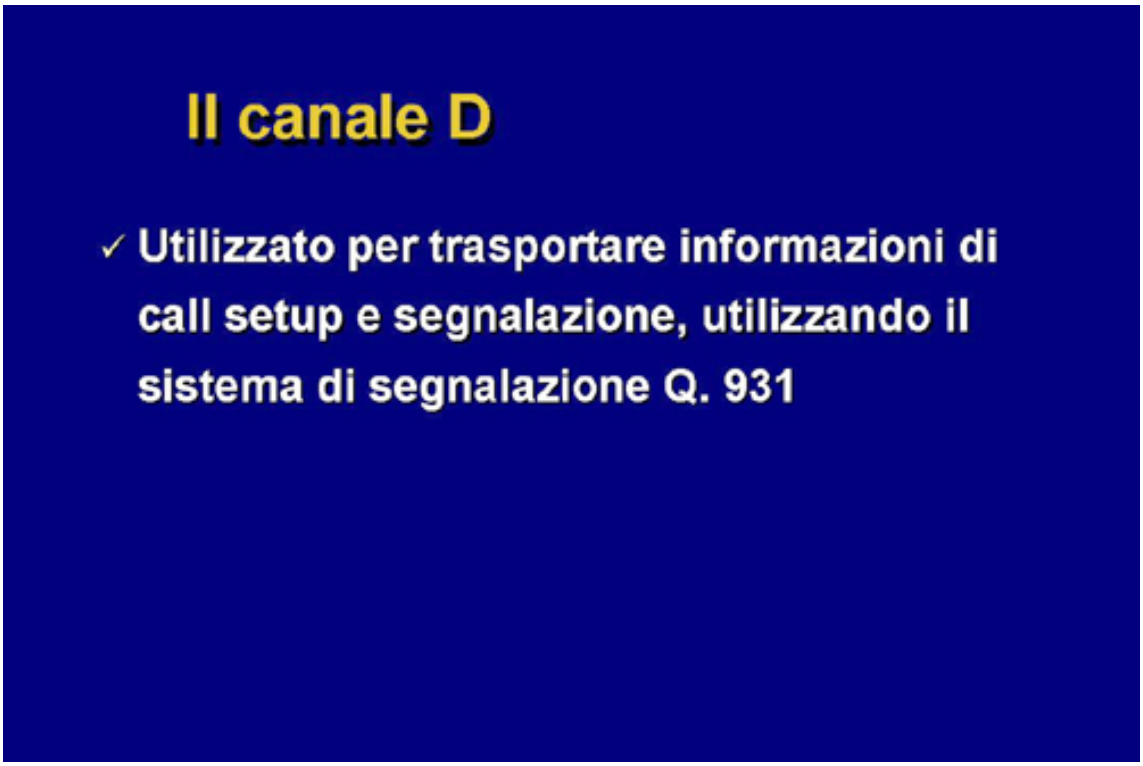
Dopodiché vediamo quali sono gli altri canali. Il canale D può avere velocità di trasmissione di 16Kbps o 64Kbps. 16Kbps nel *Basic Rate* (BRI - accesso base), 64Kbps nell'accesso primario. I canali B possono essere utilizzati per trasportare voce e dati. Come ho detto prima, in realtà trasportano dati. Un canale B non è altro che uno *stream*, cioè un flusso di bit continuo, che trasporta dati. La voce viene quindi convertita attraverso il campionamento, che abbiamo visto prima, a 64Kbps, in un insieme di dati e di conseguenza viene trasportata tranquillamente così come qualunque altra cosa, perché si può trasportare anche il video: qualunque informazione trasformabile in dati può viaggiare su un canale B.

Canali B e D (5)

Canali B e D (5)

Il canale D, viceversa, è usato per segnalazione. È usato per gestire o controllare, più propriamente, le connessioni di canali B. Non è del tutto vero anche questo, perché oggi come oggi il canale D viene anche utilizzato per trasportare dei dati. C'è la possibilità, per esempio, di trasportare i pacchetti delle reti pubbliche a commutazione di pacchetto, in particolare mi riferisco alla rete *Business Pack* in Italia, sul canale D ISDN. Però questa è una utilizzazione particolare, quella principale del canale D è la segnalazione e cioè il controllo e la gestione dei canali B.

Il canale D (1)



Il canale D (1)

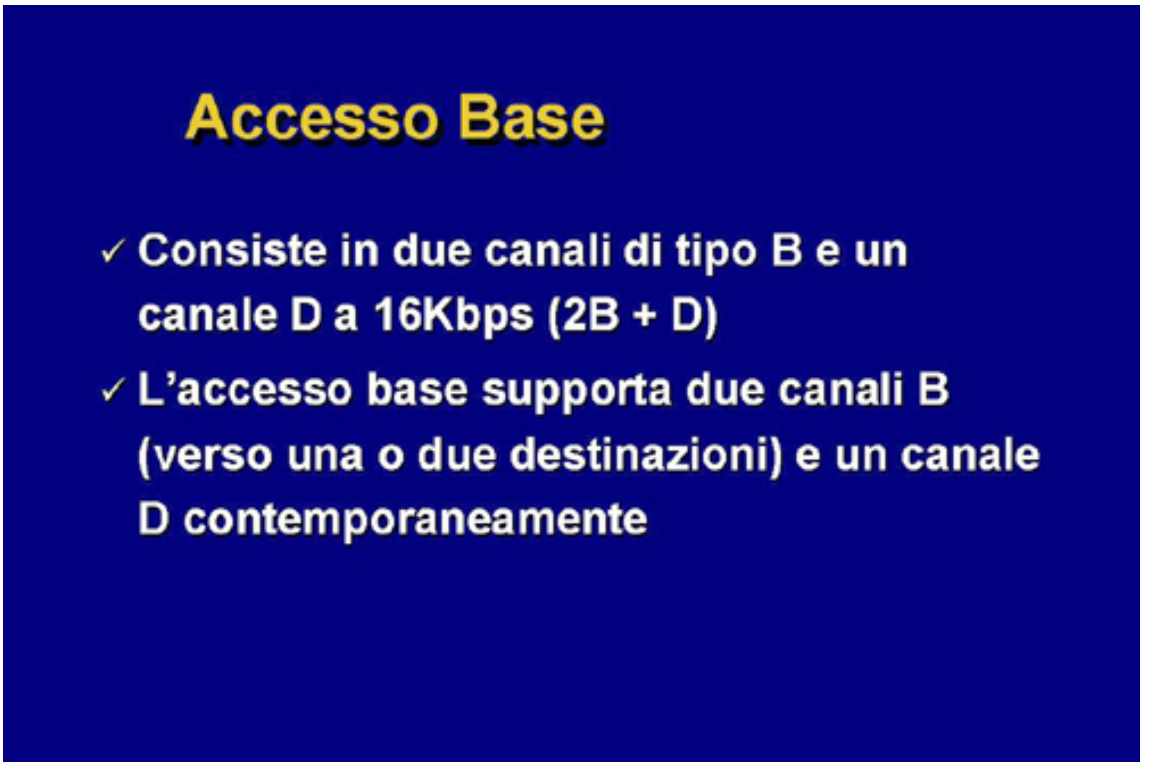
Entriamo un po' più nel dettaglio: il canale D viene usato per trasportare informazioni di *call setup*. Cosa vuol dire? Prima di tutto che cos'è la segnalazione? La segnalazione, come sappiamo, è quell'insieme di procedure attraverso le quali io controllo una comunicazione. Per esempio, io effettuo segnalazione ogni volta che alzo il ricevitore del telefono. Alzando il ricevitore del telefono nella rete telefonica normale, la centrale telefonica si accorge di questo. Questo vuol dire che il mio telefono manda un messaggio di segnalazione alla centrale telefonica. Ogni volta che io faccio un numero o schiaccio un tasto, ogni cifra è un messaggio di segnalazione. Nel caso di ISDN il canale D trasporta la segnalazione. Se, per esempio, ho un nodo terminale ISDN con un accesso *Basic Rate* (BRI), che è quello classico che tutti hanno, è come se io avessi tre fili che portano alla centrale: due per trasportare voce e uno per portare segnalazione. Quindi, ogni volta che io alzo il ricevitore del telefono, su questo canale D passa un messaggio di segnalazione che dice alla centrale telefonica: guarda che quello lì ha tirato su il ricevitore. Ogni volta che io faccio un numero, sul canale D passa un messaggio verso la centrale telefonica che dice: guarda che ha fatto quel numero, eccetera..

Il canale D (2)

Il canale D (2)

Il messaggio di *call setup* è proprio questo. Quando voglio fare una chiamata (si dice *setup* di una chiamata), vuol dire che sul canale D sta passando una segnalazione di informazione per realizzare una chiamata. Il sistema di segnalazione, quindi lo standard utilizzato per la segnalazione, è descritto nel Q. 931, che è uno standard ITU di quelli ISDN che definisce la *User-Network Interface* per ISDN. Definisce quali sono i messaggi che il mio nodo terminale deve mandare alla rete ISDN per chiamare un altro utente. Il canale D è condiviso da tutti gli utenti dell'accesso ISDN cui è associato. Torniamo al nostro discorso iniziale. Se ho la mia connessione verso una centrale, quindi ho due canali B e un canale D, in realtà il canale D viene condiviso da tutti gli utenti, che potrebbero essere fino a otto. Abbiamo detto che nel caso ISDN si prevede la possibilità di avere fino a otto utilizzatori collegati sullo stesso *bus*. Il canale D, quindi, viene visto da otto utenti.

Accesso base (1)



Accesso base (1)

Allora, l'accesso base, che è quello di cui stiamo parlando, come ho già detto, è costituito da due canali B e un canale D. I due canali B, siccome sono quelli che devono portare la voce, sono a 64Kbps, e il canale D, viceversa, è stato deciso che sia a 16Kbps, questo porta la segnalazione. Questo vuol dire che nel momento in cui io faccio una telefonata, quindi supponiamo che io ho un telefono ISDN, tiro su il ricevitore, faccio un numero, in quel momento il mio telefono segnala alla rete che sto facendo un numero e questo avviene sul canale D. Io ancora non sento niente, dopodiché viene fatta la connessione. La rete, sempre sul canale D, segnala al mio apparato che la connessione c'è e io sento il segnale di libero. Nel momento in cui il mio interlocutore tira su il ricevitore viene instaurata la comunicazione, che avviene su uno dei due canali B e quindi a questo punto la mia voce, quando io parlo, digitalizzata, cioè trasformata in dati, viaggia su uno dei due canali B, passa attraverso la rete e viene fuori da un canale B di quell'altro, arrivando così al suo terminale di rete. Durante la conversazione telefonica, la segnalazione ovviamente non serve, non ha luogo e di conseguenza il canale D è inutilizzato. Può essere utilizzato da altri terminali che sono attestati sullo stesso *bus*, per esempio, per fare un'altra chiamata sull'altro canale B. Evidentemente, essendoci due canali di tipo B, potranno esserci fino a due chiamate contemporanee: una su un canale, l'altra sull'altro. L'attivazione di queste chiamate viene gestita attraverso segnalazioni sul canale D.

Accesso base (2)

Accesso base (2)

È anche possibile, nel momento in cui io voglio un servizio di qualità maggiore, raggruppare i due canali B per vederli come se fossero un canale solo a 128Kbps. Così facendo, ovviamente, posso fare una chiamata sola, perché evidentemente, essendo insieme, ce n'è uno solo da 128. Però posso avere qualità maggiore, sempre che l'apparato dall'altra parte rispetti lo stesso tipo di standard.

Accesso primario (1)

Accesso primario (1)

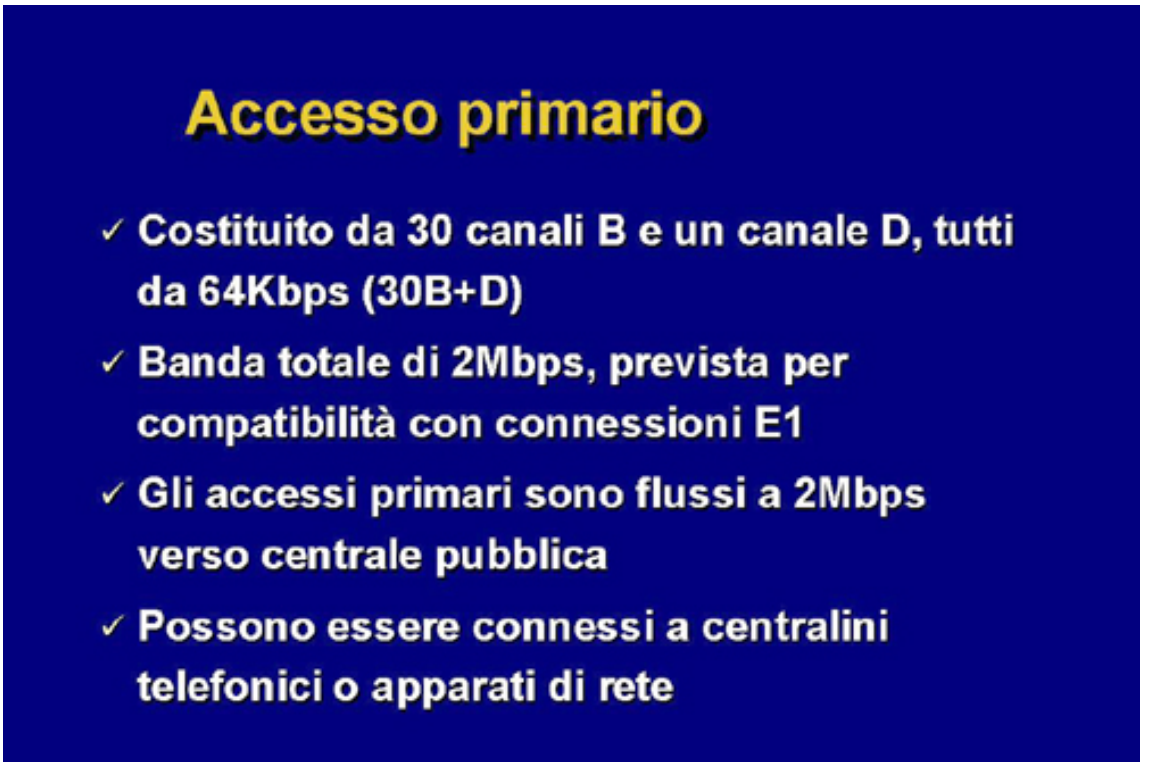
L'accesso primario è un'altra cosa. Fino adesso abbiamo visto l'accesso base, quello classico. Al posto di un servizio telefonico normale posso richiedere un accesso base ISDN, che costa uguale sostanzialmente, e poi pago gli scatti: è come avere un telefono. L'accesso primario è una cosa diversa: è per chi tipicamente ha un centralino o degli apparati che richiedono una elevata banda di trasmissione. In particolare, vedete che è costituito da 30 canali B e un canale D, quindi trenta canali B vuol dire trenta conversazioni telefoniche contemporanee, e un canale D, sempre per la segnalazione. Questo canale D serve per fare la segnalazione di tutte le chiamate. E anche il canale D in questo caso è da 64Kbps. Si dice quindi che abbiamo il 30B+D, perché abbiamo trenta canali in grado di trasportare conversazioni telefoniche a 64Kbps e un canale D, sempre da 64Kbps, per la segnalazione.

Accesso primario (2)

Accesso primario (2)

La banda totale, se uno si mette lì e fa due moltiplicazioni, si accorge che è di 2Mbps, in particolare se io faccio (30x64)+64 ottengo quasi 2 Mega e poi ci sono altri 64Kbps per la sincronizzazione di trama, che è un discorso che non abbiamo fatto e forse toccheremo in seguito. Comunque, di fatto, diciamo che 64Kbps vengono persi per la sincronizzazione, 30x64 sono conversazioni foniche e 64Kbps sono per la segnalazione. La banda di 2Mb è prevista per la compatibilità con connessioni E1. Come vi accennavo in una delle due lezioni passate, la gerarchia digitale europea prevede che dopo 64Kbps ci sia il 2Mb che viene anche chiamato E1, di conseguenza quando hanno pensato allo standard ISDN per l'Europa, hanno pensato di portarlo a 2Mb, in maniera tale che gli apparati già disponibili per il supporto delle linee dedicate a 2Mb potessero essere utilizzati, o per lo meno in parte riutilizzati, per ISDN. Perché vi dico questo? Perché negli Stati Uniti, invece, è diverso. Negli stati Uniti l'accesso primario non è 30B+D, perché la loro gerarchia digitale non prevede un E1 a 2Mb, ma prevede invece un T1 a 1,5Mb, di conseguenza anche il primario ISDN è di quel taglio.

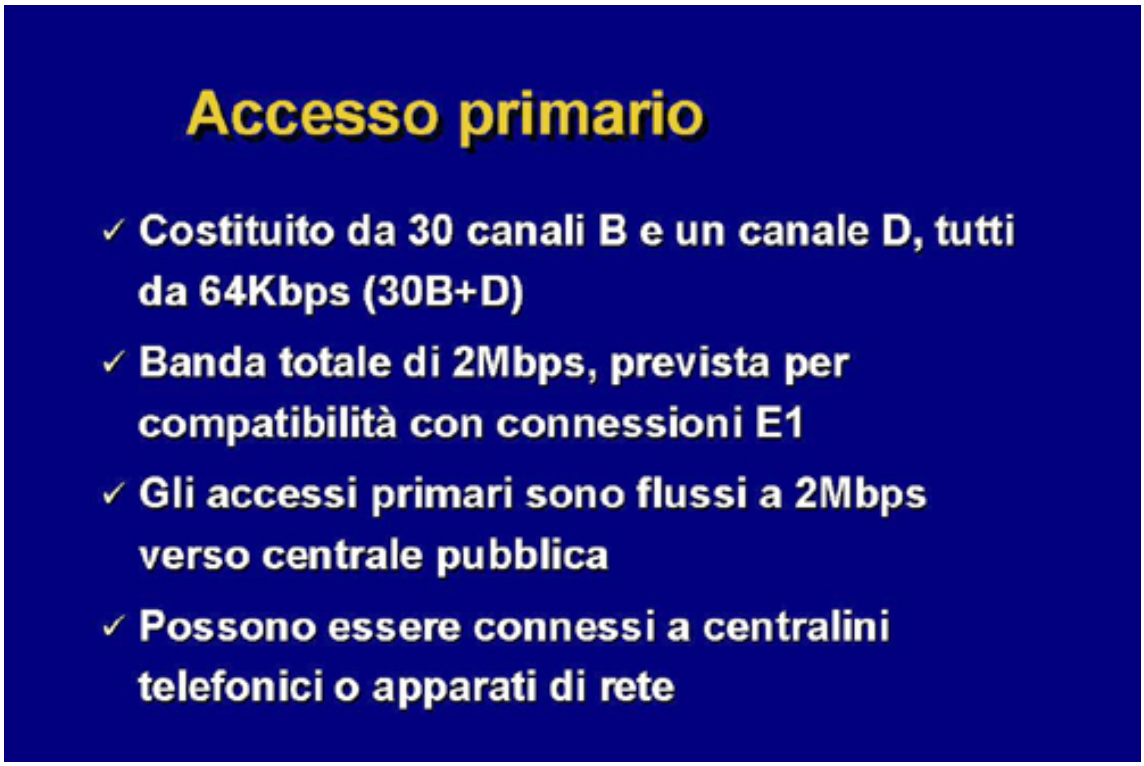
Accesso primario (3)



Accesso primario (3)

Gli accessi primari sono flussi a 2Mb verso centrale pubblica. Dobbiamo sempre ricordare il modello: noi stiamo parlando della *User-Network Interface*, cioè da me (utente) alla rete (quindi alla centrale telefonica). Se io ho un accesso primario, vuol dire che da me alla centrale telefonica è stata installata una linea a 2Mb, come se avessi un CDN, cioè un collegamento diretto numerico, ma gestito in modo ISDN. Per cui di questi 32 canali a 64Kbps uno è usato per la sincronizzazione, uno è usato per la segnalazione, 30 sono usati per la voce, o meglio per canali di dimensioni tali (64Kbps) da poter trasportare la voce. L'accesso primario, in genere, è connesso a un centralino telefonico, ovviamente, che magari al suo interno ha un elenco di qualche centinaio di interni e ha 20 o 24 o, al massimo, 30 linee esterne. Allora, invece di avere 20, 24 o 30 fili verso la centrale telefonica, ne ha uno solo, che è un accesso primario ISDN, nel quale sono multiplexate tutte le conversazioni, fino a 30 e la segnalazione è fatta su un unico canale.

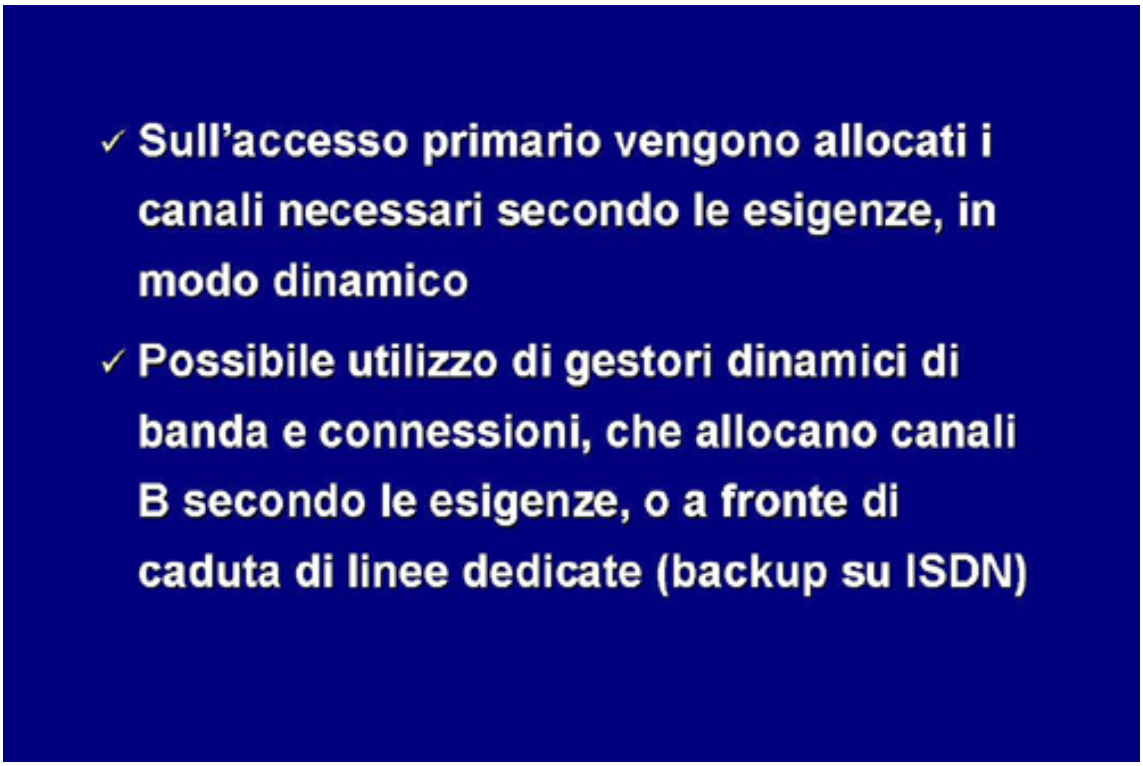
Accesso primario (4)



Accesso primario (4)

Nel momento in cui un interno solleva il ricevitore e, per esempio, preme il tasto zero per avere una linea esterna, automaticamente si alloca uno dei trenta canali ISDN. Nel momento in cui fa il numero per raggiungere un altro utente, la segnalazione del suo numero viaggia sul canale D del primario e quindi su quei 64Kbps del canale D che trasporta la segnalazione. Tutto questo dal centralino fino alla rete, cioè fino alla centrale telefonica. Quello che accade da lì in avanti sono fatti dell'operatore telefonico e non interessano a noi utenti, se noi ci mettiamo dalla parte degli utilizzatori.

Accesso primario (5)



Accesso primario (5)

Quindi, sull'accesso primario vengono allocati i canali necessari secondo le esigenze, in modo dinamico, come dicevo prima. Nel momento in cui uno schiaccia zero, automaticamente prende uno dei trenta canali oppure no. È prevista la possibilità di avere gestori di banda e connessioni che allocano canali B secondo le esigenze o a fronte di cadute di linee dedicate (*backup* su ISDN). Allora, io posso realizzare una connessione dedicata tra due siti, dopodiché ho anche una connessione ISDN. Nel momento in cui la connessione dedicata dovesse cadere per qualche ragione, perché ci sono dei guasti sulla linea, automaticamente questi apparati fanno il numero e quindi sostituiscono la connessione dedicata, che in questo momento è caduta, con una linea ISDN. Ovviamente, mentre la linea dedicata ha una tariffazione *flat*, nel senso che io pago un tot sia che la usi sia che non la usi, in questo caso, viceversa, la tariffazione è a tempo, seguendo le normali tariffe delle chiamate interurbane.

Applicazioni informatiche

Applicazioni informatiche

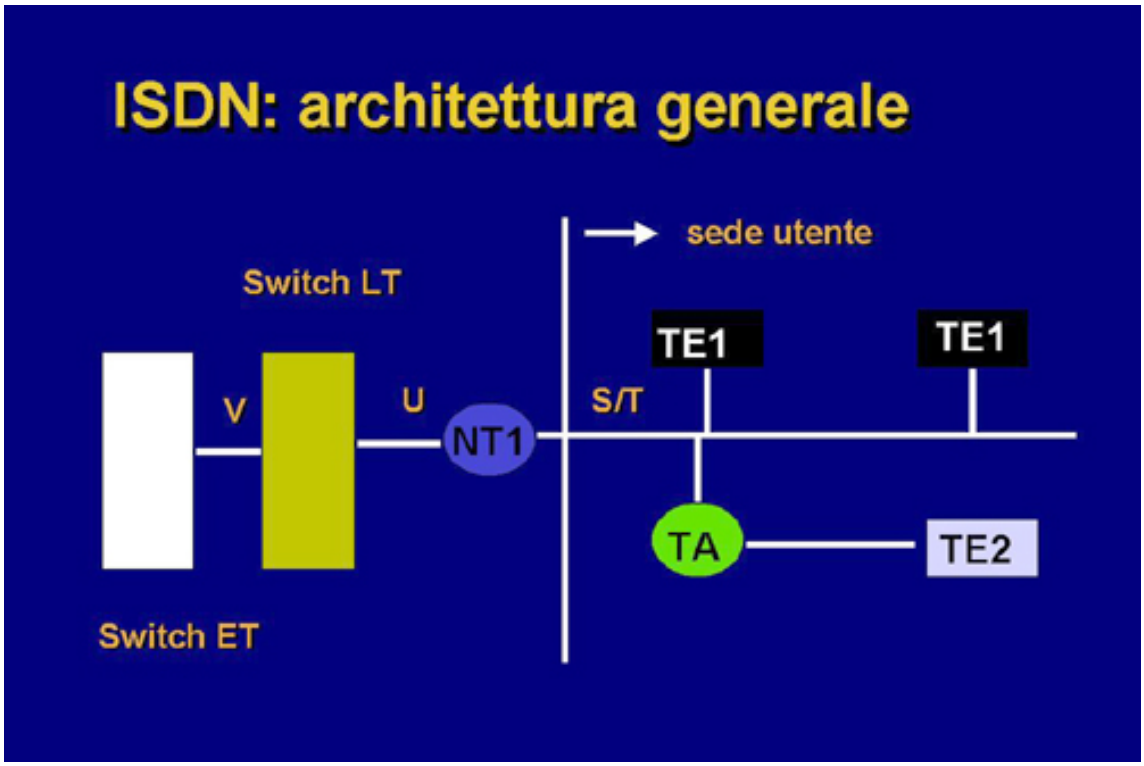
Vediamo quali sono le applicazioni informatiche. Prima di tutto ISDN non nasce per applicazioni informatiche, nasce per la voce e quindi, come poi vedremo meglio alla fine quando vi darò un cenno architetturale di come è fatta, e quindi mal si presta. Mal si presta forse è esagerato, ma diciamo che per poter essere utilizzata per la realizzazione di una rete di calcolatori, ISDN necessita di essere un attimino forzata. Il primo vantaggio, che si può vedere, è che usando ISDN nelle applicazioni informatiche non ho più le conversioni digitale/analogico e analogico/digitale, cioè non ho più bisogno dei modem. Perché ISDN trasmette in digitale, i calcolatori sono digitali e quindi non si deve più modulare né demodulare. Posso utilizzare le connessioni commutate rispetto alle connessioni dedicate in molti casi. Commutate, come vi dicevo, ha il vantaggio di costare poco, nel senso che pago solo quando uso. Mentre una linea dedicata, specialmente nel nostro paese, ha dei costi molto elevati. Va sempre diminuendo, perché l'autorità garante delle comunicazioni sta comunque costringendo gli operatori a rivedere le tariffe. Ma comunque, resta il fatto che qui in Europa, in Italia in particolare, le linee dedicate hanno dei costi proibitivi. Di conseguenza, il poter utilizzare linee commutate rende, di fatto, realizzabile tante reti che viceversa non potrebbero essere realizzate. Per questa ragione quest'ultima frase diviene possibile interconnettere singoli individui, stiamo parlando di un piccolo ufficio, per esempio, e piccole reti per le quali le soluzioni dedicate sono inutilizzabili. Penso ad esempio ai commercialisti. Voi pensate ad un ufficio di un commercialista che deve connettersi all'anagrafe tributaria o al ministero delle finanze. È chiaro che una linea dedicata non la può avere, perché il costo sarebbe elevato. È chiaro che una linea analogica, cioè una linea telefonica tradizionale, non sarebbe sufficiente a dare quelle caratteristiche di qualità necessarie. Allora, cosa rimane? Serve una linea digitale commutata, che sia da una parte digitale e dall'altra che costi poco, che quindi sia commutata.

Applicazioni

Applicazioni

Per cosa viene utilizzata? Quali sono queste applicazioni informatiche di cui stiamo parlando? Prima cosa, classica, la connessione *LAN to LAN* e *Host to LAN*. Allora, la connessione *LAN to LAN* è quella classica, che avviene quando ho due diverse reti locali situate in due uffici diversi e utilizzo degli apparati tipicamente dei *bridge* o dei *router*, cioè degli apparati di interconnessione fra le reti locali, che utilizzano come infrastruttura di trasporto ISDN. Nel caso *Host to LAN* uso i cosiddetti modem ISDN, che non sono dei modem, ma sono dei dispositivi che consentono comunque la interconnessione di un PC, per esempio, a una LAN remota. È chiaro che la connettività ha velocità superiore rispetto ai modem tradizionali, ma non così tanto perché noi sappiamo che oggi ci sono dei modem che riescono ad andare a decine di Kbps. Qui stiamo parlando di 64Kbps, quindi questa differenza non è così elevata. C'è però una grossa differenza proprio per quanto riguarda la segnalazione e quindi la realizzazione della connessione. In ISDN è molto rapida, quindi nel momento in cui viene fatta una chiamata ISDN, in pochi decimi di secondo effettivamente la connessione è su, e questo effettivamente consente di vedere, per esempio, le due LAN interconnesse in tempo reale. È chiaro che tutto questo può essere fatto anche con le tecnologie tradizionali, con le linee telefoniche classiche, ma con tempi di trasmissione e costi maggiori. I tempi di trasmissione sono maggiori nel caso delle linee telefoniche, mentre i costi risultano maggiori nel caso delle linee dedicate.

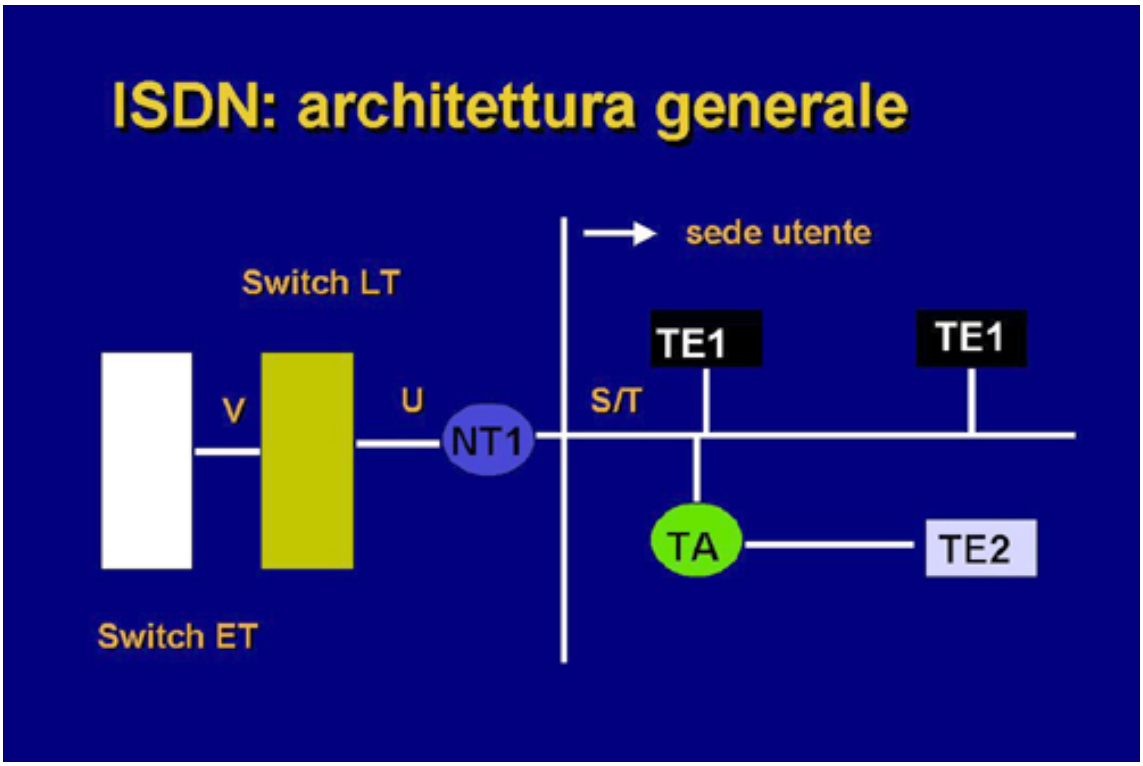
Architettura generale (1)



Architettura generale (1)

Guardiamo qualcosa di più specifico del ISDN. Noi abbiamo questa architettura generale. In questa architettura noi vediamo cosa? Vediamo la sede dell'utente, quindi questa è casa nostra, perché noi ci mettiamo dal punto di vista degli utilizzatori, e abbiamo la sede della compagnia telefonica. Quindi questa SAN, questa *Subscriber Access Network*, di cui parlavamo prima, è quella che sta tra noi e l'utente. Di fatto, è quella che comprende questo dispositivo e questo dispositivo. Allora, l'NT1, il *Network Termination One*, lo standard ISDN lo definisce NT1, altro non è che la borchia che la compagnia telefonica mi mette nel momento in cui io ordino un accesso ISDN. Loro vengono, mi mettono questa borchia nel muro, dove io attaccherò il cavo con il connettore RJ45, standard del livello fisico, giusto per fare un riferimento. Quindi NT1 non è altro che la terminazione della SAN a casa mia, della *Subscriber Access Network*.

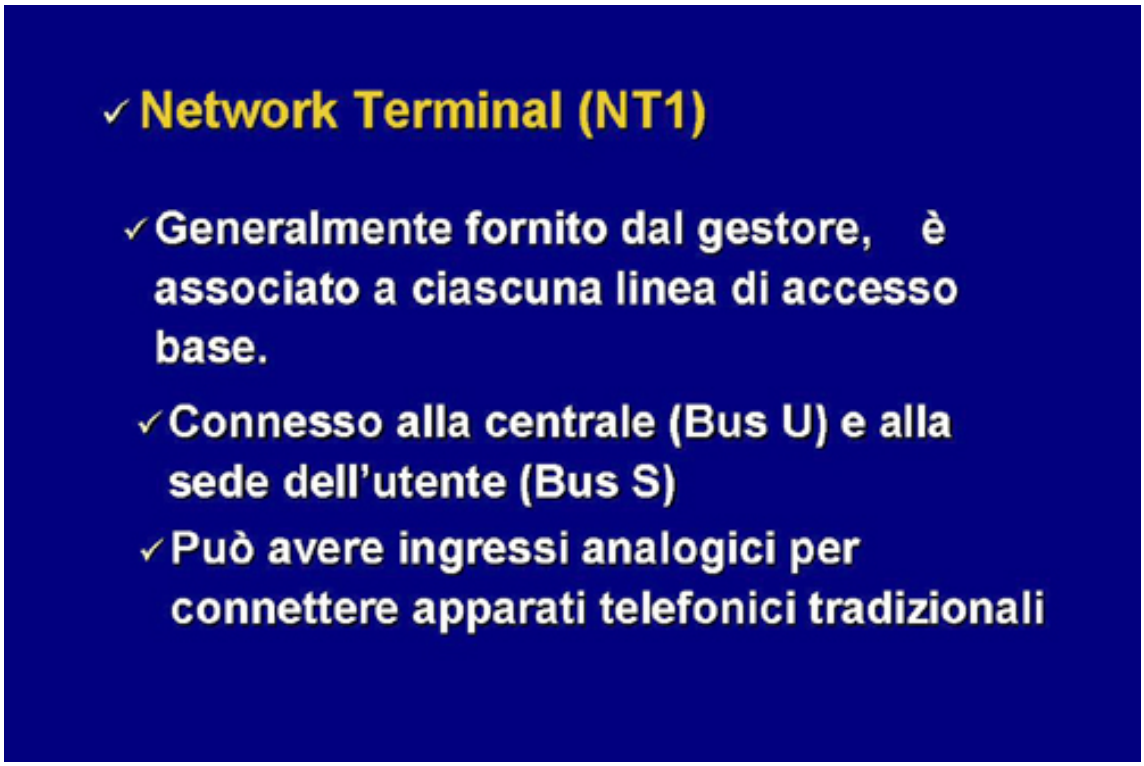
Architettura generale (2)



Architettura generale (2)

Da questa parte invece avrò la terminazione (*Line Termination*) e avrò la commutazione ISDN vera e propria. Da questa parte, che è quella che interessa a me, ho quello che viene chiamato punto o *bus* S o T. Qui posso interconnettere più apparati, che vengono chiamati *Terminal Equipment* o *Terminal End Point*. Questi sono i veri e propri apparati, ad esempio questo è un telefono, oppure questo è un calcolatore con una scheda ISDN, oppure ho un calcolatore che non ha una scheda ISDN ed è collegato, per esempio in seriale, ad un *Terminal Adapter* il quale è a sua volta collegato invece con lo standard ISDN al *bus* S o al *bus* T. In questa configurazione non è precisato, ma potrei anche avere un NT2 al posto di NT1, o meglio in aggiunta, che altro non è che un centralino telefonico, il quale all'interno si presenta sempre come ISDN. Oggi tutte le più grandi case che producono centralini telefonici, per esempio *Siemens*, Alcatel, eccetera, producono centralini che al loro interno sono compatibili con ISDN. Questa è la configurazione di riferimento ISDN dal lato utente, cioè dalla parte dell'utente.

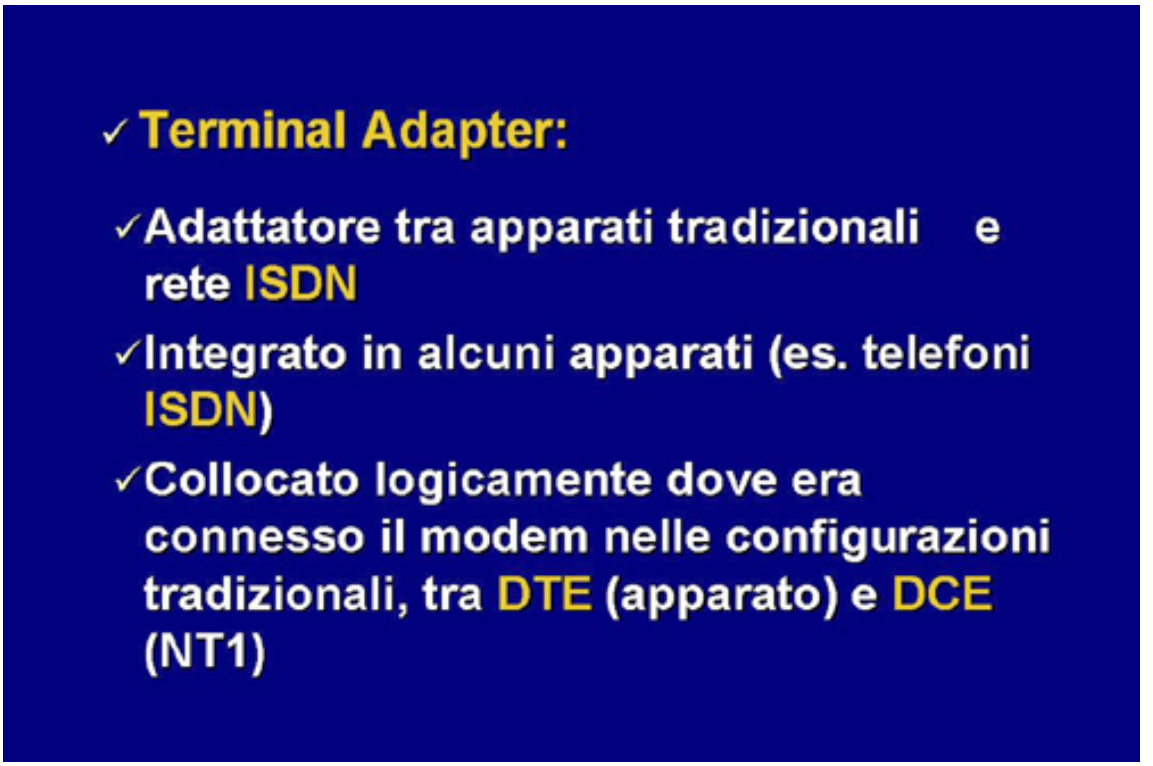
Network Terminal (NT1)



Network Terminal (NT1)

Andiamo rapidamente sulle attrezzature. Abbiamo detto che NT1 è fornito dal gestore e associato a ciascuna linea di accesso base; quindi, quando io metto la linea di accesso base, mi mettono la borchia che contiene la presa. Oltre ai punti S e T vengono definiti anche i punti *Bus* U e *Bus* S, che servono per l'interconnessione verso il gestore. Ci possono essere anche ingressi analogici per i telefoni tradizionali, questa opportunità è più un'offerta commerciale che non uno standard.

Terminal Adapter



Terminal Adapter

Il *Terminal Adapter* è quel dispositivo che consente di adattare apparati tradizionali, tipicamente seriali, a reti ISDN. In alcuni telefoni esso è già presente e sostituisce, almeno dal punto di vista della collocazione, il modem, mentre usando la rete telefonica normale si ha: il mio calcolatore, il modem e la rete telefonica. Usando una rete telefonica ISDN si ha: il mio calcolatore, il *Terminal Adapter* e la rete ISDN; è solo per questa simile collocazione che il *Terminal Adapter* viene chiamato modem ISDN, anche se in realtà non modula e non de-modula assolutamente, trattandosi sempre di reti digitali.

Segnalazione (1)

Segnalazione (1)

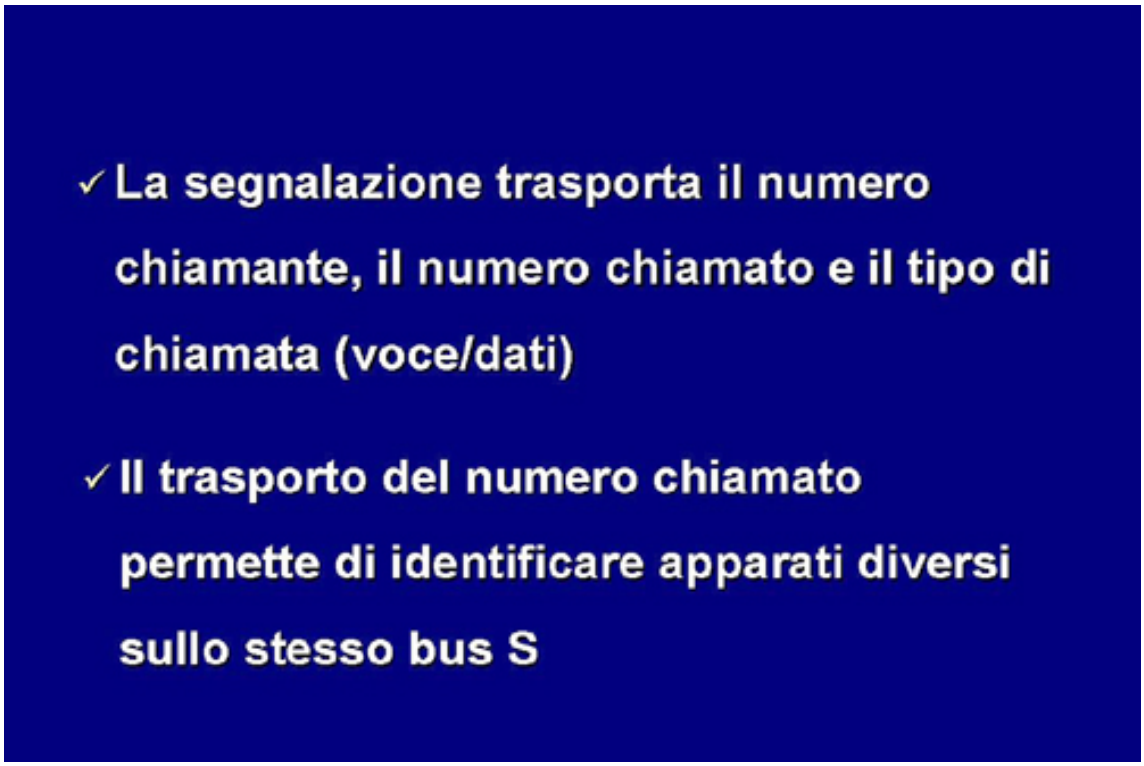
Veniamo ora alla segnalazione. Nella rete telefonica attuale la segnalazione viene fatta per via acustica secondo la codifica DTMF: nel momento in cui viene schiacciato un tasto della tastiera telefonica un particolare tono di una certa frequenza viene trasmesso alla rete telefonica. Di fatto, quindi, la segnalazione viene trasportata nel campo dell'udibile, infatti, nel momento in cui noi componiamo un numero telefonico, sentiamo le frequenze che stiamo trasmettendo. Si dice quindi che è una segnalazione in banda, perché fa parte dell'udibile.

Segnalazione (2)

Segnalazione (2)

Viceversa, ISDN usa per la segnalazione addirittura un canale diverso da quello utilizzato per parlare, cioè mentre la mia voce viaggia sul canale B, e quindi eventuali frequenze viaggiano al di sopra, sul canale D non viaggia la voce, ma soltanto la segnalazione. Quindi, nel canale D la rete invia un pacchetto verso la destinazione (*Out of Band Signalling*). Emerge per la prima volta un fatto curioso delle reti ISDN e cioè la parola pacchetto: in questo caso infatti viaggiano sul canale D dei pacchetti, cioè la segnalazione è gestita attraverso il *packet switching*, la commutazione di pacchetto, mentre invece la voce è gestita attraverso la commutazione di circuito. Questo significa che quando io parlo con un altro interlocutore, il circuito che trasporta la mia voce, da me all'interlocutore, lo fa attraverso risorse allocate staticamente e se anche io non parlo queste ultime restano comunque allocate. Vengono allocati, ad esempio, 64Kbit da me al ricevitore e se rimaniamo entrambi in silenzio li paghiamo lo stesso. In una rete ISDN, per la segnalazione non viene usata una commutazione di circuito, ma di pacchetto: ciò significa che sul canale D normalmente non viaggia nulla, nel momento in cui il terminale deve fare una segnalazione invierà dei pacchetti e poi di nuovo nulla. È quindi una gestione completamente diversa dalla precedente: la rete ISDN riesce ad esprimere molti dei concetti espressi nelle due lezioni introduttive, perché mischia la commutazione di circuito per i canali B con la commutazione di pacchetto sul canale D.

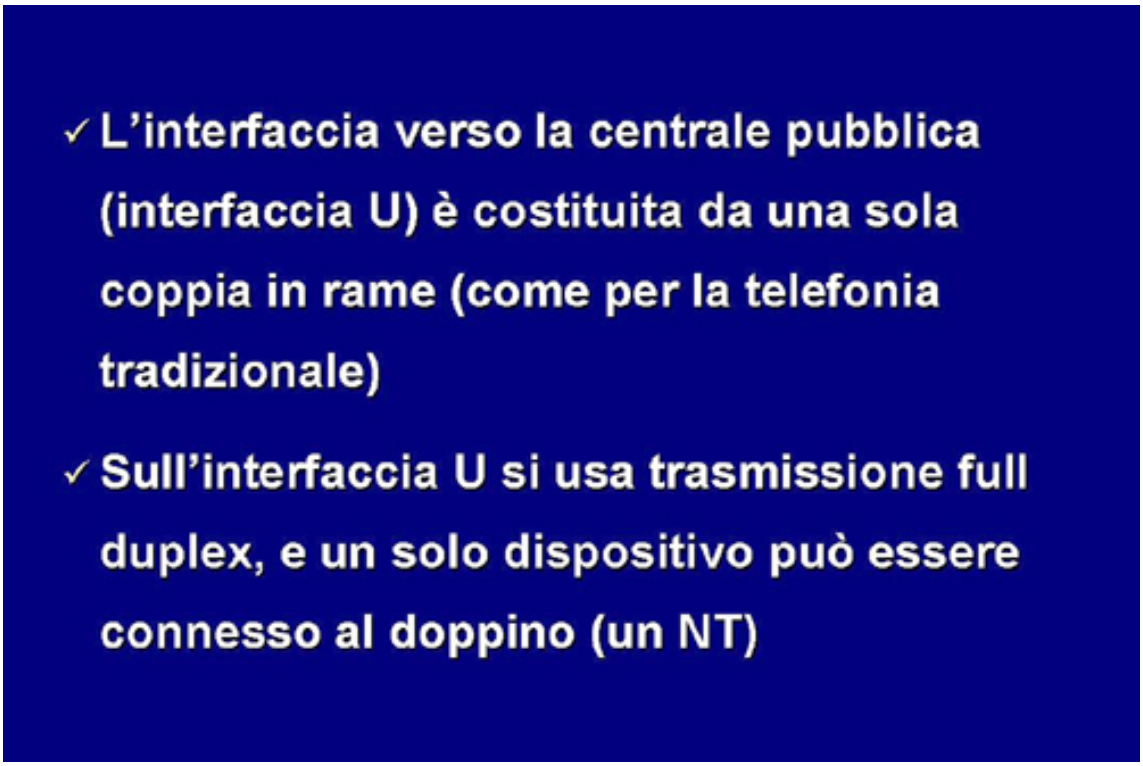
Segnalazione (3)



Segnalazione (3)

La segnalazione trasporta il numero chiamante, il numero chiamato ed anche il tipo di chiamata (voce/dati): se la chiamata è di tipo voce, cioè se il servizio che sto richiedendo alla rete è una trasmissione di voce, la rete potrebbe anche pensare di convertire il mio insieme di bit in voce e poi trasmetterlo come voce per poi riconvertirlo dall'altra parte, il che è ovviamente una forzatura. Come abbiamo detto, il numero chiamato consente di identificare più apparati sullo stesso *bus* S.

Interfacce (1)



Interfacce (1)

Parliamo ora delle interfacce. Abbiamo l'interfaccia verso la centrale pubblica: cioè, partendo dalla borchia sul muro, il punto da tale borchia (NT1) alla centrale pubblica viene chiamato U ed è costituito da una coppia di fili, come per la telefonia tradizionale. Questo significa che dal punto di vista dei cablaggi il gestore del telefono non deve installare nuovi fili per il cosiddetto ultimo miglio, nel senso che i cablaggi esistenti sono sufficienti per trasportare la rete ISDN. Sull'interfaccia U si usa la trasmissione *full duplex* e un solo dispositivo può essere connesso al doppino, proprio come nella telefonia tradizionale.

Interfacce (2)

Interfacce (2)

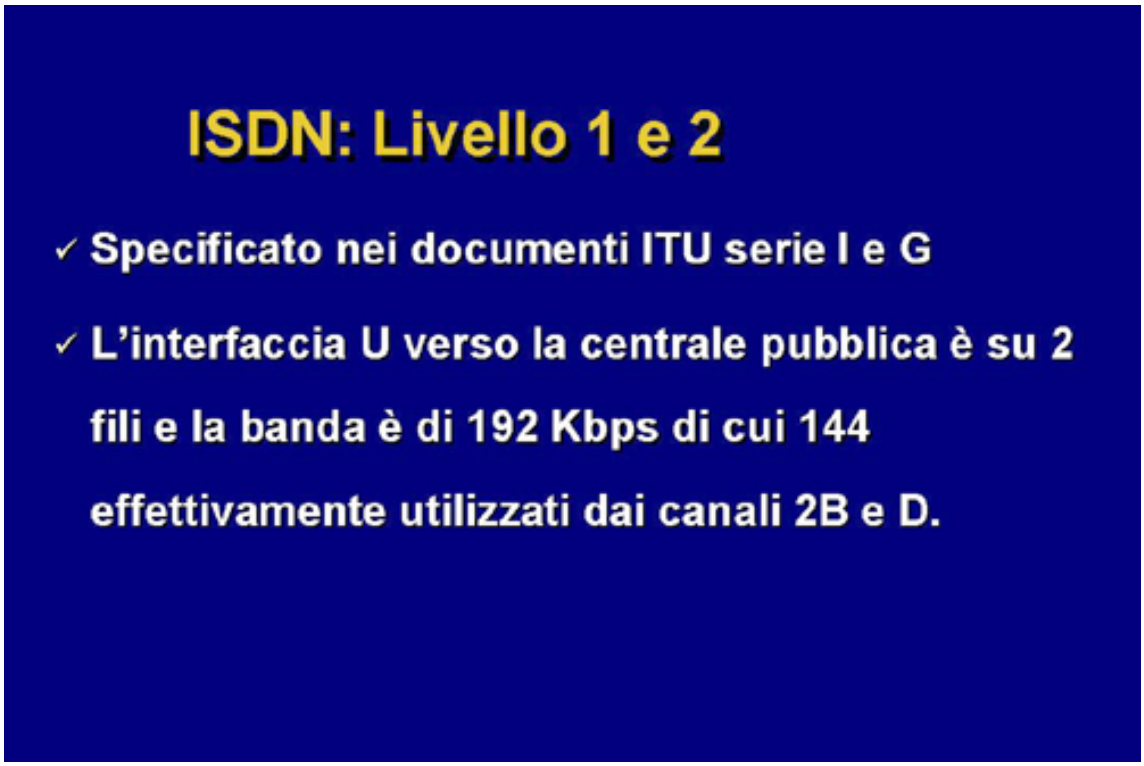
Il dispositivo NT, cioè la borchia attaccata al muro, converte, dietro al muro, questo standard a due fili passandolo a quattro fili, che è appunto lo standard ISDN che io vedo e viene chiamato *Bus* S/T. Quindi sui cavi, sui connettori di livello fisico, connettori RJ45 e cavi UTP usati per ISDN, si hanno quattro fili a partire dalla borchia sul muro verso l'interno, mentre verso la centrale pubblica ne restano due. Per ogni *Bus* S si possono avere fino ad otto fili, in tal caso si hanno due coppie di quattro fili, dove una è usata in trasmissione e l'altra in ricezione.

Interfacce (3)

Interfacce (3)

Gli apparati ISDN (telefoni, schede, *bridge*, *router*) richiedono tutti una connessione ad un *bus* S/T, cioè hanno lo standard pari alla connessione S/T.

Livello 1 e 2 (1)



Architettura generale 1 (1)

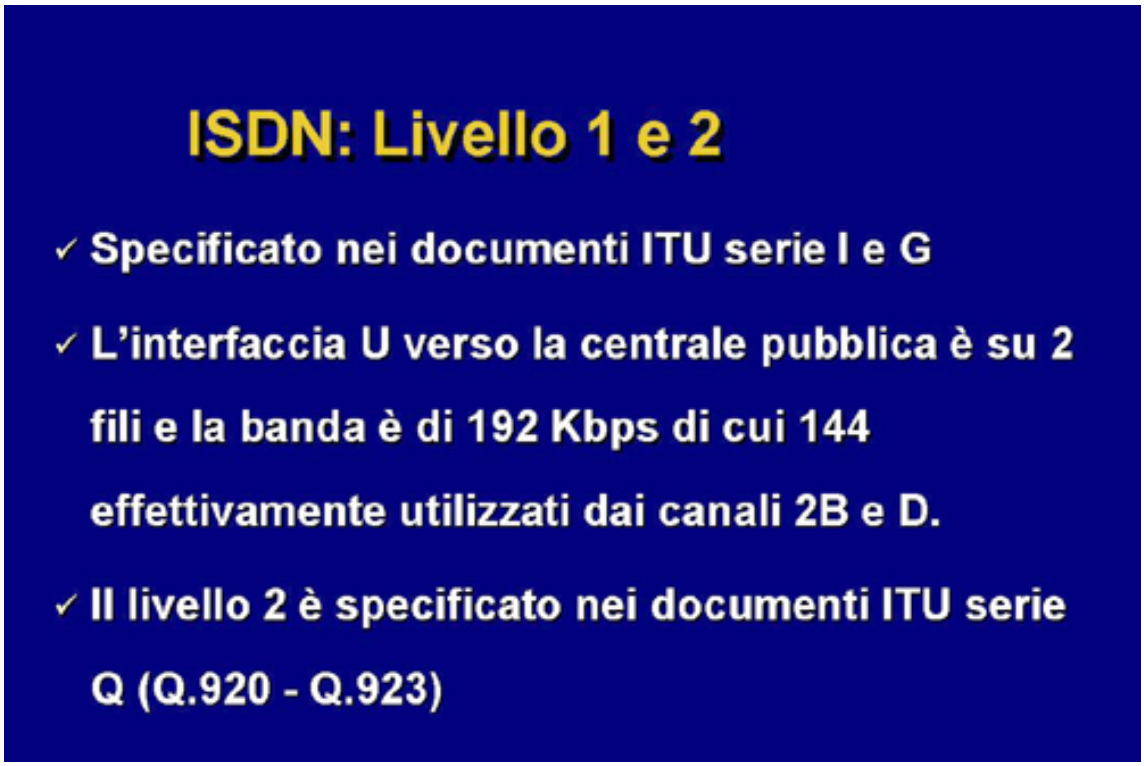
Vediamo ora rapidamente un riferimento dei livelli ISDN in relazione al modello OSI. ISDN copre tre livelli: il livello 1, il livello 2 ed il livello 3. Questi però non sono del tutto analoghi al modello OSI, proprio per tale motivo risulta molto interessante fare il raffronto tra i due modelli. Innanzitutto abbiamo l'interfaccia U verso la centrale pubblica su due fili e la banda è di 192Kbps, di cui 144 effettivamente utilizzati dai canali 2B e D. Il valore 144Kbit è ottenuto sommando i 64Kbps per ogni canale B e i 16Kbps del canale D (64+64+16=144); il *basic-rate* ha il valore di 192 kbps, anche se in realtà ne servono 144, perché a livello fisico sono necessari altri bit per la sincronizzazione e per il controllo di accesso al canale. Ci sono quindi dei bit, precisamente 48 (192-144), addizionali. Una volta che noi abbiamo su questi fili i 3 canali dobbiamo anche separarli, cioè dobbiamo prendere i nostri 3 canali e dividerli da un filo in tre parti.

Livello 1 e 2 (2)

Livello 1 e 2 (2)

Questo è essenzialmente il livello 1, che consiste nel mischiare i tre canali in maniera tale che su due fili viaggino tutti e tre; ci sarà poi dall'altra parte il livello fisico che li separa, quindi avremo tre canali diversi. Il livello 2, sempre specificato dai documenti ITU, è quello che si occupa della protezione dagli errori nelle reti normali. In questo caso il livello 2 serve per garantire una connessione priva di errori tra me e la centrale del gestore a cui sono collegato, quindi se c'è un errore di trasmissione da un punto all'altro direttamente connessi, il livello 2 lo recupera. Una cosa importante è che il livello 2 in ISDN esiste soltanto nelle trasmissioni sul canale D, per cui se c'è un errore su tale canale che porta la segnalazione, questo viene recuperato, se invece c'è un errore sulla voce, questo non è oggetto di controllo di errore, per cui i canali B, che sono a commutazione di circuito, non vedono il livello 2, che serve soltanto per la segnalazione.

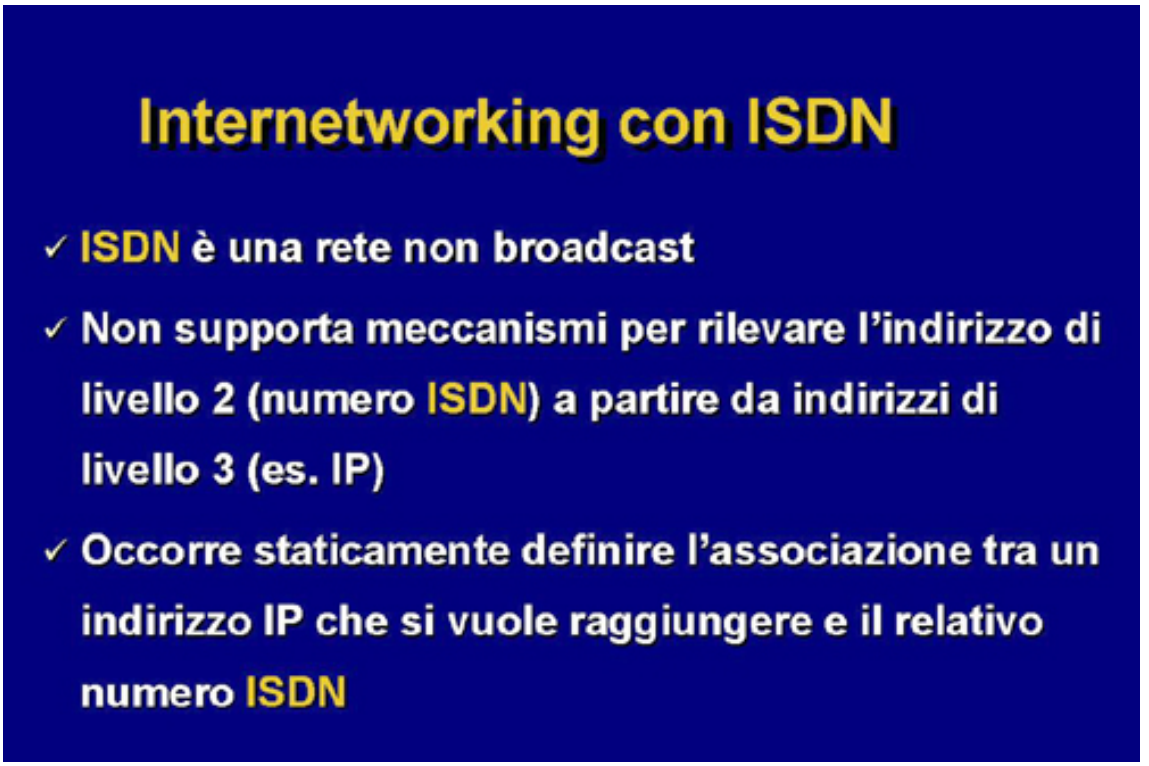
Livello 1 e 2 (3)



Livello 1 e 2 (3)

Infine arriviamo al livello 3, che è quello applicativo, cioè porta sostanzialmente informazioni legate al *setup* di una chiamata. Nel momento in cui un dispositivo vuole fare una chiamata manderà un messaggio di livello 3, questo viene incapsulato in un messaggio di livello 2 che viaggia sul canale D ISDN e arriva dall'altra parte, dove viene interpretato e ad un certo punto, se viene deciso che la comunicazione può essere fatta perché il ricevitore è libero, viene allocato un canale B. Su quest'ultimo non c'è né il livello 2 né il livello 3, ma è uno *stream* di dati, una fila di bit pura e semplice.

Internetworking con ISDN



Internetworking con ISDN

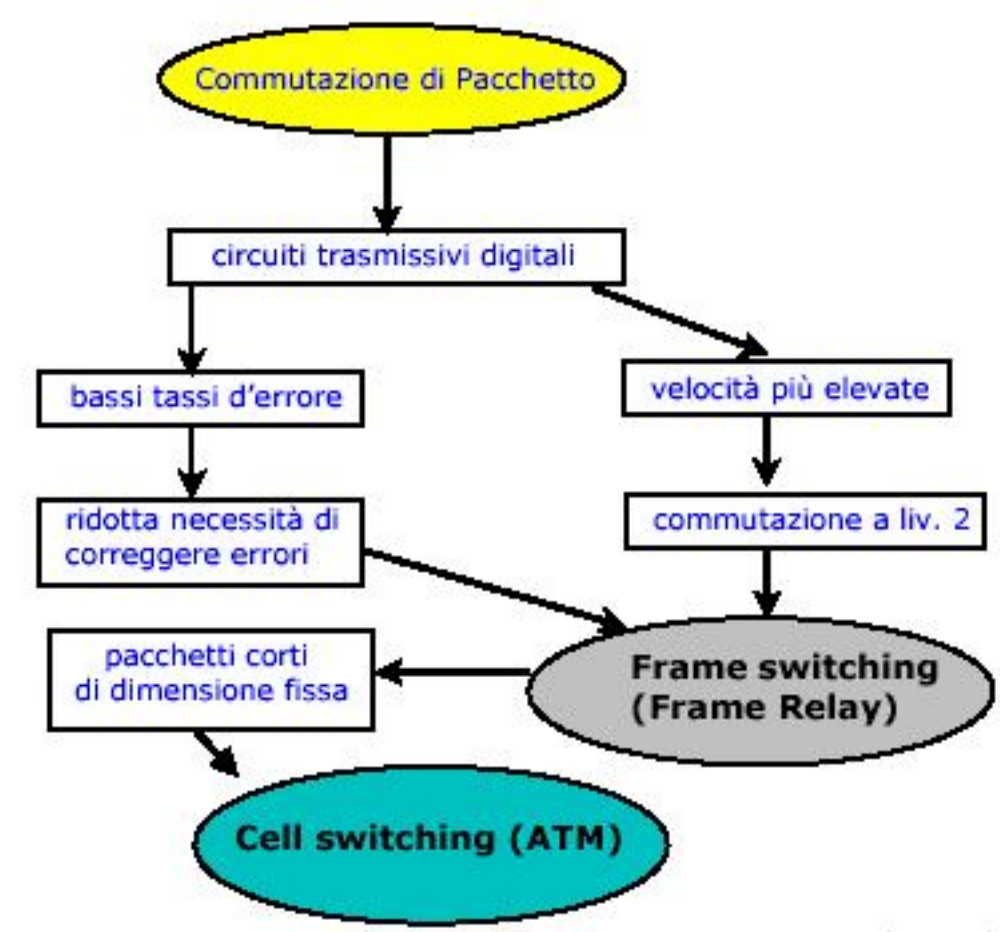
Ultima informazione che vorrei dare è l'*Internetworking* con ISDN, cioè cosa accade quando si vuole interconnettersi ad una rete IP, sia essa Internet o una Intranet o ancora una Extranet, utilizzando ISDN. Innanzitutto ISDN è una rete non *broadcast*, cioè una rete punto a punto come il telefono, quindi può e deve essere utilizzata allo stesso modo del telefono. Non supporta meccanismi per rilevare l'indirizzo di livello 2 (numero ISDN) a partire da indirizzi di livello 3, ciò vuol dire che nel momento in cui si vuol realizzare una rete IP su ISDN, utilizzo ISDN ad un livello sottostante e non c'è integrazione tra le due, cioè occorre staticamente definire l'associazione tra un indirizzo IP che si vuole raggiungere e il relativo numero ISDN. I *router*, quindi, nel momento in cui si deve realizzare il *routing* di un pacchetto, avranno cablata nelle loro tabelle di *routing* la particolare chiamata da effettuare per raggiungere quel particolare indirizzo. Questo meccanismo è ovviamente molto più complesso di come vi ho accennato, ma quel che mi importava sottolineare è la scarsa o praticamente nulla integrazione tra ISDN e IP. In sintesi, ISDN nasce come evoluzione della tradizionale rete telefonica. Il grande vantaggio è che è digitale, quindi molto rapida, per questo si presta ad essere utilizzata come infrastruttura anche per la realizzazione di reti basate su IP, oltre che per la telefonia classica.

Frame Relay

Franco Callegati
Paolo Zaffoni

8.2.1 (Distinguere tra opzioni basate su router, su switch e su bridge)

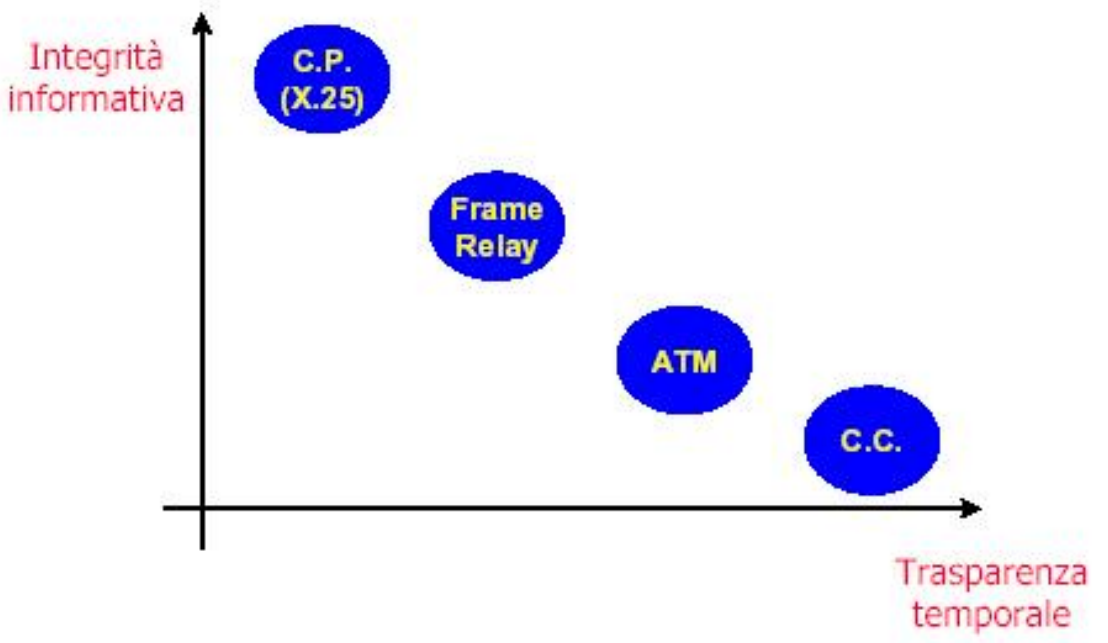
Evoluzione della commutazione di un pacchetto



Evoluzione della commutazione di un pacchetto

Modi di trasferimento

- *Frame relay*;
- ATM.



Modi di traferimento

Nel corso degli ultimi dieci anni si è assistito ad una visibile evoluzione nel settore dell'*information technology* e nel contesto delle telecomunicazioni.

Le reti di telecomunicazioni hanno adottato in maniera sempre più diffusa tecnologie trasmissive affidabili e ad alta capacità (fibre ottiche) e la capacità di elaborazione dei dispositivi di utente è andata progressivamente crescendo, consentendo all'utente l'impiego di dispositivi intelligenti (*router*, *bridge*, eccetera) e di *software* in grado di fornire teleservizi affidabili ad alte prestazioni.

Tutto ciò ha reso possibile la migrazione di alcune funzioni che in passato erano tipiche della rete, verso le apparecchiature di utente, riducendo in tal modo la complessità delle funzioni svolte dai nodi della rete e consentendo a questi ultimi di operare su capacità trasmissive sempre più alte.

È in un contesto come questo che è stato possibile lo sviluppo di tecniche come il *Frame Relay* e l'ATM, che sono caratterizzate da alte velocità e da semplificazioni protocollari, unite ad evoluti aspetti architettuali e tecnologici dei nodi di rete.

Definizione di Frame Relay

Frame relay è un modo di trasferimento dell'informazione:

- Schema di multiplazione.
- Tecnica di commutazione.
- Architettura dei protocolli.

Come tutti i modi di trasferimento, anche *frame relay* comprende le componenti di multiplazione, commutazione, profilo dei protocolli.

Lo schema di multiplazione non prevede una suddivisione della capacità trasmissiva in

slot, utilizza una modalità asincrona di assegnazione della capacità ai tributari che condividono la banda e si basa sull'utilizzo di un'etichetta (impropriamente riferita come *ADDRESS*).

La commutazione avviene con il meccanismo di *store and forward* e si basa sull'etichetta delle trame dati. Un nodo *frame relay*, ricevendo una trama da una porta di ingresso, analizza la presenza di errori e, se la trama è valida, legge il valore dell'etichetta (in particolare, il numero di canale logico), interroga una tabella di *look-up*, preliminarmente configurata dal gestore del servizio, e preleva dalla tabella stessa il link di uscita su cui deve proseguire la trama ed il numero di canale logico che la trama deve avere uscendo dal nodo; quindi ricalcola il codice di protezione da errori e trasferisce la trama sulla porta di uscita.

L'architettura dei protocolli risulta particolarmente semplificata, poiché *frame relay* si basa su una struttura di livello *data-link* ed impiega una parte (detta *core*) delle funzioni di strato *data-link*. Ciò rende il trasferimento dei dati inaffidabile, nel senso che non sono previste, da parte della infrastruttura geografica, funzioni di riscontro delle trame ricevute e procedure di ritrasmissione delle trame corrotte. Se il livello fisico è di buona qualità, implicitamente le funzioni di livello 2 possono essere semplificate perchè la probabilità di errore è praticamente nulla. In caso di errori, saranno i protocolli a bordo degli *end-system* a recuperare i dati persi o scartati.

Frame relay è anche:

- Una tecnologia.
- Un protocollo.
- Un servizio *WAN*.
- Un'interfaccia.

Frame relay non è una rete:

- Se si fa riferimento al termine rete, occorre dare il giusto peso al termine (si intende l'infrastruttura tecnologica e non il piano di indirizzamento).

Frame relay è una tecnologia perchè non impiega (almeno nella maggior parte delle implementazioni mondiali) uno schema di indirizzamento di livello 3. Le funzioni di indirizzamento che necessitano e che sono funzionali all'istradamento dei dati dal mittente al destinatario, vengono assolte dal livello 2, mediante le etichette previste.

In una rete di *computer*, infatti, è identificabile una funzione centralizzata di amministrazione degli indirizzi (come avviene, ad esempio, in una rete IP, privata o pubblica). La gestione degli indirizzi in una infrastruttura *frame relay* non è presente, poiché gli indirizzi non vengono utilizzati.

Frame relay è un servizio, dal momento che molti *carrier* lo offrono da diversi anni su base geografica per la trasmissione di dati a medie velocità (da 64 kbit/s a 1048 Mbit/s)

Frame relay è un protocollo, dal momento che prevede una serie di regole per la trasmissione delle informazioni e prevede un formato per i dati.

Frame relay è un'interfaccia, dal momento che specifica un protocollo di accesso; non impone regole per la realizzazione delle sezioni interne della rete. Infatti, le infrastrutture di molti *carrier* impiegano sugli accessi il *frame relay* come protocollo per l'utente e impiegano protocolli proprietari tra i nodi.

Frame relay è un'interfaccia anche per il seguente motivo. Un terminale di utente (*router*) accede al servizio *WAN Frame Relay* mediante interconnessione di una sua porta fisica (seriale) ad una terminazione della rete del *carrier* (costituita da un DCE). La stessa porta fisica del *router* può essere configurata per accedere ad una rete puramente trasmissiva (impostando il protocollo di livello 2 come PPP oppure un altro proprietario), oppure ad un servizio *frame relay* (impostando l'incapsulamento *frame relay* a livello 2), oppure ad una rete X.25 (impostando LAP-B come protocollo *data link*), o altro.

Origini del Frame Relay

Il *Frame relay* nacque pensando alla possibilità di poter disporre di un protocollo di rete geografica che fosse intermedio alla commutazione di pacchetto X.25 e a circuito.

Lo sviluppo si è avuto sotto la spinta di precise forze di mercato:

- sviluppo di applicazioni ad alta velocità.
- Dispositivi di utente intelligenti.
- Linee di trasmissione con bassi tassi di errore.

Ad oggi rappresenta il principale servizio di accesso in reti pubbliche di molti operatori internazionali.

A seguito della numerizzazione delle reti e con l'avvento di moderne tecnologie di trasmissione su fibra ottica, che garantiscono tassi di errore estremamente bassi sui bit trasmessi, l'ipotesi di avere a disposizione mezzi trasmissivi di qualità non elevata, che era alla base di X.25, è oggi sempre meno realistica. Queste considerazioni hanno portato a studiare nuove tecniche a commutazione di pacchetto e nuovi protocolli di interfaccia per l'utente.

Caratteristica comune di queste nuove tecniche è quella di **non prevedere alcuna procedura di recupero degli errori nei nodi interni della rete**; ciò permette di avere velocità di accesso alla rete molto alte e ritardi di attraversamento dei nodi molto bassi. Gli esempi più significativi di tali nuove tecniche sono il *Frame Relay* e l'ATM.

Caratteristiche principali del Frame Relay (1)

- Tecnica di trasferimento orientata al pacchetto (basata su tecniche di multiplazione di **pacchetti di lunghezza variabile**).
- Non nacque come protocollo di dati indipendente, ma venne definito originariamente come servizio dati in ambito ISDN.
- È stato definito per l'accesso (UNI), ma può essere impiegato nelle sezioni interne della rete (NNI).
- Offre un servizio permutato (pvc), non fornisce un servizio commutato.
- Mantiene i vantaggi dell'X.25 **semplificando i protocolli, diminuendo il ritardo, aumentando il *throughput***.

Il termine *Frame Relay* viene usato per indicare il protocollo, l'interfaccia, la tecnologia, il servizio; di fatto è un modo di trasferimento relativo ad un servizio portante, il *Frame Mode Bearer Service*.

Appena diffuso commercialmente, il servizio FR è stato disponibile su scale

geografiche limitate e in ambito di rete monogestore.

Lo scenario della standardizzazione vede attualmente protagonisti l'ITU, l'ANSI e il *Frame Relay Forum*; quest'ultimo è un ente non profit, nato nel 1990 per iniziativa della cosiddetta *gang of four* (*Cisco*, *Digital*, *Nortel*, *Stratacom*), al quale aderiscono oggi numerosi altri costruttori, *carrier*, enti di ricerca.

In tempi più recenti si è delineata la necessità di interconnettere *frame relay* di gestori diversi e di definire le funzioni ai confini tra le reti. A tale scopo, è stata definita in ambito ANSI T1.606b (1993), e successivamente in ambito ITU-T X.76, l'interfaccia tra reti (NNI) comprendente le procedure di dialogo fra interfacce di differenti gestori.

Sono, quindi, oggi disponibili 2 interfacce, quella fra utente e rete (UNI) e quella tra reti (NNI). La differenza principale è sulla simmetria o meno del dialogo di controllo. In particolare, se alla UNI (pur essendo opzionale il *polling* simmetrico) è di fatto sempre l'utente che interroga la rete (*Status enquiry*), ed è la rete a rispondere (Status), alla NNI entrambe le interfacce devono interrogarsi reciprocamente. L'*implementation agreement* del *Frame Relay Forum* FRF.2, si basa quest'ultimo concetto.

Nella versione corrente dell'accordo implementativo (FRF.2.1) il campo di indirizzo è stato esteso a 4 *byte* con un DLCI di 17 *byte* ed un campo di controllo di 8 bit per migliorare le prestazioni gestionali. Inoltre il *polling* non è su base *timer* ma asincrono, ed è inviato solo su necessità; per esempio un messaggio di Status viene inviato in modo asincrono in seguito ad eventi particolari e relativamente ai PVC che hanno subito modifiche. Ciò consente un impiego ottimale della capacità numerica alla NNI. Il supporto del servizio commutato alla NNI resta una questione ancora aperta.

La filosofia che sottende al FR è quella di trasferire le informazioni con minori elaborazioni e funzionalità nei nodi:

- Assenza di controllo di flusso e correzione di errori in rete.

Il risultato è:

- *Throughput* molto più elevati e ritardi minori dell'X.25.
- Efficiente condivisione di banda (Gestione di traffico *busrty*).
- Garanzia di banda alla UNI.
- Multiplazione a livello 2 OSI e trasparenza verso i livelli superiori.
- Assenza di elaborazione a livello 3 OSI.
- Standard consolidati.

A differenza del protocollo X.25, nel *Frame Relay* il livello 3 viene eliminato, mentre il livello 2 viene semplificato ed arricchito delle funzioni di multiplazione e commutazione. Ciò permette di realizzare un servizio di trasmissione dati a velocità più elevata, fino a 2 Mbit/s. (In Italia l'accesso *frame relay* è limitato a 2048 kbit/s; in altri paesi del mondo altri *provider* offrono servizi *frame relay* anche su accessi a 45 Mbit/s e 155 Mbit/s).

Le trame *Frame Relay* vengono multiplate statisticamente per mezzo di un campo etichetta che contiene l'indicazione del circuito virtuale a cui la trama è riferita. La commutazione viene attuata con un cambio di link fisico ed un cambio di etichetta, in base al contenuto della tabella di instradamento (*look-up*) che ciascun nodo contiene.

Servizio Frame Relay End-to-End

Il servizio trasferimento dati *Frame Relay* ha le seguenti caratteristiche:

- preserva l'ordine delle unità dati trasmesse (sotto forma di trame) attraverso un'interfaccia utente-rete quando queste unità vengono consegnate all'altra estremità;
- trasporta trasparentemente i dati d'utente:
  - (soltanto i campi etichetta e controllo di sequenza bit (FCS) della trama sono modificati dalla rete);
- rileva errori di trasmissione, di formato ed operazionali (Esempio: riconoscimento di connessione virtuale non assegnata);
- non effettua nessun riscontro dei dati ricevuti.

La tecnica *Frame Relay* è connection oriented, per cui preserva per sua natura la sequenza delle trame trasmesse.

La tecnica *Frame Relay* non garantisce però che la sequenza sia completa, in quanto, in caso di trama con errore, la trama viene scartata dal nodo che rileva l'errore senza che venga fornita notifica all'estremità trasmittente. La funzione di riscontro è demandata a protocolli di livello superiore, rispetto ai quali la rete *frame relay* è trasparente.

Possibili applicazioni

- applicazioni (per elaborazione dati) richiedenti l'interconnessione su base geografica (altre reti, per esempio ITAPAC, ISDN, CDN poco adatte, per costi oppure velocità oppure flessibilità);
- interconnessione geografica di apparati per l'*internetworking* (raramente *bridge*, spesso *router*);
- ... e inoltre trasporto di fonia.

La disponibilità di servizi *frame relay* di rete pubblica è oramai una realtà da diversi anni presso molti *carrier* che operano su mercati nazionali ed anche a livello globale. I costruttori di apparati e sistemi hanno applicato le tecnologie VLSI per realizzare prodotti specifici per telecomunicazioni, ed usufruendo delle economie di scala hanno offerto tali prodotti a costi sempre più convenienti per l'utente finale.

I gestori hanno saputo cogliere l'opportunità di questa tecnologia, disponibile sul mercato degli apparati e sistemi per offrire a clienti medio/grandi servizi di trasporto dati ad alta velocità su base geografica sempre più estesa.

La tecnologia attuale mette a disposizione interfacce ed apparati *frame relay* operanti a velocità che vanno da 64 kbps fino a 34 Mb/s ed oltre. La tecnica *frame relay* è stata applicata recentemente con successo anche a sistemi trasmissivi sincroni a 155 Mbps negli USA.

Svantaggi del frame relay

- Impossibilità di buon funzionamento in assenza di terminali di utente intelligenti o di linee trasmissive di buona qualità.
- Funzionamento adeguato ma non ottimizzato nel trasporto di applicazioni dati tradizionali (non LAN).

- Particolare criticità nel trasporto di applicazioni *delay-sensitive* (fonia e videoconferenza).

Il *frame relay* è una tecnica di trasmissione pensata espressamente per i dati, in alternativa rispetto alle tecniche tradizionali (X.25) disponibili alla fine degli anni '80. Le funzionalità che non include, e che fanno parte di un tradizionale protocollo di livello *data link*, hanno senso in uno scenario di rete trasmissiva numerica di bassa qualità.

Le moderne reti di TLC sono supportate da tecniche trasmissive numeriche che impiegano largamente fibre ottiche, per cui è giustificabile l'eliminazione nei nodi di certe funzioni legate al recupero degli errori trasmissivi.

Per quanto riguarda l'impiego del *frame relay*, già da alcuni anni sono disponibili sul mercato diversi prodotti in grado di integrare traffico dati e voce su connessioni *frame relay* (*Frame Relay Access Device*, ovvero *Frame Relay Assembler Disassembler* FRAD); tali apparati consentono di collegare due isole remote fra loro sulla stessa connessione o su connessioni logiche diverse apparati di *internetworking* (*router*, *bridge*) e centralini privati per fonia.

Il terminale *frame relay* più diffuso è il *router*.

Stato degli Standard

Il *Frame Relay* gode di uno stato di standardizzazione molto solido e ben recepito dalle diverse manufatturiere nei suoi aspetti principali.

Il fatto di lasciare opzionali alcuni servizi ne ha favorito l'implementazione.

L'architettura del protocollo prevede due piani operativi separati:

- *Control Plane* (*C-Plane*).
- *User Plane* (*U-Plane*).

C-Plane:

- responsabile dell'instaurazione, mantenimento e rilascio delle connessioni logiche.

U-Plane:

- responsabile del trasferimento dati tra utenti in modalità *end-to-end*.

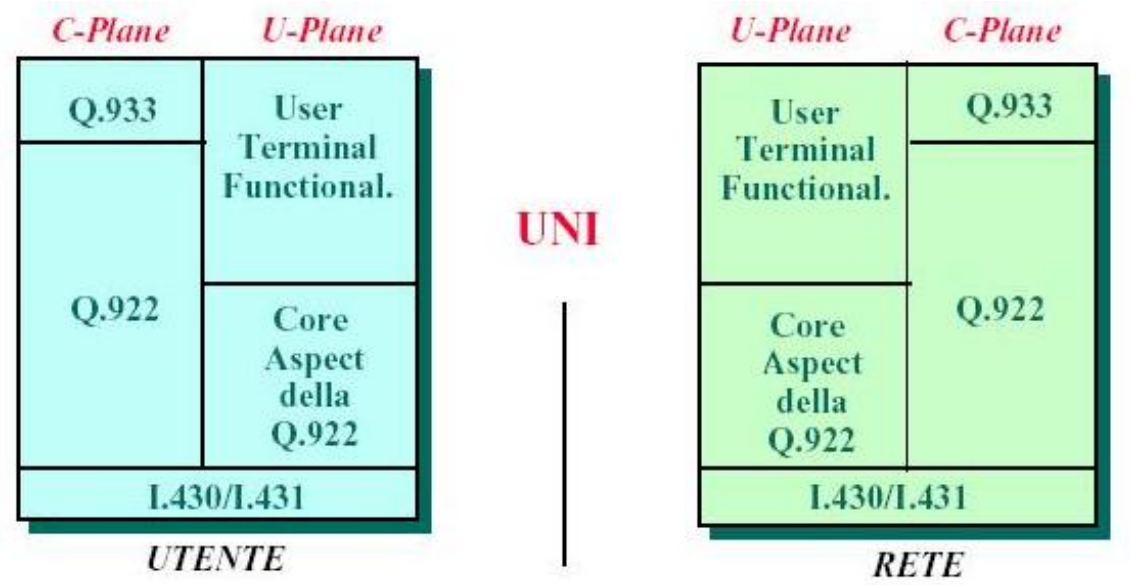
Inizialmente il *Frame Relay* è stato definito come un servizio a pacchetto in ambito ISDN, in cui è prevista una netta separazione tra il piano di controllo, che provvede alla gestione della segnalazione per la formazione e l'abbattimento delle connessioni, ed il piano di utente, che provvede alla gestione della fase dati. Nelle reti a pacchetto X.25 non vi è alcuna separazione tra il piano di controllo ed il piano di utente.

Nell'ISDN la separazione tra il piano di controllo ed il piano di utente è fondamentale per la definizione dei nuovi servizi a pacchetto. Tale separazione può essere realizzata sia utilizzando canali fisici e logici differenti per la segnalazione e per i dati di utente, sia utizzando solo canali logici differenti all'interno dello stesso canale fisico.

Per il servizio di circuito virtuale commutato SVC (*Switched Virtual Connection*) sono necessari sia i protocolli del piano di controllo che i protocolli del piano di utente.

Per il servizio di circuito virtuale permanente PVC (*Permanent Virtual Connection*) servono solo i protocolli del piano di utente, poiché il circuito virtuale viene realizzato con procedure di operatore al momento della sottoscrizione del servizio.

Architettura del Protocollo



Architettura del Protocollo

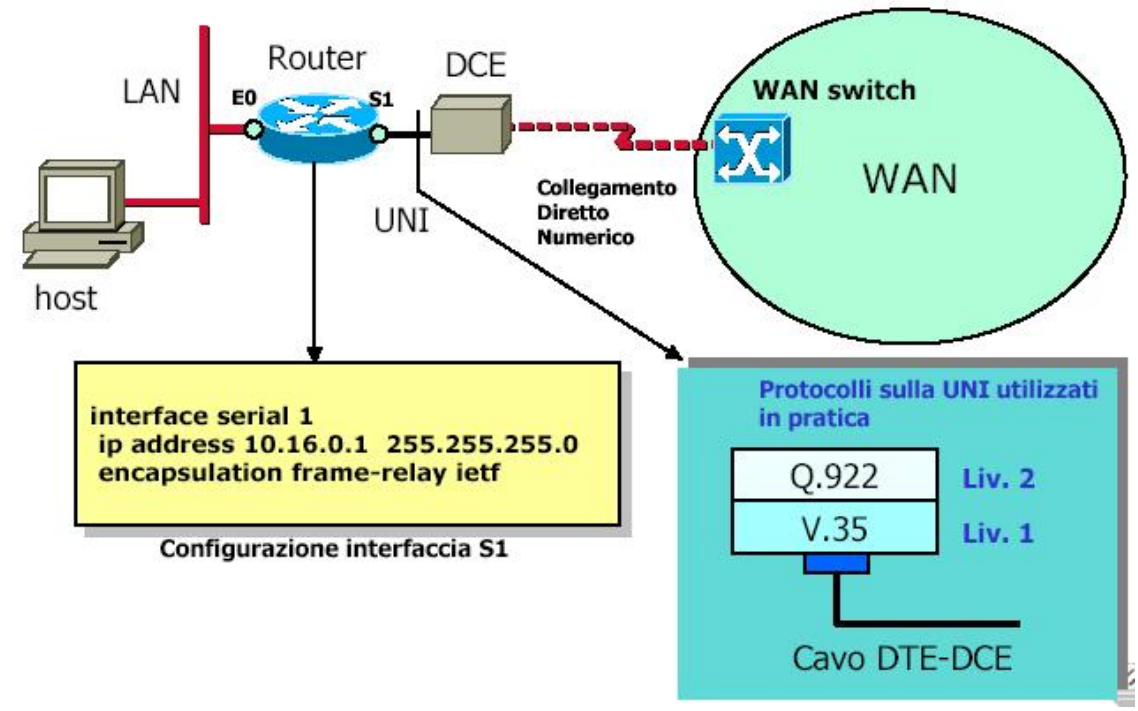
I protocolli di livello fisico sono gli stessi utilizzati in ambito ISDN, anche se molti gestori di reti pubbliche hanno recepito lo standard in maniera parziale. In particolare il livello fisico che viene utilizzato per l'acceso alla rete è quello che viene impiegato per i circuiti diretti numerici (CDN).

Lo standard Q.922, che specifica il *data link layer protocol and frame mode bearer services*, si basa sullo standard CCITT Q.921 LAPD (*Link Access Procedure on the D-channel*) e lo estende, formando il LAPF (*Link Access Procedure to Frame mode bearer services*).

Il protocollo LAPF è suddiviso in due parti:

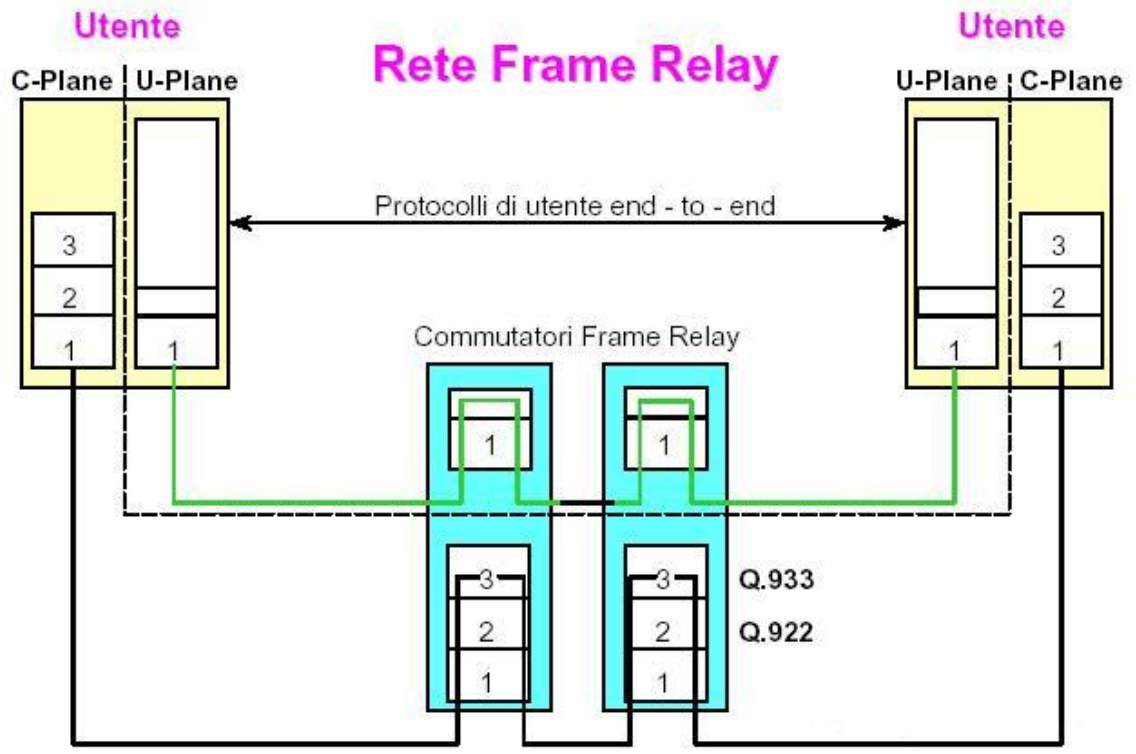
- ***DL-CORE*** (*Data Link Core protocol*) definito dalla raccomandazione CCITT I.233.
- ***DL-CONTROL*** (*Data Link Control protocol*), la rimanente parte di LAPF.

Interfaccia Frame Relay



Interfaccia Frame Relay

Profilo del protocollo



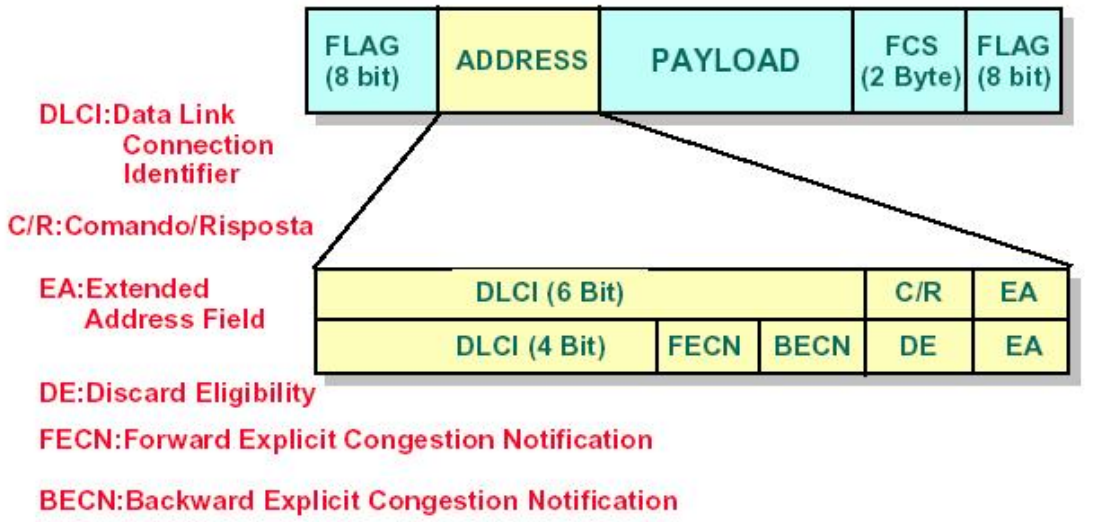
Profilo del protocollo

Il piano controllo è responsabile dell'apertura, della supervisione e del rilascio delle connessioni dati. Non sono indicati nello schema i componenti *software* che realizzano le funzioni di *call control* e che operano sia sui terminali di utente, sia sui nodi di rete.

Uno scenario in cui il *frame relay* offra un servizio SVC (*switched virtual connection*) rappresenta una soluzione in cui *frame relay* può essere considerato una rete, poiché, per poter effettuare chiamate, è necessario gestire un piano di numerazione e di indirizzamento. Tale schema di numerazione e indirizzamento è necessario per fornire agli utilizzatori connettività *any to any*.

Uno scenario in cui *frame relay* offre come servizio un collegamento logico permanente (PVC) preconfigurato e statico, consente di interconnettere un insieme limitato di terminali, ed in particolare, quelli che fanno parte del dominio dell'utente; la soluzione che si delinea è quella di una rete privata virtuale, in cui le comunicazioni sono possibili solo fra terminali appartenenti al gruppo ed i canali dati sono sempre disponibili, senza che sia necessario richiedere alla rete la loro impostazione.

Il protocollo LAPF: formato dei dati



Il protocollo LAPF: formato dei dati

Il LAPF è un protocollo completo di livello 2 e consta di due sottolivelli, denominati rispettivamente Q.922 *core* e Q.922 *upper*. **Il servizio di tipo PVC utilizza solamente il sottolivello *core* del LAPF**.

Come tutti i protocolli sincroni, la trama è delimitata da una *flag* (01111110) che può essere trasmessa continuamente, in caso di inattività della linea, per mantenere attivo il sincronismo con il nodo di accesso (*keep alive sequence*). Per evitare che nel campo dati venga simulata la *flag*, in trasmissione, prima di inviare la trama, per ogni sequenza di 4 1 consecutivi, viene inserito uno 0 che viene rimosso in ricezione prima di analizzare il contenuto della trama. La procedura, detta *bit stuffing* non viene ovviamente applicata alle sole *flag* delimitatrici.

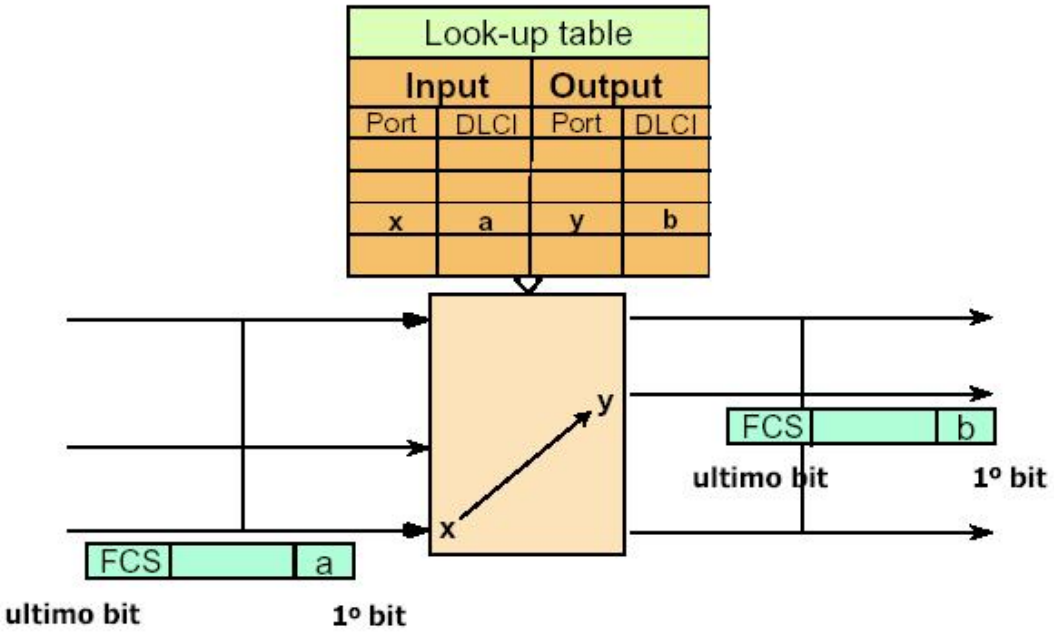
Il DLCI (*Data Link Connection Identifier*) è utilizzato per realizzare il meccanismo di indirizzamento. Il bit EA permette l'estensione del campo indirizzo. Finchè il bit EA è posto a zero, il campo indirizzo comprende il *byte* successivo. Nell'ultimo *byte* del campo indirizzo il bit EA è posto ad uno. Nelle applicazioni è generalmente usato l'indirizzo con due *byte*, in cui il bit EA è posto a zero nel primo *byte* e ad uno nel secondo *byte*.

Il bit C/R non viene gestito dalla rete; è un residuo ereditato dal protocollo LAPD. Tale bit passa inalterato attraverso la rete.

I bit BECN, FECN e DE vengono utilizzati per espletare la funzione di controllo della congestione.

Il campo FCS è utilizzato per il controllo formale delle trame ricevute.

Commutazione Frame Relay (1)



Commutazione frame relay (1)

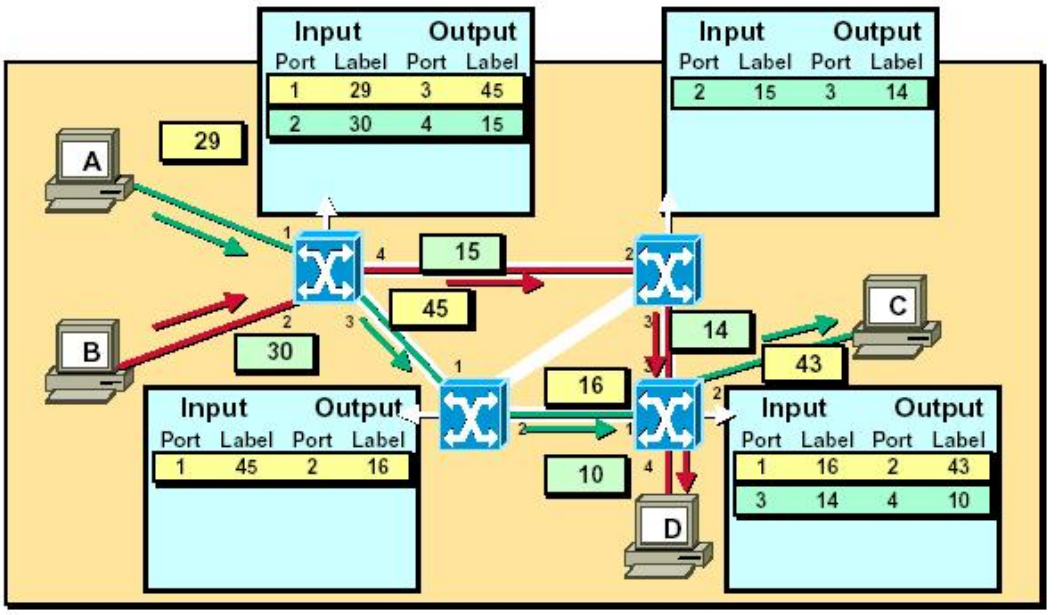
All'interno di ciascun nodo avviene una elaborazione del *frame* basata su tre passi:

- verifica integratà di trama (analisi del campo FCS); se viene rivelato un errore la trama viene scartata;
- lettura del campo DLCI con riscontro su apposita tabella; se il DLCI di destinazione non viene riscontrato la trama viene scartata;
- trasmissione del *frame*.

In breve: il *frame* viene scartato ogni qualvolta si registra:

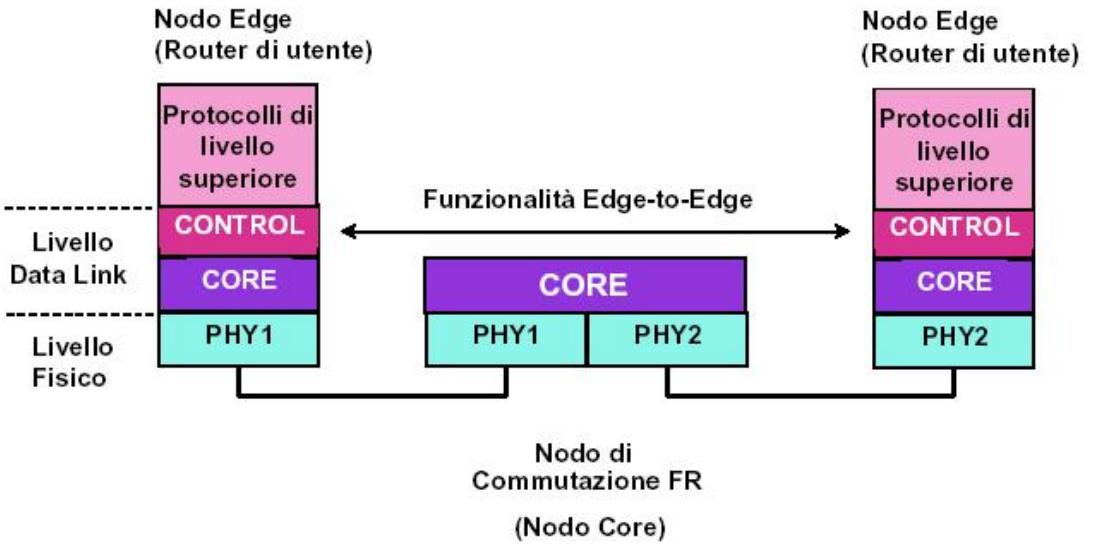
- un errore nella sua trasmissione;
- uno stato di congesione nella rete;

Il recupero delle trame scartate, non previsto dal protocollo LAPF *core*, viene demandato a protocolli di livello superiore (ad esempio LAPF *upper*) e viene effettuato da entità poste agli estremi della rete.



Commutazione frame relay (2)

Rete Frame Relay



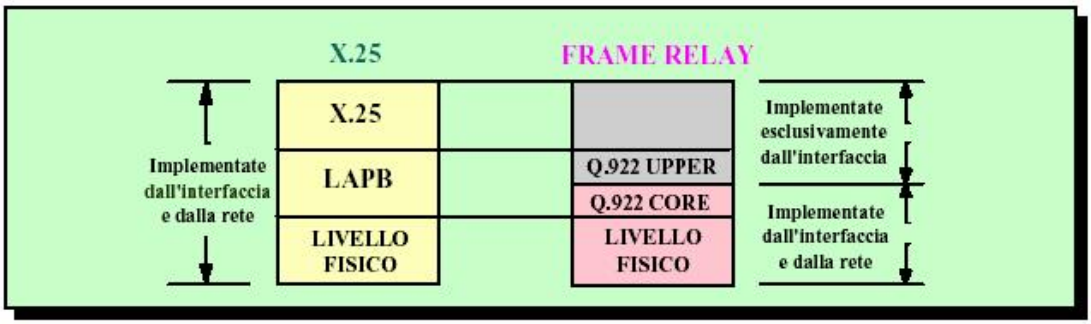
Rete frame relay

Una rete *Frame Relay* può essere realizzata da un insieme di commutatori *Frame Relay* (nodi *core*) che instradano il messaggio sulla base del DLCI, realizzando solo la

parte di LAPF detta *DL-CORE*, mentre i nodi terminali (nodi *edge*) realizzano sia il *DL-CORE* che il *DL-CONTROL*.

Tale approccio è detto *core-edge*, in quanto alcune funzionalità vengono realizzate solo *edge-to-edge* (ad esempio, recupero di errori o controllo di flusso).

Confronto Frame Relay - X.25

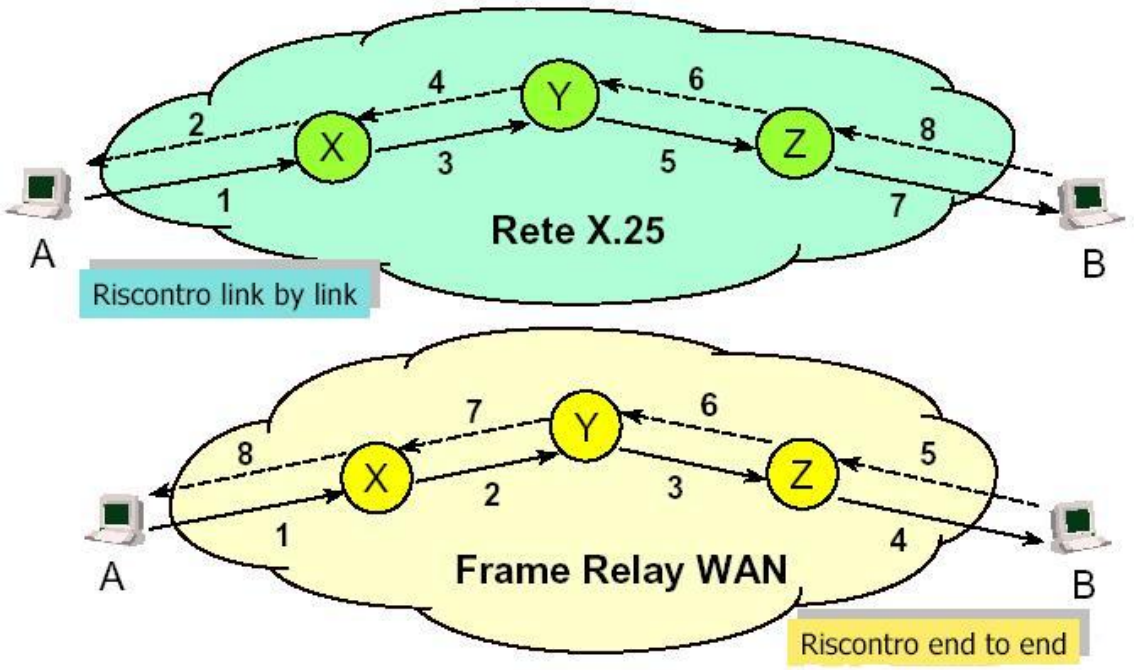


Confronto Frame Relay - X.25 (1)

Nelle reti a pacchetto X.25 sono presenti tecniche di rivelazione e correzione di errore in ogni nodo della rete. Queste procedure, basate sull'uso del protocollo LAPB (livello 2 del protocollo X.25) ed eseguite tratta per tratta, richiedono un'elevata capacità elaborativa nei nodi e causano ritardi di rete non trascurabili. Inoltre, la capacità elaborativa richiesta ai nodi X.25 per il trattamento del livello pacchetto e delle procedure di controllo di flusso non è sufficiente per supportare le apllicazione tipiche nell'interconnessione tra LAN remote (applicazioni grafiche, multimediali, trasferimento di file, eccetera).

Le limitazioni delle reti X.25 vengono superate dalla tecnica a pacchetto *Frame Relay*, dove sfruttando la disponibilità di mezzi trasmissivi a basso tasso d'errore, le funzioni di correzione di errore possono essere demandate agli apparati di utente e le procedure di controllo di flusso possono essere semplificate.

Nell'X.25 i pacchetti di controllo e quelli di trasferimento dati, appartenenti alla stessa chiamata, portano tutti lo stesso numero di canale logico di livello 3 (segnalazione in banda), nel *Frame Relay* vale il principio del separazione del controllo dal trasferimento dei dati (segnalazione fuori banda).



Confronto Frame Relay - X.25 (2)

Principi di una rete Frame Relay

Il flusso primario di dati si basa direttamente sull'intestazione del *frame* che contiene nel campo DLCI l'identificativo del circuito logico.

L'ordinamento in sequenza dei pacchetti ed il controllo degli errori deve essere gestito da protocolli di livello superiore.

Nel *Frame Relay* i circuiti sono di tipo permanente (PVC): essi sono pertanto configurati da Centro di Controllo e non dal Cliente su base chiamata.

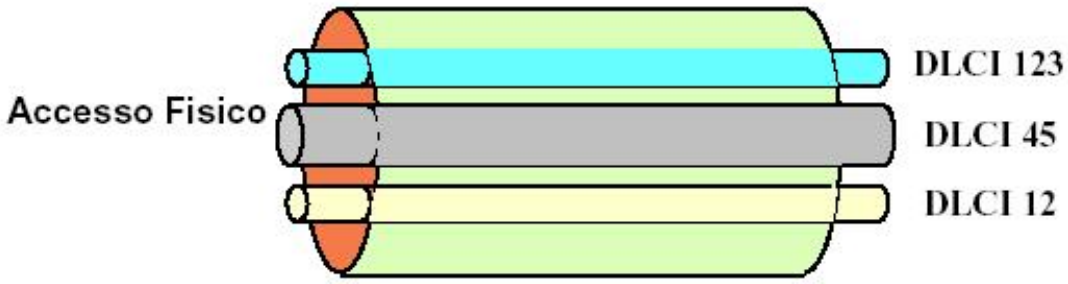
In caso di eliminazione da parte della rete di *frame* ci si affida a protocolli di livello superiore: benché automatica l'operazione di *recovery* impegna risorse elaborative e causa ritardi.

DLCI

Il DLCI non è un vero e proprio indirizzo in quanto non è associato ad una particolare destinazione.

Identifica un collegamento PVC e dunque ha significato solo locale sull'interfaccia e non globale.

Uno stesso valore di PVC può essere associato a due PVC che hanno origine in due diversi nodi, ma non per individuare due PVC configurati sulla stessa interfaccia.



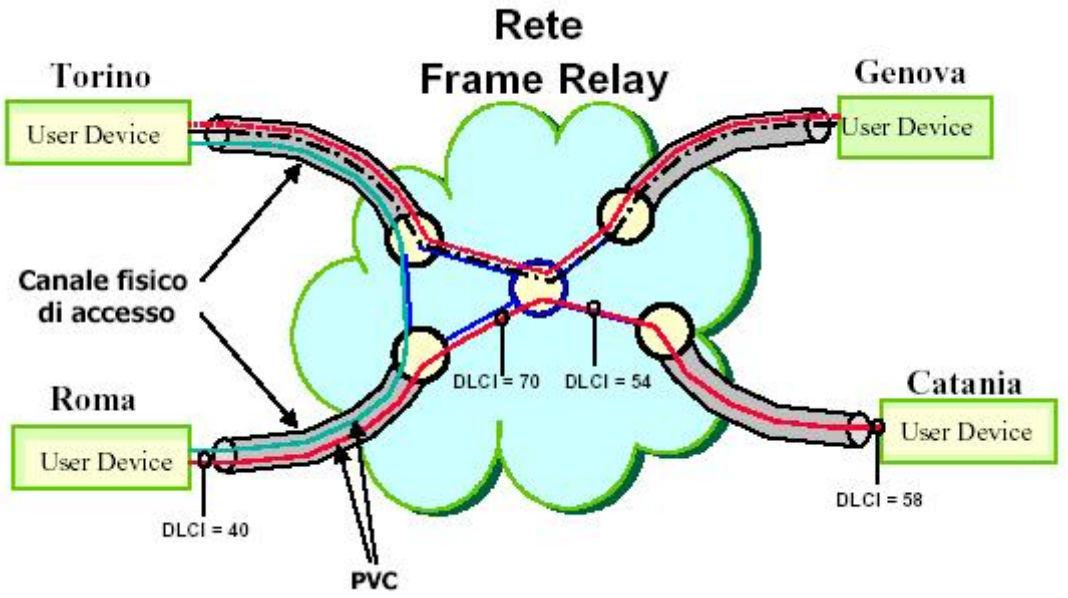
DLCI

In realtà i fondatori del *Frame Relay Forum* avevano attribuito al DLCI una valenza globale, pensando alle facilitazioni legate alla sua gestione all'interno della rete. I gestori di reti pubbliche hanno optato per una valenza locale, in quanto ciò consente di riutilizzare lo stesso DLCI per link fisici differenti ed avere una maggiore flessibilità.

Il numero di canali disponibili varia tra un massimo di 1024 (due ottetti di campo indirizzo con 10 bit disponibili per la codifica dei DLCI) ad un massimo di 8.388.608 (quattro ottetti di campo indirizzo con 23 bit disponibili per la codifica dei DLCI). La versione italiana prevede un'indirizzamento a 10 bit.

È importante osservare che il DLCI ha significato locale, nel senso che un valore di DLCI identifica una sola connessione all'interfaccia UNI. Quindi lo stesso valore di DLCI può essere associato a due o più link su linee fisiche differenti, ma non sulla stessa linea fisica.

Configurazione Logica

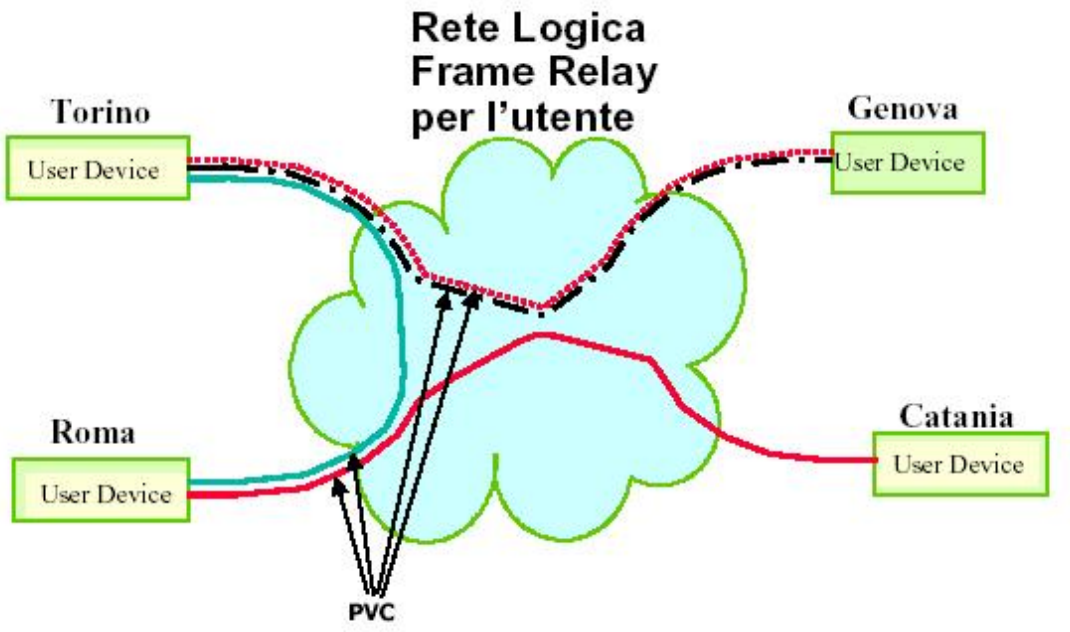


Configurazione logica (1)

La rete *Frame Relay* consente di realizzare una qualunque topologia logica fra sedi

geografiche distinte del cliente, utilizzando una topologia fisica sostanzialmente a stella per gli accessi.

Un circuito virtuale da sorgente a destinazione è individuato da una cascata di DLCI. Ogni DLCI identifica un link ed i nodi provvedono al *mapping* tra DLCI di ingresso e DLCI di uscita per le trame in transito.



Configurazione logica (2)

La figura evidenzia l'architettura della rete logica configurata per l'utente considerato. Si tratta di quattro PVC punto-punto configurati tra le seguenti sedi:
2 PVC Torino ---> Genova
1 PVC Torino ---> Roma
1 PVC Roma ----> Catania.
Si noti che i 4 PVC sono bidirezionali e possono avere caratteristiche diverse per esempio in termini di CIR. Inoltre, ciascun PVC può avere CIR diversi nelle due direzioni

Prestazioni

Il protocollo *Frame Relay* consente di multiplare attraverso il meccanismo dei canali logici (DLCI) su un unico accesso fisico bidirezionale *full-duplex* molteplici traffici relativi a diverse applicazioni.

In assenza di meccanismi di segnalazione e gestione di *throughput* non vi è modo di confinare un flusso informativo (applicazione) all'interno di una ben precisa porzione della banda complessiva disponibile sull'accesso.

Questo significa che in qualunque momento applicazioni stazionarie potrebbero venir private della propria porzione di banda da altre (*Bursty*) che eccedono il proprio *rate* nominale.

Meccanismi di Segnalazione

I meccanismi di segnalazione sono opzionali ma risultano essenziali per conseguire buone prestazioni di rete.

Tali meccanismi permettono di:

- segnalare la presenza di congestione di rete;
- fornire informazioni sullo stato delle connessioni;
- assicurare *throughput* adeguato agli utenti;
- supportare nuovi servizi (SVC).

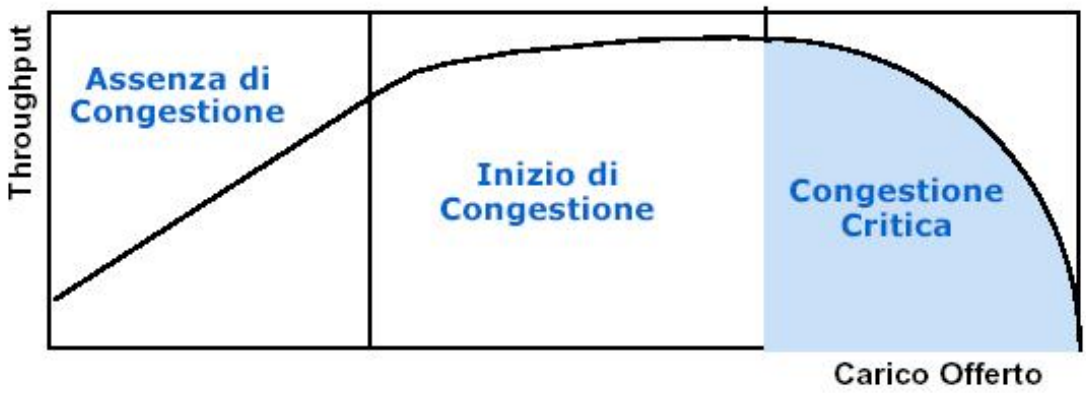
Tali meccanismi fanno uso tanto di appositi bit del *frame* tanto di appositi DLCI riservati alla segnalazione.

Controllo della Congestione

Il controllo della congestione è di fondamentale importanza nella gestione di una rete *Frame Relay*.

Sono stati previsti due meccanismi di segnalazione per segnalare agli apparati terminali la congestione:

- notifica esplicita tramite i campi BECN e FECN (Addendum a ANSI T1.606);
- notifica implicita (ANSI T1.618).



Controllo della Congestione

Riguardo al trattamento della congestione le normative CCITT non esprimono una regola ben precisa da adottare, ma si limitano a suggerire possibili algoritmi di controllo di tipo cooperativo. Tali algoritmi presuppongono un'azione di rallentamento del flusso da parte dei terminali di utente secondo determinate soglie e sono caratterizzati da isteresi.

Il controllo della congestione coinvolge principalmente due aspetti. Il primo è la prevenzione della congestione, riguardante quei meccanismi preposti a scongiurare l'insorgere della congestione. Il secondo riguarda i meccanismi di recupero della congestione, che agiscono in aggiunta ai precedenti per risolvere una situazione di

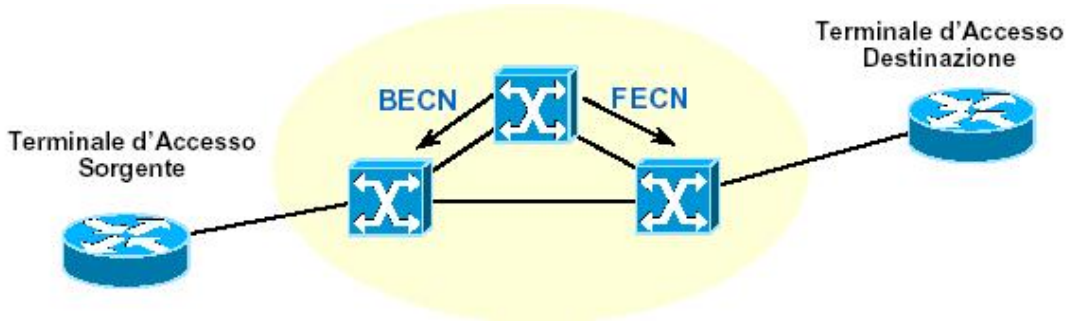
grave congestione.

I meccanismi di controllo e di recupero della congestione previsti per il *Frame Relay* sono basati sull'uso dei bit DE, FECN e BECN. In linea di principio esistono due modalità per controllare il livello di congestione della rete: la notifica implicita ICN (*Implicit Congestion Notification*), e la notifica esplicita ECN (*Explicit Congestion Notification*). Con la ICN si demanda totalmente all'utente la risoluzione del problema. Con la ECN la rete interviene a segnalare agli utenti lo stato di congestione della rete, anche se poi sarà cura degli utenti regolare il traffico in maniera tale da riportare la rete in condizioni di normalità.

Controllo della congestione: FECN e BECN

La caratteristica di gestire traffico *bursty* rende la congestione un fenomeno potenzialmente frequente in una rete *Frame Relay*.

Nel momento in cui un nodo di rete registra l'inizio di uno stato di congestione notifica ai terminali tramite la modifica dei bit FECN e BECN presenti nell'etichetta del *frame*.



Controllo della congestione: FECN e BECN (1)

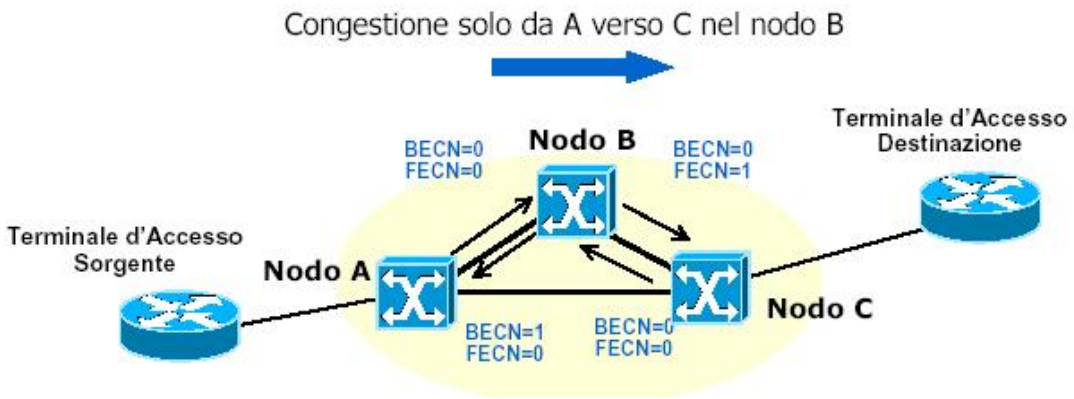
La rete *Frame Relay* supporta il meccanismo ECN (*Explicit Congestion Notification*) mediante l'uso dei bit FECN e BECN, contenuti nel campo indirizzo della trama.

Con riferimento alla figura si supponga che il nodo B sia in congestione, in questo caso esso provvederà ad impostare il bit FECN ad 1 in tutte le trame dirette verso il nodo C, notificando così la situazione anomala alle destinazioni. Al tempo stesso il nodo B provvederà a impostare il bit BECN ad 1 in tutte le trame dirette al nodo A, notificando così la situazione anomale anche alle sorgenti di traffico.

Durante il periodo di congestione le sorgenti dovrebbero ridurre gradualmente il traffico offerto, mentre con il ritorno alla normalità, segnalata dalla ricezione dei bit FECN e BECN posti a zero, le sorgenti di traffico potrebbero gradualmente aumentare il traffico offerto fino ai valori normali.

La rete non può fare affidamento sul comportamento degli utenti per controllare una situazione di congestione. La rete deve essere in grado di proteggersi da situazioni che possono generare congestione, controllando il valore del *throughput* concesso ad ogni chiamata virtuale ed eliminando le trame relative alle chiamate che non rispettano il valore del CIR.

Un nodo FR può essere in congestione solo in una determinata direzione di traffico.



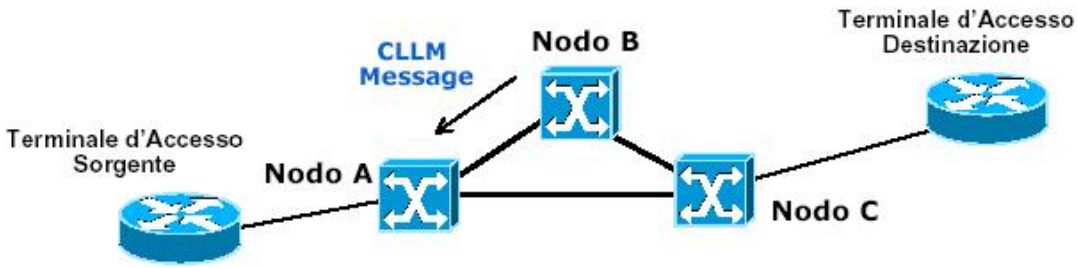
Controllo della congestione: FECN e BECN (2)

L'utilizzo di due bit (FECN e BECN) in ciascuna trama permette di segnalare l'insorgere di situazioni di congestione indipendentemente nelle due direzioni di transito delle trame FR. La figura mostra i valori dei due bit nelle trame entranti/uscenti nelle due direzioni nel/dal nodo B in congestione nel caso che questa insorga solo da A verso C. Si noti che il bit FECN è utile se il controllo della congestione è affidato al ricevitore (è il caso del protocollo TCP), mentre il bit BECN è utile se il controllo dei dati trasmessi è detenuto dall'*host* in trasmissione (per esempio, il protocollo SNA).

Controllo della congestione: CLLM

In tale metodo viene impiegato un messaggio di rete *Consolidated Link Layer Management* veicolato su un DLCI di servizio (1023) tramite il quale si segnala al dispositivo terminale la presenza di congestione.

È un meccanismo scarsamente implementato ma utile qualora il traffico sia monodirezionale.



Controllo della congestione: CLLM

Stato delle Connessioni

La segnalazione di controllo all'interfaccia utente-rete sullo stato delle connessioni (PVC) può avvenire per mezzo di due diversi meccanismi:

- impiego del protocollo *Local Management Interface* (LMI) con segnalazione di tipo asimmetrico dall'interfaccia d'utente alla rete sul DLCI 1023. Risulta esclusivo rispetto a messaggi CLLM;

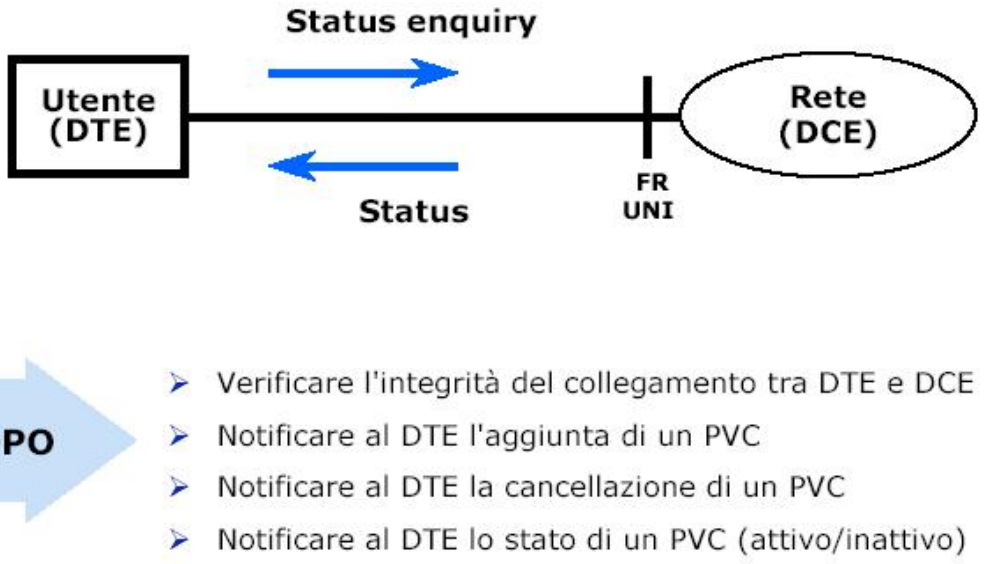
- meccanismo definito dall'ANSI (T1.617 Annex D) che estende la specifica LMI con segnalazione di tipo simmetrico sul DLCI 0.

Benché LMI risulti più semplice dal punto di vista dell'implementazione il T1.617 consente la segnalazione anche tra interfacce rete-rete.

Nel servizio FR la specifica LMI definisce il protocollo e le procedure per la gestione dei circuiti virtuali permanenti all'interfaccia UNI. La specifica LMI, proposta inizialmente dal *Frame Relay Forum*, è stata incorporata prima nello standard ANSI T1.617 (Annesso D) ed infine, con lievi modifiche, nella raccomadazione CCITT Q.933 (annesso A).

Il protocollo LMI utilizza un meccanismo di *polling*: l'apparato di utente interroga la rete per ottenere informazioni sullo stato dei PVC definiti all'interfaccia UNI. Lo scambio dei messaggi avviene al livello 2 con modalità senza connessione, utilizzando solo trame informative non numerate secondo il protocollo LAPD (Raccomandazione Q.921). I messaggi di livello 3 sono strutturati secondo la Raccomandazione Q.931, come in ISDN.

PVC Management



PVC Management (1)

Uno degli aspetti più articolati del *Frame Relay* è sicuramente il processo di PVC *Management*.

Per PVC *Management* si intende un processo di scambio di messaggi tra il DTE (apparato d'utente, tipicamente un *router*), ed il DCE (l'interfaccia di rete *Frame Relay*). È sostanzialmente un processo di *polling* iniziato sempre, nel caso di interfaccia UNI (*User-to-Network Interface*) dal DTE. Il DTE stimola il DCE inviando messaggi di *Status Enquiry*, a cui la rete deve rispondere opportunamente mediante messaggi di Status. Il processo di *polling* avviene sulla base di contatori e *timer* configurati sia sul DTE che sul DCE, e che devono essere congruenti per consentire un corretto scambio dei messaggi.

Lo scopo di questa procedura è quello di:

- verificare l'integrità del link tra DTE e DCE, ossia tra apparato d'utente e interfaccia di rete, in termini di collegamento fisico (linee, modem, eccetera); un corretto scambio di messaggi *Status Enquiry* / Status tra utente e rete implica che la connessione fisica tra DTE e DCE è corretta;
- notificare il DTE con informazioni riguardanti lo stato dei PVC (e dei relativi DLCI) disponibili per la trasmissione dati; infatti, mediante i messaggi di Status inviati dalla rete, il DTE è in grado di:
  - capire quali sono i PVC configurati all'interfaccia e di conseguenza rilevare l'aggiunta e la cancellazione di un PVC;
  - individuare lo stato di un PVC, dove lo stato può essere attivo e quindi disponibile per la trasmissione dati o inattivo e quindi non disponibile per la trasmissione di trame dati *Frame Relay*.

La segnalazione di controllo all'interfaccia utente-rete sullo stato delle connessioni (PVC) può avvenire per mezzo di tre diversi meccanismi:

- Lo standard LMI (*Local Management Interface*, normalizzato in ambito *Frame Relay Forum*) con segnalazione di tipo asimmetrico dall'interfaccia d'utente alla rete sul DLCI 1023.
- Lo standard ITU-T (Q933 Annex A) anch'esso basato sullo scambio asimmetrico dei messaggi di segnalazione sul DLCI 0.
- Lo standard definito dall'ANSI (T1.617 Annex D) che estende la specifica LMI con segnalazione di tipo simmetrico sul DLCI 0 (consente la segnalazione anche tra le interfacce rete-rete NNI).

Gestione del Throughput

L'impiego del bit DE unitamente a meccanismi di CIR consente al dispositivo di utente o ai nodi di rete di contrassegnare i *Frame* che prioritariamente la rete può scartare in caso di:

- congestione di rete;
- comportamenti anomali da parte di apparati terminali che non rispettino determinati profili di servizio.

In tal modo è possibile garantire anche ad utenti che sviluppano un traffico modesto di non venire penalizzati nella contesa alle risorse di rete (banda trasmissiva) da sorgenti di traffico più intense.

Il fatto che lo standard non obblighi ad implementare la gestione del bit DE porta sovente a spostarne l'impiego sui nodi di rete

Committed Information Rate - CIR

Il bit DE trova la sua applicazione nel meccanismo del CIR.

Il CIR rappresenta la capacità di banda trasmissiva che ogni utente contrattualizza con la rete e che ritiene necessaria e sufficiente in condizioni stazionarie di traffico per veicolare le proprie applicazioni su un determinato canale logico (DLCI).

Qualora l'utente superi in determinati istanti di tempo il valore di CIR definito, ma non il limite massimo messogli a disposizione dall'accesso, la rete pone a 1 il bit DE.

In condizioni normali di rete scarica, i *Frame* con DE=1 vengono comunque inoltrati e recapitati.

In caso di congestione tali *Frame* vengono scartati per primi.

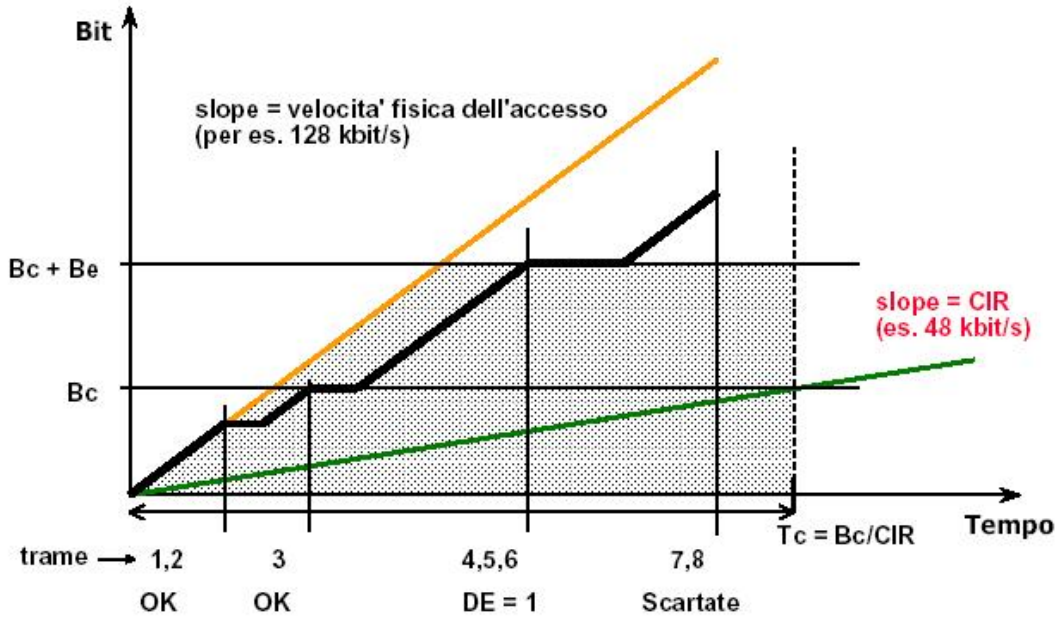
La prestazione CIR è disponibile sugli accessi Plus della rete.

Viene definito sulla base delle seguenti grandezze:

- Bc (*Committed Burst Size*): rappresenta la massima quantità di *byte* che l'utente può offrire alla rete in un tempo prefissato Tc e di cui la rete garantisce la consegna.
- Be (*Excess Burst Size*): è la massima quantità di *byte* al di sopra del valore Bc che un utente può inoltrare in un tempo Tc e di cui la rete non garantisce la consegna. I *byte* che vengono inoltrati tra i valori Bc e Be hanno DE=1.
- Tc: intervallo di tempo di osservazione.
- CIR: viene definito come Bc/Tc. Per valori di *rate* superiori al CIR i *frame* verranno marcati con DE=1 e scartati dalla rete se le risorse non fossero sufficienti.
- Il CIR verrà garantito attraverso un meccanismo di controllo all'interfaccia di accesso in rete.

L'implementazione del CIR non solo è un importante servizio per garantire qualità al Cliente ma anche uno strumento di controllo e gestione della rete per l'operatore

Committed Information Rate - CIR (1)

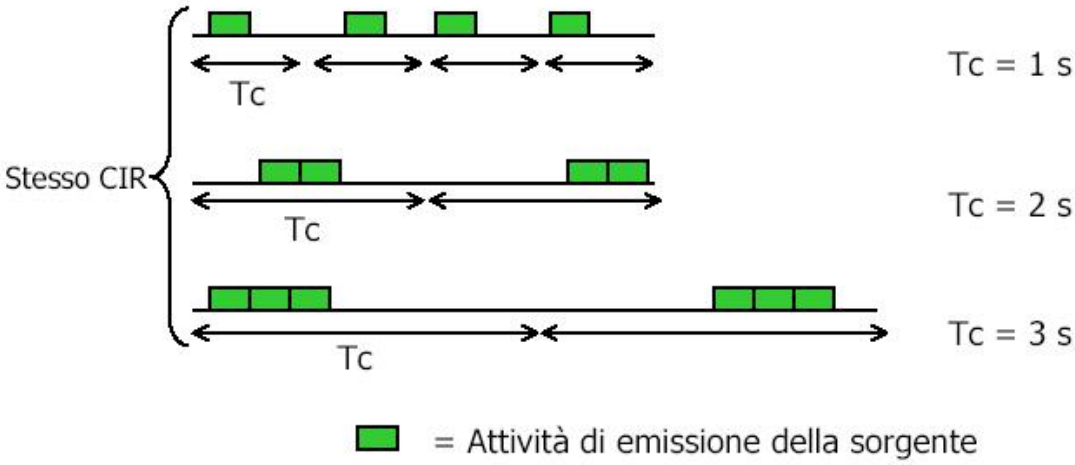


Committed Information Rate - CIR (2)

L'utente trasferisce trame alla velocità consentita dalla porta di accesso. La rete controlla il traffico in ingresso e lascia passare inalterate tutte le trame, finché non viene superato il parametro Bc.

Le trame in eccesso (sino al valore Be) sono accettate, ma il loro bit DE viene posto ad 1 dalla rete; così facendo esse saranno le prime ad essere scartate in caso di congestione della rete.

Ulteriori trame, inviate dall'utente nell'intervallo Tc quando il valore Bc + Be è stato superato, sono scartate dal nodo di accesso, senza la necessità di segnalare tale operazione all'utente. Trascorso l'intervallo di tempo Tc, ha inizio un nuovo ciclo di controllo del traffico, rappresentabile con un nuovo diagramma in cui l'origine degli assi risulta traslato a destra di una quantità pari a Tc.



Committed Information Rate - CIR (3)

A parità di CIR, la definizione di un Tc più alto permette all'utente di trasmettere la stessa quantità di dati in modo più intermittente, come è evidenziato nella figura. Definire un Tc più basso vuol dire costringere l'utente a trasmettere la stessa quantità di dati in modo più regolare, cioè più distribuita nel tempo.

L'intervallo di tempo Tc è fissato dal gestore della rete geografica ed è pari, tipicamente ad 1 s. Valori più piccoli di tale limite tutelano il service *provider*, poiché consentono di ridurre la durata del *burst*; valori maggiori favoriscono l'utilizzatore.

Parametri di QoS per il Frame Relay

Tali parametri per servizi offerti su PVC sono ancora in corso di definizione.

L'unico riferimento è la Raccomandazione ITU-T X.144 che definisce per il *Frame Relay* i seguenti parametri di QoS:

- FTD (*Frame Trasfer Delay*): Intorno ai 5:25 ms in funzione della lunghezza di trama.
- FLR (*User Information Frame Loss Ratio*): *Frame* perse/(S Fr. perse+Fr. trasferite+Fr. trasferite con errore nel *payload*).
- *Extra Frame Rate*.
- RFER(*Residual Frame Error Ratio*): *Frame* TX corr.te/(S Fr. TX corr.te+Fr. TX con errore nel *payload*).
- PVC *Availability*: Si fissano valori superiori per FLR, EFR e RFER e si dichiara indisponibile il PVC se solo uno dei parametri supera la soglia.

CIR e QoS

Anche il parametro CIR (assieme ai parametri Bc e Be) può essere sicuramente considerato tra i fattori che determinano la Qualità del Servizio *Frame Relay*.

- Il CIR indica il tasso di trasferimento dei dati che la rete garantisce di trasferire in condizioni operative normali.

Tuttavia per la natura statistica del servizio questa garanzia non può che essere fornita in termini di probabilità di garanzia sul trasferimento dei dati.

Valutare numericamente questa garanzia può risultare molto complesso, come altrettanto complesso può essere il metodo per misurarla in esercizio.

Al momento non esistono valori di riferimento per questa probabilità.

Multiprotocol su frame relay

La RFC 1490 specifica l'incapsulamento di protocolli di livello 2 (*bridged*) e di livello 3 (*routed*) in trame *Frame Relay*.

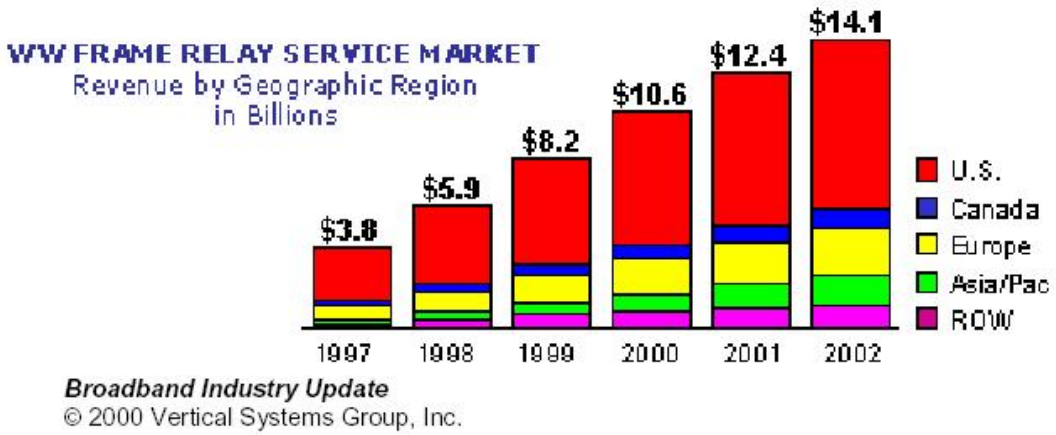
In particolare, la figura mostra l'incapsulamento di datagrammi IP in trame FR.

UI= Unnumbered Information
NLPID = Network Layer
Protocol Identifier

Multiprotocol su frame relay

La normativa RFC 1490 definisce le modalità di incapsulamento di pacchetti IP in trame FR e consente di multiplare più *stack* di protocolli sulla stessa connessione *frame relay*, demandando al *router* ricevente la separazione dei pacchetti appartenenti a *stack* diversi ed il loro smistamento verso apparati di utente specializzati. La multiplazione di più *stack* di protocolli avviene mediante il campo dati NLPID che specifica il tipo di livello 3 trasportato nel campo informativo che segue nella trama.

Trend del frame relay



Trend del frame relay

Il mercato mondiale dei servizi *Frame Relay* continua a crescere in maniera stazionaria con proiezioni al 2002 che prevedono ricavi di circa 14 miliardi di dollari. Questo dato si riferisce all'offerta di servizi con *access rates* compresi tra 56/64 kbps a T3/E3.

Il mercato degli U.S.A. è quello più fiorente, ma le proiezioni al 2002 mostrano che il mercato non U.S.A. tende a crescere più rapidamente: il tasso di crescita annuale composto è del 29% contro quello degli U.S.A. che è del 22%.

Asynchronous Transfer Mode (ATM)

Franco Callegati
Paolo Zaffoni

8.2.1 (Distinguere tra opzioni basate su router, su switch e su bridge)

Introduzione

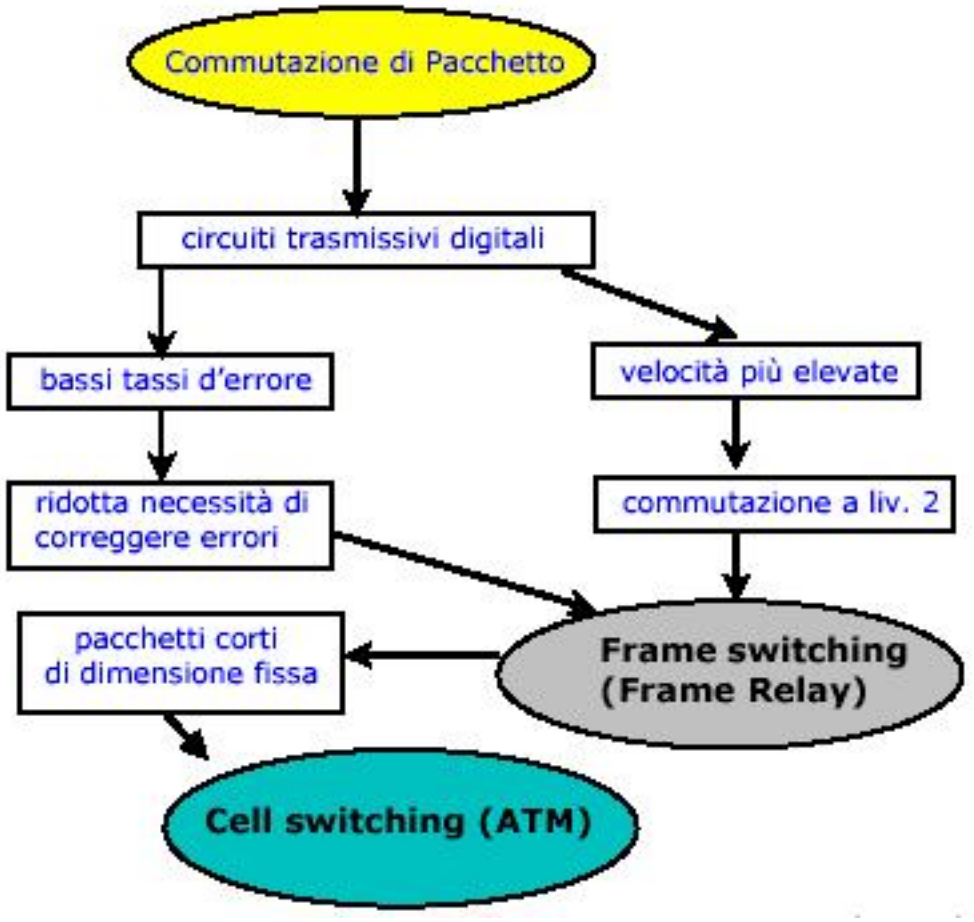
È importante precisare fin da questo punto che *Asynchronous Transfer Mode* si delinea come tecnologia, piuttosto che come rete. Non prevede infatti, nella maggior parte delle implementazioni, un'assegnazione di indirizzi.

Come servizio *WAN*, fornisce all'utente informatico, ovvero all'utente amministratore delle reti corporate, un tubo non trasparente e numerato (con etichetta e non con indirizzo) per il trasporto di aggregati di pacchetti dati che terminano in apparati predefiniti.

Con ATM si realizzano infrastrutture in cui gli estremi della comunicazione sono preliminarmente identificati e le connessioni vengono impostate da un sistema di gestione dell'infrastruttura su una base contrattuale e statica.

La maggior parte dei servizi ATM forniti da *carrier* in tutto il mondo è di tipo PVC (Permanent Virtual Connection).

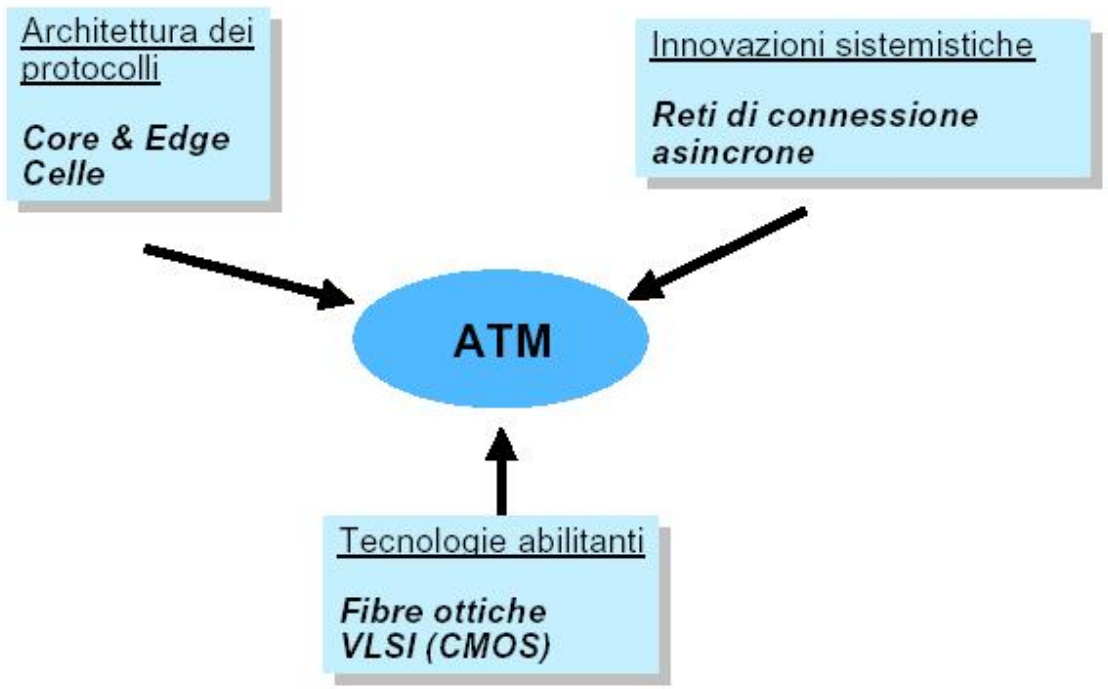
Evoluzione della commutazione di un pacchetto



Evoluzione della commutazione di un pacchetto

Agli inizi degli anni '80 i gestori delle TLC pubbliche hanno avviato la realizzazione sistematica di reti specializzate per l'offerta commerciale del trasferimento dati a pacchetto (reti X.25). Successivi miglioramenti sono stati apportati a queste reti, in particolare per aumentare la velocità di trasporto dei dati (reti ***Frame Relay***). Oggi il traffico dati è in forte crescita e nel giro di pochi anni sorpasserà il traffico telefonico: V.Cerf (MCI) ha posto il punto di incrocio a metà del 2001 (per quel che riguarda la rete MCI). Una conseguenza di questo fatto è che nel futuro sarà conveniente progettare una rete ottimizzata per i dati (e quindi a pacchetto), in grado di trasportare anche la voce. La tecnologia ATM permette di trasferire in modo integrato tutti i tipi di traffico numerico su una singola infrastruttura.

Genesi di ATM



Genesi di ATM

Anche ATM, al pari di *frame relay* predilige la velocità a discapito dell'affidabilità. In ATM le funzioni svolte dai nodi di rete sulla presenza di errori nei dati trasportati, si esplicano solo sull'etichetta dei pacchetti (detti celle) e non sul contenuto informativo.

La realizzazione di una infrastruttura ATM presuppone quindi un'alta affidabilità delle tecnologie trasmissive adottate (fibre ottiche fra nodi e fino alla sede di utente).

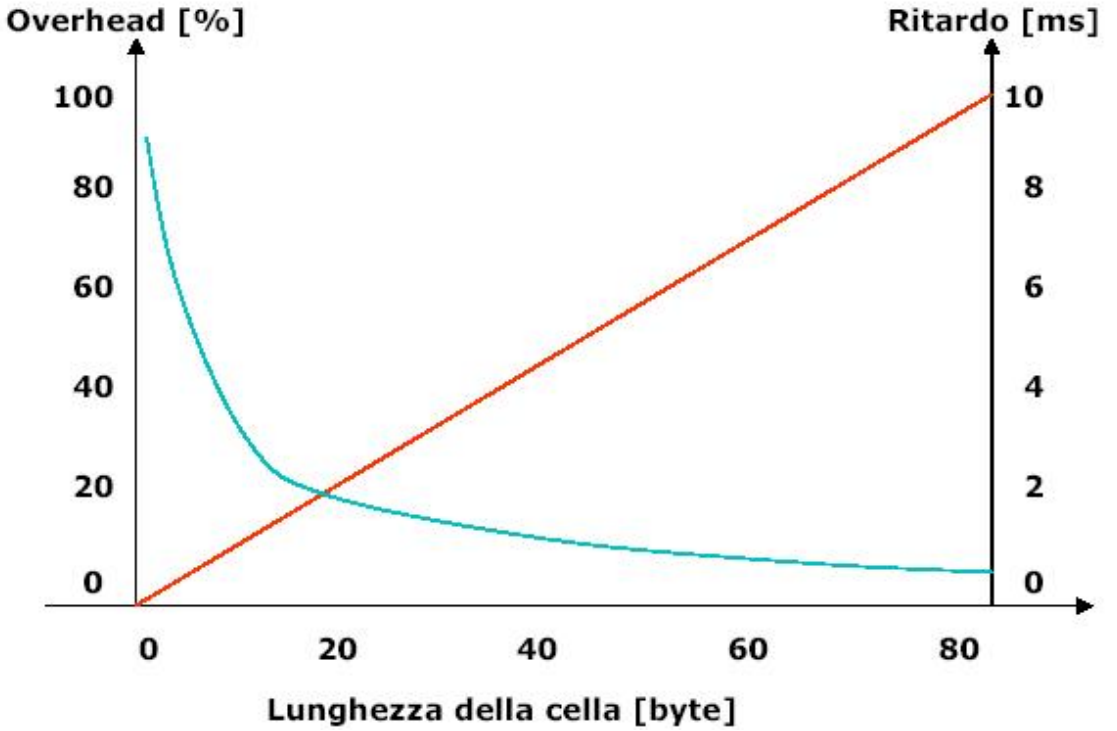
Requisiti di ATM

- Alta velocità (centinaio di Mb/s).
- Allocazione di banda dinamica.
- Granularità fine nell'assegnazione della banda.
- Supporto anche di traffico di tipo *bursty*.
- Adattabilità sia ad applicazioni sensibili al ritardo che alla perdita.
- Possibilità di connessioni multipunto e *broadcast*.

ATM è stato concepito per superare le limitazioni dell'ISDN a banda stretta:

- banda statica e fissa (64 kbit/s).
- Assenza di supporto per il *burst* dei dati.
- Assenza di un modo di trasferimento unico.
- Utilizzazione non ottimale della banda trasmissiva.

Lunghezza cella ATM (ritardo di pacchettizzazione per la voce)



Lunghezza cella ATM (ritardo di pacchettizzazione per la voce)

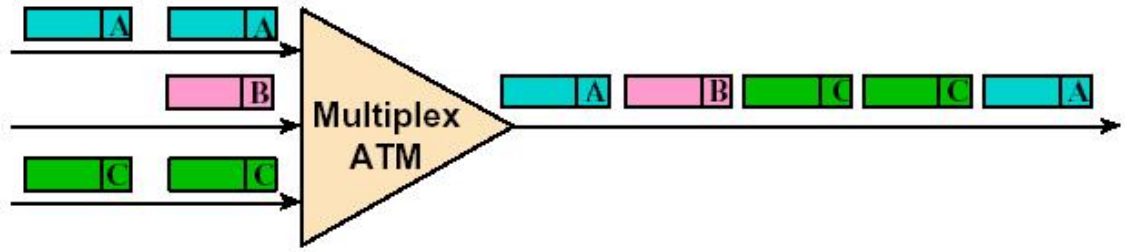
ATM effettua il trasferimento delle informazioni in formato numerico tramite unità elementari di trasporto, denominate celle, di lunghezza fissa (53 *byte*). Tutti i tipi di informazione (voce, fax, dati, immagini, video) sono trasportate tramite lo stesso formato di cella. I criteri che hanno guidato la scelta della lunghezza delle celle possono essere ricondotti essenzialmente a due motivi che giustificano rispettivamente l'adozione della lunghezza fissa rispetto a quella variabile, nonchè la scelta della specifica lunghezza impiegata per la cella. I principali vantaggi della lunghezza fissa della cella, rispetto a quella variabile, sono connessi alla significativa riduzione di complessità degli apparati di commutazione e multiplazione, nonchè alla semplificazione del trattamento delle celle nella rete, venendo meno la necessità di determinare in ogni nodo le loro lunghezze. Per quanto riguarda l'impatto sugli apparati, occorre tener presente che la necessità di trasportare flussi ad alta (155 Mbit/s, 622 Mbit/s) o altissima velocità (2,5 Gbit/s ed oltre) richiede che la multiplazione e la commutazione siano effettuate in *hardware* e non in *software* (come per esempio accade nel *Frame Relay*, ove appunto per questa scelta la velocità rimane limitata intorno ai 2 Mbit/s). La complessità di commutatori di cella a lunghezza variabile avrebbe costituito un collo di bottiglia rispetto alla crescente capacità di trasporto permessa dalla tecnologia della trasmissione su fibra ottica. La lunghezza della cella è stata determinata tenendo presente quattro fattori principali: l'effetto sulla voce del ritardo per la costituzione dei pacchetti, l'efficienza del trasporto (incidenza dell'*header*), la velocità della commutazione e il tempo complessivo di attraversamento dei nodi di rete. Il ritardo introdotto per la formazione dei pacchetti della voce è tanto minore quanto più corta è la lunghezza della cella; ma tanto più corta è la cella tanto maggiore è la perdita di efficienza nel trasporto per l'aumento dell'incidenza dell'*header* (*overhead*). Per i servizi voce su ATM, ITU-T raccomanda l'impiego di cancellatori d'eco se il ritardo *end-to-end* supera i 24 ms. La scelta della lunghezza della cella è stata lungamente discussa in ambito internazionale ed il valore di 48 *byte* (per il

payload) è risultato dal compromesso finale tra la proposta europea di 32 *byte* e quella americana di 64 *byte*. Le preferenze degli americani per una cella di 64 *byte*, tendente a una utilizzazione più efficiente della banda di trasmissione, era giustificata dal fatto che, vista l'estensione geografica del continente, erano già presenti cancellatori d'eco sulle reti tradizionali. Al contrario in Europa la rete è organizzata e progettata proprio per evitare il controllo dell'eco e quindi per escludere i cancellatori d'eco.

Schema di multiplazione ATM

ATM: *Asynchronous Transfer Mode*:

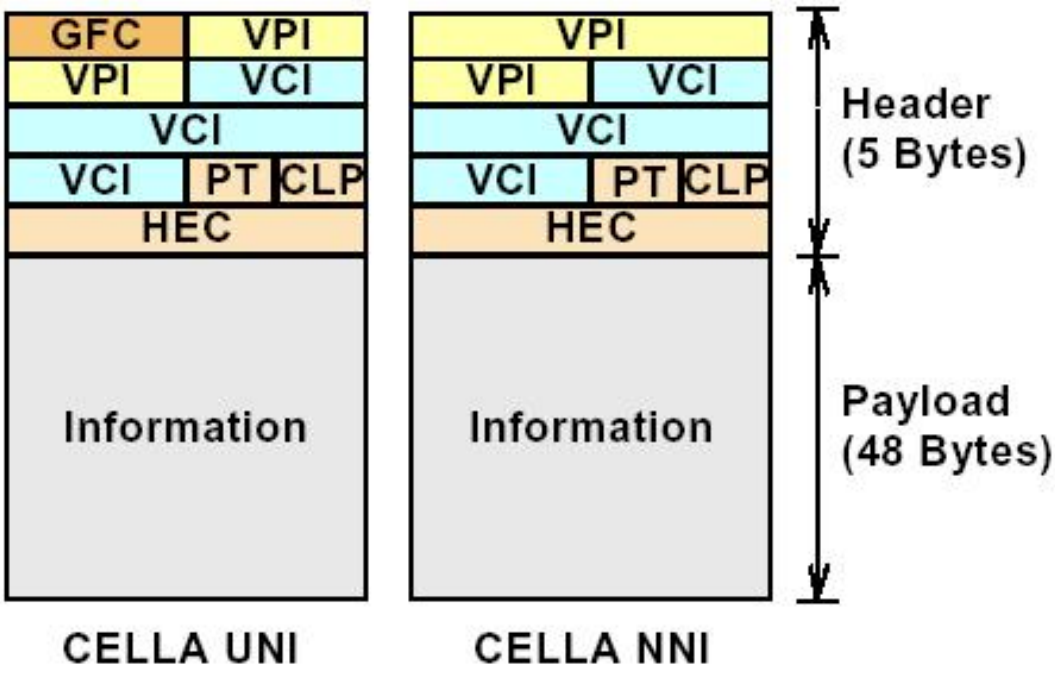
- Multiplazione asincrona
  - flussi informativi slottati in celle di lughezza fissa (53 *byte*);
  - allocazione di banda dinamica.



Schema di multiplazione ATM

ATM (*Asynchronous Transfer Mode*) una tecnica di trasporto, multiplazione e commutazione pensata per flussi informativi ad alta velocità. Non trova un esatto riscontro nei livelli OSI, ma si può posizionarla tra il livello 1 ed il livello 2. È descritta nella Raccomandazione I.361 *B-ISDN ATM layer specification* dell'ITU-T (ex CCITT). ATM è una tecnica efficiente di multiplazione e commutazione, basata su di un principio di commutazione veloce di pacchetto. Essa utilizza unità informative di lunghezza fissa (48 *byte* di dati e 5 *byte* di intestazione o *header*), denominate celle ATM. La caratteristica principale della tecnica ATM risiede nella sua flessibilità nel meccanismo di allocazione della banda, mediante la multiplazione asincrona di differenti flussi informativi (celle). Lo schema di multiplazione adottato è caratterizzato dalla suddivisione della banda trasmissiva del multiplo (e dei segnali tributari) in slot di ugual misura (celle): nella maggior parte dei casi in assenza di celle valide da trasmettere, un apparato ATM genera in trasmissione celle vuote (riconoscibili dal particolare valore dell'*header*) che hanno unicamente funzione di riempitivo e che vengono rivelate in ricezione e cestinate senza consegnarle ai livelli opportuni di elaborazione (per riassemblaggio di un messaggio lungo o di uno *stream* o per le operazioni di commutazione).

Formato della cella ATM



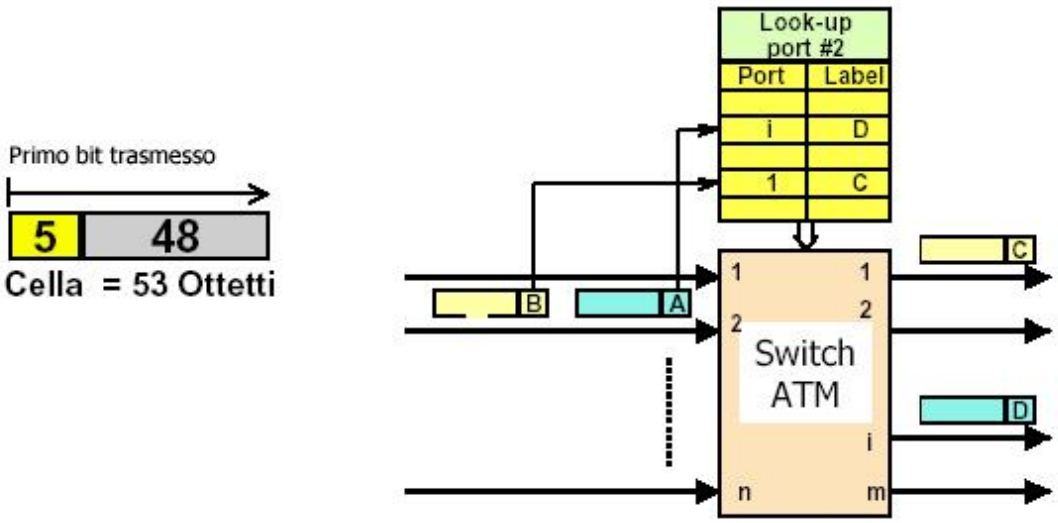
Formato della cella ATM

Sono previsti due formati dell'intestazione: uno per l'interfaccia tra i terminali e la rete (UNI) e l'altro per l'interfaccia tra nodi della rete (NNI). L'intestazione della cella ATM contiene le informazioni che consentono alla rete ATM di trasferire il carico informativo dalla sorgente a destinazione e sono quindi informazioni critiche dal punto di vista del corretto instradamento delle informazioni. Un errore nell'intestazione potrebbe indirizzare la cella verso una destinazione errata. Per questo motivo è stato definito un meccanismo molto robusto di controllo degli errori sul contenuto dell'intestazione, che utilizza il campo HEC ed è in grado sia di rivelare gli errori sia, quando possibile, di correggerli. La cella ATM non contiene meccanismi di controllo dei dati d'utente trasportati nella parte *payload* della cella stessa: il controllo degli errori sui dati d'utente è fatto al livello AAL ai bordi della rete.

Tecnica di commutazione ATM

ATM: *Asynchronous Transfer Mode*:

- Commutazione con connessione
  - identificatore della connessione nell'etichetta della cella;
  - attraversamento eseguito in *hardware*.



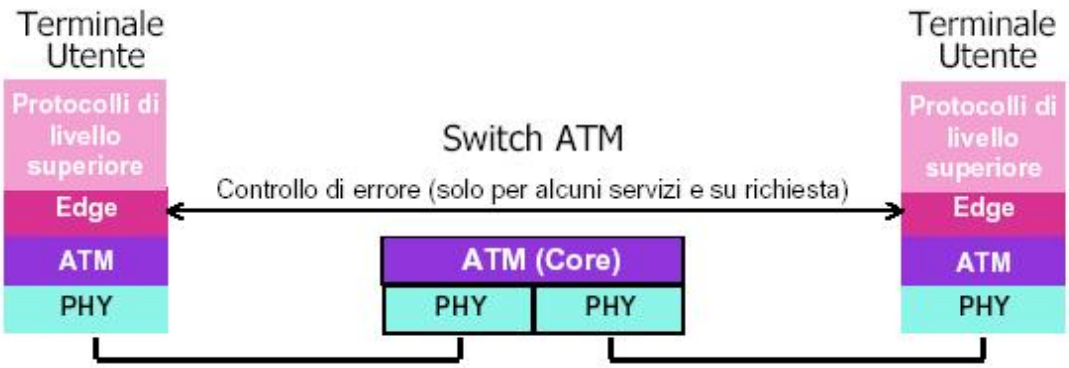
Tecnica di commutazione ATM

L'ATM è una tecnica orientata alla connessione: prima della effettiva trasmissione delle informazioni occorre predisporre il cammino che sarà seguito dalle celle attraverso tutta la rete tra i due estremi che devono comunicare tra loro. La connessione ATM è di tipo virtuale nel senso che non risulta fisicamente trasparente. In relazione alla modalità di costruzione, si possono avere *Permanent Virtual Circuit* (PVC) e *Switched Virtual Circuit* (SVC). Le PVC vengono aperte e chiuse (non in tempo reale) dal gestore della rete con opportune operazioni di configurazione tramite un sistema di gestione della rete. Le SVC sono invece aperte, modificate e chiuse dinamicamente, su richiesta dell'utente, mediante una procedura di segnalazione, come avviene nel caso di connessione telefonica o di una connessione a chiamata virtuale X.25.

Architettura dei protocolli ATM

Protocolli: principio del *Core and Edge*

- nei nodi sono eseguite solo le funzioni essenziali (commutazione e multiplazione) a livello ATM (1-2 della pila OSI).
- Le funzionalità residue, specifiche per i diversi tipi di servizio, sono svolte agli estremi.



Architettura dei protocolli ATM

Il profilo dei protocolli dell'architettura ATM è stato concepito nell'ottica *core and edge*. In altri termini la rete si semplifica notevolmente e quindi può fornire servizi di trasporto ad alta velocità, ma deve impiegare mezzi trasmissivi affidabili e di ottima qualità, consentendo così alti *throughput* e relegando ai margini delle connessioni (terminali *edge* di utente) le funzioni di adattamento alle applicazioni, recupero degli errori, e quant'altro la rete non implementa).

Caratteristiche generali di ATM

- Mezzi trasmissivi veloci (purchè con basso tasso di errore):
  - tipicamente >= 150 Mb/s.
- Bassi ritardi:
  - idoneo per dati, voce e immagini video.
- Meccanismi sofisticati per il controllo di flusso (i tradizionali meccanismi a finestra non sono efficienti).
- Segnalazione sofisticata (capace di gestire connessioni *multiparty*).
- Tecnica di trasferimento adatta a realizzare LAN e *WAN*.
- Tecnica di trasferimento scelta per la B-ISDN.

La commutazione ATM è detta anche ad etichetta, per via delle funzioni che ciascun nodo realizza sul flusso di unità informative che riceve.

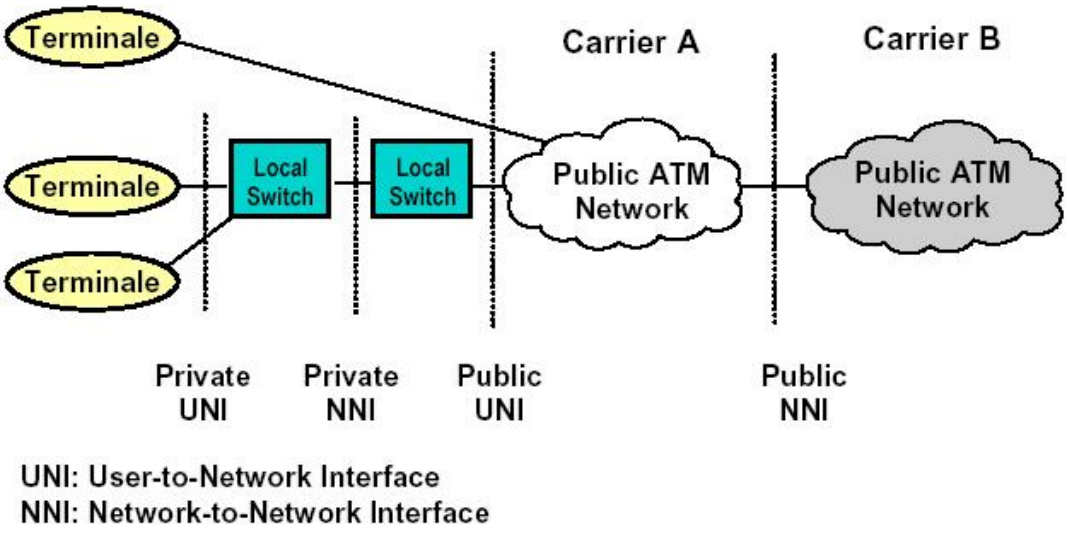
In particolare la commutazione ATM si esplica mediante:

- traduzione dell'etichetta dal valore entrante al valore uscente;
- trasferimento della cella dal *multiplex* entrante verso il *multiplex* uscente;
- eventuale replica della cella verso più uscite fisiche (con etichette distinte e tradotte), nel caso di connessioni multicast.

Per poter operare a velocità di parecchie centinaia di Mbit/s, la funzione di commutazione deve avvalersi di appositi organi *hardware* (analogamente a quanto accade nei nodi per la commutazione di circuito, dove però la presenza di organi di connessione *hardware* è legata a garantire l'allocazione di un cammino fisico fra la linea entrante e la linea uscente).

Tale organo interno al nodo, nella maggior parte dei casi è progettato e realizzato con componenti ASIC ottimizzati per l'elaborazione delle celle ATM ed opera senza l'intervento di *software* di controllo a livello di singola cella. In altri termini una proprietà spesso implementata per tali organi (reti di connessione asincrone) consente l'*autorouting* all'interno del nodo.

Interfacce di rete



Interfacce di rete

La rete ATM è costituita da un insieme di nodi di commutazione e da un insieme di nodi terminali. I nodi o celle di commutazione sono collegati tra loro con linee punto-punto e formano il *core* della rete con una topologia a maglia. I nodi terminali sono connessi ai nodi di commutazione con una topologia a stella. L'interfaccia tra nodo di commutazione e terminale viene detta *User to Network Interface* (UNI), mentre quella tra nodi di commutazione è detta *Network to Network Interface* (NNI). Le UNI e le NNI sono le specifiche formali su cui basarsi per rrealizzare una rete ATM: si possono collegare delle stazioni solo a nodi di commutazione con UNI compatibili come, d'altra parte, è possibile interconnettere tra loro solo nodi di commutazione con NNI interoperabili.

Funzioni dei protocolli ATM

- Fornire, se necessario, un servizio *connectionless*.
- Fornire ai diversi servizi connessioni che rispondano agli specifici requisiti prestazionali.
- Segmentare e riassemblare i flussi informativi in celle.
- Commutare le celle nei nodi.
- Adattare il flusso di celle al particolare tipo di supporto trasmissivo.
- Inviare i singoli bit sul portante fisico.

L'architettura dei protocolli ATM è stata ideata seguendo la filosofia *core and edge*, la quale prevede che le funzionalità di protocollo atte al trasporto di informazioni tra gli utenti della rete non sono implementate nello stesso modo su tutti i nodi attraversati dalla comunicazione. Nella parte interna della rete, nel *core*, sono operativi solo i protocolli di livello più basso, che svolgono solo il minimo delle operazioni necessarie al trasferimento delle informazioni; nei punti terminali sono presenti anche i protocolli di livello superiore, che forniscono altre funzionalità al trattamento dei flussi informativi. La filosofia *Core and Edge* è concettualmente molto diversa da quella su cui si basano le reti a pacchetto più vecchie come X.25 dove, per mantenere un elevato livello qualitativo della comunicazione nonostante l'infrastruttura poco affidabile, bisogna

effettuare un controllo di errore su ogni collegamento interno alla rete; l'evoluzione dei mezzi trasmissivi ha ridotto drasticamente il tasso di errore e ciò consente alla rete di effettuare il controllo di errore solo agli estremi (*Edge*) e di alleggerire dal punto di vista computazionale i compiti dei nodi interni (*Core*): ciò comporta una semplificazione dell'architettura e quindi un miglioramento dal punto di vista delle prestazioni. L'architettura protocollare di ATM è simile a quella del modello di riferimento OSI, con uno sviluppo tridimensionale dovuto alla presenza di 3 piani di lavoro: *User Plane*, per il trasporto delle informazioni di utente, *Control Plane*, per il trasporto e il trattamento della segnalazione e un *Management Plane* suddiviso in un *Layer Management*, con lo scopo di gestire i flussi informativi di *Operation And Manteinance* (OAM) per configurazione e manutenzione della rete e quelli di segnalazione, e in un *Plane Management*, per il coordinamento dei piani di lavoro precedenti. Ogni piano, a differenza dell'ultimo che si occupa del coordinamento e quindi è trasversale agli altri, è diviso in tre livelli: *Physical Layer*, *ATM layer* (ATM), *ATM Adpater Layer* (AAL).

Modello di riferimento dei protocolli

<table>
<tr><td>Convergence sublayer</td><td rowspan="2">Livello adattamento AAL (ATM Adaptation Layer)</td><td rowspan="2"></td></tr>
<tr><td>Segmentation and Reassembly</td></tr>
<tr><td>Cell header generation/extraction</td><td rowspan="3">Livello ATM</td><td rowspan="3"></td></tr>
<tr><td>Cell VPI/VCI translation</td></tr>
<tr><td>Cell multiplex/demultiplex</td></tr>
<tr><td>Cell rate decoupling</td><td rowspan="4">Transmission Convergence</td><td rowspan="5">Liv. fisico</td></tr>
<tr><td>HEC generation/verification</td></tr>
<tr><td>Cell delineation</td></tr>
<tr><td>Transmission frame generation</td></tr>
<tr><td>Bit timing</td><td rowspan="2">Physical medium dependent</td></tr>
<tr><td>Bit TX/RX</td></tr>
</table>

Modello di riferimento dei protocolli

Da questa rappresentazione si vede chiaramente come il Livello Fisico comprenda tutte le funzionalità comprese tra il mezzo fisico (tipicamente ottico o elettrico) ed il livello ATM (che controlla le funzioni di commutazione e multiplazione).

Il Livello Fisico è inoltre suddiviso in due sottolivelli: il sottolivello più basso (PM: *Physical Medium*) dipende dalle caratteristiche del mezzo trasmissivo ed ha il compito di convertire le caratteristiche fisiche del segnale (ovvero variazioni nel tempo di un segnale ottico o elettrico) in flusso continuo di bit, salvaguardardone l'integrità e la sequenza di trasmissione. Il sottolivello più alto (TC: *Transmission Convergence*) ha il compito di identificare all'interno della sequenza di bit i confini delle celle, separare quelle di pertinenza del Livello Fisico da quelle utilizzate dal livello ATM e passare

queste ultime al circuito che svolgerà le opportune funzioni. Il TC si preoccupa anche della generazione/estrazione di celle vuote, utilizzate per riempimento del flusso trasmissivo.

In pratica è opportuno suddividere il sottolivello TC in due sottolivelli separati, a seconda che le funzioni svolte siano tipiche del sistema trasmissivo utilizzate o più strettamente legate alla presenza di celle ATM. Infatti, mentre le funzionalità legate al sistema trasmissivo possono essere comuni anche ad altri apparati (non ATM) che utilizzano lo stesso sistema e le stesse modalità di organizzazione dei bit presenti in linea, le funzioni legate alla presenza di celle ATM sono indipendenti dal sistema trasmissivo utilizzato e sono comuni a tutti gli apparati in tecnica ATM.

La pila ATM si completa con il protocollo di adaptation, che sarà descritto più avanti e che ha il compito di isolare i protocolli applicativi (anche IP, *frame relay*, ecc.) dal tipo di trasmissione a celle.

UNI ATM

Private ATM			***Public ATM***		
ATM Forum Physical Layer UNI Interfaces			*ATM Forum Physical Layer UNI Interfaces*		
Frame Format	*Bit rate/Line rate*	*Transmission Media*	*Frame Format*	*Bit rate*	*Transmission Media*
Cell Stream	25.6 Mb/s	UTP-3	DS1	1.544 Mb/s	*Twisted Pair*
STS-1	51.84 Mb/s	UTP-3	DS3	44.736 Mb/s	*Coax Pair*
FDDI	100 Mb/s	MMF	STS-3c, STM-1	155.520 Mb/s	SMF
STS-3c, STM-1	155.52 Mb/s	UTP-5, STP	E1	2.048 Mb/s	*Twisted or Coax Pair*
STS-3c, STM-1	155.52 Mb/s	SMF, MMF, Coax	E3	34.368 Mb/s	*Coax Pair*
Cell Stream	155.52 Mb/s - 194.4 Mbaud	MMF / STP	J2	6.312 Mb/s	*Coax Pair*
STS-3c, STM-1	155.52 Mb/s	UTP-3	NxT1	Nx 1.544 Mb/s	*Twisted Pair*
STS-12, STM-4	622.08 Mb/s	SMF, MMF	NxE1	Nx 2.048 Mb/s	*Twisted Pair*

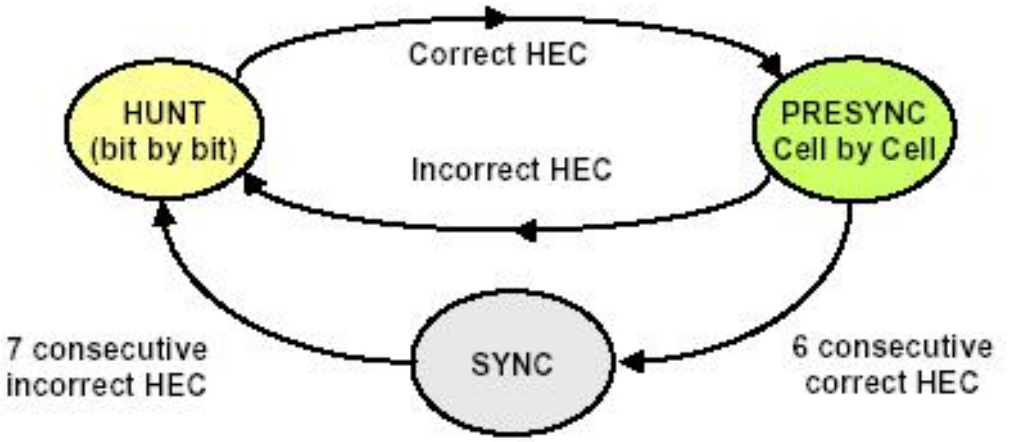
Il primo livello fisico per il trasporto delle celle ATM è stato normalizzato dall'ITU ed è relativo al trasporto su sistemi trasmissivi SDH con velocità 155 Mbit/s nei primi anni `90.

Successivamente, ad opera dell'ATM Forum sono stati proposti altri livelli fisici per impieghi in ambito locale o per ambito di reti pubbliche meno esigenti dal punto di vista delle funzioni di strato fisico.

In particolare la figura riassume le UNI normalizzate in ambito ATM Forum. Una parte delle interfacce per reti pubbliche sono state normalizzate in tempi recenti anche in ambito ITU; in particolare quelle per sistemi trasmissivi plesiocroni (E1, E3, T1, T3).

Cell delineation

- *Cell Delineation*: identificazione dei confini delle celle all'interno del *payload* in ricezione mediante analisi del campo HEC:
  - HUNT (stato iniziale). Il ricevitore sposta una finestra di cella bit per bit e calcola HEC.
  - PRESYNC. La finestra si sposta di cella in cella fino a quando non sono stati rilevati 6 HEC corretti consecutivi.
  - SYNC. Struttura del *payload* identificata; HEC viene ora usato per il rilevamento degli errori. Il sincronismo é perso quando si rilevano 7 HEC errati consecutivi.



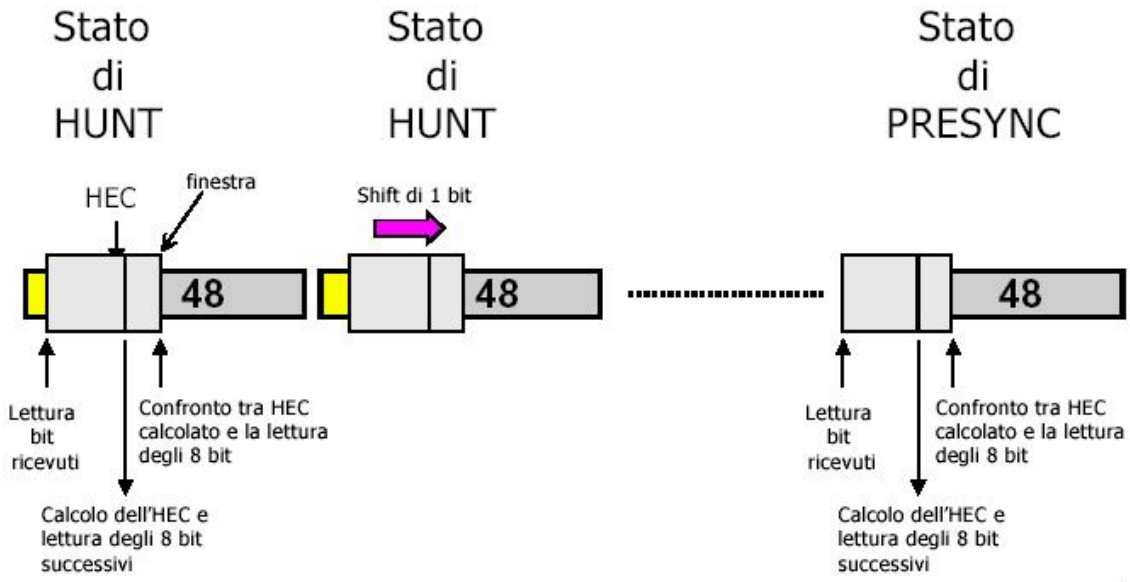
Cell delineation (1)

Il codice HEC consente di rivelare errori presenti nell'etichetta della cella ed opera anche con lo scopo di garantire l'allineamento di cella fra il trasmettitore ed il ricevitore (esempio: *router* di utente e nodo di rete).

Lo stato iniziale in cui si attiva il ricevitore è quello di ricerca. Il ricevitore cattura 4 *byte* e calcola il codice HEC come se i 4 *byte* ricevuti fossero l'etichetta di una cella. A tale scopo utilizza la stessa regola di calcolo standard impiegata dal trasmettitore. Se i due codici (il quinto *byte* ricevuto e il *byte* calcolato in base ai primi 4 ricevuti) coincidono, ciò che è stato ricevuto potrebbe essere l'inizio di una cella. Il passo successivo consiste nel memorizzare l'evento (cambio di stato e incremento di un contatore inizialmente a 0). Il ricevitore si sposta di cella in cella (e non più di *byte* in *byte*) ed effettua di nuovo il controllo sul codice HEC. Ad ogni buon esito del confronto, si ha permanenza nello stato di pre-sincronismo e l'incremento del contatore degli eventi. Se il conteggio giunge al valore 6 (standard), si dichiara agganciato il sincronismo di cella. Se il confronto fallisce, il contatore viene reimpostato a 0 e lo stato assunto è di nuovo quello di ricerca iniziale.

Dallo stato di sincronismo, 1 errore manifesto sull'etichetta di 7 celle consecutive farà perdere il sincronismo e determinerà il ritorno allo stato di ricerca iniziale.

La conseguenza è: *burst* di errori determinano la perdita del sincronismo di cella.



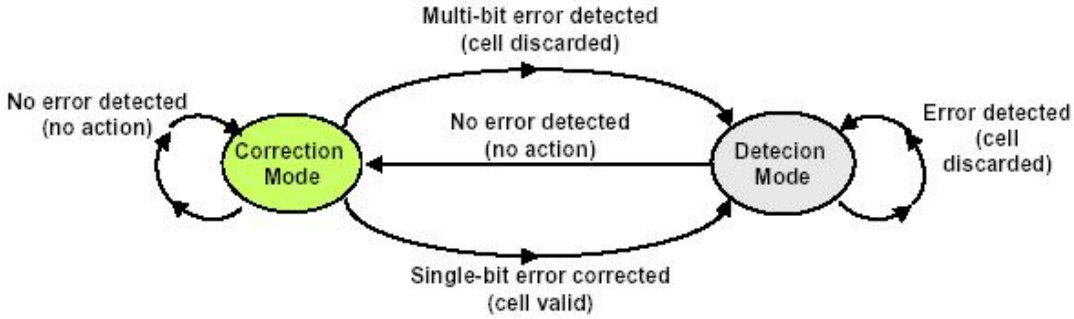
Cell delineation (2)

La figura illustra il meccanismo di scorrimento della finestra di 5 *byte*, prima bit a bit (stato HUNT) e poi cella a cella (stato PRESYNC).

HEC generation/verification

HEC *Generation* / *Verification*:

- TX: generazione del campo HEC delle celle utilizzando il polinomio generatore:
  - X8 + X2 + X + 1.
- RX: rilevamento errori multipli e correzione errori singoli:
  - Stato iniziale ed errori sporadici: *Correction Mode*;
  - *Burst* di errori: *Detection Mode*.



HEC generation/verification

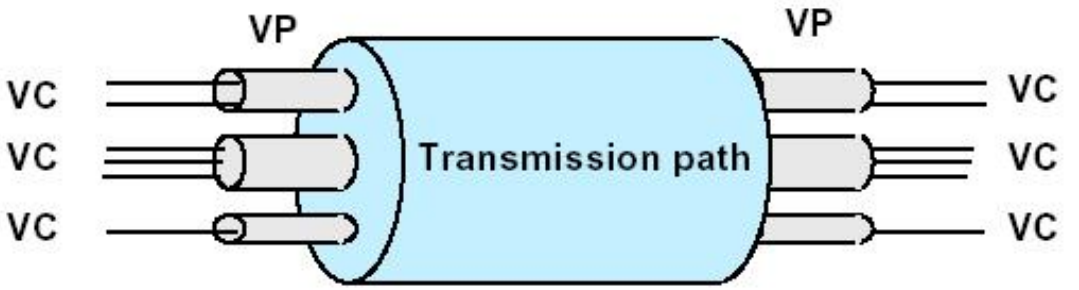
Per ciascuna cella ricevuta, il ricevitore controlla sempre la presenza di errori nell'intestazione.

Un errore singolo riscontrato nello stato *CORRECTION MODE* determina la correzione dello stesso errore ed il cambiamento di stato in *DETECTION MODE*.

Un singolo errore riscontrato sulla etichetta di una cella ricevuta durante la permanenza nello stato di *DETECTION MODE* determina l'eliminazione della cella e l'incremento del contatore di sincronismo.

Il ripristino dello stato di *DETECTION* si ha non appena si riceve una cella priva di errori di etichetta. Ciò determina la reimpostazione a 0 del contatore del sincronismo. La cella viene quindi considerata valida e passata agli strati di protocollo successivi (strato ATM).

Virtual Path e Virtual Channel



Virtual Path e Virtual Channel

VP = *Virtual Path*: rappresenta un flusso di celle, all'interno di un canale trasmissivo, caratterizzate da uno stesso valore di VPI.

VC = *Virtual Channel*: flusso di celle con lo stesso valore di VPI e VCI.

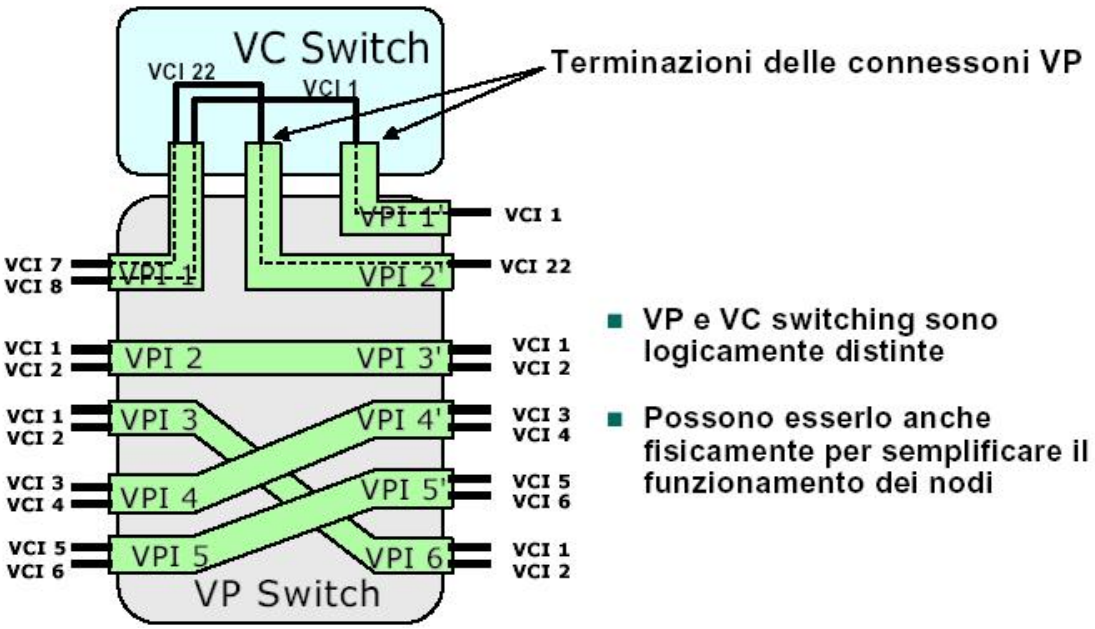
Un VP può contenere uno o più VC.

Il concetto di fascio di canali virtuali è analogo al gruppo di canale logico presente nell'architettura X.25.

In questo caso il VP può essere utilizzato per diversi scopi:

- come strumento di pianificazione a disposizione del gestore della rete pubblica, al fine di suddividere la banda trasmissiva dei link fisici internodali tra più fasci, almeno in prima approssimazione;
- come strumento di esercizio della rete, sempre per il gestore, al fine di affasciare insiemi omogenei di canali logici assegnati a differenti utenti (canali virtuali accomunati da caratteristicheparticolari), ad esempio:
  - originati da interfacce afferenti allo stesso nodo A e terminati tutti su interfacce dello stesso nodo B;
  - relativi a categorie di traffico simili;
- come *facility* per semplificare il trattamento dell'etichetta delle celle nei nodi della rete (nel senso che una cella appartenente ad una connessione VP viene elaborata per tutte le parti dell'etichetta ad eccezione del VCI che resta inalterato da estremo ad estremo).

VP/VC switch

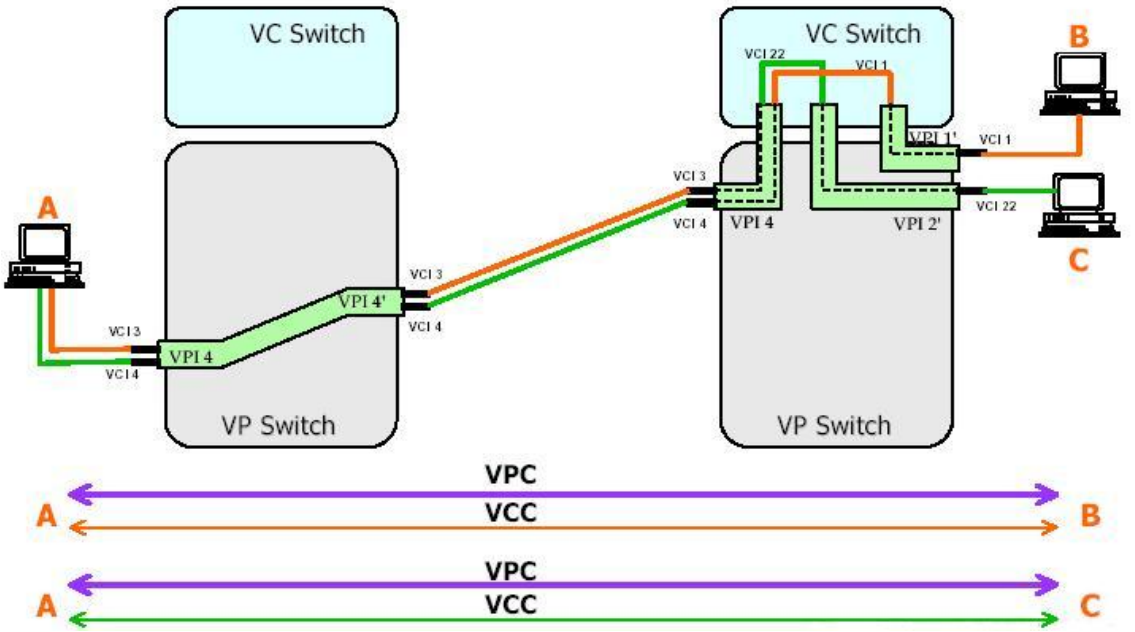


VP/VC switch (1)

- VP e VC *switching* sono logicamente distinte
- Possono esserlo anche fisicamente per semplificare il funzionamento dei nodi

In una connessione VP, i nodi interessati dal transito sono trasparenti rispetto all'attivazione e al consumo di banda da parte degli stessi VC. Inoltre, in un VP *switch*, le celle appartenenti ad una connessione VP vengono modificate solo nel campo VPI, lasciando inalterato il campo VCI. Si dice in tal caso che i VC sono di tipo *end to end*.

Un VC *switch* opera un controllo più fine, poiché modifica l'etichetta sia nel campo VPI sia nel campo VCI. Le funzioni di controllo del traffico vengono attuate per singolo VC.



VP/VC switch (2)

La tecnologia attuale consente di realizzare nodi (disponibili sul mercato già da diversi anni) in grado di realizzare sia le funzioni di VP *switch*, sia le funzioni di VC *switch*.

La differenza di operazioni risiede in opzioni di configurazioni che l'amministratore della rete imposta.

Perché VP e VC

- Ma perché due identificativi (VPI e VCI) e non uno solo (Il protocollo X.25 utilizza il Numero di Canale Logico - NCL, il FR utilizza il DLCI).
- L'utilizzo di due identificativi permette di aggregare più flussi di celle (VC) in un unico VP.
- Grazie all'aggregazione di VC in VP è possibile:
  - instaurare tra due terminali una connessione VP su base semipermanente e le connessioni VC su base chiamata (si pensi a due PABX connessi con una VPC);
  - fare una prima ripartizione della capacità trasmissiva (canale) di un collegamento tra vari VP e, successivamente, ripartire la capacità di ciascun VP fra diversi VC; ciò facilita la pianificazione e la progettazione della rete;
  - aggregare VC aventi caratteristiche simili (per esempio, in termini di QoS); ciò facilita la gestione della rete.

Qualità del servizio di rete ATM

L'utente può chiedere ad una rete ATM di trasferire i dati secondo diverse qualità di servizio

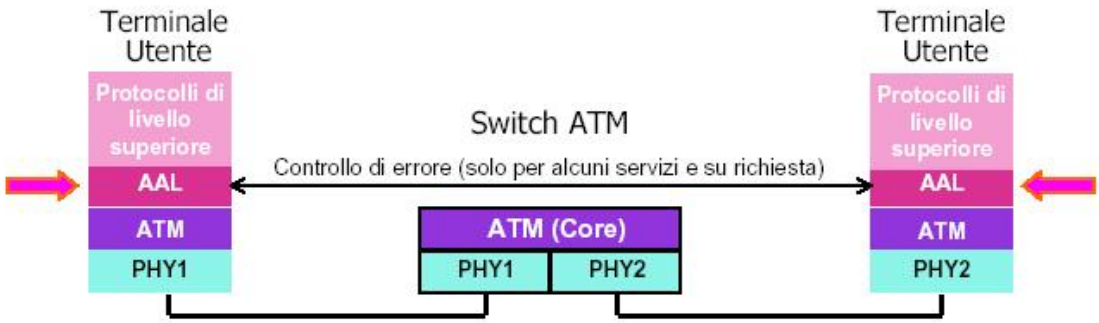
Sono realizzati due tipi di QoS:

- *Specified* QoS:
  - Garantisce le prestazioni in termini di ritardo massimo delle celle, variazione del ritardo e numero di celle perse.
  - Garantisce la banda in termini di *Peak Cell Rate*, *Sustained Rate*, *PeakBurst Length*.
- *Unspecified* QoS:
  - *Best Effort Delivery*.

ATM: lo strato di adattamento

L'indipendenza delle funzioni dello strato ATM dalle caratteristiche del teleservizio (applicazione d'utente) si ottiene con un ulteriore strato protocollare detto *ATM Adaptation Layer* (AAL).

Lo strato AAL svolge infatti funzioni specifiche per una certa applicazione (per esempio, controllo e recupero di errori multipli nelle applicazioni dati).



ATM: lo strato di adattamento (1)

Le varie applicazioni esistenti si basano in genere su protocolli di comunicazione diversi da ATM che impongono procedure e funzioni specifiche per l'adattamento al trasporto su celle di dimensione ridotta e fissa, quali la segmentazione ed il riassemblaggio dei pacchetti dati. Le applicazioni, inoltre, presentano caratteristiche peculiari e differenziate delle quali bisogna tener conto al fine di renderne efficiente il trasporto (non sempre si riesce ad ottemperare a questa esigenza). È quindi necessario definire alcuni protocolli intermedi in grado di tradurre messaggi in sequenze di celle e viceversa di riassemblare in messaggi significativi per i protocolli alti, le sequenze di celle ricevute. Dal punto di vista della stratificazione protocollare, l'AAL è posto tra il livello ATM ed il livello immediatamente superiore, in ciascuno dei piani: Utente, Controllo e Gestione.

Esempi dei servizi forniti dall'AAL comprendono: la gestione degli errori di trasmissione; la gestione dell'effetto di quantizzazione dovuto alla dimensione della cella ATM; la gestione della perdita o inserzione di celle; il controllo di flusso e della temporizzazione sorgente-destinazione.

In teoria, occorrerebbe definire uno strato AAL per ogni tipo di teleservizio. In realtà si è proceduto a raggruppare tutti i possibili teleservizi secondo criteri ben definiti.

Si sono così ottenute solo quattro Classi di Servizio (A, B, C, D) e, quasi corrispondentemente, quattro tipi di AAL:

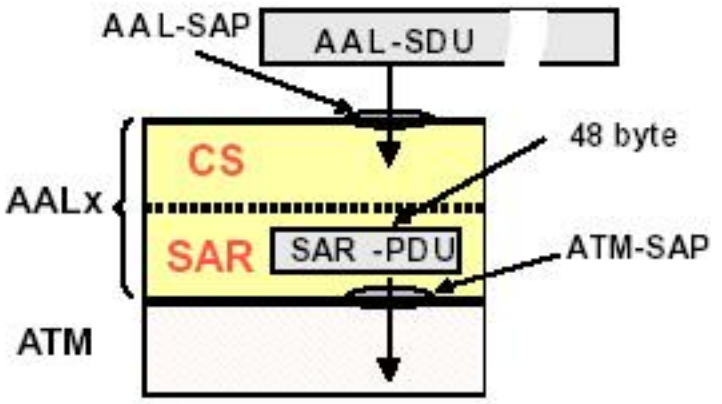
- AAL1;
- AAL2;
- AAL3/4;
- AAL5.

Una funzione comune a tutti gli AAL è quella di *segmentation and reassembling* (SAR).

Le funzioni del livello AAL sono organizzate in due sottolivelli denominati SAR (*Segmentation and Reassembly*) e CS (*Convergence Sublayer*).

Il SAR provvede alla segmentazione delle PDU del livello superiore nel campo informativo delle celle ATM (48 *byte*) e viceversa alla loro ricostruzione.

Il CS è dipendente dal servizio e fornisce, in corrispondenza dell'AAL-SAP, lo specifico servizio AALx.



ATM: lo strato di adattamento (2)

Classi di servizio e AAL

Le caratteristiche considerate per raggruppare i teleservizi sono:

- necessità di avere requisiti di tempo reale (per esempio, la fonia e il video);
- variabilità o meno del *bit rate* (per esempio alcuni servizi video sono a *bit rate* variabile);
- modalità di connessione (per esempio, i servizi IP sono *connectionless*).

Classe di Servizio	Classe A	Classe B	Classe C	Classe D
Relazione temporale tra sorgente e destinazione	Richiesta		Non Richiesta	
Bit rate	Costante		Variabile	
Modo di connessione	*Connection Oriented*			*Connectionless*
Tipo di AAL	1	2	3/4, 5	3/4, 5

Anche se teoricamente possibile, non sono stati sviluppati *Adaptation Layer* specifici per ciascun servizio. Le funzionalità di adattamento sono state definite in base alla classificazione dei vari servizi in un ristretto numero di categorie denominate classi di servizio. La definizione di queste classi è stata effettuata tenendo presente tre parametri:

- Relazione temporale tra sorgente e destinazione.
- Costanza/variabilità del *bit rate*.
- Modalità di connessione (riferita al servizio *end to end*, dal momento che la tecnica ATM prevede servizi di rete a connessione).

Classe A: relazione temporale tra sorgente e destinazione, *bit rate* costante, teleserviziorientati alla connessione; sono compresi in tale classe l'emulazione di circuito (trasporto su rete ATM di un segnale a velocità Nx64 kb/s, 2 Mb/s, 34 Mb/s, eccetera), voce, video a *bit rate* costante.

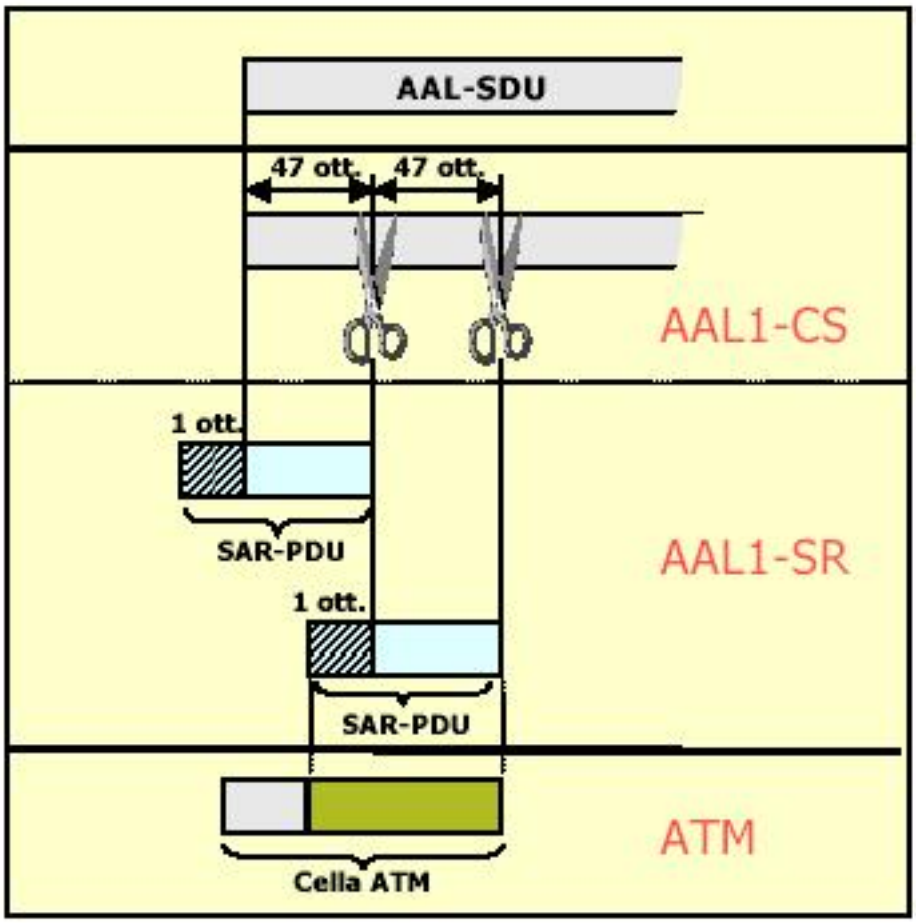
Classe B: relazione temporale tra sorgente e destinazione, *bit rate* variabile, teleserviziorientati alla connessione; la classe comprende le applicazioni video e audio a *bit rate* variabile (esempio: MPEG su ATM).

Classe C: assenza di relazione temporale tra sorgente e destinazione, *bit rate* variabile, servizio orientato alla connessione; la classe comprende teleservizi di trasporto dati *connection-oriented*, segnalazione.

Classe D: simile alla classe C, per teleservizi senza connessione; sono compresi teleservizi di trasporto dati *connectionless* (IP su ATM, *LAN emulation*, eccetera).

Per implementare le funzioni di adattamento necessarie al supporto delle quattro classi di servizio definite, l'ITU-T aveva originariamente individuato quattro differenti AAL (numerati da 1 a 4), ciascuno funzionalmente adatto al supporto di una specifica classe. A tale impostazione sono seguite una serie di modifiche. In particolare, per quanto riguarda le classi C e D, vista la loro somiglianza strutturale e funzionale, lo sviluppo dei rispettivi AAL è confluito in un unico protocollo denominato AAL 3/4; successivamente, vista la complessità dell'AAL 3/4, è stato definito un nuovo AAL, funzionalmente più semplice, denominato AAL 5.

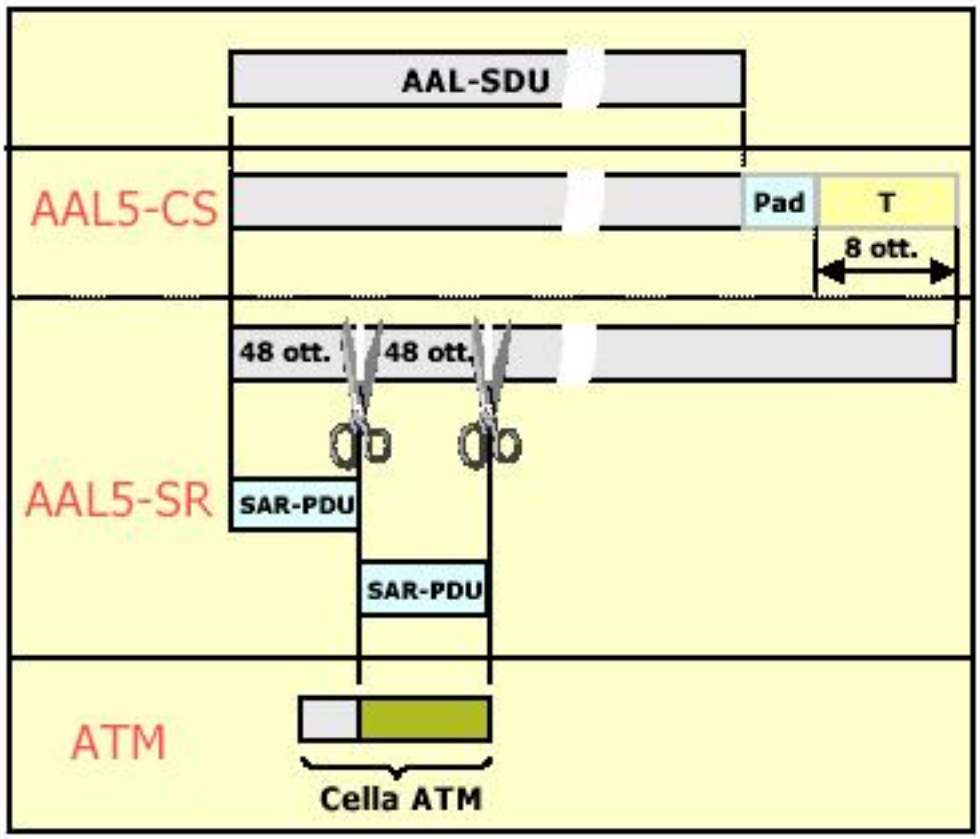
AAL 1



AAL 1

Tutti i tipi di AAL sono organizzati in due sottostrati di protocollo: il *Segmentation And Reassembly* (SAR), funzionalmente legato allo strato ATM, che provvede alla segmentazione delle PDU del sottostrato superiore in blocchi da 48 *byte* (*payload* delle celle ATM) e viceversa alla loro ricostruzione, e il *Convergence Sublayer* (CS), che è *service dependent*, in quanto le funzioni svolte dipendono dalla specifica classe di servizio. La Raccomandazione ITU-T I.362 descrive alcuni protocolli AAL, che consistono in combinazioni di funzioni SAR e CS, per il supporto di servizi appartenenti alle classi A..D. I servizi CBR utilizzano l'AAL di Tipo 1. Servizi di tipo *connectionless* utilizzano l'AAL di Tipo 3/4 o 5; altre funzioni necessarie a fornire il servizio CL (esempio indirizzamento e *routing* a livello di rete), devono essere svolte da un protocollo di convergenza di livello superiore. Nel caso degli AAL di tipo 3/4 e 5, è stata introdotta una ulteriore suddivisione, all'interno del livello CS, in due sottolivelli: *Common Part Convergence Sublayer* (CPCS) e *Service Specific Convergence Sublayer* (SSCS).

ALL 5



AAL 5

L'AAL 5 nasce da un'esigenza di semplificazione e di miglioramento dell'efficienza dell'AAL 3/4 (per semplicità non trattato in questo documento). In particolare, le caratteristiche salienti dell'AAL 5 sono:

- non prevede il *multiplexing* di messaggi: su ciascuna connessione ATM ogni messaggio non può essere intercalato con altri (il *multiplexing* può essere effettuato eventualmente da altri protocolli montati sull'AAL);
- utilizza un'unico bit per identificare le AAL-PDU nel flusso di celle;
- usa il CRC a livello CPCS anzichè a livello SAR (può essere uno svantaggio, nel momento in cui i supporti trasmissivi non sono caratterizzati da ottima qualità, cosa che del resto non consente a monte di impiegare ATM; lo svantaggio si verifica anche quando ciascuna *protocol data unit* è di lunghezza tale da far superare in trasmissione il limite stabilito nel contratto di traffico, ed in tal caso le celle di coda vengono scartate dalla rete, costringendo il trasmettitore a ripetere l'emissione di tutta la trama: il risultato è che il terminale sorgente non riesce a trasmettere);
- allinea le PDU a multipli interi del *payload* di una cella.

Parametri di QoS

La Raccomandazione Q.2931 (segnalazione B-ISDN) prevede 7 parametri che definiscono la QoS per una connessione ATM:

- *cell error rate*;
- *serious cell block errors*;
- *cell loss ratio*;
- *cell misinsertion rate*;
- *cell delay*;
- *mean cell transfer delay*;
- *cell delay variation*.

L'utente può chiedere ad una rete ATM di trasferire i dati secondo diverse qualità di servizio. In particolare per le reti ATM sono previsti due tipi di QoS:

- *Specified* QoS
  - Garantisce le prestazioni in termini di ritardo massimo delle celle, variazione del ritardo e numero di celle perse.
  - Garantisce la banda in termini di *Peak Cell Rate*, *Sustained Rate*, *PeakBurst Length*.
- *Unspecified* QoS
  - *Best Effort Delivery*.

La Raccomandazione Q.2931 (segnalazione B-ISDN) prevede lo scambio fra elementi della rete di 7 parametri che definiscono la QoS per una connessione ATM:

- *cell error rate*;
- *serious cell block errors*;
- *cell loss ratio*;
- *cell misinsertion rate*;
- *cell delay*;
- *mean cell transfer delay*;
- *cell delay variation*.

In una chiamata effettuata da un terminale per richiedere una SVC (connessione virtuale commutata) il mancato soddisfacimento di uno dei parametri specificati dal messaggio di chiamata determina il rifiuto della richiesta di connessione.

Alcuni parametri di QoS importanti e definibili tra due estremi (UNI) A e B di una connessione ATM sono:

- *Cell Loss Ratio* (CLR) = numero di celle perse tra le N inviate da A verso B in un tempo T diviso il numero totale N di celle inviate (ovviamente, il valore di CLR dipende da T).
- *Cell Transfer Delay* (CTD) = intervallo di tempo intercorrente tra l'istante di immissione del primo bit di una cella in A e l'istante di ricezione dell'ultimo bit della medesima cella in B.
- *Cell Delay Variation* (CDV) = esprime la variabilità dei ritardi di trasferimento da A a B (la sua valutazione è specificata dalla Raccomandazione I.371 e dalla *Traffic Management Specification* di ATM Forum).

Parametri di QoS

Descrittori di traffico

Si dividono in:

- Descrittore di traffico di una sorgente ATM (STD, *Source Traffic Descriptor*); è un set di parametri che descrivono le caratteristiche del traffico di una sorgente ATM:
  - Tipo e Categoria di Servizio;
  - *Peak Cell Rate* (PCR);
  - *Sustainable Cell Rate* (SCR);
  - *Maximum Burst Size* (MBS);
  - *Minimum Cell Rate* (MCR).
- Descrittore di traffico di una connessione ATM (CTD, *Connection Traffic Descriptor*); è un set di parametri che descrivono le caratteristiche proprie di una connessione:
  - Descrittori di traffico di sorgente ;
  - *Cell Delay Variation Tolerance* (CDVT);
  - Definizione di conformità.

Quindi il CTD contiene anche il STD.

Il Descrittore di Traffico (DdT) può essere riferito a una sorgente di traffico ATM o a una connessione virtuale ATM.

Il DdT di una sorgente ATM è costituito da un insieme di parametri di traffico che descrivono in modo opportuno le caratteristiche del traffico della sorgente ATM. Esso viene utilizzato durante la fase di instaurazione di una chiamata, per specificare le caratteristiche della connessione ATM richiesta.

Il DdT di una connessione ATM costituito dall'insieme di parametri che descrivono le caratteristiche proprie di una connessione. Esso include in genere:

- il DdT della sorgente;
- il valore di *Cell Delay Variation Tolerance* definito all'interfaccia UNI/INI;
- la definizione di conformità adottata per specificare il criterio di conformità delle celle alla connessione ATM.

Su queste informazioni si baseranno le procedure di accettazione della chiamata (CAC) per allocare le risorse di rete e derivare i parametri necessari alla funzione di *policing* UPC/NPC). Tutti i parametri di traffico contenuti in un descrittore di traffico pertanto dovrebbero essere:

- semplici e facilmente comprensibili da parte dell'*end system*;

- utili alla funzioni di CAC per l'allocazione del risorse di rete;
- utili alla rete per il raggiungimento e il mantenimento degli obiettivi di QdS;
- definiti in modo tale che la funzione di UPC/NPC li possa far rispettare.

Questi parametri costituiscono il riferimento in base al quale un terminale scambia informazioni all'atto dell'instaurazione di una connessione e successivamente invia traffico in rete. Essi devono inoltre consentire i test sul flusso di celle ATM utili alla verifica di conformità della connessione ATM.

Definizione di conformità

In una richiesta di connessione (VCC o VPC) si dichiarano certe caratteristiche di Traffico tramite i Descrittori di Traffico.

Per una richiesta di connessione accettata, la rete applica dei controlli sulla UNI (*Usage Parameter Control*) per verificare il rispetto dei dati di traffico dichiarati.

L'algoritmo usualmente utilizzato per il monitoraggio dei dati di traffico (per esempio PCR e CDVT) è il *Generic Cell Rate Algorithm*.

Se i dati di traffico non sono rispettati, la rete intraprende determinate azioni tendenti a mantenere la QoS di altre connessioni che rispettano invece i dati di traffico dichiarati.

Il Descrittore di Traffico di una connessione ATM deve sempre contenere una Definizione di Conformità (DdC).

La DdC si applica alle celle che, transitando attraverso una interfaccia UNI pubblica, vengono valutate in base ad uno specifico algoritmo per verificarne l'effettiva rispondenza al contratto di traffico; superati tali criteri medianti appositi test le celle sono dichiarate conformi.

La DdC costituisce uno strumento di verifica per l'utente di quanto gli viene reso disponibile dalla rete, oltre a proteggere la rete, mediante i sistemi di UPC/NPC da condizioni di traffico anomale.

La regola operativa generale adottata per la DdC è basata sull'algoritmo GCRA (*Generic Cell Rate Algorithm*), illustrato nel seguito.

Categoria di servizio

Category of Service (secondo ATM Forum) e *Transfer Capability* (secondo ITU-T) indicano entrambe diverse modalità di allocazione e gestione delle risorse (di elaborazione, di memoria e di capacità di trasferimento) da parte della rete per le connessioni ATM.

Tale differenziazione è stata fatta in modo complementare alle funzioni AAL con l'intento di ottimizzare l'impiego delle risorse di rete.

Sia ATMF che ITU-T hanno fatto una classificazione delle CoS/TC in base ai descrittori di traffico e ai parametri di QoS.

L'introduzione del concetto di Categoria di Servizio ATM (CoS) deriva dall'aver riconosciuto che le molteplici tipologie di connessioni che possono presentarsi in una

rete ATM non possono essere discriminate solo in base al valore di certi parametri di traffico o di QdS, ma possono richiedere l'impiego di parametri diversi e l'applicazione di funzioni di controllo diverse.

Le diverse CoS ATM sono state quindi introdotte per raggruppare le connessioni a cui può essere applicata la stessa descrizione parametrica e lo stesso insieme di funzioni di controllo.

Categorie di servizio vs Transfer capabilities

ITU-T ATM *Transfer Capability*	ATM Forum ATM *Service Category*
Deterministic Bit Rate (DBR)	*Constant Bit Rate* (CBR)
Statistical Bit Rate (SBR)	*Variable Bit Rate-real time* (VBR-rt) *Variable Bit Rate-non real time* (VBR-nrt)
Available Bit Rate (ABR)	*Available Bit Rate* (ABR)
ATM *Block Transfer* (ABT)	-
-	*Unspecified Bit Rate* (UBR)

Categorie di servizio attualmente definite per ATM (specifica ATM Forum UNI 4.0) sono:

- ***Constant Bit Rate* (CBR)**. Una connessione con classe di servizio CBR mette a disposizione una banda garantita (definita dal parametro PCR, *Peak Cell Rate*) per tutto il suo tempo di vita. È adatta al trasferimento affidabile di traffico *real-time* in quanto la rete offre prestazioni garantite in termini di CDV, max CTD e CLR. Tale classe si applica al traffico a *bit-rate* costante pari al PCR (esempio: *circuit emulation*) e trasmissione di traffico a *bit rate* variabile con banda di picco inferiore a quella allocata (sfruttamento non ottimale delle risorse).
- ***Real time Variable Bit Rate* (rt-VBR)**. È stata pensata esplicitamente per applicazioni *real-time* che generano traffico a *bit-rate* variabile cioé che necessitano di ritardi contenuti ed il più possibile costanti. La sorgente deve comunicare le caratteristiche del traffico generato in termini di PCR, SCR e MBS (*MaximumBurst Size*, dimensione massima del blocco di celle consecutive trasmesse). La rete offre una QoS garantita con riferimento a: tempo di trasferimento a destinazione (max CTD) e tasso di perdita sulla connessione (CLR). Tale categoria consente la multiplazione statistica di più flussi informativi e si adatta ad applicazioni di trasferimento di audio e video interattivi su ATM (esempio MPEG2 su ATM).
- ***non real time Variable Bit Rate* (nrt-VBR)**. Pensata esplicitamente per applicazioni non *real-time* che generano un traffico di tipo *bursty*. La sorgente deve comunicare le caratteristiche del traffico generato in termini di PCR, SCR e MBS. La rete offre una QoS garantita con riferimento al tasso di perdita sulla connessione (CLR). Non viene garantita alcuna prestazione con riferimento al tempo di trasferimento Le applicazioni possibili riguardano il trasferimento di traffico dati a *bit-rate* variabile (esempio traffico generato da WWW).

- ***Unspecified Bit Rate* (UBR)**. Una connessione UBR offre un servizio di trasporto dati di tipo *best-effort*. La sorgente può trasmettere un flusso di celle a *bit-rate* variabile fino ad una valore massimo specificato e pari al PCR. La rete non garantisce alcuna prestazione con riferimento al tasso di perdita ed al tempo di trasferimento delle celle. Le applicazioni riguardano il trasporto su ATM del traffico dati generato dai protocolli attualmente utilizzati in ambito di rete locale (esempio: *Ethernet*, IP).
- ***Available Bit Rate* (ABR)**. La categoria di servizio ABR offre un servizio di tipo *best-effort* controllato, in grado di sfruttare in modo più efficiente le risorse di rete. Tale categoria prevede un meccanismo di controllo di flusso mediante il quale la rete può sollecitare la sorgente a (la sorgente può stimolare la rete a): ridurre il *bit-rate* trasmesso in caso di congestione incrementare il *bit-rate* trasmesso (a valori specificati dalla stessa sorgente e comunque fino ad un valore massimo pari al PCR) se vi sono risorse disponibili. Nella fase di *set-up* della connessione può essere specificata anche la banda minima che si vuole sia garantita dalla rete (MCR).

Le applicazioni riguardano il trasporto su ATM dei protocolli (esempio: IP) attualmente utilizzati sulle reti locali di tipo tradizionale (esempio: *Ethernet*).

Categorie di servizio ATM Forum

Categorie di servizio ATM					
	CBR	Rt-VBR	Nrt-VBR	UBR	ABR
Parametri di Traffico					
PCR	Specificato			Specificato	Specificato
SCR, MBS	N/A	Specificati		N/A	
MCR	N/A				Specificato
Parametri di QoS					
CDV picco-picco	Specificato			Non Specificato	
CDT max	Specificato			Non Specificato	
CLR	Specificato			Non Specificato	Eventualmente Specificato

Constant Bit Rate (CBR): la rete assicura la disponibilità di un *data rate* costante per la durata della connessione; adatta per applicazioni che richiedono basso tasso di perdita delle celle e minimo ritardo (assoluto e sua variazione).

Variable Bit Rate - real time (VBR-rt): adatta per traffico con *data rate* variabile e *delay-sensitive* con requisiti di basso tasso di perdita di celle.

Variable Bit Rate - non real time (VBR-nrt): adatta per traffico con *data rate* variabile con requisiti di basso tasso di perdita di celle.

Unspecified Bit Rate (UBR): servizio di tipo *best-effort*; non è garantita alcuna capacità trasmissiva e qualsiasi cella può essere scartata.

Available Bit Rate (ABR): è garantito un minimo di capacità, con la possibilità di emettere, quando possibile, ad un *rate* superiore minimizzando la probabilità di perdita di celle.

Contratto di traffico

Quando si richiede una connessione (VCC o VPC) si specificano alla rete:

- le caratteristiche del traffico (Descrittori di Traffico della Connessione);
- i parametri di QoS per la connessione (non nel caso di UBR).

La rete, mediante la funzione di *Call Admission Control* (CAC) stabilisce se può accettare tale richiesta, cioè se può assegnare risorse alla connessione e mantenere nel contempo la QoS concordata per le altre connessioni.

Se la richiesta è accettata, la rete si impegna a garantire il servizio concordato in termini di traffico e QoS solo se l'utente rispetta i dati di traffico dichiarati.

Tutto ciò ha le caratteristiche di un contratto, detto appunto Contratto di Traffico.

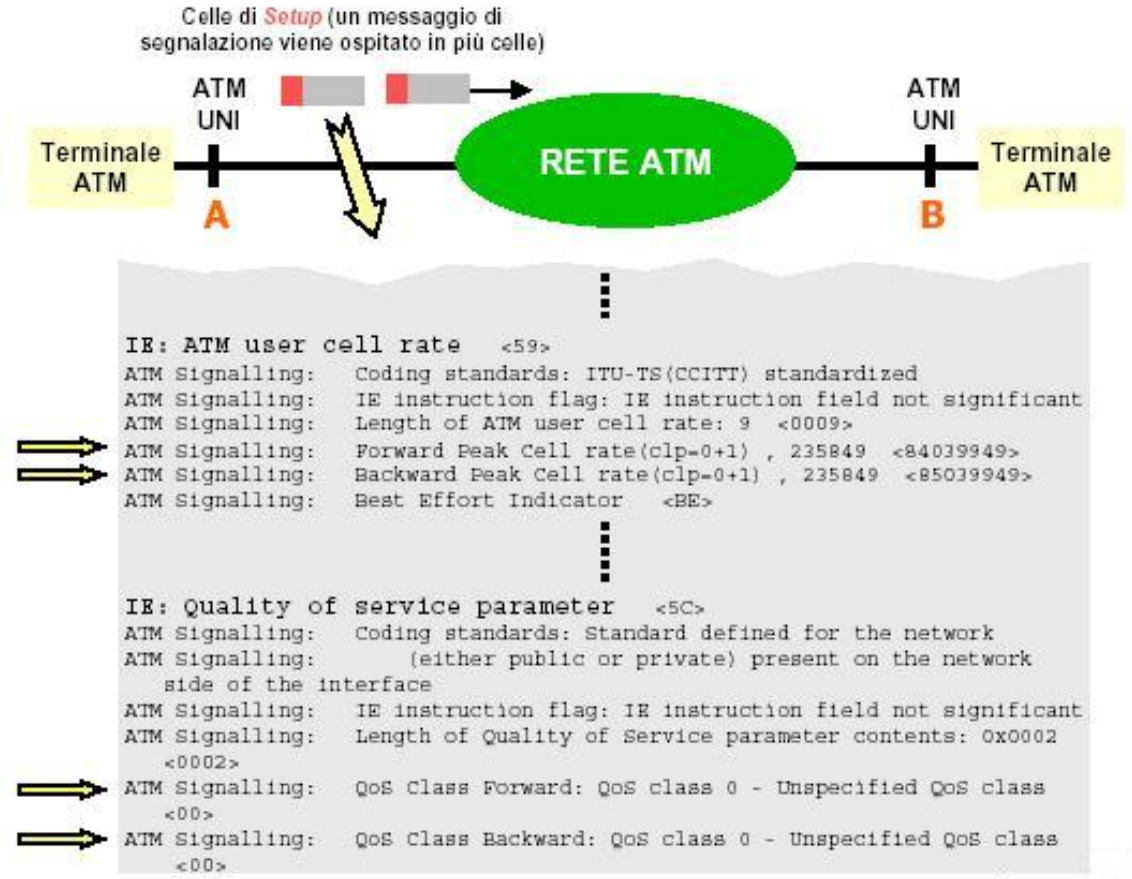
Il problema della gestione del traffico é molto importante nelle reti ATM in quanto il traffico trasportato può essere di natura eterogenea (audio, video, dati, eccetera) e l'utente può richiedere alla rete servizi a qualità (QoS) garantita e servizi di tipo *best-effort*. Gli obiettivi della gestione del traffico sono volti a garantire all'utenza la QoS concordata, ottimizzando l'utilizzo delle risorse di rete (banda, *buffer*, eccetera) e minimizzando la complessità degli apparati d'utente. Nella fase di *set-up* della connessione l'utente stipula con la rete un contratto di traffico (*Traffic Contract*) costituito dai seguenti elementi:

- ***Traffic Descriptor*** - insieme di parametri che definiscono le caratteristiche del traffico che sarà generato dalla sorgente.
- ***Requested QoS*** - insieme di parametri che definiscono le prestazioni che ci si attende siano garantite dalla rete.
- ***Conformance Definition*** - definizione della regola da utilizzare per stabilire quali celle siano conformi al *Traffic Contract*.

Le risorse di rete vengono allocate in modo da realizzare le prestazioni richieste fintanto che il traffico generato é conforme al *Traffic Contract*. I parametri di QoS attualmente definiti nella specifica UNI 4.0 di ATM Forum sono i seguenti:

- ***peak-to-peak Cell Delay Variation* (*peak-to-peak* CDV)** - massima tolleranza per la variabilità del tempo di trasferimento delle celle.
- ***Maximum Cell Transfer Delay* (max CTD)** - massimo tempo di trasferimento tollerato per le celle trasmesse sulla connessione.
- ***Cell Loss Ratio* (CLR)** - massimo tasso di perdita tollerato sulla connessione.

Esempio di richiesta di connessione (su segnalazione)



Esempio di richiesta di connessione (su segnalazione)

Controllo del traffico e della congestione

Il Controllo del Traffico serve per evitare l'insorgenza di fenomeni di congestione.

Il Controllo della Congestione serve per ridurre l'intensità, l'estensione e la durata della congestione.

Le funzioni di controllo (del traffico e della congestione) possono avere tempi di risposta su diverse scale temporali:

- tempo di inserzione di cella: reagiscono nel tempo di cella;
- tempo di propagazione A/R: reagiscono nel tempo di vita di una cella nella rete;
- durata della connessione: su questa base temporale si determina se una nuova connessione può essere accettata e con quali prestazioni (esempio: CAC);
- lungo termine: il controllo agisce su più connessioni.

Per controllo del Traffico si intende l'insieme di azioni intraprese dalla rete per evitare situazioni di congestione:

- ***Connection Admission Control* (CAC)**: insieme di azioni intraprese dalla

rete durante la fase di segnalazione per verificare se una nuova connessione chiesta su chiamata, possa essere accettata. La verifica avviene sulla base degli elementi del *Traffic Contract* specificati dall'utente. La chiamata viene accettata solo se sono presenti risorse sufficienti a garantire la QoS richiesta senza degradare la QoS delle connessioni già esistenti. Il CAC non fa parte della UNI e quindi i suoi schemi implementativi sono a completa discrezione del *Network Provider*.

- ***Usage Parameter Control* (UPC) o *policing***: insieme di azioni intraprese dalla rete per assicurare che il traffico offerto da un utente sia conforme al *Traffic Contract* negoziato nella fase di *set-up* della connessione. Il *policing* evita che comportamenti scorretti di un utente abbiano ripercussioni negative sulla QoS di connessioni appartenenti ad altri utenti. Il *policer*, localizzato alla UNI dal lato della rete (scheda ATM su cui risulta attestato l'utente), svolge due compiti:
  - verifica la validità del VPI/VCI di ogni cella;
  - verifica se il traffico entrante nella rete attraverso VCC/VPC valide stia violando i parametri negoziati nel *Traffic Contract*.
- ***Traffic Shaping***: é un meccanismo che consente ad un apparato di utente (terminale ATM oppure apparato *general purpose* con interfaccia ATM verso rete pubblica) di rendere conformi al *Traffic Contract* negoziato i flussi di celle che immette in rete. Lo *shaper* é localizzato alla UNI dal lato dell'utente ed adotta ovviamente lo stesso algoritmo del *policer*. Esempi di *traffic shaping*: abbassamento del PCR, limitazione della durata dei *burst* e opportuna spaziatura delle celle nel tempo.

Call Admission Control e Usage Parameter Control

Agisce su base durata della connessione.

Se la rete, mediante la funzione CAC, accetta una richiesta di connessione, attiva una funzione di monitoraggio dei dati di traffico (UPC, *Usage Parameter Control*).

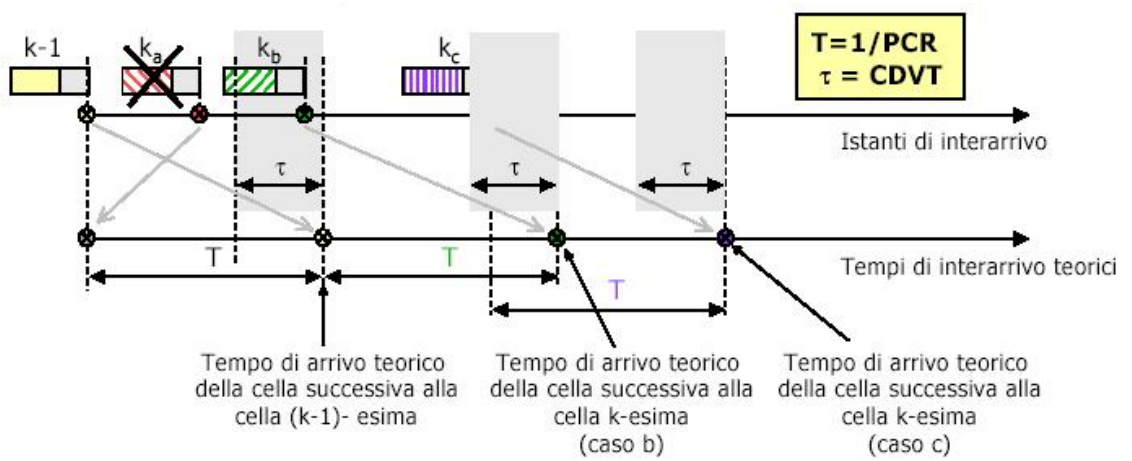
Se ci sono violazioni rispetto al contratto di traffico, la rete intraprende delle azioni tendenti, principalmente, a garantire la QoS alle connessioni attive e conformi al relativo Contratto di Traffico (per esempio scartando alcune celle della connessione non conforme) -> *Policing*.

Un primo tipo di UPC riguarda PCR e CDVT e l'algoritmo che lo implementa si chiama *Virtual Scheduling Algorithm*.

Lo stesso algoritmo si applica per il monitoraggio di SCR e MBS; per questo motivo l'algoritmo è stato chiamato anche *Generic Cell Rate Algorithm*.

Generic Cell Rate Algorithm

Consideriamo l'esempio di GCRA applicato al parametro PCR:



Generic Cell Rate Algorithm

N.B.: se una cella arriva come la k c il tempo di arrivo teorico viene calcolato a partire dall'effettivo tempo di arrivo della cella stessa; ciò serve per evitare la trasmissione di celle *back-to-back* (trasmissione di celle alla piena capacità del link trasmissivo) dopo periodi di *idle*.

L'algoritmo GCRA (*Generic Cell Rate Algorithm*) è la regola pratica adottata da ITU-T e ATM Forum per la verifica di conformità delle celle di una connessione ATM all'interfaccia UNI/INI pubblica. Tale algoritmo viene usato per effettuare la funzione di UPC/NPC. Anche se le due cose sono strettamente legate, è comunque importante distinguere i due concetti: la definizione di conformità, e le azioni conseguenti ad essa (*policing*).

L'algoritmo di GCRA utilizza in ingresso due variabili: una variabile incremento (I), ed una variabile limite (L): la prima tiene conto della velocità delle celle, mentre la seconda considera il massimo valore di CDV tollerato. Il risultato è dunque funzione dei valori assunti da queste due variabili:

GCRA = GCRA (I,L)

Nella Raccomandazione ITU-T I.371 sono definite due versioni equivalenti dell'algoritmo di GCRA:

- *Virtual Scheduling* (VS);
- *Continuous State Leaky Bucket*.

I due algoritmi sono equivalenti nel senso che, per ogni sequenza di istanti di arrivo di celle, essi individuano come o non conformi le stesse celle.

È opportuno precisare che il GCRA è utilizzabile per il controllo sia della frequenza di picco delle celle sia per quella media purché si programmino in modo opportuno i due parametri d'incremento e limite.

L'algoritmo GCRA applicato al parametro PCR permette di definire se una cella è conforme o meno ai dati di traffico dichiarati (PCR e CDVT).

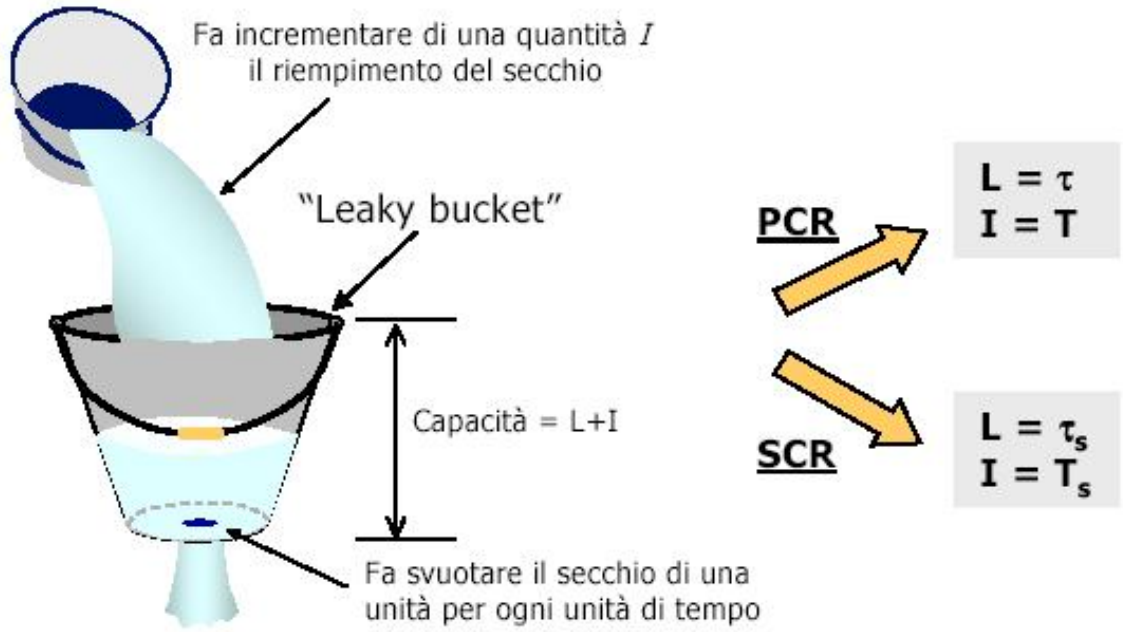
Inoltre, l'algoritmo GCRA applicato al parametro PCR pone un limite al numero massimo di celle trasmesse *back-to-back* alla piena capacità del link; tale numero è fissato dalla relazione seguente:

N = int [1 + t/(T - d)]

Dove:

- T = 1 / PCR;
- t = CDVT;
- d = tempo di trasmissione di cella alla velocità del link fisico;
- int [x] vuol dire il numero intero uguale o immediatamente inferiore a x.

Algoritmo del secchio bucato



Algoritmo del secchio bucato

L'algoritmo GCRA può essere espresso anche come algoritmo del secchio bucato (*leaky bucket*). L'algoritmo considera un contatore, il cui valor minimo è zero, che viene incrementato di una quantità pari ad I ogni volta che arriva una cella conforme e che viene decrementato di una unità per ogni unità di tempo. Il valore massimo del contatore è pari a (L+I); una cella che fa superare tale valore massimo viene dichiarata non conforme. Il nome dell'algoritmo deriva dal fatto che è equivalente al caso di un secchio bucato la cui capacità sia pari a (L+I) e che perda liquido con un tasso di una unità di capacità per unità di tempo e che riceva ad ogni immissione una quantità di liquido pari a I.

Policing

L'azione che consegue alla verifica di conformità delle celle nonché la verifica stessa costituiscono la funzione di *Policing* del traffico ATM.

Se, applicando l'algoritmo GCRA, una cella viene dichiarata non conforme, essa può essere:

- marcata (*tagging*);

- scartata (*discarding*).

Esistono diverse possibili alternative per applicare il *Policing*.

L'obiettivo del controllo del traffico a livello di cella è quello di garantire che il multiplatore soddisfi determinati requisiti di prestazioni o, in altre parole, sia in grado di offrire la QdS desiderata. Il problema è analogo a quello che si presenta in una qualsiasi rete a commutazione di pacchetto; deve però essere particolarizzato tenendo conto delle specificità delle reti ATM, in particolare:

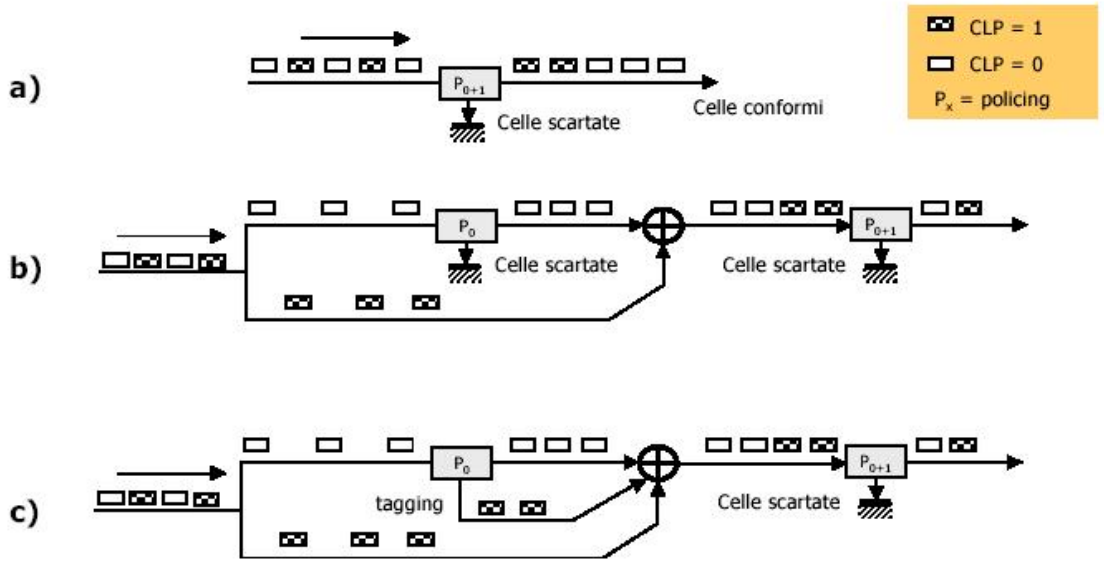
- la dimensione piccola e costante dei pacchetti;
- l'estrema variabilità delle caratteristiche che può avere il traffico generato dalle diverse connessioni;
- la necessità di garantire diversi gradi di QdS alle diverse connessioni.

Le particolarità della tecnica ATM hanno richiesto lo sviluppo di soluzioni specifiche per il controllo del traffico, che comprendono l'uso combinato di funzioni di controllo di tipo sia preventivo che reattivo.

Le funzioni di controllo preventivo hanno l'obiettivo di limitare a priori il traffico immesso nel multiplatore e sono rese indispensabili dalla necessità di soddisfare requisiti di QdS stringenti quali quelli richiesti da servizi con vincoli di tempo reale. Queste funzioni consistono essenzialmente nel valutare se il multiplatore possiede risorse sufficienti a supportare una nuova connessione sulla base delle caratteristiche del traffico offerto dalla connessione (controllo di accettazione delle connessioni), e nel verificare che il traffico immesso nel multiplatore non superi quello atteso (funzione di *policing*, ossia, controllo dei parametri di traffico). È possibile inoltre utilizzare particolari funzioni di sagomatura del traffico (*Traffic Shaping*) per dare al traffico in ingresso caratteristiche tali da ridurne la criticità per le risorse del multiplatore.

Le funzioni di controllo reattivo costituiscono invece un ulteriore strumento di protezione applicabile alle connessioni prive di vincoli di tempo reale nel caso si verifichino problemi di congestione. Queste funzioni consistono essenzialmente in controlli di flusso ad anello chiuso (*feedback*) che consentono di limitare il traffico inviato al multiplatore in modo da facilitare il ristabilirsi delle normali condizioni operative.

Alternative di policing



Alternative di policing

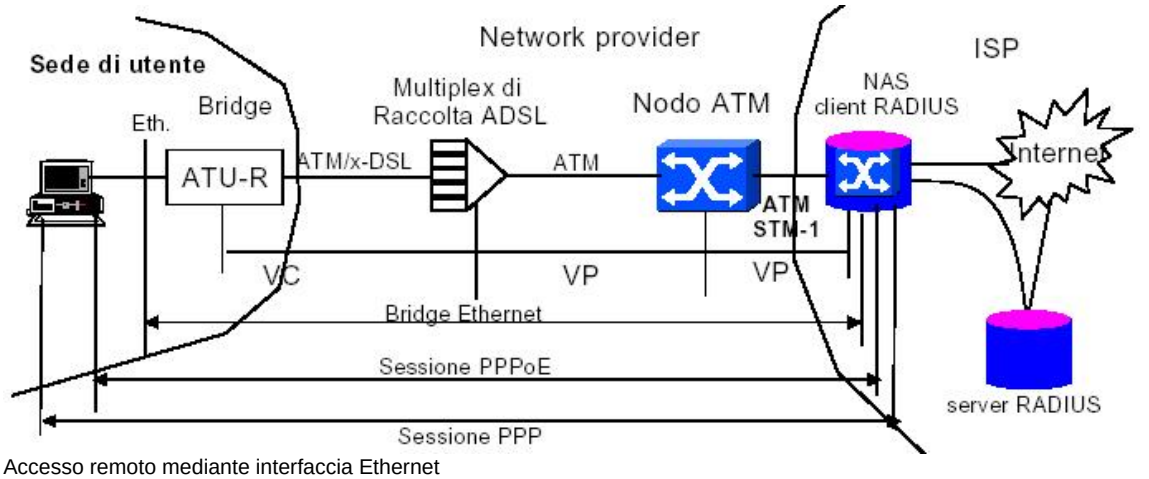
Il disegno si riferisce a tre possibili alternative di *Policing* del traffico ATM. Il caso a) è relativo all'applicazione del criterio di conformità al flusso aggregato di celle ATM (celle con CLP = 0 insieme a celle con CLP = 1).

I casi b) e c) si riferiscono alla possibilità di trattare in modo differenziato celle ad alta (CLP = 0) e a bassa priorità (CLP = 1). La differenza tra il caso b) e quello c) è che nel secondo le celle CLP = 0 non conformi vengono marchiate con CLP = 1 (*tagging*) e non scartate e subiscono un'altra verifica insieme alle celle che hanno CLP = 1 sin dall'origine.

Applicazioni di ATM per l'utente

- *Desktop* (una scheda ATM costa troppo rispetto ad una scheda LAN).
- Integrazione di servizi sul *backbone* di una rete *Corporate* (possibile, ad oggi ancora economicamente praticabile, poco utilizzato).
- Per trasporto di traffico IP (poco efficiente; il *celltax* non tollerato dall'utente).
- Se l'utente della rete ATM è un *Internet Service Provider* importante e non ha una sua struttura di rete capillare per la raccolta dell'utenza...

Accesso remoto mediante interfaccia Ethernet



Accesso remoto mediante interfaccia Ethernet

NAS = *Network Access Server* (tipicamente è un *router* fornisce l'indirizzo IP all'utente)
ATU-R = *ADSL Terminating Unit - Remote*)
ADSL = *Asymmetrical Digital Subscriber Loop*
RADIUS = standard per autenticazione, autorizzazione, *accounting* centralizzati
STM-1 = standard trasmissivo di tipo sincrono (*Synchronous Time-division Multiplexing*)
VC = *Virtual Channel*
VP = *Virtual Path*

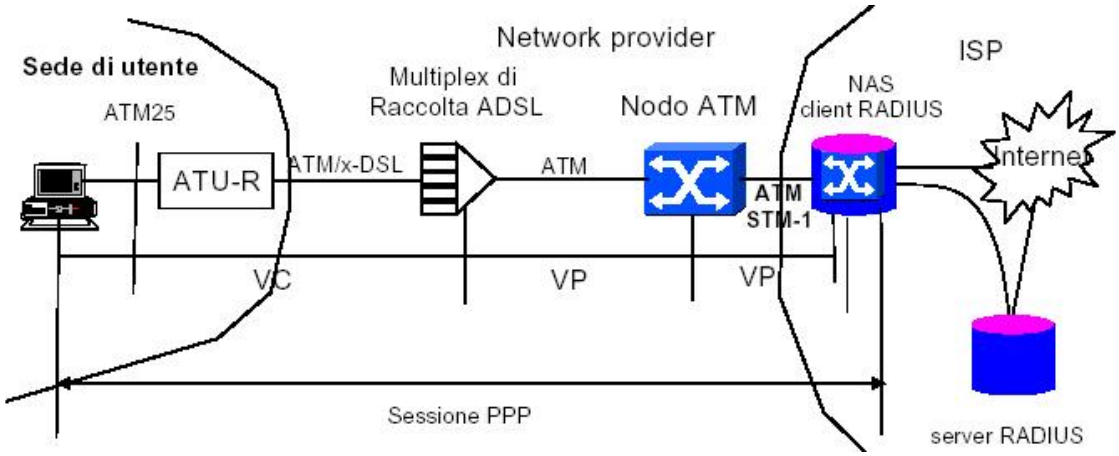
La figura illustra uno scenario di impiego di ATM come tecnologia per l'accesso a reti dati IP ad alta velocità, per utenti di tipo residenziale. In tale scenario si ipotizza l'esistenza di diverse entità:

- *network provider*.
- *Service provider*.

Il *Network provider* fornisce la connettività di tipo ATM fra il modem di utente e il punto di presenza del *service provider* designato da quest'ultimo.

Il *service provider* utilizza un servizio di rete di tipo ATM VP e raccoglie i singoli flussi a livello di VC provenienti dagli utilizzatori finali. I flussi dati vengono concentrati su un apparato del *service provider*, dove hanno luogo le funzioni di elaborazione a livelli più alti (autenticazione, assegnazione dell'indirizzo IP, *accounting*, eccetera).

Accesso remoto mediante interfaccia ATM25

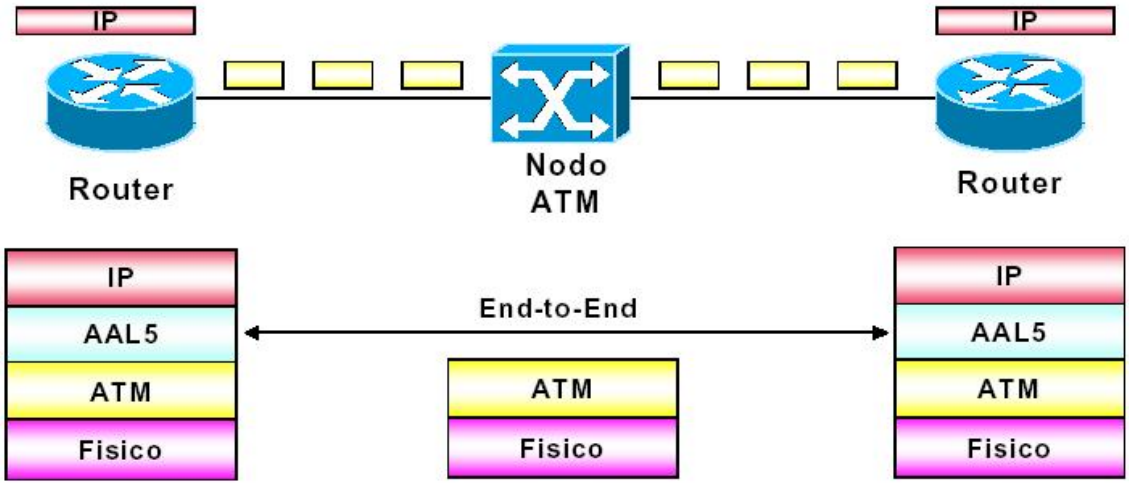


Accesso remoto mediante interfaccia ATM25

NAS = *Network Access Server* (tipicamente è un *router* fornisce l'indirizzo IP all'utente)
ATU-R = *ADSL Terminating Unit - Remote*)
ADSL = *Asymmetrical Digital Subscriber Loop*
RADIUS = standard per autenticazione, autorizzazione, *accounting* centralizzati
ATM25 = interfaccia ATM a 25Mbit/s
STM-1 = standard trasmissivo di tipo sincrono (*Synchronous Time-division Multiplexing*)
VC = *Virtual Channel*
VP = *Virtual Path*

Lo scenario di rete rappresentato è simile a quello precedentemente illustrato, con la differenza che in questo caso, l'*end system* di utente accede alla rete con una scheda ATM piuttosto che *Ethernet*.

IP su ATM



IP su ATM

Il trasporto di IP su ATM avviene attraverso dispositivi (*host*/*router*) dotati di interfaccia verso un nodo di commutazione ATM. Il pacchetto IP subisce un processo di

segmentazione in celle ATM che vengono trasportate fino al dispositivo successivo (*host*/*router*) da una rete ATM e qui i vari segmenti del pacchetto vengono riassemblati, il pacchetto viene ricomposto e consegnato al dispositivo di destinazione.

Due questioni molto importanti sono l'adattamento dei pacchetti IP allo strato ATM:

- l'imbustamento in celle ATM;
- la corrispondenza degli indirizzi IP con quelli ATM tramite traduzione.

L'imbustamento dei pacchetti IP nelle celle ATM avviene tramite uno strato di adattamento ATM (AAL). A tale scopo si utilizza sempre lo strato di adattamento AAL5, poiché questo introduce meno *overhead* rispetto agli altri strati AAL. Per distinguere il tipo di protocollo trasportato e permettere una corretta estrazione dei pacchetti imbustati ci sono due metodologie, VC *multiplexing* e LLC/SNAP.

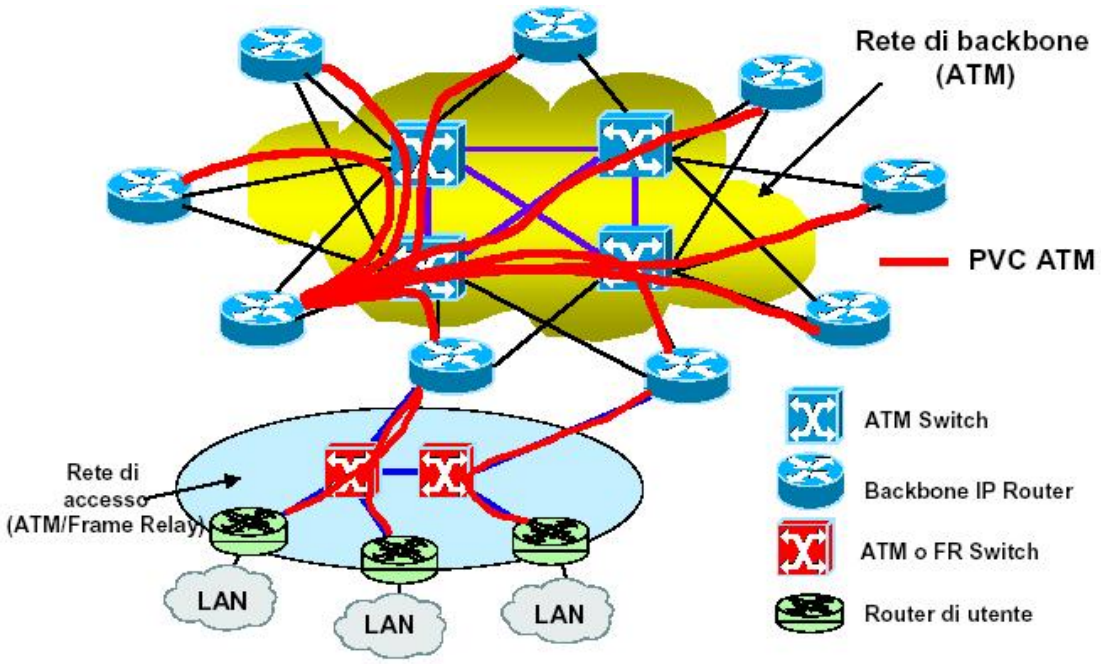
La corrispondenza degli indirizzi IP-ATM avviene o su base configurazione, oppure introducendo in rete un ARP *server* (ATMARP *server*).

Tra i vantaggi dell'utilizzo della tecnica ATM per il trasporto del protocollo IP si ricordano:

- alta velocità nel trasporto e nella commutazione dei pacchetti;
- possibilità di realizzare con ATM una piattaforma multiservizio non limitandosi al solo trasporto di flussi IP;
- possibilità di sfruttare i meccanismi di QdS ATM per offrire servizi differenziati;
- granularità di banda offerta da ATM;
- protezione e affidabilità offerta dal trasporto su infrastrutture ATM.

Il trasporto di IP su ATM ha però anche delle limitazioni, di cui si parlerà nel seguito, che mettono in dubbio per il futuro la possibile integrazione di queste due tecnologie.

IP classico su ATM: architettura di rete IP



IP classico su ATM: architettura di rete IP

Il modello IP classico su ATM ha riscosso un largo successo nella modalità con soli PVC grazie all'interesse per il supporto di ATM come *backbone* geografico di una rete IP. Diversi IBP (*Internet Backbone Provider*) di grandi dimensioni adottano ormai su larga scala collegamenti (virtuali permanenti) ATM per collegare i *router* di un *backbone* IP, sui quali gira il protocollo InATMARP per la risoluzione degli indirizzi IP sulle connessioni PVC ATM.

Nell'applicazione tipica, i *router* comunicano tra di loro attraverso un insieme di PVC ATM, che quindi funzionano come circuiti logici che garantiscono la connettività tra i *router* di *edge*. I *router* non conoscono la topologia fisica della rete, hanno conoscenza soltanto dei PVC, che appaiono quindi a loro come semplici collegamenti punto-punto. Viene attivato su ciascun PVC un protocollo di *routing* in modo che i *router* possano stabilire delle adiacenze (*peer relationships*) e scambiarsi le informazioni di *routing*. Tipicamente tutti i PVC sono di tipo UBR o ABR.

È da notare che se N è il numero dei *router* di *backbone*, nel caso si volesse magliare completamente la rete bisogna configurare N(N-1)/2 PVC (oltre ad eventuali necessari PVC di *backup*). Ciò comporta un elevato numero di adiacenze, che potrebbero portare ad uno stress del protocollo di *routing* IP in caso di fuori servizio contemporaneo di molti PVC (cosa che potrebbe facilmente verificarsi, ad esempio, con il fuori servizio di un nodo ATM).

Il modello IP classico su ATM è stato adottato dai grandi IBP a partire dalla metà degli anni '90, quando la richiesta di banda da parte degli ISP (*Internet Service Provider*) si è fatta sempre più pressante per rispondere al crescente volume di traffico Internet.

Intorno alla metà degli anni '90 quindi, gli IBP sono stati costretti a migrare le loro reti in modo da supportare link a velocità maggiore di 155 Mbit/s. Molti IBP hanno quindi realizzato reti utilizzando *router* con interfacce ATM SAR a 155 Mbit/s all'*edge* e *switch* ATM con interfacce a 155 Mbit/s nel *backbone* della rete.

Successivamente (dopo circa un anno) i link tra gli ATM *switch* del *backbone* sono stati portati a 622 Mbit/s, mentre oggi alcuni grandi IBP (esempio: UUNET) hanno portato i link internodali del *backbone* a 2,5 Gbit/s e stanno già preparando la transizione a 10 Gbit/s. Le interfacce trasmissive seguono gli standard SONET/SDH.

Routing IP

Franco Callegati
Paolo Zaffoni
8.3.1 (Definire le componenti software fondamentali di una WAN)

Tecnica

La funzione fondamentale dell'**instradamento** (*routing*) consiste nell'inoltro (*forwarding*) di pacchetti ed avviene generalmente in modalità *store-and-forward* (memorizza ed inoltra). La necessità di ricevere completamente il pacchetto prima di ritrasmetterlo introduce un tempo di latenza pari al tempo di trasmissione.

Le tecniche fondamentali di inoltro, che differiscono per il metodo di analisi del problema instradamento, sono le seguenti:

- *Routing by network address*. L'indirizzo di un sistema, che deve essere univoco sulla rete, è scritto direttamente nel pacchetto. Gli **IS** (*Intermediate System*) usano tale indirizzo come chiave di ricerca nella loro tabella di instradamento e determinano lungo quale cammino il pacchetto debba essere ritrasmesso. Tale tecnica è usata nei ***transparent-bridge*** (livello OSI 2), e in IP. È in generale adottata dai protocolli non connessi.
- *Label swapping*. È generalmente usata nei protocolli connessi e trova applicazioni in **ATM**. Ogni pacchetto è marcato con una *label* che serve come chiave in una tabella di instradamento sull'IS. L'IS, prima di ritrasmettere il pacchetto, sostituisce la *label* con una nuova *label*. Le *label* devono quindi essere univoche solo all'interno di un dato link. Se il protocollo è connesso, le *label* altro non sono che gli identificativi delle connessioni.
- *Source routing*. È una tecnica usata tramite una opzione del protocollo IP (per esempio, dai *bridge Token Ring*). Nel *source routing* la lista degli IS da attraversare, è scritta nel pacchetto dal nodo mittente, che lo chiede ad un IS o lo scopre con meccanismi di "*route location*".

La tecnica presa in esame in questa trattazione sarà la prima, poiché è quella adottata negli schemi di instradamento IP, e quindi integrata nei protocolli e nei *router* IP.

Definizioni

Con la dicitura rete fisica si indica un insieme di calcolatori aventi le interfacce di rete attestate su una stessa sottorete, in cui una particolare tecnologia di trasporto assicura la connessione. Una rete logica è l'insieme delle interfacce, a cui è stato assegnato lo stesso indirizzo di ***'subnet'***, che possono comunicare senza dover passare attraverso un ***router*** (instradatore).Tale condizione viene detta di *routing* implicito. IP assumeva originariamente una corrispondenza biunivoca tra reti fisiche e logiche; realizzazioni più moderne ammettono anche più reti logiche nella stessa rete fisica. Il *routing* tra reti logiche diverse è esplicito ed è gestito dai *router* tramite tabelle di instradamento.

IP adotta i concetti di destinazioni dirette e indirette nella sua logica di *routing*. Un *host* diretto è una stazione collegata direttamente alla rete ed al *router* della rete, mentre un *host* indiretto è un *host* di destinazione situato su una rete diversa da quella dell'*host* di

origine; questo significa che il **datagramma** deve essere inviato ad un *router* intermedio prima di essere consegnato all'*host* di destinazione.

Il modo in cui IP gestisce gli indirizzi e decide i percorsi di *routing*, richiede che una macchina esamini solo la parte di indirizzo di rete dedicata all'indirizzo di destinazione, per determinare se l'*host* di destinazione è collegato direttamente o indirettamente alla rete dell'*host* di origine: in altri termini, la macchina verifica la corrispondenza della parte 'rete' dell'indirizzo di destinazione e sceglie se effettuare un *forwarding* diretto o *forwarding* indiretto.

Forwarding diretto: la trasmissione di un datagramma IP tra due *host* connessi su una singola rete logica IP (stesso *netid*): non coinvolge i *router*. Il trasmettitore incapsula il datagramma nel frame fisico e lo invia direttamente all'*host* destinatario.

Forwarding indiretto: i datagrammi passano da un *router* all'altro finché non raggiungono un *router* che può trasmetterli direttamente. I *router* realizzano l'interconnessione tra le diverse reti.

Classificazione

Gli **algoritmi di *routing*** possono essere classificati per tipo:

- ***Statici*** o ***dinamici***: negli algoritmi statici, le tabelle di *routing* che vengono memorizzate sono compilate da una persona (amministratore di rete) e i valori di tali tabelle non cambiano per nessun motivo fino a quando l'amministratore di rete non li cambia, mentre negli algoritmi dinamici le tabelle vengono continuamente aggiornate e cambiate a seconda dei cambiamenti della rete (caduta di una rete, inserimento di una rete).
- *Gerarchici*: i *router* gerarchici hanno funzioni diverse da quelli che non lo sono, in quanto vengono suddivisi più nodi in gruppi logici chiamati domini di *routing*, *autonomous system* o aree. Solo alcuni di questi *router* possono interagire con ulteriori *router* di altri domini di *routing*, mentre altri possono interagire con *router* appartenenti allo stesso dominio.
- ***Link-State*** o ***Distance-Vector***: *link-state* (conosciuto anche come *shortest path first*) trasferisce tutte le informazioni di *routing* a tutti i nodi: ogni *router* invia solo la porzione di tabella che descrive lo stato dei suoi link. Gli algoritmi del tipo *distance-vector* inviano tutta o parte della tabella ai soli router vicini. Quindi *link-state* spedisce piccoli aggiornamenti a tutti, *distance-vector* spedisce grossi aggiornamenti ma solo ai *router* vicini: i *link-state* richiedono più risorse *hardware* (CPU e memoria) rispetto ai *distance-vector*, ma sono meno propensi ai *routing loop*.

Tabella di instradamento

Ogni *router* contiene una tabella di instradamento, visto che se un pacchetto viene destinato al *router* questo dev'essere instradato. Ogni riga nella tabella deve contenere almeno i seguenti tre elementi:

- un indirizzo di destinazione: il *router* può avere più di un percorso per la stessa destinazione.

- l'interfaccia su cui inoltrare i pacchetti.
- il costo per raggiungere la destinazione sul percorso, che inizia con l'interfaccia indicata nella riga.

Il costo consentirà all'IS di scegliere tra eventuali percorsi alternativi; l'unità di misura di questo costo dipende dal protocollo utilizzato. Si indicano genericamente con il termine *route* le informazioni presettate su una riga della tabella di *routing*.

Quando il *router* deve inoltrare un pacchetto, scorre la tabella per individuare la riga corrispondente al destinatario del pacchetto stesso. Mediamente il tempo di ricerca (*table lookup*) è pari alla metà del numero di righe. Considerando che tale operazione viene eseguita ogni volta che si deve inoltrare un pacchetto, diventa molto critica la complessità della tabella ai fini delle prestazioni dell'apparato.

Routing

Nel ***routing by network address***, la ricerca non verrà basata sull'intero indirizzo del destinatario, ma su un prefisso, molto spesso di lunghezza variabile. La ricerca dovrà essere eseguita nei confronti di quella riga che specifica il *route* con più lungo prefisso comune all'indirizzo del destinatario (*longest prefix matching*).

Affinché i pacchetti arrivino a destinazione è indispensabile che le tabelle nei vari IS siano coerenti tra di loro, al fine di evitare l'invio di pacchetti in percorsi ciclici (*routing loop*). In tal caso i pacchetti girerebbero a vuoto, consumando inutilmente risorse computazionali e trasmissive dei vari *router*.

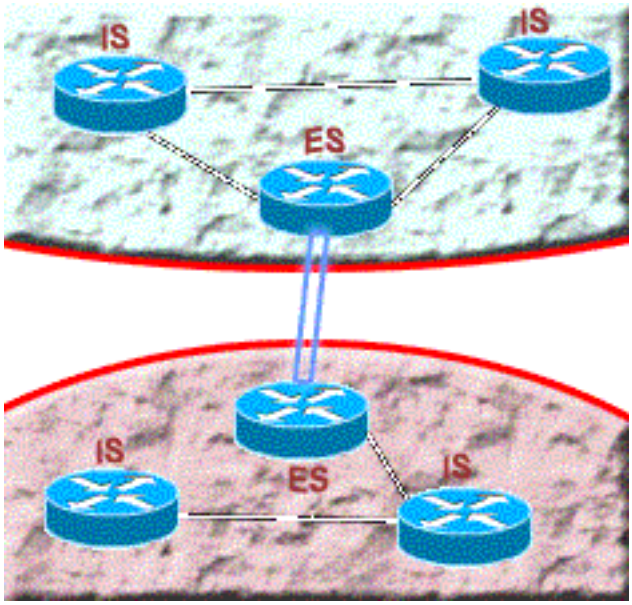
Dal percorso lungo il quale un pacchetto viene inoltrato, dipendono il ritardo che esso subirà, la probabilità che venga scartato a causa di eventuali congestioni del IS e il fatto che esso raggiunga o no la destinazione. Inoltre se la rete contiene maglie, una destinazione potrà essere raggiunta attraverso una o più percorsi alternativi; in presenza di guasti, la scelta di un percorso che eviti nodi o collegamenti non funzionanti consentirà alla rete di continuare a recapitare dati.

Dunque la scelta del percorso, cioè il ***routing***, sarà un fattore chiave per il buon funzionamento della rete e per la sua robustezza (*fault-tolerance*).

Neighbor greetings

Un altro problema è rappresentato dal ***neighbor greetings***, cioè del fatto che gli IS collegati ad una **LAN** devono conoscere gli ES collegati alla stessa LAN e viceversa. Questo è indispensabile per due motivi:

- gli IS devono conoscere gli ES per inserirli nelle tabelle di instradamento e propagare l'informazione della loro raggiungibilità agli altri IS;
- gli ES devono conoscere gli IS presenti sulla LAN per sapere a chi inviare i messaggi non destinati a nodi collegati alla stessa LAN.



La soluzione a quest'ultimo problema deve essere tale da ammettere LAN prive di *router*, LAN con un solo *router* o LAN con più *router*.

Routing statico

Il *routing* statico prevede che i percorsi di inoltro dei pacchetti siano determinati ed eventualmente cambiati dall'amministratore della rete tramite la configurazione degli apparati di *internetworking*. Quindi la tabella di *routing* è sotto la responsabilità dell'amministratore che deve gestire l'inserimento, la modifica e l'eliminazione delle righe.

Principalmente si ha lo svantaggio della mancanza di reattività ai cambiamenti topologici della rete, ed i percorsi non saranno automaticamente adattati alle variazioni dello stato di funzionamento di nodi e collegamenti.

Il ***fixed directory routing*** e il ***flooding***, sono le principali tecniche di *routing* statico.

Routing statico: fixed directory routing

Il *fixed directory routing* prevede che ogni nodo abbia una tabella di instradamento che metta in corrispondenza il nodo da raggiungere con la linea da usare, e che tale tabella sia scritta manualmente dal gestore della rete nel *router* tramite un'operazione di *management*.

Il gestore ha il totale controllo dei flussi di traffico sulla rete, ma è necessario un suo intervento manuale per il reinstradamento di detti flussi in presenza di guasti. Questo approccio è spesso utilizzato in TCP/IP per le parti non magliate della rete e le regole di instradamento specificate su ogni singolo *router* prendono il nome di *route* statiche.

Esiste una variante, detta quasi-statica, che adotta tabelle con più alternative da scegliere secondo un certo ordine di priorità, in funzione dello stato della rete. Questo

approccio, che consente di avere cammini alternativi in caso di guasto, è adottato, ad esempio, dalla rete SNA.

Occorre comunque evidenziare che la gestione manuale delle tabelle risulta molto complessa e difficoltosa, soprattutto per reti di grandi dimensioni.

Routing statico: flooding

Il *flooding* è un altro algoritmo non adattativo, in cui ciascun pacchetto in arrivo viene ritrasmesso su tutte le linee, eccetto quella su cui è stato ricevuto. Concepito per reti militari a prova di sabotaggio, se realizzato nel modo sopra descritto massimizza la probabilità che il pacchetto arrivi a destinazione, ma induce un carico elevatissimo sulla rete.

Si può cercare di ridurre il carico utilizzando tecniche di selective *flooding*, in cui i pacchetti vengono ritrasmessi solo su linee selezionate. Un primo esempio, senza applicazioni pratiche, è l'algoritmo *random walk* che sceglie in modo pseudo-casuale su quali linee ritrasmettere il pacchetto. Una miglioria più efficace si ha scartando i pacchetti troppo vecchi, cioè quelli che hanno attraversato molti *router*: a tal scopo nell'***header*** del pacchetto viene inserito un *age-counter*.

Un'ultima miglioria, ancora più significativa, consiste nello scartare un pacchetto la seconda volta che passa in un nodo: in tal modo si realizza una tecnica per trasmettere efficientemente la stessa informazione a tutti i nodi, qualsiasi sia la topologia. Lo svantaggio è che bisogna memorizzare tutti i pacchetti su ogni nodo per poter verificare se sono già passati.

Routing dinamico

Negli algoritmi di instradamento dinamico (adattativo) le tabelle dipendono dalle informazioni raccolte sulla topologia della rete, sul costo dei cammini e sullo stato degli elementi che la compongono. Gli algoritmi adattativi possono essere **centralizzati**, **isolati** o **distribuiti**.

Routing dinamico centralizzato

Il *routing* centralizzato è quello che più si avvicina al *fixed directory routing*. Presuppone l'esistenza di un **RCC** (**R**outing **C**ontrol **C**enter) che conosce la topologia della rete, riceve da tutti i nodi informazione sul loro stato e su quello dei collegamenti, calcola le tabelle di instradamento e le distribuisce.

È un metodo che consente una gestione della rete molto accurata, in quanto permette di calcolare le tabelle anche con algoritmi molto sofisticati, ma implica l'esistenza di un unico gestore, ipotesi questa oggi molto spesso non realistica.

Il RCC, per ragioni di affidabilità, deve essere duplicato e la porzione di rete intorno ad esso è soggetta ad un elevato volume di traffico di servizio: informazioni di stato che arrivano al RCC e tabelle di instradamento che escono dal RCC.

In caso di guasti gravi possono verificarsi situazioni in cui il RCC perde il contatto con una parte periferica della rete e si verificano quindi degli aggiornamenti parziali di

tabelle che possono determinare situazioni di *loop*.

Routing dinamico isolato

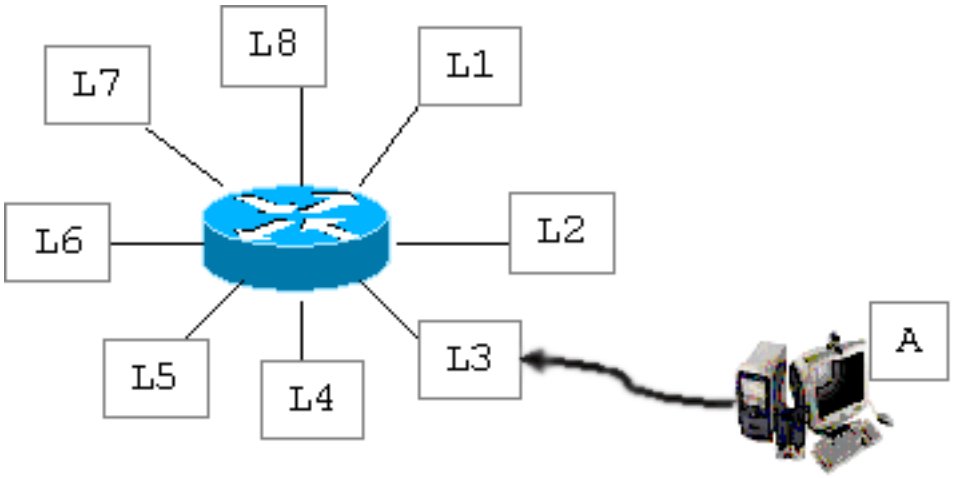
Ogni IS calcola in modo indipendente le tabelle di instradamento senza scambiare informazioni con gli altri IS. Esistono due algoritmi di *routing* isolato riportati in letteratura: "*hot potato*" e "*backward learning*".

hot potato

Ogni IS considera un pacchetto ricevuto come una patata bollente e cerca di liberarsene nel minor tempo possibile, ritrasmettendo il pacchetto sulla linea con la coda di trasmissione più breve.

backward learning

L'IS acquisisce una conoscenza indiretta della rete analizzando il traffico che lo attraversa: se riceve un pacchetto proveniente dal nodo A sulla linea L3, il *backward learning* impara che A è raggiungibile attraverso la linea L3. è possibile migliorare il *backward learning* inserendo nell'*header* del pacchetto un campo di costo inizializzato a zero dalla stazione mittente ed incrementato ad ogni attraversamento di un IS. In tale modo gli IS possono mantenere più alternative per ogni destinazione, ordinate per costo crescente.



Il limite di questo metodo consiste nel fatto che gli IS imparano solo le migliorie e non i peggioramenti nello stato della rete: infatti se cade un link e si interrompe un cammino, semplicemente non arrivano più pacchetti da quel cammino, ma non giunge all'IS nessuna informazione che il cammino non è più disponibile. Per tale ragione occorre limitare temporalmente la validità delle informazioni presenti nelle tabelle di instradamento: ad ogni *entry* viene associata una validità temporale che viene inizializzata ad un dato valore ogni volta che un pacchetto in transito conferma l'*entry*, e decrementata automaticamente con il passare del tempo. Quando la validità temporale di un'*entry* giunge a zero, questa viene invalidata ed eliminata dalla tabella di instradamento. Qualora ad un IS giunga un pacchetto per una destinazione ignota, l'IS ne fa il *flooding*. Il *backward learning* può generare *loop* su topologie magliate, per cui, ad esempio nei *bridge*, lo si integra con l'algoritmo di *spanning tree* per ridurre la

topologia magliata ad un albero.

Questo metodo metodo è utilizzato per calcolare le tabelle di instradamento nei *bridge* conformi allo standard IEEE 802.1D.

Routing dinamico distribuito

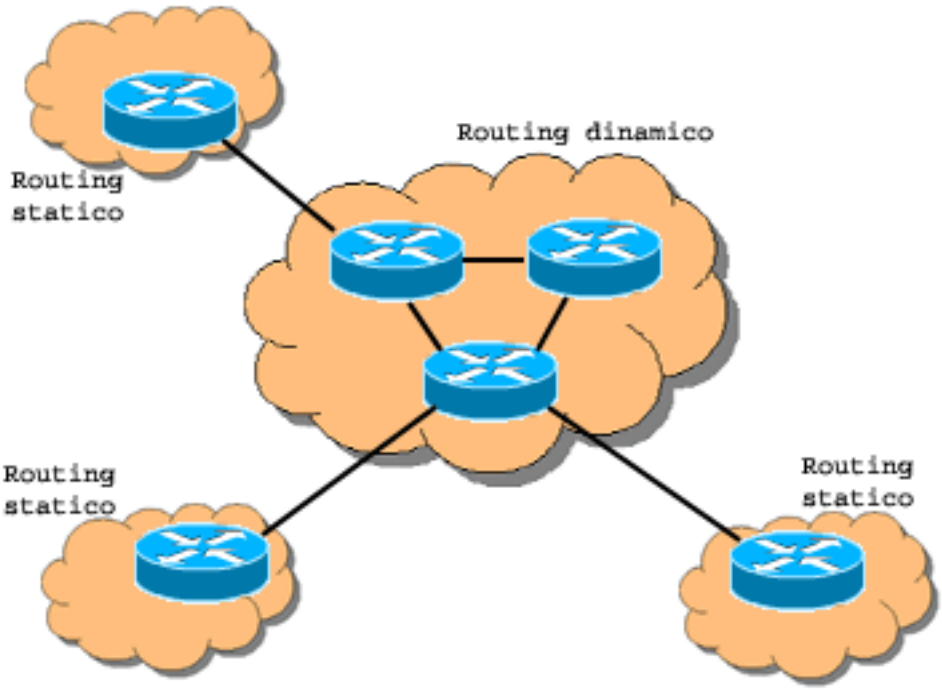
Il *routing* distribuito si pone come una scelta di compromesso tra i due precedenti: non esiste un RCC, ma le sue funzionalità sono realizzate in modo distribuito da tutti gli IS della rete, che a tal scopo usano un protocollo di servizio per scambiare informazioni tra loro ed un secondo protocollo di servizio per scambiare informazioni con gli ES. Tali protocolli vengono detti di servizio in quanto non veicolano dati di utente, che sono gestiti da un terzo protocollo, ma solo informazioni utili al calcolo delle tabelle di instradamento e al ***neighbor greetings***.

Le tabelle di instradamento vengono calcolate a partire dai due **parametri di ottimalità** (costo e *hop*). Il costo di ciascuna linea di ciascun *router* è un parametro che viene impostato dal *network manager* tramite il *software* di gestione dei *router* stessi.

Gli algoritmi di *routing* distribuito sono oggi adottati da DECnet, TCP/IP, OSI, APPN, ecc., e si suddividono ulteriormente in due famiglie: algoritmi ***distance vector*** e algoritmi ***link state*** *packet*.

Routing statico e dinamico: dove impiegarli

Dovendo trarre delle conclusioni, il ***routing*** **statico** meglio si adatta a situazioni periferiche di dimensione ridotta; dove cioè sia semplice gestire l'intera rete. Al contrario dove siano possibili frequenti cambi di topologia, variazioni di prestazioni in dipendenza da carico di lavoro del *router* e dei link, guasti, percorsi magliati ... allora sarà bene utilizzare ***routing*** **dinamico**.



Algoritmi e protocolli di routing: criteri di ottimalità

Per la scelta di un algoritmo di instradamento esistono più criteri di ottimalità spesso contrastanti, ad esempio minimizzare il ritardo medio di ogni pacchetto o massimizzare l'utilizzo delle linee. A tal fine si dovrà disporre di una serie di parametri misurabili in base ai quali le caratteristiche di un percorso possano essere confrontate, per scegliere il migliore tra due cammini alternativi.

Gli unici due parametri universalmente accettati sono:

- il numero di salti effettuati (*hop*), cioè il numero di IS attraversati lungo un percorso;
- il costo, cioè la somma dei costi di tutte le linee attraversate lungo un percorso.

Entrambi questi parametri sono di demerito, in quanto il costo di una linea è assegnato in modo inversamente proporzionale alla velocità della linea stessa, e gli *hop* indicano *router* attraversati e quindi ritardi introdotti.

Il metodo appena introdotto, non tiene conto dei problemi di carico dinamico della rete. Le tecniche più moderne consentono di operare un bilanciamento del traffico (*load splitting*) tra cammini paralleli, eventualmente attivando circuiti commutati, quali quelli forniti da una rete alternativa in presenza di un guasto (ad esempio, funzionalità di *backup* di un **CDN**) o per gestire un eccesso di traffico su di un link (traffico di trabocco).

La scelta dell'algoritmo di instradamento ottimale è anche complicata dalle limitate risorse di memoria e CPU disponibili oggi sui *router*, specialmente se confrontate con la complessità delle reti ed in particolare con l'elevato numero di nodi collegabili con una topologia qualsiasi. Algoritmi troppo complessi, operanti su reti molto grandi,

potrebbero richiedere tempi di calcolo inaccettabili.

Algoritmi e protocolli di routing: caratteristiche degli algoritmi

Riassumendo, le caratteristiche che in generale si richiedono ad un algoritmo di *routing* sono:

- **Semplicità**: l'algoritmo deve essere funzionalmente efficiente con un minimo *software* e una bassa utilizzazione delle risorse *hardware*, poiché i *router* hanno CPU e memoria finite e devono impiegare la maggior parte del loro tempo a instradare pacchetti, e non a calcolare nuove tabelle di instradamento.
- **Robustezza/adattabilità**: di fronte a guasti *hardware*, variazioni di topologia, alto traffico, l'algoritmo deve continuare a lavorare.
- **Ottimizzazione**: è l'abilità dell'algoritmo a scegliere la strada migliore. La strada dipende dalla metrica (unità di misura per calcolare la lunghezza del percorso).
- **Stabilità**: quando ad esempio una rete diviene irraggiungibile, i *router* distribuiscono messaggi di aggiornamento di tale cambiamento a tutta la rete nel più breve tempo possibile, perché in caso contrario si potrebbero verificare dei "*routing loop*". Inoltre l'algoritmo deve sempre convergere velocemente ad un instradamento stabile, cioè non deve modificare le tabelle di instradamento se non a fronte di una variazione di topologia.
- **Equità**: nessun nodo deve essere privilegiato o danneggiato.

Gli algoritmi di *routing* non adattativi (**statici** e deterministici) utilizzano criteri fissi di instradamento mentre gli algoritmi adattativi (**dinamici** e non deterministici) calcolano le tabelle di instradamento in funzione della topologia della rete e dello stato dei link.

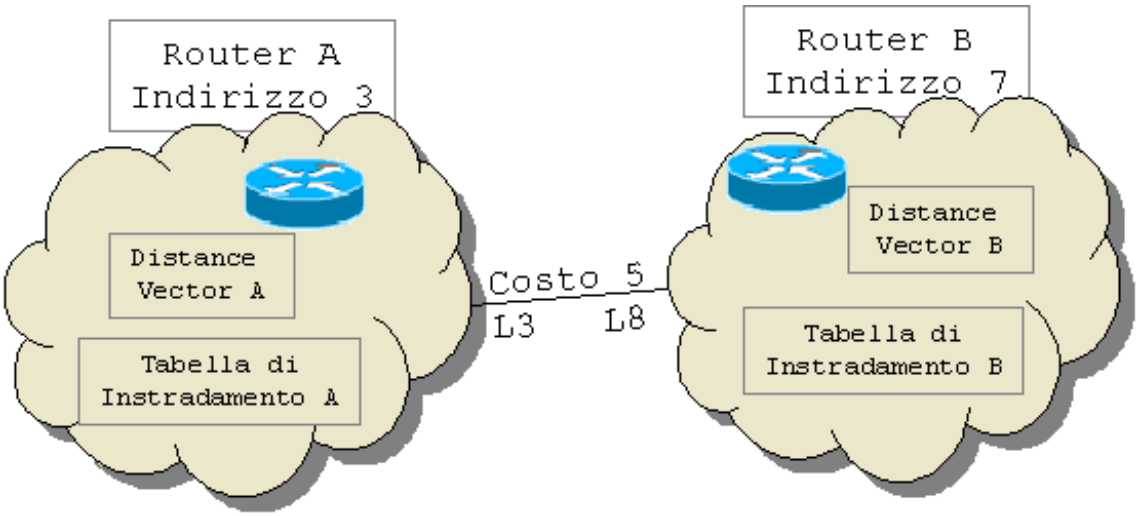
Algoritmi e protocolli di routing: Distance Vector (Bellman-Ford)

L'algoritmo *distance vector* è anche noto come algoritmo di *Bellman-Ford*. Abbiamo visto che ogni *Intermediate System* mantiene una tabella di instradamento, che gli indica attraverso quale interfaccia raggiungere la destinazione collegata. Il *router*, per realizzare tale algoritmo, gestisce una ulteriore struttura dati, detta vettore delle distanze (*distance vector*), per ogni linea.

Il vettore delle distanze è una struttura dati composta da:

- indirizzo di destinazione;
- minimo costo associato ad un *route* verso la destinazione.

L'algoritmo prevede che l'*Intermediate System* invii su tutte le proprie interfacce l'elenco delle destinazioni che è in grado di raggiungere e la distanza da esse. Il *distance vector* associato a ciascuna linea nella tabella di instradamento, contiene informazioni ricavate dalla tabella del *router* collegato all'altro estremo della linea (si veda figura).



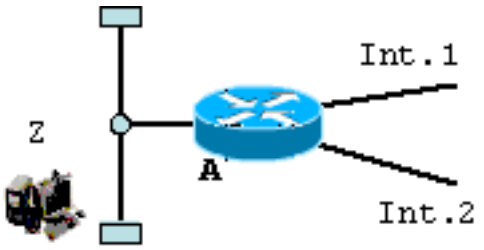
Un apparato riceve un *distance vector* da ognuno dei suoi vicini e lo memorizza dopo aver sommato alle distanze annunciate la distanza tra sé e il vicino che ha inviato il vettore. Nel caso in figura, le distanze verranno sommate al costo del percorso tra il *router* A e B, ossia 5.

A partire da tale conoscenza l'apparato costruisce la propria tabella di *routing* con una semplice operazione di fusione (*merge*) dei vettori.

Algoritmi e protocolli di routing: Distance Vector - merge

Quando un *router* memorizza un *distance vector* nella sua struttura dati locale, verifica se sono presenti variazioni rispetto al *distance vector* precedentemente memorizzato: in caso affermativo ricalcola le tabelle di instradamento fondendo (*merge*) tutti i *distance vector* delle linee attive. Analoga operazione di ricalcolo avviene quando una linea passa dallo stato ON allo stato OFF o viceversa. Alcune implementazioni di protocolli *distance vector* inviano anche i *distance vector* periodicamente, ad esempio il RIP, invia il *distance vector* ogni 30 secondi.

La fusione avviene secondo il criterio di convenienza del costo: a parità di costo secondo il minimo numero di *hop* e a parità di *hop* con scelta casuale. Se la tabella di instradamento risulta variata rispetto alla precedente, il *distance vector* relativo viene inviato ai *router* adiacenti.
Con 'Ind', identifichiamo un IS on un ES, senza alcuna distinzione. Quindi le lettere identificano *host* o *router* della rete.



Interfaccia 1		Interfaccia 2	
Ind	Costo	Ind	Costo
A	9	A	10
B	11	B	0
C	0	C	11
D	2	D	9
E	7	E	4
F	8	F	6
X	3	X	10
Y	10	Y	11
W	10	W	8
Z	13	Z	10
Costo Int.1 +9		Costo Int.2 +10	

Nella tabella qui sopra sono riportati i vettori delle distanze ricevuti dal *router* A su due sue interfacce. La numero uno ha un costo di 9 e la due un costo di dieci.

Nella fase di *merge*, si compone una tabella di *routing* come di seguito visualizzata.

Ind	Costo	Interfaccia
A	0	se stesso
B	10	2
C	9	1
D	11	1
E	14	2
F	16	2
X	12	1
Y	19	1
W	18	2
Z	20	2

La fusione, in particolare, tiene conto anche di percorsi fittizi, quali le destinazioni collegate direttamente e l'apparato stesso. Nell'ipotesi in cui la destinazione 'Z' sia direttamente collegata all'apparato, nella tabella risultante, troveremo scritto un valore 0 come costo.

Al termine della fusione, se questa implica la modifica della tabella precedentemente memorizzata sul *router*, avverrà la trasmissione su tutte le interfacce del nuovo vettore delle destinazioni.

Il modo di operare fin qui esaminato, se esaminato su ampia scala, genera un fenomeno simile a quello generato da una pietra che cade in uno stagno d'acqua:

- La pietra è una variazione dello stato della rete,
- lo stagno è la rete,
- le onde generate dalla caduta della pietra sono i *distance vector* che si dipartono dal luogo di impatto, arrivano ai bordi della rete, si specchiano e tornano verso il centro e ancora verso la periferia e poi verso il centro, con un moto che si ripete più volte prima di giungere a stabilità (acqua ferma).

Algoritmi e protocolli di routing: Distance Vector - cold start

Si supponga di produrre una inizializzazione contemporanea di tutti i nodi, un *cold start*. Durante questo stato, ciascuno dei nodi è caratterizzato unicamente da una conoscenza locale, il che significa che ciascun nodo conosce il proprio indirizzo, ma ignora totalmente la topologia della rete. La tabella di *routing* sarà quindi minima, e conterrà il route dell'apparato stesso e di tutti gli *End System* ad esso collegato. La conoscenza degli ES è dovuta, con IP, a configurazione manuale, o a specifici protocolli di ***neighbor greetings***.

Dalla tabella minimale, sarà estratto il vettore delle distanze che verrà trasmesso a tutti i vicini. All'atto della ricezione, gli apparati sono in grado di costruire una tabella comprendente un numero maggiore di destinazioni e di trasmettere vettori delle distanze sempre più ricchi.

Con il susseguirsi di questi scambi, ogni apparato della rete raccoglie informazioni sulla raggiungibilità delle destinazioni di tutta la rete. La convergenza dei vari distance-vector in transito, porta alla stabilizzazione.

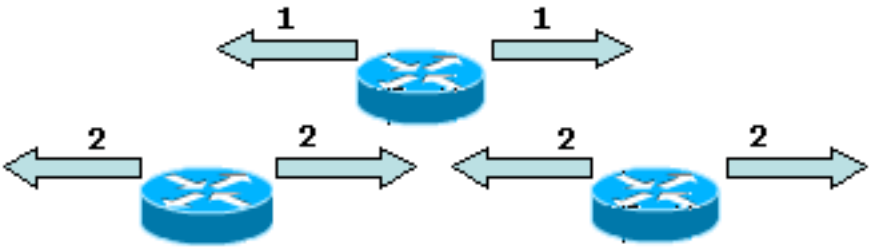
Solo ora le informazioni contenute nelle tabelle di instradamento sono da ritenersi corrette.

Algoritmi e protocolli di routing: Distance Vector - updates

Si ripresenta la similitudine con l'effetto generato da una pietra che cade in uno stagno d'acqua (in questo caso per la propagazione delle modifiche topologiche). Nell'ipotesi di un guasto ad un'interfaccia, il *router* dovrà ricalcolare la tabella di instradamento dai vettori delle distanze delle interfacce rimaste funzionanti. In caso di una tabella risultante diversa da quella precedentemente calcolata, si dovrà optare per la ritrasmissione a tutti della nuova tabella.

In questo modo, anche gli altri apparati, dovranno verificare le novità apportate. La verifica non tiene conto della topologia, bensì calcola semplicemente l'unione delle tabelle come precedentemente mostrato (***merge***).

Analogamente al malfunzionamento, anche un cambio della topologia volontario, provoca una reazione a catena che comincia nella zona circostante al cambiamento e si propaga sino ai bordi della rete, e contemporaneamente anche nella direzione opposta.



Il modo in cui l'algoritmo reagisce ad un cambiamento topologico, produce le seguenti conseguenze:

- un IS può ricalcolare più di una volta la propria tabella di *routing* a seguito di un cambiamento topologico.
- non si può considerare raggiunta la convergenza, fino a che tutti gli IS non abbiano terminato di calcolare le proprie tabelle, determinando che non è più necessario ritrasmettere il vettore delle distanze ai vicini.

In ultima analisi, si determina che la velocità di convergenza è limitata dall'apparato e dai collegamenti più lenti della rete.

Algoritmi e protocolli di routing: Distance Vector - sommario

L'algoritmo *distance vector* è fatto di meccanismi estremamente semplici, che ne hanno determinato la facile distribuzione e lo sviluppo sugli apparati di instradamento. Al contempo, dimostra una serie di limiti tra i quali:

- Gli apparati per *internetworking* non sono in grado di rendersi conto se stanno cercando di inoltrare il pacchetto su di un percorso chiuso (*loop*).
- La quantità di informazioni di servizio che gli IS si scambiano è considerevolmente alta e sottrae risorse al traffico dati. Va ricordato che il vettore delle distanze contiene informazioni su tutte le destinazioni che esso è in grado di raggiungere.
- La complessità dell'algoritmo, fa sì che la convergenza del *routing* si raggiunga solo dopo un numero di operazioni pari al quadrato o al cubo dei nodi della rete. Questa differenza tra valore minimo e massimo dipende dal numero di collegamenti che la rete dispone. Ad esempio una rete che cresce di 10 volte avrà complessità in aumento da 100 a 1000 volte.

Algoritmi e protocolli di routing: Link State

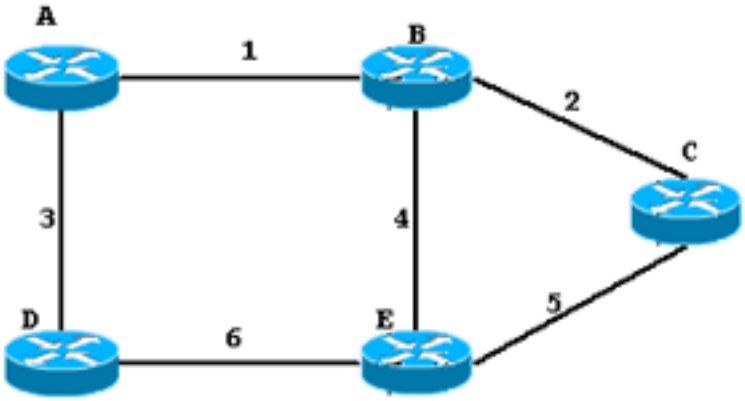
I protocolli di tipo *Link State* sono basati sul concetto di mappa distribuita:

- tutti i nodi posseggono una copia della mappa della rete, che viene regolarmente aggiornata.

Gli IS, dalla copia locale della mappa della rete, eseguono un calcolo completo dei migliori percorsi. La mappa è contenuta in un *database*, dove ciascun *record* rappresenta un link nella rete.

Ciascun *record* contiene un identificatore di interfaccia e le informazioni che descrivono lo stato del link (la destinazione e la distanza o metrica). Con queste informazioni ogni nodo può facilmente calcolare il percorso più breve da sé stesso a tutti gli altri nodi. Con la velocità degli attuali *computer*, è necessario solo un brevissimo tempo, tipicamente una frazione di secondo se la rete non è molto estesa, per calcolare tali percorsi.

Poiché tutti i nodi contengono lo stesso database ed eseguono lo stesso algoritmo di *route-generation*, i percorsi sono coerenti e non si verificano *loop*.



Da	A	Link	Distanza
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	a	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Si consideri l'invio di un pacchetto dal nodo A al nodo C nella rete riportata in figura, e ci si basi sui calcoli tra i nodi A e B. Il nodo A può rilevare, tramite il database, che il percorso più corto verso il nodo C passa attraverso il nodo B e quindi invia il pacchetto sul link numero 1. Il nodo B, a sua volta, invierà il pacchetto sul link numero 2.

Algoritmi e protocolli di routing: Link state - LSP, adiacenze e flooding

Ai fini della costruzione della mappa della rete, i *router* invieranno e riceveranno speciali pacchetti di servizio detti ***L**ink **S**tate **P**acket* (**LSP**).

Il *Link State Packet* (LSP) contiene:

- Stato di ogni link connesso al *router*.
- Identità di ogni vicino connesso all'altro estremo del link.
- Costo del link.
- Numero di sequenza per il LSP
  (a seguito di frequenti variazioni di topologia un *router* può ricevere un LSP vecchio dopo uno nuovo, quindi deve essere in grado di valutare il più recente).
- ***Checksum***.
- *Lifetime*
  (la validità di ogni LSP è limitata nel tempo; diversamente un errore sul numero di sequenza potrebbe rendere un LSP valido per anni).

La generazione del LSP avviene su base periodica, oppure quando viene rilevata una variazione nella topologia locale (adiacenze), ossia:

- Viene riconosciuto un nuovo vicino.
- Il costo verso un vicino è cambiato.
- Si è persa la connettività' verso un vicino precedentemente raggiungibile.

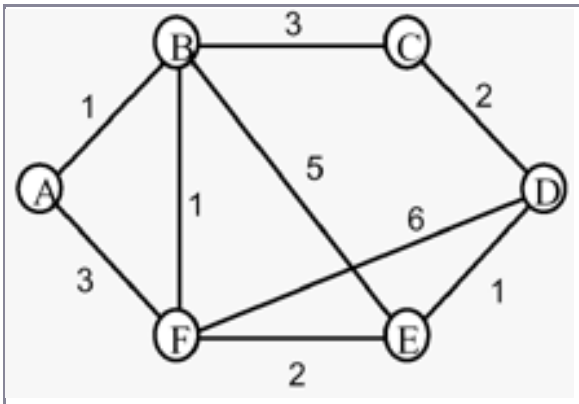
Il LSP è trasmesso in ***flooding*** selettivo su tutti i link del *router* e tutti i *router* del dominio di *routing* ricevono il LSP. All'atto del ricevimento di un LSP il *router* compie le seguenti azioni:

- Se non ha mai ricevuto LSP da quel *router* o se il LSP è più recente di quello precedentemente memorizzato (campo *Sequence Number*), memorizza il pacchetto e lo ritrasmette in *flooding* su tutte le linee eccetto quella da cui l'ha ricevuto.
- Se il LSP ha lo stesso numero di sequenza di quello posseduto non fa nulla.
- Se il LSP è più vecchio di quello posseduto trasmette al mittente il pacchetto più recente.

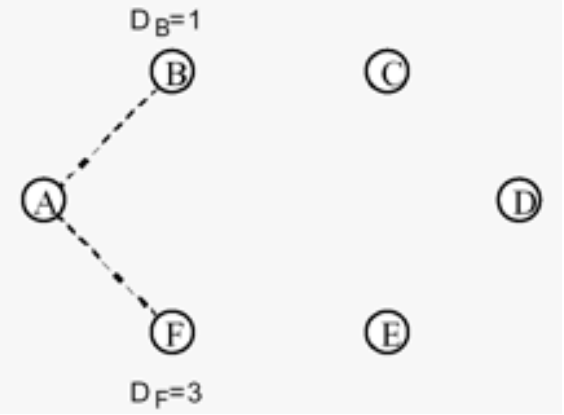
Algoritmi e protocolli di routing: Link state - algoritmo Dijkstra e SPF

Ogni nodo ha a disposizione il grafo pesato della rete ed assegna a tutti gli altri nodi un'etichetta che rappresenta il costo massimo per la raggiungibilità del nodo in esame; l'algoritmo di calcolo modifica tali etichette cercando di minimizzarle e di renderle permanenti.

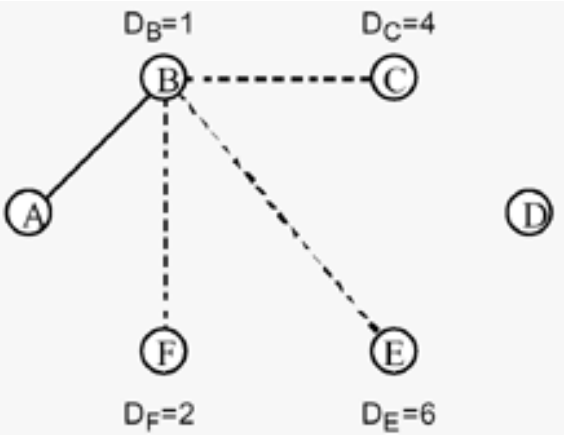
L'*intermediate System* che voglia calcolare il proprio elenco di percorsi più brevi, si basa sui principi dell'algoritmo Dijkstra. Si visita ogni nodo della rete a partire da se, un nodo alla volta; il successivo nodo visitato è quello più vicino a uno dei nodi già visitati (da ciò il nome **SPF**, ***S****hortest* ***P****ath* ***F****irst*: prima il percorso più breve).



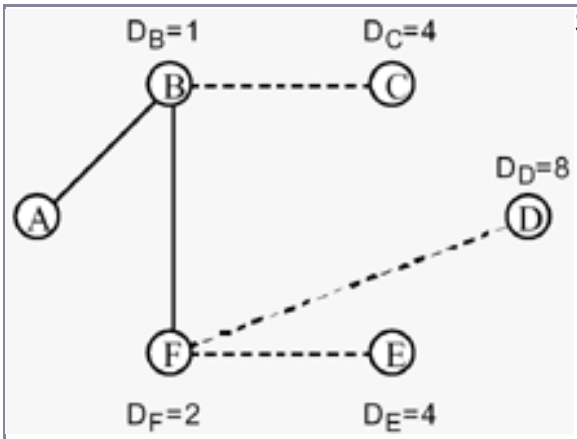
Si consideri l'insieme di nodi a fianco mostrato.



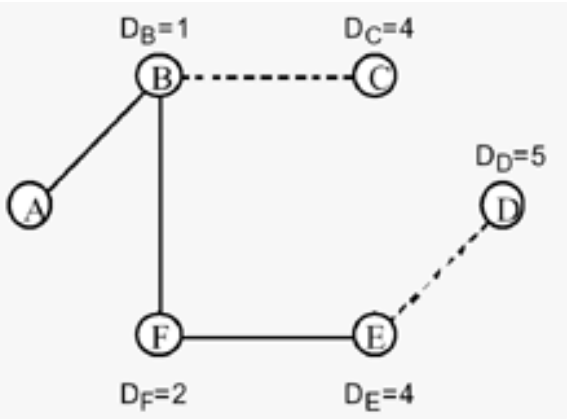
Il punto di partenza arbitrariamente designato, è il nodo A. Quindi sarà analizzato il grafo pesato a partire dal nodo A.
Notare il costo del percorso evidenziato sopra ogni nodo di destinazione.



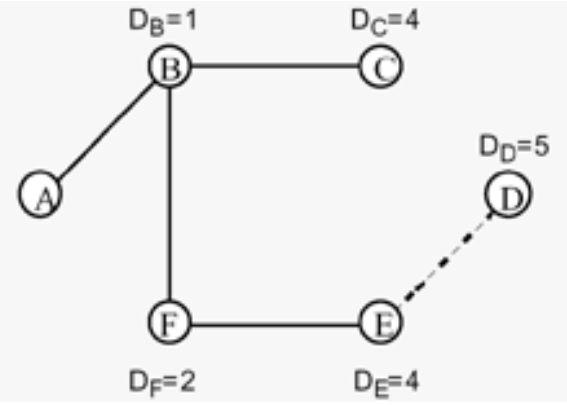
Si procede scegliendo sempre per primi i percorsi meno costosi. In questo caso il passo meno costoso, sta nel passaggio per B.



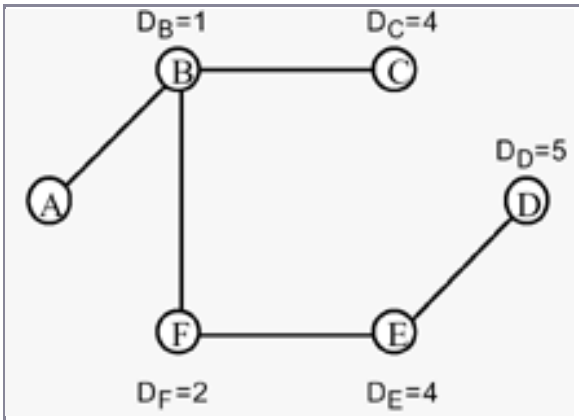
Sempre con il medesimo principio viene scelto in nodo F.



Nodo scelto: E



Nodo scelto: C



Nodo scelto: D
Alla fine, tutti i nodi sono raggiungibili senza maglie (*loop*).

Algoritmi e protocolli di routing: Link state - proprietà

Il *routing link-state* è caratterizzato da una elevata stabilità, cioè brevi tempi di convergenza e bassissima inclinazione a creare dei *loop* di instradamento. Questi pregi derivano da:

- il numero di passi che portano allo *shortest path* è dell'ordine di C log N; dove C è il numero di collegamenti e N è il numero di nodi.
  Diversamente da ***Bellman Ford***, dove si cresceva in passi da compiere quadraticamente o cubicamente, in questo caso si avrà aumento lineare nei confronti della complessità topologica e logaritmico della dimensione della rete.
- avvenendo il calcolo dello *shortest path* in maniere indipendente tra gli apparati, la lentezza di calcolo di un singolo non pregiudicherà le prestazioni degli altri.
- i tempi di propagazione degli LSP non pregiudicano la consegna dei pacchetti paganti, poiché tanto più gli apparati sono lontani dalla zona dove è avvenuto un ipotetico cambiamento, tanto più sarà probabile che non porti alla modifica delle tabelle di *routing*.

Va aggiunto il non indifferente vantaggio della ridotta dimensione dei pacchetti LSP; in tali pacchetti di servizio verranno incapsulate solo informazioni relative ai collegamenti con i vicini, piuttosto che con il vettore distanza dove è necessario inserirvi tutta la tabella di instradamento.

Autonomous System

Ogni *Router* vede crescere la propria tabella di *routing* in maniera lineare rispetto al numero di reti da instradare direttamente. Nei primi anni ottanta, Internet era considerata come una *single network* (SN) dalle dimensioni geografiche molto estese. L'introduzione di sistemi che permettessero ai *router* di adattarsi dinamicamente al mutare della tipologia di connessione costituiva una soluzione parziale al problema.

Se da una parte diminuiva il tasso di errore e la necessità di coordinamento umano, aumentando contemporaneamente la tolleranza ai guasti delle singole connessioni,

dall'altra ci si rese conto che questa scelta non risolveva il problema della crescita delle tabelle di *routing*.

La decisione intrapresa fu quella di abbandonare il modello SN per suddividere Internet in un certo numero di *Autonomous System* (**AS**) costituiti da un insieme di *router* e LAN sotto la medesima amministrazione.

Questa definizione, non molto flessibile, è stata in seguito sostituita da una più funzionale che prevede che i *router* all'interno di ogni AS siano reciprocamente raggiungibili. Le informazioni di raggiungibilità sono, preferibilmente, scambiate mediante uno o più protocolli adattivi detti ***Interior Protocol***. Inoltre si prevede che gli AS si scambino informazioni sulla rispettiva raggiungibilità utilizzando un protocollo opportuno genericamente designato come ***Exterior Protocol***.

La definizione classica di un sistema autonomo è quella di un insieme di *router* sotto una singola organizzazione ed amministrazione tecnica, utilizzante un IGP e metriche ad *hop* per gestire l'instradamento interno all'AS stesso, e un EGP per gestire il *routing* verso altri AS.

In molte monografie il termine *Autonomous System* viene utilizzato parallelamente alla dizione Dominio di *routing*, per intendere una zona sottoposta alla gerarchia di un *router* principale (*Master*).

Se un *gateway* deve instradare un messaggio a un altro *gateway* appartenente allo stesso AS, avrà nella propria tavola di *routing* l'informazione di raggiungibilità idonea. Se al contrario è necessario raggiungere un *gateway* che appartiene ad un differente AS il messaggio viene inviato attraverso una coppia di *router* particolari detti *exterior router*, almeno uno per AS.

Ciascun *exterior router* conosce le reti raggiungibili utilizzando i link che lo collegano agli altri *exterior router* ma non conosce il modo in cui queste reti sono di fatto connesse all'interno dei rispettivi AS (secondo un modello tipicamente gerarchico).

Autonomous System: IGP - la famiglia di protocolli inter-dominio

Il protocollo IGP (*Interior Gateway Protocol*) maggiormente utilizzato oggi su Internet è senza dubbio il protocollo RIP.

RIP è l'acronimo di ***R****outing* ***I****nformation* ***P****rotocol* ed è un protocollo relativamente semplice appartenente alla famiglia di protocolli di tipo *distance vector*.

Sviluppato dalla *Xerox* per XNS, nel 1982 il RIP è stato adattato per il TCP/IP con lo *UNIX* BSD.

Si tratta di un protocollo di *routing* intradominio basato su un algoritmo di tipo *distance vector*, ed è stato adottato dall'IETF nello RFC 1058 (1988) e nello RFC 1388 (1993). È inoltre utilizzato in diverse implementazioni proprietarie tra cui lo RTMP di *AppleTalk*.

È il protocollo di *routing* IP più vecchio ancora in uso: la versione IP esiste in due versioni, la seconda versione aggiunge nuove funzionalità a questo protocollo. RIPv1 è di tipo *classfull* mentre RIPv2 è *classless*.

L'algoritmo ***distance vector*** è stato sviluppato da *Bellman*, *Ford* e *Fulkerson* nel lontano 1969, poi è stato integrato dalla *Xerox Network System* nei suoi protocolli come XNS RIP. Tale protocollo è stato precursore di comuni protocolli di *routing*come *Novell* IPX RIP, *AppleTalk Routing Table Maintenance Protocol* (RTMP) e naturalmente IP RIP. Nella versione 4.2 del *Berkley UNIX* rilasciata nel 1982 il RIP è implementato come un processo demone chiamato *routed*: ancora molte versioni di *UNIX* implementano il RIP.

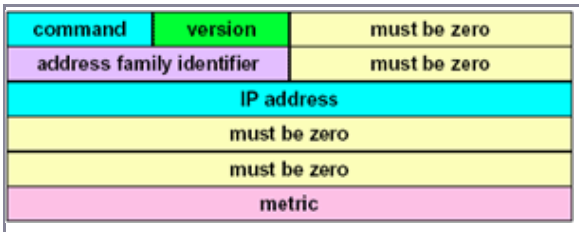
Autonomous System: IGP - RIP (caratteristiche)

Gli indirizzi presenti nelle tabelle RIP sono indirizzi Internet a 32 bit. Una voce (*entry*) nella tabella di *routing* può rappresentare un *host*, una rete o una sottorete. Non sono presenti specifiche sul tipo di indirizzo nei pacchetti RIP; è compito dei *router* analizzare l'indirizzo per capire di cosa si tratta. Questi, prima separano la parte di rete dalla parte sottorete + *host* in funzione della classe dell' indirizzo (A, B o C). Se la parte sottorete+*host* è nulla, l'indirizzo rappresenta una rete, viceversa può rappresentare sia una sottorete che un *host*. Al fine di discriminare tra queste 2 possibilità, è necessario conoscere la *subnet mask*; se la parte *host* è nulla, si tratta dell'indirizzo di una sottorete, di un *host* viceversa.

Di *default*, RIP utilizza una **metrica** molto semplice: la distanza (*hop count*) è il numero di links che vengono attraversati per raggiungere la destinazione. Questa distanza è espressa come un numero intero variabile tra 1 e 15; 16 rappresenta una distanza infinita.

RIP supporta sia i links punto a punto che le reti di tipo *broadcast* come *Ethernet*. I pacchetti RIP vengono impacchettati nei pacchetti **UDP** e IP; i processi RIP utilizzano la **porta 520** sia per la trasmissione che per la ricezione. Utilizzando una porta specifica, minore di 1024, si è in linea con le protezioni di sistema di BSD-*Unix*.

I pacchetti normalmente sono inviati in modalità *broadcast*, ovvero saranno ricevuti da tutti i *routers* connessi alla rete. Normalmente i pacchetti vengono inviati ogni 30 secondi, o meno, nel caso di aggiornamenti alle tabelle. Se una *route* non viene aggiornata entro 3 minuti, la distanza viene fissata ad infinito e l'*entry* verrà successivamente rimossa dalle tabelle. Allo scopo di evitare aggiornamenti troppo frequenti , questi vengono regolati da un *timer* casuale, che può variare tra 1 e 5 secondi.



Command
request = 1 utilizzato in fase di inizializzazione per richiedere un *distance vector*
response = 2 n utilizzato normalmente per distribuire i *distance vector*

Come osservabile dal pacchetto, il protocollo RIP prevede un comando di richiesta e uno di risposta o aggiornamento. Normalmente RIP opera in risposta, a intervalli regolari di 30 secondi, o in seguito a richieste di aggiornamento delle tabelle di *routing*. Il processo RIP, in seguito alla ricezione di un messaggio di risposta, aggiorna la propria tabella. Ogni voce della tabella sarà al limite composta da:

- Indirizzo di destinazione.
- Metrica associata con la destinazione.
- Indirizzo del *next router*.
- Un *recently updated flag*.
- Numerosi *time*.

Autonomous System: IGP - RIP

Elaborando le risposte in arrivo, il *router* esaminerà le voci una ad una ed eseguirà una serie di controlli, quali la verifica della validità dell'indirizzo e l'appartenenza ad una delle classi A, B o C, che il numero identificante la rete non sia 127 (*loop-back*) o zero (ad eccezione dell'indirizzo di *default* 0.0.0.0), che la parte *host* non sia un indirizzo *broadcast* e che la metrica non sia maggiore di 15 (infinito). In ogni caso voci non corrette vengono ignorate.

Se la metrica in arrivo risulta diversa da infinito, viene incrementata di 1 per il successivo *hop*, quindi la tabella di *routing* viene scandita per una voce corrispondente alla destinazione e viene quindi eseguito il generico processo **distance vector**, di seguito illustrato.

- Se la e la voce non è presente e la sua metrica nel messaggio ricevuto non è infinito, la aggiunge alla tabella, inizializzando la metrica al valore ricevuto ed il *next router* al mittente del messaggio, prima di avviare un *timer* per quella voce.
- Se la voce è presente con una metrica più grande, aggiorna i campi della metrica e del *next router* e riavvia il *timer* per quella voce.
- Se la voce è presente ed il *next router* è il mittente del messaggio di risposta, aggiorna la metrica se questa differisce dal valore memorizzato e, in tutti i casi, riavvia il *timer*.
- In tutti gli altri casi, il messaggio ricevuto è ignorato.

Se la metrica o il *next router* cambiano, l'*entry* viene marcata come aggiornata. Un messaggio di risposta viene inviato ad intervalli regolari di 30 secondi o può essere attivato in seguito ad un aggiornamento alle tabelle di *routing*. Questo può essere causa di eccesso di traffico sulla rete e l' RFC-1058 specifica una serie di precauzioni da adottare a questo preciso scopo. La risposta non dovrebbe essere inviata immediatamente in seguito alla ricezione dell' aggiornamento ma, piuttosto, dopo un piccolo intervallo *random*, variabile tra 1 e 5 secondi. Questo permette ai relativi aggiornamenti provenienti dai nodi vicini di venire riassunti nel successivo messaggio, limitando così il carico di rete.

Un messaggio di risposta separato viene preparato per tutte interfacce connesse. L'informazione può variare in seguito al processo di *split horizon*. Il messaggio normalmente include le coppie indirizzo e metrica per tutte le voci della tabella ma, se il messaggio è inviato come un aggiornamento, non deve necessariamente includere tutte le voci, ma solo quelle che sono state aggiornate rispetto all'ultima trasmissione. Il massimo formato del pacchetto è 512 *bytes*, che permette di avere sino a 25 voci per messaggio. Nel caso di un maggiore numero di voci, RIP invierà più pacchetti. L'indirizzo sorgente del messaggio dovrebbe sempre coincidere con l'indirizzo IP associato con l'interfaccia alla quale il messaggio è inviato.

I processi RIP possono anche ricevere messaggi di richiesta. Una richiesta viene

normalmente inviata quando un *router* inizia le operazioni allo scopo di ottenere dai suoi vicini il valore iniziale delle tabelle di *routing*. Esistono 2 possibili forme di richiesta, quella per una lista completa delle tabelle di *routing* o quella per sole specifiche *routes*.

Una delle richieste di lista completa delle tabelle di *routing* si ha specificando solo le coppia indirizzo + metrica per l'indirizzo di *default* 0.0.0.0 con una metrica pari ad infinito. In questo caso, il *router* replicherà con una tipica risposta, simile a quelle inviate periodicamente nelle normali operazioni del protocollo, incluso il processo di *split horizon*.

Qualsiasi altra forma di richiesta prevede la lista delle sole voci specificate. La risposta verrà inviata in modalità *point to point* al richiedente e conterrà una copia esatta dell'informazione di distanza nelle tabelle di *routing*, senza eseguire il processo di *split horizon*. Questa forma di richiesta ha poco significato per le normali operazioni, mentre ne assume molto per scopi di *debugging*.

Autonomous System: IGP - RIP (parametri)

Timers

Il protocollo RIP gestisce i seguenti *timers*:

- *Routing update timer* (*default* 30 s): intervallo di tempo per l'invio degli annunci.
- *Route invalid timer* (*default* 90 s): intervallo di tempo dopo il quale una *route* è dichiarata irraggiungibile (distanza posta ad infinito).
- *Route flush timer* (*default* 270 s): intervallo di tempo dopo il quale la *route* è cancellata dalla *routing table*.
- *Triggered updates*: sono inviate con un ritardo casuale compreso tra 1 e 5 secondi, per evitare intasamenti della rete e per far sì di poter eventualmente comunicare più cambi di *route* con un messaggio solo.

Stabilità

Al fine di assicurare una buona stabilità e, quindi, di evitare situazioni di *loop* e, di conseguenza, possibili congestioni della rete, RIP utilizza una serie di tecniche, di seguito descritte:

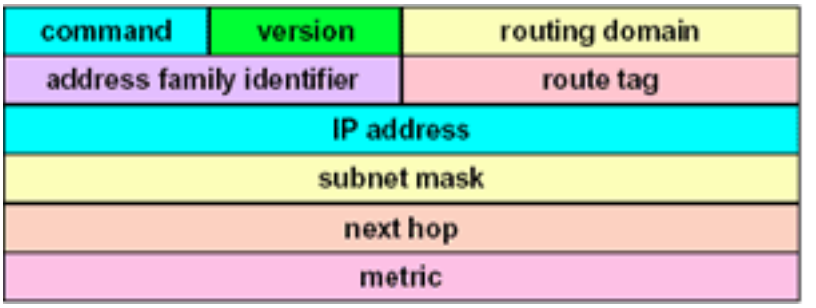
- *Triggered updates*: si tratta di messaggi di *routing update* fatti anzitempo, causati da variazioni di connettività. Nel caso vengono inviate solo le informazioni relative alle *route* che sono cambiate (non viene trasferito il *distance vector* completo). Evitano alla rete di trovarsi in uno stato incoerente, fino allo scadere del *Routing Update Timer*, quando i *distance vector* sarebbero inviati comunque.
- *Hop Count Limit*: fa sì che le destinazioni più distanti di 15 siano considerate irraggiungibili e permette quindi di evitare il *count to infinity problem*.
- *Hold Downs*: Il *router* mette in quarantena le *route* che utilizzavano il link guasto. Inoltre, il *router* che ha rilevato il guasto non può partecipare ad alcun *loop*, almeno fino alla scadenza dell'*Hold Down timer*.
- *Split Horizon*: Se A raggiunge la destinazione X attraverso B, non ha senso

per B cercare di raggiungere X attraverso A. Previene il *loop* tra 2 nodi ma non elimina i *loop* con 3 nodi

- *Split Horizon with Poisonous Reverse*: Migliora lo *Split Horizon* puro. Nel momento in cui si verifica una variazione di *route*, lo *Split Horizon* semplice non comunica più la *route* (che continua però a valere sino alla scadenza del *timer*). Viceversa, con il *Poisonous Reverse* la *route* viene comunicata con costo infinito, quindi gli altri *router* apprendono immediatamente che non possono usare la *route* (non devono aspettare lo scadere del *route invalid timer*). Le *route* non più valide non vengono rimosse dagli annunci ma sono annunciate con metrica 15 (*infinity*).

Autonomous System: IGP - RIPv2

A causa di alcune limitazioni è stato sviluppato un nuovo RIP descritto nella RFC 1723. Questa nuova versione consente l'interoperabilità con RIPv1 ed ha in più la possibilità di trasferire anche le *netmask* e quindi di instradare anche su sottoreti differenti.



RIPv1 non è un protocollo sicuro. Qualsiasi *host* che invia pacchetti dalla porta UDP 520 verrebbe considerato un *router* dai propri vicini, mentre invece solo un utente privilegiato dovrebbe avere il diritto di utilizzare questa porta.

A questo scopo, RIPv2 prevede una procedura di autenticazione che specifica che la prima *entry* in un pacchetto può essere rimpiazzata da un *authentication segment*. Il pacchetto conterrà quindi l'*header* iniziale a 32 bit, seguito da un segmento di autenticazione composto da:

- Il campo **AFI** (***A**ddress **F**amily **I**dentifier*) settato a 0xFFFF.
- Un campo *Authentication Type* (2 *bytes*) che identifica il tipo di algoritmo di autenticazione in uso.
- 16 *bytes* di dati di *autentication*.
- 24 coppie di campi destinazione-metrica.

I *routers* RIPv1 semplicemente rilevano che il campo AFI non risulta pari a 2 e quindi non considerano l'*authentication segment*, procedendo con le 24 rimanenti *entries*. Alla ricezione del pacchetto, il *router* RIPv2 verifica la presenza dei campi di autenticazione e, in caso affermativo, ne rivela l'origine.

Esistono svariati algoritmi di autenticazione, definiti attraverso il campo *Authentication Type*; in comune vi è la protezione dei messaggi di aggiornamento attraverso una *password* cifrata.

Autonomous System: IGP - IGRP

IGRP (***I**nternal **G**ateway **R**outing **P**rotocol*) nasce come un'evoluzione del RIP. Alcune delle modifiche apportate al protocollo originale sono sotto il *copyright* della *Cisco*. Alla base dello IGRP vi è un aggiornamento periodico inviato in *multicast* delle informazioni possedute da ciascun *gateway*. Ogni informazione di *routing* è composta da quattro attributi principali destinati al calcolo del miglior cammino sorgente-destinazione:

- Ritardo (R): indica la somma di tutti i ritardi accumulati nel cammino intrapreso. Il valore può essere calcolato staticamente in base al tipo di reti attraversate o modificato dall'amministratore di sistema.
- Banda (B): rappresenta la banda disponibile sul più lento link attraversato. Il calcolo di questo valore avviene quasi sempre staticamente.
- Affidabilità (A): rappresenta una stima dell'errore medio presente su ogni connessione fisica ed è calcolato dinamicamente monitorando, istante per istante, le condizioni presenti sulla rete.
- Carico (C): indica il valore di picco registrato sul *router* più occupato attraversato nel cammino sorgente-destinazione.

Durante l'applicazione del classico algoritmo *distance-vector*, la scelta tra due cammini che conducono alla medesima destinazione si effettua in base ad una funzione di metrica che tiene conto dei quattro parametri appena descritti e di alcune costanti che condizionano il risultato finale. Si ricordi che il RIP, usa una metrica molto meno complessa e basata esclusivamente sulla distanza intesa come numero di link attraversati. Uno dei principali problemi evidenziati dal RIP è la difficoltà di rilevare la creazione di un ciclo nel caso di caduta di un link.

A *counting to infinity* il RIP affianca i metodi *Split Horizon*, *Trigger Updtate* e *Poisonous Reverse*. IGRP utilizza, al posto del *Poisonous Reverse*, uno fra i due metodi descritti di seguito:

Path HoldDown si basa su una semplice ma efficace osservazione: la creazione di *loop* si verifica se e solo se un *router*, che ha già rilevato la caduta di un link, riceve l'informazione contraria (per errore o per ritardi di propagazione delle informazioni) da un *router* che non ha rilevato la mutata condizione topologica, prima di riuscire ad immettere sulla rete questo nuovo stato. Partendo da quest'osservazione *Path HoldDown* non appena rileva la caduta di un link l impone un periodo di quarantena durante il quale non è accettato alcun *update* esterno relativo ad l.

Route Poisonous: osserva che la tecnica *counting to infinite*, in presenza di creazione di un *loop*, fa crescere progressivamente la metrica associata ai cammini coinvolti. Se questa condizione si verifica improvvisamente, IGRP considera in modo conservativo il cammino inutilizzabile sino al prossimo *trigger update*. Un effetto collaterale di questa strategia è la creazione di problemi temporanei quando la topologia fisica della rete viene modificata intenzionalmente (ad esempio aggiungendo nuovi *router*).

Oltre ai metodi di prevenzione dei *loop*, una nuova caratteristica del protocollo è la possibilità di mantenere nelle tavole di *routing* non solo il miglior cammino verso una specifica destinazione ma anche percorsi ausiliari da utilizzare come *backup*. L'algoritmo del RIP è leggermente modificato per consentire la memorizzazione anziché lo scarto di un link con metrica maggiore, in caso di presenza di un link con metrica minore.

Autonomous System: IGP - EIGRP

EIGRP è la risposta *Cisco* ad OSPF. La casa americana, pur avendo implementato nei propri *router* quest'ultimo protocollo, sostiene che ci siano dei validi motivi per non abbandonare la tecnologia *distance-vector*. L'*Enhanced* IGRP è essenzialmente un algoritmo *distance-vector* cui è affiancato il metodo conosciuto come *Diffusion Update Algorithm* (DUAL) sviluppato da J.J.Garcia-Luna-Aceves sulla base di un precedente algoritmo di E.W. Dijkstra.

Supposto che ogni *router* h mantenga per ciascuna destinazione j la metrica d (k, j). La metrica è propagata attraverso ciascuno dei k *router* direttamente connessi con h. Sia inoltre l (h, k) il costo associato al link che collega lo stesso h con k.

In condizioni normali per raggiungere j, il *router* h seleziona il *router* direttamente connesso x tale che sia minimizzata la funzione di costo [FC] d (h, j) = l (h, x) + d (x, j).

In altre parole, h privilegia il *router* x che è raggiunto con minore costo rispetto sia al link di connessione sia alla distanza per la destinazione j.

Supponiamo che h, dopo aver ricevuto come update dal *router* y i due valori d1(y, j) ed l1(h, y), verifichi che la nuova somma risulta minore del valore precedentemente calcolato (analiticamente l1(h, y) + d1(y, j) < d(h, j)). Il *router* h selezionerà semplicemente come *next hop* y al posto di x. Al di là delle notazioni, il funzionamento è fino a questo momento semplice e, sorprendentemente, vi sono poche possibilità di creare dei *loop* in caso di caduta di un link. Vediamo perché: se viene effettuato un *update*, che incrementa il costo di un link (come nel caso di creazione di un ciclo), non si modifica la tavola di *routing* perché in ogni caso l'algoritmo impone la scelta del cammino di costo minimo. La condizione precedente è vera con l'eccezione del caso in cui l'*update* incrementa il costo del *router* x correntemente selezionato, in altre parole del cammino con costo di per sé già minimo. In tal caso EIGRP è più complicato poiché cerca nuovamente l'esistenza di un *router* vicino k tale che d (k, j) < d (h, j) dove d (h, j) è il vecchio valore di metrica che ha permesso di selezionare il *router* x prima dello *update* che ha incrementato il costo verso x. Se la selezione da luogo ad un insieme I di *router*, si sceglie quello che minimizza FC. Se non esiste nessun *router* che soddisfi la condizione richiesta la tavola di *routing* per la *entry* relativa alla destinazione j è messa in quarantena fotografando la situazione precedente all'*update* incriminato. Da quel momento viene eseguita la parte *Diffusion* dell'algoritmo. Si noti che in questa fase, in modo poco conservativo, si considera il link verso x come valido anche se, di fatto, esso potrebbe essere interrotto.

Un *router* entrato in modo *Diffusion* invia una *query* a tutti i vicini con l'esclusione del *router* x, segnalando la nuova distanza d1(h, j) = l1(h, x) + d1 (x, j). Chi riceve il messaggio, supponiamo il *router* z, ha due comportamenti: se z non possiede alcun'informazione circa la raggiungibilità della destinazione j o, se al contrario, non solo possiede informazioni su j ma è stato in grado di selezionare un vicino per raggiungere x risponde alla *query* immediatamente con la propria tavola di *routing*. In caso le condizioni precedenti non siano vere anche z invia una *query* ai propri vicini passando in modo *Diffusion*.

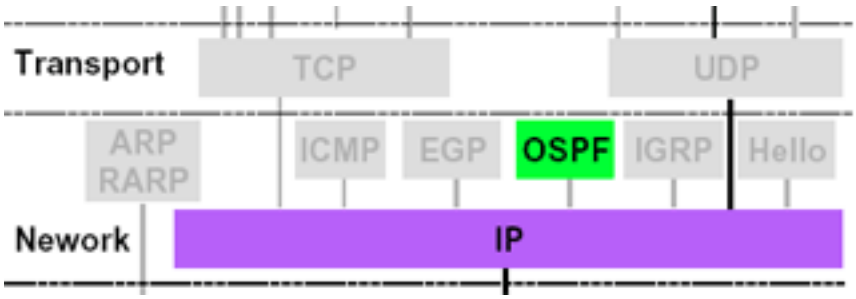
Un *router* ritorna in modo normale non appena riceve una risposta ad una sua *query*, preoccupandosi di propagare la stessa a tutti quei vicini da cui aveva eventualmente ricevuto richieste. A quel punto la tavola è tolta dalla quarantena ed aggiornata con le

nuove informazioni acquisite.

E-IGRP non è semplice. Tuttavia, esso rappresenta, a ragion veduta, lo stato dell'arte per ciò che riguarda la classe d'algoritmi *vector distance*. Semplificando, il suo obiettivo è quello di minimizzarne gli effetti negativi ricorrendo in caso d'insuccesso ad una forma di elaborazione distribuita.

Autonomous System: IGP - OSPF

OSPF (***O**pen **S**hortest **P**ath **F**irst*) adopera pacchetti di servizio specifici di tipo IP ed è definito dall'IETF con la RFC 1247 (1991) e la RFC 1583 (1994).



Contrariamente ai protocolli *distance-vector*, OSFP è basato su una differente tecnica detta *link state* che, in media, richiede minore tempo per convergere ad una soluzione stabile. Semplicemente ciascun *router* R deve:

- Scoprire l'indirizzo di ogni *router* vicino Ri.
- Misurare il costo necessario per raggiungere ciascun Ri usando una funzione idonea allo scopo.
- Inviare un messaggio a tutti i *router* diffondendo le informazioni acquisite.
- Calcolare, in locale, il cammino minimo verso ogni altro *router* utilizzando un algoritmo sviluppato da Dijkstra.

La maggiore differenza rispetto agli algoritmi *vector distance* è che gli algoritmi *link state* richiedono che ciascun *router* sia informato circa la completa topologia e i ritardi presenti nella rete.

Sulla base di queste informazioni ogni *router* calcola in locale il cammino minimo verso ogni destinazione conosciuta.

Scoperta dei vicini e misura dei costi di raggiungimento

Per conoscere l'intera topologia di una rete sono necessari alcuni accorgimenti: in primo luogo conoscere da ciascun punto l'insieme di *router* direttamente connessi. Per far questo non appena un *router* è inserito, invia un pacchetto di *HELLO* su tutte le connessioni *point-to-point* disponibili e riceve come risposta dai *router* direttamente connessi il relativo identificativo. Per calcolare il costo di connessione con i *router* vicini, è utilizzato un pacchetto *ECHO* misurando il tempo necessario ad ottenere la relativa risposta.

Diffusione delle informazioni ed algoritmo di *flooding*

Una volta che le informazioni necessarie allo scambio sono state collezionate, ciascun

router costruisce un pacchetto contenente l'identità di chi invia, un numero di serie e la lista dei vicini (ciascuno con la stima di costo associata). L'algoritmo di diffusione serve a fare conoscere ad ogni ***gateway*** la completa topologia di rete a cui si è stati connessi. Esso si basa su di una tecnica di **flooding** in cui il numero di serie è usato per verificare la consistenza delle informazioni. Supponiamo che un pacchetto Psr (s indica il numero di serie ed r l'origine) giunga su di un *router* R:

- Se il pacchetto non è mai stato acquisito o se, al contrario, già esiste un pacchetto Ptr (con numero di serie t e con origine r) tale che s > t allora Psr viene forwardato su tutte le connessioni *point-to-point* con l'eccezione di quella da cui è arrivato.
- Se s = t il pacchetto è duplicato e si procede al suo scarto.
- Se il numero di serie del pacchetto s è inferiore di t si procede egualmente allo scarto per obsolescenza.

In base a questo semplice sistema, dopo un certo numero di iterazioni, ogni *gateway* conosce la topologia di rete cui è stato connesso.

Autonomous System: IGP - OSPF cammino minimo con algoritmo di Dijkstra

Non appena tutti i *router* hanno acquisito la topologia della rete, possono costruire un grafo pesato G che rappresenta le connessioni: ciascun link fisico è rappresentato da una coppia d'archi di direzione opposta e con pesi eventualmente differenti. Su G applicano l'algoritmo per il calcolo dei cammini minimi su grafo sviluppato da Dijkstra. Riportiamo di seguito i passi principali dell'algoritmo:

Dato l'insieme di tutti i nodi nella rete (i *router* che utilizzano OSPF) si definisca E l'insieme dei nodi già considerati ed R l'insieme di quelli rimanenti. Poniamo inizialmente E={r} (con r nodo rappresentante il *router* locale). Sia O l'insieme dei cammini uscenti da r; poniamo inizialmente O pari all'insieme dei cammini uscenti da r di lunghezza uno. Si ordini per metrica crescente O. Se O è vuoto o contiene soltanto cammini con metrica infinita, allora si possono marcare tutti i nodi in R come irraggiungibili. L'algoritmo termina.

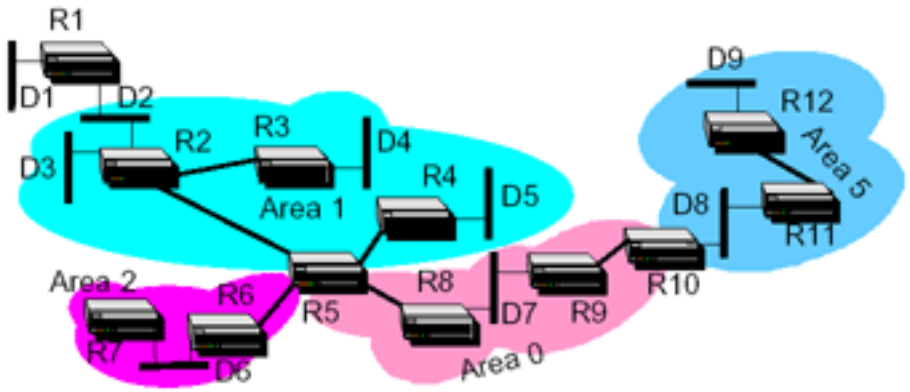
Si esamini il cammino di minore lunghezza P contenuto in O e lo si rimuova da questo insieme. Sia v l'ultimo nodo di P. Se v è già in E si ritorna al passo precedente altrimenti si è certi che P sia il più breve cammino da s ad v. Si sposta, quindi, v dall'insieme R all'insieme E. Si costruisca un nuovo insieme NP di j cammini candidati, collegando a P ciascun nodo fra gli nj adiacenti al nodo v. Il costo associato è pari al costo di P sommato al costo di nj. L'insieme NP è unito all'insieme O mantenendo l'ordinamento per costi crescenti. Si continua dal secondo passo.

L'algoritmo appena descritto si chiama *Shortest Path* (da cui il nome *Open Shortest Path First*) perché costruisce in modo incrementale l'insieme dei cammini minimi. Ad ogni passo si prova a verificare se sono soddisfatte le condizioni di *Bellman* (vedi riquadro). L'ordinamento dei cammini è realizzato mantenendo i nodi in una coda di priorità. Pur senza approfondire il perché, può essere utile citare il fatto che la complessità computazionale dell'algoritmo è pari ad O(MlogM) con M pari al numero di link di connessione contenuti nel *network* in esame. A titolo di paragone ricordo che i protocolli ***distance vector***, basandosi sull'algoritmo di *Bellman-Ford*, convergono in

O(MN) dove N è il numero di *router* presenti nel *network*.

Autonomous System: IGP - gerarchia OSPF e pacchetti

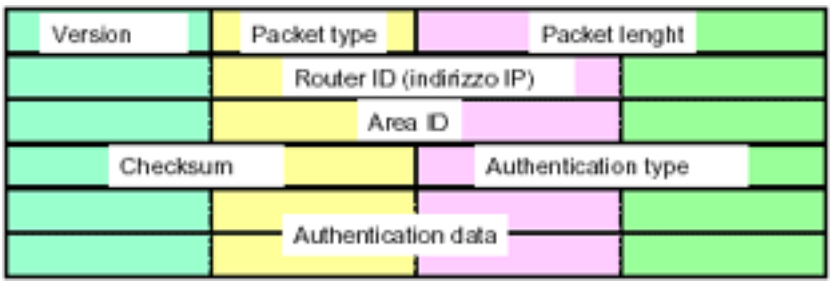
OSPF ha il concetto di gerarchia, dispone infatti del concetto di AS, il quale è ulteriormente suddiviso due livelli: *local area* (gruppo di reti contigue) e ***backbone*** (area particolare non necessariamente contigua).



Gli avvertimenti *Link-state* non lasciano rispettive aree. Segue un elenco dei termini comunemente menzionati nel contesto OSPF:

- *Backbone*: area di transito tra le altre aree
- *Backbone router*: *router* che è nel *backbone*
- *Area border router* (ABR): *backbone router* che si affaccia su più aree esegue una copia dell'algoritmo per ogni area
- *Internal router*: *router* che fa parte di un area diversa dal *backbone*.

Pacchetti OSPF



Version: specifica la versione del protocollo

Packet Type: identifica il tipo del pacchetto.

- *Hello*
- *Database Description*
- *Link state Request*
- *Link State Update*

- *Link State Acknowledgement*

router-id:
un numero su 32 bit che identifica univocamente il *router* che ha generato il pacchetto

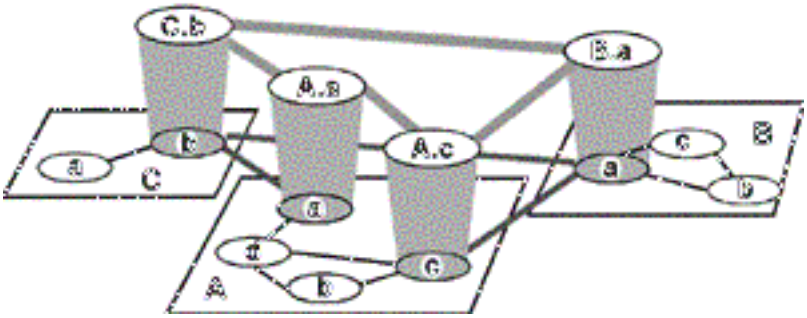
Area-id:
un numero su 32 bit che identifica a quale area appartiene il pacchetto

Authentication Type:
0 nessuna *authentication*
1 password in chiaro
altri valori riservati per usi futuri

Autonomous System: EGP - protocolli inter-dominio

Internet consiste di Sistemi Autonomi (***Autonomous Systems* AS**) interconnessi fra loro:

- *Stub* AS: piccolo
- *Multihomed* AS: grande (no *transit*)
- *Transit* AS: *provider*



BGP (*Border Gateway Protocol*) è un protocollo di *routing* tra domini, correntemente utilizzato sul *backbone* di Internet ed è in pratica il successore del protocollo **EGP** (*Exterior Gateway Protocol*).

In effetti questo protocollo viene usato soprattutto su Internet dove diversi AS sono collegati a questa grande rete attraverso strutture chiamate *Internet Service Provider* (ISP).

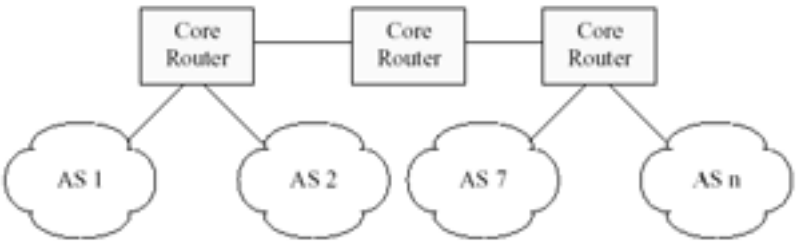
Il protocollo BGP costruisce un grafo di *autonomous system* basato sulle informazioni che si scambiano i *router*: questo grafo viene chiamato anche albero in cui ciascun AS viene identificato con un numero univoco. La connessione tra due AS si chiama percorso e una collezione di percorsi forma a sua volta un percorso che viene utilizzato per raggiungere la destinazione.

Autonomous System: EGP - exterior gateway protocol

L'***E**xterior **G**ateway **P**rotocol* è il primo **EGP** ad essere stato ampiamente utilizzato all'interno della rete Internet. Specificato con RFC 904 nell'aprile 1984 è oggi

ampiamente disponibile su tutti i *router*, anche se è ormai considerato un protocollo obsoleto e Internet lo sta sostituendo con il BGP.

EGP è simile ad un algoritmo *distance vector*, ma invece del concetto di costo specifica solo se la destinazione è raggiungibile oppure no. Questo ne impedisce il funzionamento su topologie magliate.
Esiste il concetto di un *core system* formato da una interconnessione di *core router*.



EGP genera dei pacchetti di *routing update* che contengono informazioni di *network reachability*, cioè annunciano che certe reti sono raggiungibili attraverso certi *router*. I pacchetti di *routing update* sono inviati ai *router* vicini ad intervalli di tempo regolari e raggiungono tutti i *router* EGP. L'informazione in essi contenuta è utilizzata per costruire le tabelle di instradamento.

I limiti di EGP sono molti e gravi: EGP non ha una metrica associata alle linee e quindi basa le sue decisioni esclusivamente sulla raggiungibilità; EGP non ammette la presenza di magliature nella topologia, e tutti gli AS devono essere collegati in modo stellare ad un *core system*; i pacchetti di *routing* update possono essere molto grandi.

Autonomous System: EGP - la famiglia BGP

Il ***B**order **G**ateway **P**rotocol* (**BGP**) è un *exterior gateway protocol* pensato per rimpiazzare il protocollo EGP ormai obsoleto. Il BGP è specificato per la prima volta dal RFC 1105 nel 1988, rispecificato come BGP-2 nel RFC 1163 nel 1990 e rispecificato ancora come BGPv3 nel RFC 1267 del 1991.

I *router* BGP comunicano tra loro utilizzando un livello di trasporto affidabile. Il BGP è un algoritmo di tipo *distance vector*, ma invece di trasmettere il costo di una destinazione, trasmette la sequenza di *autonomous system* da attraversare per raggiungere quella destinazione. Ogni *router* calcola il suo instradamento preferito verso una data destinazione e lo comunica ai *router* BGP adiacenti tramite un *distance vector*. La politica con cui tale calcolo avviene è configurabile su ogni *router* BGP.

Autonomous System: EGP - BGPv4 Funzionamento di base

Due sistemi danno luogo ad una connessione TCP tra di loro, dopodiché si scambiano messaggi per aprire e confermare le modalità di connessione.

All'inizio inviano entrambi nel flusso dei dati la loro intera tabella di instradamento. Man mano che si presentano variazioni dinamiche della tabella, inviano aggiornamenti al

peer. BGP non richiede periodici invii dell'intera tabella, quindi è necessario che ogni dispositivo che comunica mediante BGP gestisca e mantenga in memoria tutta la tabella di ognuno dei suoi *peer*.

Vengono comunque impiegati dei sistemi di *timeout* propri di BGP con dei messaggi di tipo *KeepAlive* per assicurare il mantenimento della connessione. Inoltre vi sono degli altri tipi di notifica nel caso vi siano condizioni di errore o altri particolari eventi.

È importante sottolineare che secondo l'RFC del protocollo, non è detto che entrambi i capi della connessione debbano per forza essere *router*. È infatti possibile che uno dei due sia un *host* che voglia solo controllare o gestire la propria tabella in maniera dinamica, o magari che faccia da punto di unione tra un segmento di rete di un AS ed il *router* perimetrale di un altro AS, senza per questo fungere da *gateway* del suo sistema autonomo.

a) Le *route*

Una *route* viene definita come una coppia che abbini ad una destinazione gli attributi del percorso per quell'indirizzo. Queste vengono emanate come messaggi BGP di tipo *UPDATE*: la destinazione è quel o quei sistemi i cui indirizzi IP sono riportati nell'NLRI o *Network Layer Reachability Information*, ed il percorso è l'informazione riportata nei campi di attributi della *route* che vengono riportati all'interno dello stesso *UPDATE*.

Le *route* vengono inserite e mantenute all'interno del RIB o *Routing Information Bases*. BGP permette anche di poter cancellare una *route* precedentemente immessa in un RIB remoto mediante 3 modi differenti:

- Inserendo il prefisso IP della destinazione nel campo *WITHDRAWN ROUTES* di un messaggio *UPDATE*;
- emanando una *route* alternativa che contenga lo stesso campo NLRI;
- terminando la connessione TCP, fatto che comporta la rimozione di tutte le *route* che i due *peer* avevano emanato l'un l'altro.

b) *Routing Information Bases* (RIB)

Questo consta di tre parti:

- *Adj-RIBs-In*: contiene le *route* ricevute mediante messaggi remoti di *UPDATE*. Queste verranno prese in considerazione dal processo decisionale di BGP.
- Loc-RIB: contiene le *route* presenti localmente che il processo decisionale BGP ha scelto tra quelle contenute in *Adj-RIBs-In*.
- *Adj-RIBs-Out*: contiene le *route* che il processo ha selezionato per essere emanate ai propri *peer*.

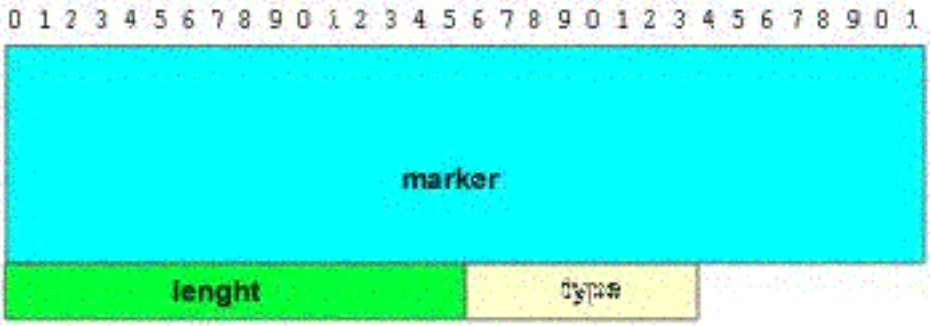
In pratica, *Adj-RIBs-In* contiene informazioni ricevute mediante *UPDATE* e non ancora processate, Loc-RIB quelle scelte come facenti parte della tabella di *routing*, mentre *Adj-RIBs-Out* quelle da inviare mediante propri messaggi di *UPDATE*.

Autonomous System: EGP - BGPv4 formato dei messaggi

Vediamo ora gli *header* BGP ed i tipi di messaggio che i *router* scambiano tra di loro

per iniziare una connessione BGP e mantenere il loro RIB. I messaggi possono essere lunghi fino ad un massimo di 4096 ottetti e non più corti di un *header* BGP, o 19 ottetti.

Nella figura seguente viene mostrato l'*header* BGP:



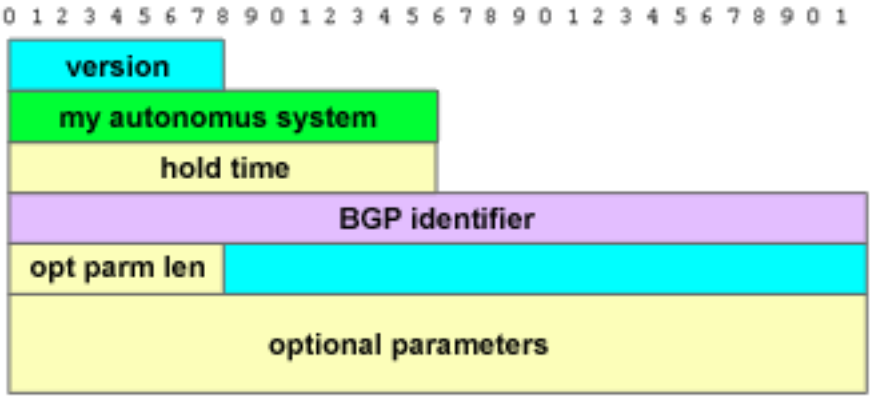
Il *marker* contiene un valore di 128 bit che può essere predetto dal *peer*. Infatti nel caso di un messaggio di tipo *OPEN*, o se un messaggio *OPEN* non contiene alcuna informazione relativa ai meccanismi di autenticazione, allora sarà composto da soli 1. Altrimenti deve contenere un valore computabile secondo gli algoritmi di autenticazione usati. Questo viene usato non solo per autenticare i messaggi in arrivo, ma anche per stabilire stati di mancata sincronizzazione tra i *peer* BGP.

La lunghezza (*length*) indica mediante un *unsigned* int la lunghezza totale del messaggio in *byte*, incluso l'*header* (minimo 19, massimo 4096, non deve contenere *padding*). Il tipo (*type*) di messaggio può essere uguale a:

- *OPEN*.
- *UPDATE*.
- *NOTIFICATION*.
- *KEEPALIVE*.

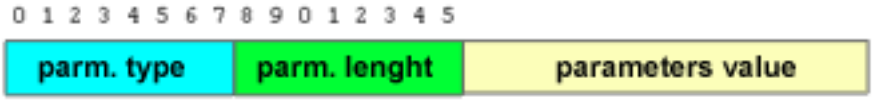
[1] Messaggio di tipo *OPEN*

Dopo una connessione TCP, il primo messaggio inviato da entrambi i *peer* è un *OPEN*. Nel caso sia accettabile, viene inviato un *KEEPALIVE* di conferma. Dopo che la sequenza *OPEN* è stata portata a termine, sarà possibile utilizzare i messaggi di tipo *UPDATE, NOTIFICATION* e *KEEPALIVE*. Oltre all'*header* BGP, sono presenti anche i seguenti dati:



La versione (*version*) è un *unsigned* int che identifica la versione del protocollo. Ora è uguale a 4. Dopo, un *unsigned* int di 16 bit contiene il valore dell' AS (*my autonomos system*). *Hold Time* contiene il valore in secondi entro cui bisogna inviare un messaggio *UPDATE* o *KEEPALIVE*, pena la caduta della connessione BGP. Impostando 0, non ci saranno *timeout*. L'RFC specifica un valore di almeno 3 secondi come base del valore al di fuori di 0.

L'identificativo BGP (BGP *identifier*) è in pratica un indirizzo IP associato ad una interfaccia utilizzata per l'invio dei messaggi. Ad esempio *Cisco* calcola questo valore usando il numero IP maggiore associato al *router*. Opt Parm Len indica la lunghezza in *byte* degli eventuali Parametri Opzionali. I parametri opzionali (*optionals parameters*) vengono rappresentati da una tripletta che contiene:



il tipo di parametro, la sua lunghezza ed il suo valore.

L'RFC di BGP4 specifica un solo parametro opzionale:

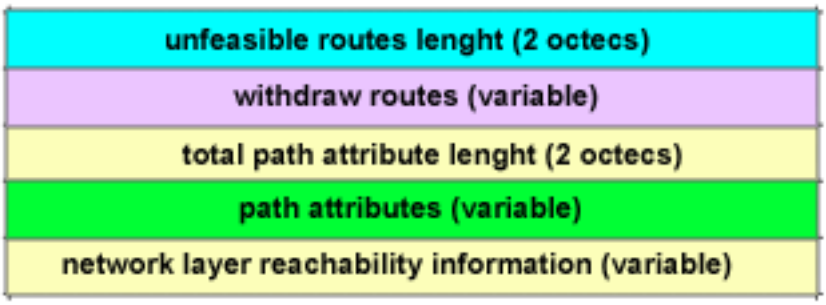
a) *Authentication Information* (*Parameter Type* 1)
Il campo *value* del parametro opzionale in questo caso contiene un codice di autenticazione seguito da un campo variabile contenente i dati necessari alla autenticazione:

È stato anche proposto l'uso di un'opzione TCP come la MD5 *Signature Option* per la gestione sicura dei segmenti TCP nel corso di sessioni BGP per poter ovviare a problemi di *hijack* e *man-in-the-middle*.

Il messaggio di tipo *OPEN* deve essere lungo almeno 29 *byte* compreso l'*header* del messaggio.

[2] Messaggio di tipo *UPDATE*

Questo messaggio viene usato per emanare o ritirare una singola *route* dal servizio. Può farlo anche insieme specificando differenti *route*. Contiene sempre l'*header* BGP e può anche contenere i seguenti campi opzionali:



I primi due ottetti contengono la lunghezza totale in *byte* del campo successivo e deve permettere di calcolare la lunghezza del NLRI [vedi sotto]. Un valore uguale a 0 indica che nessuna *route* viene ritirata e che quindi il campo *WITHDRAWN ROUTES* non è presente nel messaggio *UPDATE*.

Il campo successivo contiene una lista di prefissi IP codati come una coppia <lunghezza, prefisso> secondo questo schema:

Length (1 *octet*) - *Prefix* (*variable*)

la lunghezza indica il valore in bit del prefisso IP. Se uguale a 0 specifica tutti gli indirizzi IP. Ad esempio <24, 192.168.1.3> sta per 192.168.1.0/24 secondo la notazione CIDR, o *Classless Internet Domain Routing*, il prefisso invece l'indirizzo IP da associare ai bit di mascheramento.

Segue il campo di lunghezza totale degli attributi di percorso che indica in *byte* la lunghezza del campo che lo segue. Deve permettere di calcolare anche il NLRI. Un valore uguale a 0 indica che non c'è NLRI.

Gli attributi di percorso sono presenti in ogni messaggio di tipo *UPDATE*. Ogni attributo consta di una tripletta <tipo, lunghezza, valore>.

Il tipo consta di due ottetti, uno per delle *flag*, e l'altro per il codice vero e proprio:

Attr. *Flags* (1 *octect*) - Attr. *Type Code* (1 *octect*)

Il primo bit delle *flag* è il bit delle opzioni. Se uguale a 1 il parametro è opzionale, se uguale a 0 è invece riconosciuto. Il secondo bit identifica invece se un parametro debba essere propagato o meno, ovvero se sia transitivo, con valore uguale a 1, o non transitivo, con valore uguale a 0. Se il primo è uguale a 0, il secondo è sempre uguale a 1. Ovvero i parametri riconosciuti sono sempre transitivi. Il terzo bit specifica se i parametri opzionali e transitivi (11) sia parziale, se uguale a 1, o totale, se uguale a 0. Il quarto specifica se la lunghezza degli attributi sia di un solo ottetto, uguale a 0, o di due, se uguale a 1. Questo quarto bit può essere usato solo in caso di lunghezza superiore a 255 *byte*. Gli ultimi 4 bit non sono utilizzati e devono essere settati a 0. Se il 4 bit è uguale a 0, allora la lunghezza sarà specificata nel terzo ottetto della tripletta degli attributi; nel quarto se invece sarà uguale a 1.

I codici di attributo sono:

ORIGIN (*Type Code* 1) Contiene un valore per identificare la origine del'attributo di percorso.

- ottenuto mediante IGP;
- ottenuto mediante EGP;
- INCOMPLETO - ottenuto altrimenti.

AS_PATH (*Type Code* 2) contiene triplette per indicare le sequenze di percorso in segmenti della rete di AS. La tripletta prende la forma <tipo, lunghezza, valore>. Il tipo è un *long* con questi valori:

- *AS_SET* : set disordinato di AS.
- *AS_SEQUENCE* : set ordinato.

La lunghezza indica il numero di AS, mentre il valore contiene i numeri di AS percorsi.

NEXT_HOP (*Type Code* 3) Indica il prossimo *hop* da usarsi per arrivare alle destinazioni listate nell'NLRI.

MULTI_EXIT_DISC (*Type Code* 4) Opzionale non transitivo serve per il processo decisionale BGP per scegliere tra diversi punti di uscita verso il vicino AS.

LOCAL_PREF (*Type Code* 5) Serve per far conoscere il grado di preferenza del *router* mittente per una certa *route*.

ATOMIC_AGGREGATE (*Type Code* 6) Di lunghezza 0, serve per informare i *peer* che il sistema locale ha scelto una *route* non specifica, o meno, scartandone una più specifica contenuta in essa.

AGGREGATOR (*Type Code* 7) Opzionale transitivo, contiene il numero di AS che ha aggregato una *route*, seguito dal numero di *router* che ha aggregato.

COMMUNITY (*Type Code* 8) Implementato da *Cisco*, permette di definire la propagazione delle *route* al di fuori di comunità virtuali tra AS o *router*.

ORIGIN ID (*Type Code* 9) Implementato da *Cisco*, serve per identificare il *router* originante di una *route* dopo una sua riflessione in un AS per evitare doppioni nel caso venisse riflessa anche all'origine.

CLUSTER LIST (*Type Code* 10) Implementato da *Cisco*, contiene una lista di ID attraversati durante la riflessione di una *route* al di fuori del *cluster* di *client*.

Dopo segue l'NLRI. La sua lunghezza viene calcolata sulla base della lunghezza degli altri campi del messaggio *UPDATE*, secondo questo algoritmo:

Lunghezza del messaggio *UPDATE* - 23 - *Path Attributes* - *Withdrawn Routes* dove 23 è la lunghezza dell'*header* BGP, e dei due campi che contengono la lunghezza degli attributi e dei *WITHDRAWN*.

Le informazioni sulle destinazioni sono contenute mediante coppie di valori <lunghezza, prefisso>:

Length (1 *octet*) - *Prefix* (*variable*)

questi due valori sono gli stessi utilizzati anche per rimuovere le *route*, come visto in precedenza. La lunghezza minima è di 23 *byte* come visto prima, 19 per l'*header* BGP, 2 per *Unfeasible Routes Length* e 2 per *Total Path Attribute Length*.

Ogni messaggio *UPDATE* può al massimo emanare una sola *route*, come riportato nel

campo dell' NLRI, e può descriverla con una serie di attributi. Può invece ritirare più di una *route* con il campo *Withdrawn Routes*.

[3] Messaggio di tipo *KEEPALIVE*

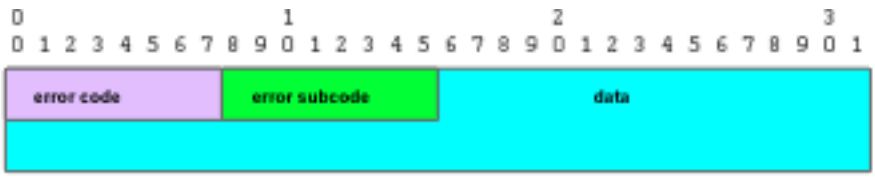
BGP non usa il sistema di *timeout* e *retransmission* proprio di TCP per gestire e controllare la propria connessione tra *peer*, ma utilizza questo messaggio per valutare lo stato delle *route* emanate da ogni *peer*, prima che il livello di trasporto rilevi dei problemi di rete.

I messaggi di tipo *KEEPALIVE* non devono essere inviati più velocemente di uno per secondo, ma devono comunque raggiungere il *peer* prima che il suo *Hold Timer* raggiunga la fine. Un valore ragionevole è un terzo del *Timer* remoto. Se il valore è stato negoziato uguale a 0, allora i messaggi non devono essere inviati.

Questo messaggio consiste del solo *header* BGP di 19 *byte*.

[4] Messaggio di tipo *NOTIFICATION*

Questo messaggio viene inviato appena venga rilevata una condizione di errore. Dopodiché la connessione BGP viene conclusa. Oltre all'*header* BGP, questo messaggio contiene i seguenti dati:



il codice di errore è un *unsigned* int che può contenere i seguenti valori:

- *Message Header Error*.
- *OPEN Message Error*.
- *UPDATE Message Error*.
- *Hold Timer Expired*.
- *Finite State Machine Error*.
- *Cease*.

il sottocodice contiene dei valori associati ad ognuno dei codici per meglio definire il tipo di errore [come succede nel caso dei pacchetti ICMP]. Se non contiene alcun valore deve essere settato a 0.

Ecco i sottocodici assegnati dall'RFC:

Message Header Error subcodes:

- *Connection Not Synchronized*.
- *Bad Message Length*.
- *Bad Message Type*.

OPEN Message Error subcodes:

- *Unsupported Version Number*.
- *Bad peer* AS.

- *Bad BGP Identifier*.
- *Unsupported Optional Parameter*.
- *Authentication Failure*.
- *Unacceptable Hold Time*.

UPDATE Message Error subcodes:

- *Malformed Attribute List*.
- *Unrecognized Well-known Attribute*.
- *Missing Well-known Attribute*.
- *Attribute Flags Error*.
- *Attribute Length Error*.
- *Invalid ORIGIN Attribute*
- *AS routing Loop*.
- *Invalid NEXT_HOP Attribute*.
- *Optional Attribute Error*.
- *Invalid Network Field*.
- *Malformed AS_PATH*.

il campo dati invece contiene elementi necessari per diagnosticare l'errore. Dipende strettamente dal codice e dal sottocodice di errore. Per maggiori informazioni controllare l'RFC di BGP4.

Autonomous System: EGP - Negoziazioni e UPDATE

Negoziazioni

I dispositivi paritari BGP dispongono della possibilità di negoziare durante il messaggio *OPEN* la versione d BGP supportata da entrambi. Nel caso di una versione non supportata, ogni *router* avrà la versione richiesta dal *peer* con l'*OPEN*, quella da lui tentata, quelle ricevute dal *peer* mediante un *NOTIFICATION* e quelle disponibili localmente. In questo modo saranno in grado di scegliere una versione comune per la connessione BGP. D'altra parte è ovvio che future versioni di BGP, per poter supportare la negoziazione della versione, dovranno mantenere l'uso e la conformazione dei messaggi *OPEN* e *NOTIFICATION*.

Elaborazione dei Messaggi *UPDATE*

Dopo che la connessione TCP sia stata effettuata, i messaggi *OPEN* e *KEEPALIVE* scambiati, il *timer* inizializzato, è possibile ricevere messaggi di tipo *UPDATE*.

Se il messaggio contiene un campo *WITHDRAWN ROUTES* non vuoto, allora le *route* precedentemente emanate con quei prefissi IP saranno eliminate da *Adj-RIB-In*. Dopodiché il processo decisionale penserà a cancellare la *route* dal *Local-RIB* o a sostituirle con altre possibili.

Se il messaggio contiene una *route* accettabile allora succederà questo:

- se la nuova è identica ad una già presente in *Adj-RIB-In*, allora quella vecchia verrà sostituita e quindi cancellata, costringendo il processo BGP a sostituirla in *Local-RIB*.
- se la nuova *route* fa parte di una precedente contenuta in *Adj-RIB-In* allora

il processo decisionale farà prevalere la nuova in quanto più specifica di quella precedente

- se la nuova ha uguali attributi ma è più specifica allora non servono altre azioni
- se la nuova ha un NLRI non presente nelle vecchie *route* verrà inserita subito nel *Adj-RIB-In*
- se la nuova è una meno specifica di una precedente allora il processo decisionale valuterà solo il set di IP raggiungibili con la nuova *route*.

Come funziona questo processo decisionale? Esso è suddiviso in tre fasi distinte. Nella prima BGP deve calcolare un livello di preferenza per ogni *route* disponibile in *Adj-RIB-In*. Per *route* interne all'AS spesso il solo parametro *LOCAL_PREF* verrà usato, altrimenti i parametri di percorso saranno valutati secondo il PIB o *Policy Information Base*, i cui dati sono a cura di ogni AS.

La seconda fase serve per scegliere la *route* considerata più efficiente tra quelle disponibili per ogni destinazione. Dopo averla scelta, per miglior livello di preferenza, od in quanto nuova ed unica *route* per una destinazione, la porrà in *Local-RIB*. Esistono comunque delle regole specifiche nel caso diverse *route* siano in parità per quanto riguarda la preferenza.

Nella terza fase BGP si occupa di valutare *Local-RIB* per poter ottimizzare secondo le *policy* dell'AS il *Adj-RIB-Out*. Nel momento in cui la valutazione di *Local-RIB* ed il processo di *Adj-RIB-Out* siano completi, allora potrà avere luogo l'invio di messaggi *UPDATE*.

Configurazione di un router

Franco Callegati
Paolo Zaffoni

8.2.1 (Distinguere tra opzioni basate su router, su switch e su bridge)

Accesso e configurazione di base

L'accesso al *router* può avvenire:

- via rete (***telnet***);
- collegando un terminale (o un PC) alla *console* (fisicamente è una porta seriale asincrona classica) del *router*.

Nel primo caso è possibile la gestione da remoto; nel secondo caso è necessario essere in locale (oppure collegato al *router* tramite modem) e impostare i corretti parametri del programma di emulazione terminale (*HyperTerminal* nel mondo *Windows*). L'accesso in locale è obbligatorio nella fase di configurazione iniziale del *router*; successivamente è possibile utilizzare anche l'accesso *telnet* (se il *router* è raggiungibile).

Una terza possibilità è quella dell'utilizzo di strumenti di configurazione remota, quali gli strumenti di gestione basati su SNMP. Questa soluzione non consente tuttavia la completa configurazione della macchina; per alcuni aspetti è comunque necessario intervenire sulla configurazione del *router* manualmente.

Un'ultima possibilità, utilizzata soprattutto per controllare alcuni parametri di base del *router*, è quella di accedere alla macchina attraverso un *browser web* (il servizio può essere abilitato tramite il comando IP http *server* dalla modalità di configurazione). Questa modalità offre tuttavia funzionalità estremamente limitate (status di ogni interfaccia, ...) ed è utilizzato più come controllo che come strumento di configurazione.

Comandi fondamentali

Anziché presentare in questa sezione l'elenco dei principali comandi possibili, si presenta un esempio di configurazione reale di un *router* spoglio.

- *Enable*
  Entra in modalità amministrazione (richiede una *password*).
- *Erase startup_config*
  Cancella la configurazione della NVRAM ed azzera la configurazione del *router*. È importante notare che questo comando va impartito in modalità privilegiata e non in modalità di configurazione.
- *Configure terminal* | *memory* | *network*
  Entra in modalità di configurazione; i comandi di configurazione verranno impartiti dal *medium* specificato dalla seconda parola chiave. In altre parole, se viene specificata la *keyword terminal* i comandi verranno accettati dalla tastiera; se viene indicata la parola *memory* verrà copiata la configurazione di *startup* in quella volatile ed eseguita, mentre con la parola chiave *network* verrà ricercato un *server* TFTP sul quale è memorizzata la configurazione la quale verrà quindi caricata in memoria ed eseguita.

- *Hostname name*
  Assegnazione del nome al *router*.
- *Enable password* ena_pwd
  Abilitazione (e configurazione) della *password* del *router* locale (quella richiesta alla digitazione del comando *enable*).
- *Username name password* passwd
  Associa *password* a nomi. Può essere utilizzato sia per accedere ad un *router*, sia per configurare il *router* ad accedere in *dial/up* ad un altro apparato. Nel secondo caso, il *router* usa come *password* quella associata al proprio nome.
- *Line* vty 0 4
  Configura i terminali virtuali: il primo numero dopo il VTY indica il numero del primo terminale virtuale; il secondo indica il numero dell'ultimo terminale virtuale (in questo caso è stata configurata la possibilità di 5 accessi contemporanei al *router*).
- *Login*
  Imposta l'obbligo di una fase di *login* nell'accesso via *telnet* (ma non impone una *password*).
- *Password telnet*_pwd
  Abilita (e configura) della *password* di accesso al *router* via *telnet*.
- *Exit*
  Esce dalla modalità di configurazione dei terminali virtuali.
- *Exit*
  Esce dalla modalità di configurazione.
- *Show running-config*
  Visualizza su *monitor* l'attuale configurazione (RAM) del *router*.
- *Copy running-config startup-config*
  Salva nella NVRAM la configurazione attiva.
- *Show startup-config*
  Visualizza su monitor la configurazione salvata su NVRAM.

NOTA: Nella visualizzazione di una configurazione (ad esempio *sh run*) vengono riportate solo le opzioni che non sono al valore standard.

Altri comandi fondamentali (attivabili solo in modalità *enable*)

- *copy running-config* tftp:nome*file*
  Salva su un *server* TFTP la configurazione attiva.
- *Reload*
  Effettua il *reboot* del *router*.

Modifiche tra IOS

Nelle più recenti versioni di IOS (fondamentalmente > 12.0) sono stati modificati pesantemente i comandi relativi alla visualizzazione e alla gestione della configurazione. In figura sono riportati i principali comandi e le principali variazioni del settore. I comandi tradizionali sono ancora supportati, ma è preferibile utilizzare i nuovi.

In breve i comandi sono stati unificati sotto le tre voci *copy*, per la gestione dei *file* di configurazione, *more*, per la loro visualizzazione, e *erase* per la loro cancellazione.

L'argomento di questi comandi include quindi il *device* fisico dove il *file* è memorizzato (ad esempio tftp:, system:, nvram:, etc) e la sua locazione all'interno di questo *device*.

Vecchi comandi	Nuovi comandi
configure network	*copy ftp: system:running-config*
copy rcp running-config	*copy rcp: system:running-config*
copy tftp running-config	*copy tftp: system:running-config*
configure overwrite-network	*copy ftp: nvram:startup-config*
copy rcp startup-config	*copy rcp: nvram:startup-config*
copy tftp startup-config	*copy tftp: nvram:startup-config*
show configuration | *show startup-config*	*more nvram:startup-config*
write erase | *erase startup-config*	*erase nvram:*
write memory | *copy running-config startup-config*	*copy system:running-config nvram:startup-config*
write network	*copy system:running-config ftp:*
copy running-config rcp	*copy system:running-config rcp:*
copy running-config tftp	*copy system:running-config tftp:*
write terminal | *show running-config* *more*	*system:running-config*

Controllo e debug

I principali comandi di utilità, controllo e *debugging* sono solitamente disponibili solo in modalità privilegiata.

- *show* comando
  Visualizza i parametri relativi a comando;
- *show ?* (oppure *show* ip)
  Elenca ciò che è possibile visualizzare;
- *term mon* (*term no mon* per la sua disattivazione)
  Attiva il *debugging* sul monitor (necessario solo via *telnet*, per attivare l'*output* su terminale locale e non sulla *console* del *router*);
- *debug* comando
  Per attivare il *debug* su una funzione specifica;
- *debug ?*
  Mostra le attività su cui il *debugging* può essere attivato;
- *debug ip packet dump*
  Stampa su *monitoring* il *dump* esadecimale dei pacchetti che passano nel *router*; è un comando molto pericoloso per la sua capacità di saturare il *router*;
- *no debug all*
  Per disabilitare tutti i comandi *debug* attivati in precedenza.

Il *debug* deve essere lanciato con cura evitando di saturare la CPU e la capacità trasmissiva (nel caso di *debug* remoto) a disposizione del *router*. Non è infrequente che il *router* risulti saturato dalla gestione dei messaggi di *debug* e che non riesca più ad accettare altri comandi di nessun tipo. In queste condizioni il *debug* provoca la totale perdita di controllo sul *router* che può essere riattivato solamente attraverso l'utilizzo della *console* dello stesso.

Altri strumenti di controllo

Altri comandi utili per il controllo dell'operabilità del *router* sono quelli classici dell'ambiente TCP/IP, e cioè:

- *ping* [indirizzo]
  Controlla la raggiungibilità di indirizzo;
- *trace* [indirizzo]
  Visualizza il percorso verso la destinazione; nel caso di più percorsi, li visualizza tutti;
- *telnet* [indirizzo]
  Apre un terminale virtuale con la destinazione.

Può essere importante ricordare che questi strumenti di diagnostica sono molto approssimativi. Ad esempio una mancanza di risposta al comando *ping* non implica automaticamente la mancanza di una *route* per raggiungere la destinazione, ma può anche essere l'eventuale mancanza di una *route* per il ritorno. È quindi importante accertarsi in prima battuta che i vicini al *router* in esame siano raggiungibili, per poi proseguire il *debug* secondo cerchi concentrici a raggio sempre maggiore.

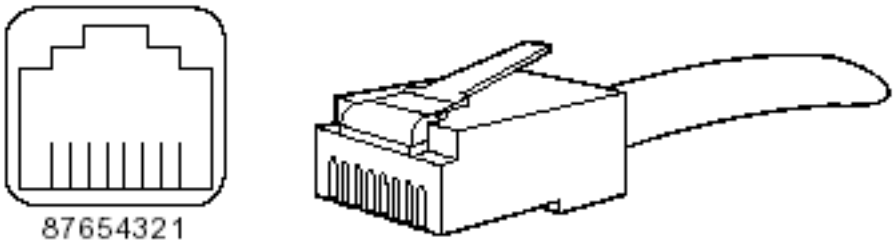
Interconnessione con la rete fisica

Cavi RJ-45

I prodotti *Cisco* utilizzano i seguenti tre tipi di cavi RJ-45:

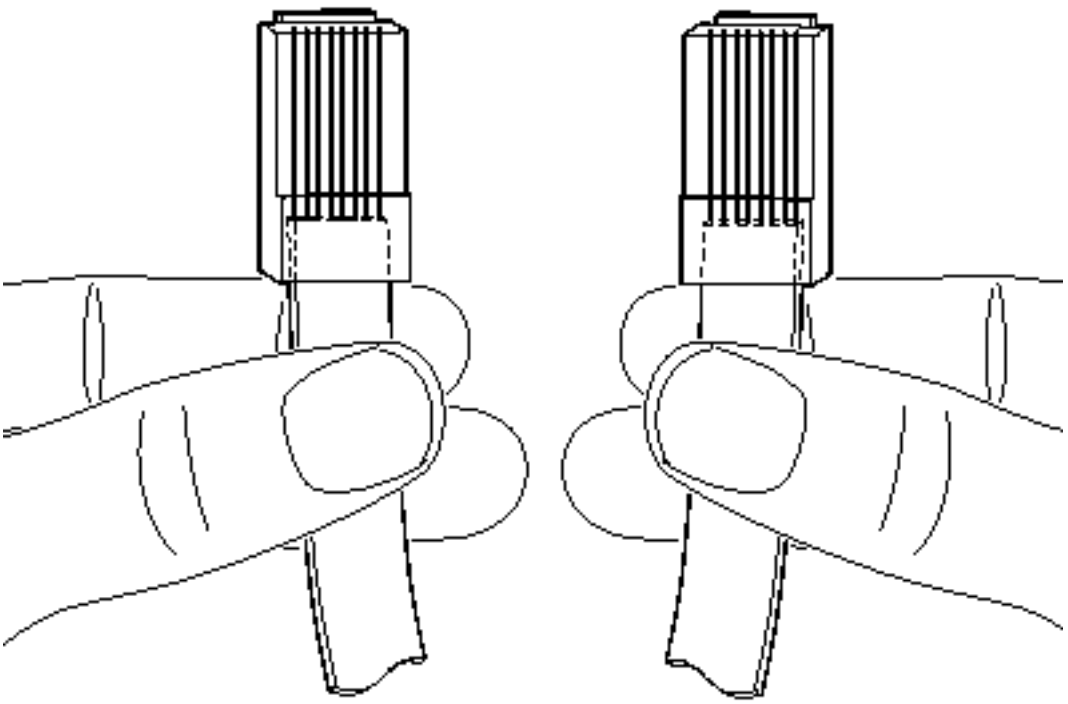
- *Straight-through* (dritto).
- *Crossover* (incrociato).
- *Rolled*.

Cisco tipicamente non li fornisce e sono in genere largamente disponibili da terze parti.



RJ-45 connector

Per identificare il tipo di cavo RJ-45, si deve tenere in mano i due estremi del cavo in modo da poter vedere la colorazione dei cavi all'interno del *plug*, come mostrato in figura.



Si esamini la sequenza di colorazione dei cavi internamente al *plug* RJ-45 (trasparente):

- *Straight-through* [dritto]
  la colorazione dei cavi nel *plug* procede identicamente in ambedue gli estremi.
- *Crossover* [incrociato]
  il primo (più a sinistra) colore di cavo su di una lato del cavo è il terzo sull'altro lato.
- *Rolled*
  i colori dei cavi nel *plug* di un estremo del cavo sono in ordine esattamente opposto a quelli dell'altro estremo.

Layout cavi

Cavo straight-trought

Signal	*Pin*	*Pin*	*Signal*
Tx+	1	1	Tx+
Tx-	2	2	Tx-
Rx+	3	3	Rx+
-	4	4	-
-	5	5	-
Rx-	6	6	Rx-
-	7	7	-
-	8	8	-

Cavo cross-over

Signal	Pin	Pin	Signal
Tx+	1	3	Rx+
Tx-	2	6	Rx-
Rx+	3	1	Tx+
-	4	4	-
-	5	5	-
Rx-	6	2	Tx-
-	7	7	-
-	8	8	-

Cavo straight-trought

Signal	Pin	Pin	Signal
-	1	8	-
-	2	7	-
-	3	6	-
-	4	5	-
-	5	4	-
-	6	3	-
-	7	2	-
-	8	1	-

Configurazione delle interfacce

L'IOS assegna ad ogni interfaccia fisica di rete un identificativo univoco all'interno del sistema. Questo identificativo è formato dal loro nome tecnologico più un identificativo numerico (quindi si troveranno *Ethernet0, Ethernet1, Serial0, Serial1*) in ordine crescente. Nel caso di apparati composti da *chassis*, il numero dell'interfaccia comprende anche il numero dello *chassis* (ad esempio *Ethernet0/1* indica la seconda *Ethernet* del primo *chassis*).

Da questo punto in poi si seguiranno le seguenti regole:

- i comandi, eccetto quando chiaramente specificato, saranno comandi disponibili solamente in modalità configurazione (o da un suo sottomenu)
- a questa regola fanno eccezione i comandi inizianti per *show*, i quali sono disponibili esclusivamente in modalità *enable*.

Comandi generali

Sono quei comandi che, impostati in modalità *enable*, permettono successivamente la configurazione opportuna delle interfacce vere e proprie.

Sono normalmente dei comandi di tipo generale che hanno validità per tutto il *router*.

ip *subnet-zero*

Abilita l'uso della *subnet* zero sulle interfacce e sulle *routing updates*. In mancanza di questo comando le reti terminanti con "0" non sono ammesse se non con *netmask* naturali (/24, /16 e /8); ad esempio non è ammessa la rete 130.192.1.0/30, mentre lo è la 130.192.1.4/30

Comandi di interfaccia

Sono comandi che vanno dati all'interno della configurazione delle interfacce.

- *interface name*
  Entra nel sottomenu di configurazione dell'interfaccia *name*. Questo comando permette l'entrata nel sottomenu di configurazione delle interfacce abilitando quindi la digitazione dei comandi successivi
- ip *address* indirizzo maschera [*secondary*]
  Assegna all'interfaccia l'indirizzo indirizzo. L'opzione *secondary* indica che l'indirizzo è secondario e permette la configurazione di più indirizzi IP sulla stessa interfaccia fisica
- *description* descrizione_interfaccia
  Assegna una stringa letterale per la descrizione dell'interfaccia
- *shutdown*
  Disabilita il funzionamento di quell'interfaccia (può essere utilizzato ad esempio dalle interfacce ISDN per forzare la terminazione della chiamata corrente); per riattivare l'interfaccia è necessario digitare *no shutdown*
- mtu valore
  Definisce una MTU diversa rispetto a quella standard
- ip *proxy* arp
  Abilita il *proxy* arp su quell'interfaccia

Indirizzi delle interfacce

Per configurare ed attivare le interfacce di rete è necessario seguire alcuni semplici ma indispensabili passi. Le interfacce dei *router Cisco* vanno configurate ponendo attenzione alla tipologia delle stesse e alla porta fisica.

Router(config)# **interface tipo porta**

Ad esempio per configurare la porta *Ethernet0* i comandi da eseguire saranno i seguenti:

Router<enable
Router#config
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface ethernet0
Router(config-if)#
Router(config-if)# media-type 10baset

Lo stato del *prompt* cambia in *Router(config-if)#* per permettere ulteriori configurazioni dell'interfaccia stessa. La specifica *media-type* è d'obbligo per tutti i *router* che hanno disponibilità di interfacce *Ethernet* con connettore sia RJ45 che AUI. Per *default* il media-type è di tipo AUI.

Proseguendo nella configurazione è necessario assegnare l'indirizzo IP (diamo per scontato che si utilizzi il protocollo IP) e la *subnetmask*.
La sintassi impone questa metodologia di imputazione dei comandi:

Router(config-if)# **ip *address netmask address***

Per configurare la nostra interfaccia *Ethernet* con l'IP privato 192.168.150.1 dovremo eseguire i seguenti comandi

Router(config-if)#ip address 192.168.150.1 255.255.255.0
Router(config-if)#no shutdown
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet 0, changed state to up
Router(config-if)#

Nella configurazione di un *router Cisco* assume importanza l'identificazione e l'impostazione dei protocolli di *routing*.

Nomi delle interfacce

In un *router* ogni interfaccia ha un indirizzo IP ben preciso e differente dalle altre (tralasciando i casi di interfacce *unnumbered* o *negotiated*). È buona pratica assegnare un nome simbolico ad ogni interfaccia, inserendolo sia nel *router* tramite il comando IP *host* che in un DNS. Ad esempio:

ip *host* s0_bbone 10.25.23.3
ip *host* e0_bbone 10.25.24.1
ip *host* roma_bbone 10.25.24.1
ip *domain-name* rete.it

Scegliamo nel caso sopra come interfaccia principale del *router* l'interfaccia *Ethernet0* (e0). Quindi roma_bbone.rete.it corrisponderà allo stesso indirizzo di e0_bbone.rete.it. Occorre utilizzare un dominio fittizio (rete.it) oppure un dominio reale assegnato dal NIC (questo a seconda che sia una rete privata o una rete pubblica).

Nella configurazione delle interfacce è inoltre buona regola:

- Utilizzare sempre il comando ***shutdown*** se l'interfaccia non è attiva.
- Ricordare di rimuovere indirizzi, descrizione, ecc. se l'interfaccia è stata disattivata. Questo soprattutto se la gestione della rete è affidata a più persone.
- Utilizzare sempre il comando ***description***, che permette di assegnare ad una interfaccia una descrizione. È buona regola in questo caso inserire nella descrizione informazioni aggiuntive, come la tecnologia del collegamento (ISDN, *Frame-Relay*, CDN, ecc.) il numero della linea (DLCI, N. CDN, N. ISDN), punti collegati o cliente collegato all'interfaccia, eventualmente la velocità. Ad esempio: *interface Serial0*
  description Collegamento Terni-L'Aquila - CDN N. xxxxxx/yy - Vel. 128KBit.
  Inoltre se si ha un buon sistema di gestione o ci si vuole cimentare nello *scripting*, si può prelevare queste informazioni tramite SNMP e costruire automaticamente una tabella *excel* o un *database* dei collegamenti attivi. Molto utile anche per generare inventari sempre aggiornati, che alternativamente andranno prodotti manualmente.
- Specificare sempre tramite il comando *bandwidth* la velocità del collegamento. In questo modo le statistiche di occupazione che si possono

visualizzare tramite il comando *show interface* indicheranno il carico corretto della linea (xxx/255). Ad esempio: *interface Serial0* *bandwidth* 128.

Notare che lo specificare la velocità del collegamento sull'interfaccia non limita in alcun modo le *performance* del collegamento ma serve solo a fini statistici per aggiornare in modo corretto le stime di occupazione. Quindi se ad esempio si specifica *bandwidth* 1024 su un collegamento a 2MBit/Sec (2048KBit/Sec) non si limita l'utilizzo del collegamento ad un solo MBit/secondo, ma semplicemente le statistiche di occupazione saranno calcolate dal *router* come se il collegamento avesse una velocità di un solo MBit/secondo.

Infine se si dispone di reti libere a disposizione (ad esempio si dispone di una rete privata e con molti o tutti gli indirizzi liberi) utilizzare sempre una interfaccia di *loopback* configurata su ogni *router*.

In questo modo si avrà a disposizione una interfaccia sempre *up* e raggiungibile, indipendentemente dal fatto che le altre siano *up* o *down*:

interface loopback0
ip *address* 10.149.250.5 255.255.255.252

I vantaggi dell'utilizzare una interfaccia di *loopback* su ogni *router* sono diversi, questi alcuni:

- Si dispone di una interfaccia sempre *up* a cui far fare riferimento dai sistemi di gestione per fare i *poll* SNMP, prelevare le configurazioni, andare in *telnet*, ecc.
- Si può utilizzare l'indirizzo assegnato alla *loopback* per rendere *unnumbered* delle interfacce ad esempio seriali o ISDN (non consigliabile se si dispone di indirizzi a disposizione poiché rende più complessa la gestione) ma se si è a corto è una soluzione.
- Si può usare la *loopback* come peer del processo BGP per renderlo più stabile (l'interfaccia è sempre UP).
- *Tunneling*: usare l'interfaccia di *loopback* come estremo di un tunnel rende il tunnel stesso più stabile (se si usa ad esempio STUN o GRE o IPSEC).

Nomi degli apparati

Un'altra cosa da prendere in considerazione sono i nomi che vengono dati agli apparati. Si vedono spesso reti nelle quali gli apparati hanno nomi di fantasia (ad esempio Zeus, Athena, ecc.). Questo tipo di pratica è senz'altro divertente ma molto meno efficace dal punto di vista della gestibilità della rete. Molto meglio dare agli apparati nomi che denotino, ad esempio, la loro locazione fisica. Sarà poi più semplice per il gestore di rete capire su quale apparato e in quale punto della rete è collegato in un certo momento.

Per esempio, un utilizzo sensato del comando *hostname* dell'IOS è il seguente.

hostname roma_bbone

hostname terni_access

hostname aquila_ISDN

denotando ad esempio la locazione fisica e l'utilizzo che viene fatto del *router* (*router* di *backbone*, *router* di accesso ISDN, ecc.).

Inoltre l'*host name* può essere reperito tramite SNMP da un eventuale sistema di gestione: questo permetterà di rappresentare la mappa fisica della rete su, ad esempio, HP *Openview* NNM in modo molto più semplice. È buona norma documentare in una tabella *excel* i nomi degli apparati, la locazione fisica, gli indirizzi, ecc; quindi occorre configurare un DNS dove inserirete i nomi degli apparati con i rispettivi indirizzi IP: sarà in questo modo molto più semplice raggiungerli utilizzando nomi, invece di indirizzi. Se si utilizza un DNS, bisogna configurare tutti i *router* in modo che facciano riferimento ad esso per risolvere i nomi:

ip *domain-server* 10.5.4.1

dove 10.5.4.1 è l'indirizzo IP del vostro DNS. Se invece non si dispone di un DNS *server* si inserisca il comando:

no ip *domain-lookup*

per evitare che il *router* cerchi di risolvere ogni nome a lui sconosciuto che si inserisce da linea di comando.

Dial on Demand

Le funzionalità di *Dial on Demand* sfruttano la possibilità di avere connessioni alla rete telefonica pubblica (ad esempio utilizzando un modem, un *Terminal Adapter* per la rete **ISDN**, una scheda di rete ISDN, la rete **X.25**) che vengono utilizzate per attivare dei circuiti di comunicazione di tipo temporaneo tra due *end points*. In generale, questo collegamento è attivato sulla base di un certo evento predefinito, ad esempio la caduta di un circuito fisico primario, la necessità di trasportare un determinato pacchetto dati, e altro. L'attivazione del collegamento comporta l'invio di opportuni comandi di segnalazione da parte dell'interfaccia fisica verso la rete pubblica, in modo da permettere lo scambio dati.

Le motivazioni che portano all'impiego del *Dial on Demand routing* sono principalmente:

- Connessioni *Dial-up*: vengono utilizzate per attivare un canale di comunicazione primario ad-hoc nel caso in cui questo canale non debba essere permanentemente attivo; tipico esempio di impiego è una sede periferica di un'azienda la quale attiva un collegamento per lo scambio dati con la sede centrale solo nel momento in cui ve ne è la necessità. Questa scelta presuppone un traffico ridotto in volume e concentrato in alcuni momenti particolari; ad esempio non è la soluzione ideale per collegare un *server web* ad Internet.
- *Backup*: è utilizzato in caso di soluzione temporanea in caso di caduta di un altro link primario. Il collegamento di *backup* consente di ovviare al disservizio utilizzando un canale temporaneo per lo scambio di dati. Normalmente la connessione di *backup* viene abbattuta nel momento in cui

il collegamento primario ritorna operativo.

- Trabocco: è utilizzato quando la capacità del link primario non è più sufficiente a smaltire il traffico. Viene quindi attivato un nuovo collegamento in parallelo al primo in grado di espandere le capacità del canale (inteso come aggregato di due canali). Questa soluzione può essere valida nel caso in cui il traffico in eccesso è concentrato in particolari momenti, e non conviene quindi adottare un canale dedicato ad-hoc, allocato permanentemente.

È da rimarcare come *Backup* e *Dial on Demand routing* siano due cose separate: è possibile abilitare le funzionalità di *backup* su qualunque link, anche se normalmente i link di tipo *dial-up* sono quelli più utilizzati. In linea di principio è quindi possibile utilizzare due collegamenti seriali in parallelo, permanentemente allocati, di cui uno *backup* dell'altro. Il DDR è conveniente nel momento in cui l'ammontare di traffico è limitato ed è concentrato in brevi periodi di tempo. Anche nel caso di funzionalità di *backup*, è necessario che la linea primaria non abbia disservizi troppo spesso, altrimenti può essere più conveniente una seconda linea dedicata di *backup* anziché una linea ISDN.

Una criticità per il DDR è, ad esempio, l'attivazione di protocolli di *routing* sul collegamento *dial-up* in quanto lo scambio di messaggi di servizio tra *router* adiacenti può mantenere attivo il collegamento 24 ore su 24.

Attivazione e disattivazione di una chiamata

Il DDR si basa su un concetto di evento: il collegamento *dial-up* viene instaurato automaticamente nel momento in cui l'evento viene verificato. Esempi di eventi possono essere la perdita del segnale di portante su un link primario, il superamento di una certa soglia di traffico su un link, oppure più banalmente la richiesta di attivazione manuale da parte dell'utente.

Un evento abbondantemente utilizzato in pratica è la definizione di alcune categorie di traffico (ad esempio alcuni pacchetti dati) che possono scatenare la chiamata. Ad esempio, si potrebbe volere che solamente i pacchetti destinati ad un indirizzo IP (quello del *server* aziendale) abbiano il diritto di far partire la chiamata verso la sede centrale, mentre tutte le altre destinazioni devono essere ignorate.

La fase di riconoscimento dell'evento viene seguita dalla fase di attivazione del canale; i dati verranno inoltrati solo a canale instaurato. È evidente come l'utilizzo in produzione di funzionalità DDR richiede che la fase di connessione sia veloce onde non indisporre gli utenti. Una connessione DDR attraverso un modem, tecnicamente possibile, ha dei tempi di collegamento decisamente elevati e quindi sono preferite connessioni di tipo **ISDN**.

La disattivazione del canale ha un andamento speculare alla sua attivazione: viene verificato un determinato evento (ad esempio la riduzione del traffico sotto una certa soglia, la riattivazione del canale primario, la mancanza di traffico per un certo tempo) al seguito del quale il canale viene abbattuto.

L'evento di chiamata può essere verificato più volte (ad esempio nel caso di soglia di traffico, oppure mediante l'indirizzo IP destinazione); nel caso in cui la chiamata sia già attiva, il nuovo evento viene ovviamente ignorato e il traffico relativo ad esso viene

veicolato sulla connessione precedentemente instaurata.

Opzioni di chiamata

Una connessione *dial-up* (di tipo **ISDN**) necessita di alcuni parametri di configurazione:

- Numero di telefono del chiamato: è necessario specificare il numero di telefono da comporre per l'attivazione della chiamata. Il numero di telefono può essere unico, oppure diverso a seconda di alcune scelte. Ad esempio, una sede centrale che ha la necessità di collegarsi con più sedi periferiche può utilizzare un solo accesso ISDN. La chiamata alla sede periferica giusta verrà effettuata in base all'indirizzo IP destinazione contenuto nel pacchetto che scatena la chiamata. In questo modo è possibile risparmiare sull'*hardware* installato, ma è necessario che il *router* abbia più numeri di telefono chiamabili e che sia in grado di discriminare tra essi in base ad un opportuno parametro (ad esempio l'indirizzo IP di destinazione del pacchetto in transito).
- Protocolli di autenticazione: spesso è necessario che la chiamata venga autenticata; è possibile ad esempio specificare un controllo sul numero di telefono del chiamante, oppure richiedere l'utilizzo di *username - password* (ad esempio mediante **PAP** o **CHAP**), o altro.

Backup

I *router Cisco* hanno due possibili soluzioni per la realizzazione del *backup* di un link:

- *backup* fisico: l'interfaccia di *backup* (ISDN) è logicamente collegata alle sorti di un'altra interfaccia fisica (ad esempio una seriale) ed entra in azione immediatamente non appena il *router* rileva una mancanza di connettività sul link primario (ad esempio la caduta della portante)
- *backup* logico: l'interfaccia di *backup* è indicata come strada possibile verso la destinazione remota al pari del collegamento primario. A differenza di questo, però, il costo di raggiungimento della destinazione è maggiore, quindi i pacchetti dati scelgono normalmente la prima strada per il raggiungimento della destinazione.

Il *backup* fisico ha il vantaggio di intervenire non appena si rileva un problema sul collegamento primario. Questo, però, può essere anche un fatto negativo, in quanto l'utente potrebbe non essere interessato a pagare l'instaurazione di una connessione anche nel momento in cui non ci sia traffico. D'altro canto questa è l'unica strada possibile nel caso di scambio dati con protocolli che non hanno un proprio livello 3 (ad esempio netbeui).

Dal punto di vista della configurazione degli apparati, il *backup* fisico comprende la configurazione dell'interfaccia primaria con la parola chiave *backup* <*interface*>, il che indica che, a fronte di un guasto su quell'interfaccia, quella di *backup* deve essere attivata immediatamente. Il *backup* logico invece funziona principalmente nell'ambito di rete IP e consiste nella definizione di una *route* statica verso la destinazione remota ad un costo superiore alla *route* standard. A differenza della *route* standard, che instrada sull'interfaccia primaria, la *route* statica secondaria instrada attraverso l'interfaccia ISDN. Il vantaggio è che la connessione viene stabilita solo quando c'è un'effettiva

necessità di scambio di dati, ma, d'altra parte, funziona con *route* IP e quindi non è in grado di attivare la connessione a fronte di traffico di altro tipo. In ogni caso è necessario che l'interfaccia primaria sia in grado di rilevare un malfunzionamento del link fisico. Questo è facilmente ottenibile nel caso in cui il link primario sia gestito attraverso un'interfaccia seriale; viceversa può essere più critico accorgersi di un link interrotto nel caso di tecnologia *Ethernet*.

Sia nel caso di trabocco che nel caso di *backup* è possibile definire il ritardo tra la caduta della linea primaria e l'attivazione della linea secondaria, e il ritardo tra la riattivazione della linea primaria e l'abbattimento della connessione di *backup*.

Trabocco

La gestione del trabocco è molto simile concettualmente a quella del *backup*. L'attivazione del trabocco richiede la definizione di due soglie di traffico, la prima che indica quando deve essere attivato il collegamento supplementare, il secondo che indica a che livello di traffico questo collegamento dovrà essere abbattuto. Il *router* misura la quantità di traffico attraverso un'opportuna media in modo da nascondere variazioni istantanee del carico.

Dialer Profiles

Il sistema operativo IOS mette a disposizione una funzionalità interessante che può essere vista come una virtualizzazione di interfacce di tipo *dial-up*. I *Dialer Profile*s, infatti, sono una nuova interfaccia virtuale che può essere utilizzata per mascherare la mancanza di un numero adeguato di interfacce fisiche. Un esempio chiarisce meglio la situazione. Si supponga un *router* con 10 collegamenti geografici. Ipotizzando di abilitare un *backup* fisico su ogni collegamento, sono necessarie 10 interfacce ISDN: il *backup* fisico richiede che sia rigidamente definita un'interfaccia come *backup* di un'altra. Tuttavia, è altamente improbabile che si guastino 10 collegamenti geografici tutti insieme: è molto probabile che un paio di interfacce ISDN siano più che sufficienti a mettere il *router* al riparo da sorprese.

L'idea di *Dialer Profile*s permette quindi di definire delle nuove interfacce *dial-up* virtuali in modo che sia possibile assegnare ad ogni interfaccia fisica un'interfaccia di *backup*. Queste, poi, condividono uno stesso insieme di interfacce fisiche da cui attingono in caso di bisogno. Le interfacce fisiche, a questo punto, non dispongono di parametri di rete propri, ma acquisiscono quelli delle interfacce virtuali nel momento in cui queste vengono mappate su quelle fisiche.

Nel momento in cui viene rilevata la necessità di attivazione di un'interfaccia *dial-up*, il *dialer* viene attivato e i suoi parametri (indirizzo IP, numero di telefono da chiamare, ...) vengono trasferiti alla prima interfaccia fisica disponibile, attivando così la chiamata. Se esiste già un collegamento remoto verso quella destinazione la chiamata non viene effettuata, con la stessa modalità già vista in precedenza per le interfacce fisiche.

Configurazione ISDN

I passi fondamentali per la configurazione di ISDN sono:

- configurazione delle interfacce;
- configurazione dei gruppi;

- configurazione delle eventuali *access-list*.

I principali comandi di interfaccia sono:

isdn switch-type basic-net3

Imposta il tipo di *switch* con cui operare (euro-ISDN). Nelle ultime versioni di IOS questo comando è diventato comando di interfaccia (prima era globale), per cui diventa possibile avere interfacce ISDN attaccate a diversi tipi di centralini con segnalazione diversa.

dialer string num

Utilizza sempre il numero num per aprire una chiamata. Questo comando (oppure in alternativa quello successivo) è sempre obbligatorio.

dialer map prot indir [name name] num

Permette di definire più numeri a cui instradare la chiamata a seconda del pacchetto che si presenta all'interfaccia, con il significato per instradare un pacchetto del protocollo prot verso il *next-hop* indir apri una connessione con il numero ISDN num. Il pacchetto deve essere di interesse per l'interfaccia; inoltre la corrispondenza tra indir e num è utilizzata anche in fase di accettazione delle chiamate (Se devo raggiungere l'indirizzo indir devo comporre il numero num ma anche: Riconosco come pacchetti validi provenienti da indir solo quelli che provengono da una connessione col numero num).

dialer-group num

Indica il tipo di filtro da applicare ai pacchetti che attraversano l'interfaccia (specifica il gruppo di accesso cui appartiene l'interfaccia); è un comando obbligatorio.

dialer idle-timeout sec

Tempo dopo il quale, se non viene rilevato traffico su quell'interfaccia, il collegamento viene disattivato. Se non specificato viene adottato quello standard (120 sec).

isdn answer1 num, isdn answer2 num

Indica che può accettare chiamate provenienti dal numero ISDN num; è usato per il controllo accessi. È possibile indicare fino a due numeri chiamanti (oppure num:subaddr). È possibile impostare *wilcards* (isdn *answer1* 345:2x).

isdn caller num

Come il precedente, ma lo *screening* viene fatto sul valore dell'ISDN caller ID. Se ne possono impostare fino a 64 per ogni interfaccia, ed accetta *wilcard* (isdn *caller* 345xx).

Esempio di configurazione

- *configure terminal*: Entra in modalità configurazione
- *interface* BRI0: Entra in configurazione interfaccia
- *isdn switch-type basic-net3*: Definisce il tipo di ISDN utilizzato
- ip *address* 10.0.0.33 255.255.255.240: Definisce l'indirizzo IP

- *encapsulation* ppp: Definisce l'*encapsulation* di livello 2
- *dialer map* ip 10.0.0.34 5178046: Abbina il numero telefonico da chiamare con l'indirizzo IP remoto
- *dialer-group* 1: Assegna l'interfaccia al gruppo numero 1
- *exit*: Esce dalla configurazione dell'interfaccia
- ip *route* 10.0.0.0 255.255.255.0 10.0.0.34: Configura le *route* statiche necessarie per raggiungere il resto della rete attraverso ISDN

route statiche necessarie per raggiungere il resto della rete attraverso ISDN.

- *dialer-list* 1 *protocol* ip *list* 101
  Dialer-list: definisce come interessanti tutti i pacchetti IP, quindi associa il *dialer-group* 1 alla *access-list* 101 per un miglior raffinamento della politica di accesso
- *access-list* 101 *permit* ip *any any*
  Filtro (numero 101) da applicare ai pacchetti. In questo caso, una *access-list* così semplice è superflua in quanto sarebbe bastato il semplice comando *dialer-list*. Una *access list* migliore potrebbe essere: *access-list* 101 *permit* ip 126.0.0.0 0.255.255.255 128.16.64.0 0.0.0.255, in cui si accettano in ingresso i pacchetti della rete 126.x diretti verso la 128.16.64.x
- *end*
  Termina la configurazione corrente

Accesso ATM tramite interfaccia Seriale 1

Questo paragrafo descrive come configurare *router* che utilizzano una interfaccia seriale per accesso **ATM** attraverso una ***A****TM* ***D****ata* ***S****ervice* ***U****nit* (**ADSU**). Si procederà con:

- abilitare l'incapsulamento *Asynchronous Transfer Mode-Data Exchange Interface* (ATM-DXI),
- selezionare un metodo di incapsulamento multiprotocollo utilizzando ATM-DXI,
- impostare un PVC per l'incapsulamento selezionato.

Accesso ATM tramite interfaccia Seriale

Nei *router* con interfaccia seriale, un ADSU è necessario per fornire l'interfaccia ATM alla rete, convertire i pacchetti uscenti in celle ATM, e riassemblare le celle ATM entranti in pacchetti.

Ogni interfaccia seriale può essere configurata per l'incapsulazione multiprotocollo su ATM-DXI, come specificato nella RFC 1483. Al ADSU, l'intestazione DXI viene eliminata, e i dati del protocollo vengono segmentati in celle per il trasporto sulla rete ATM.

La RFC 1483 descrive due metodi di trasporto traffico di interconnessione di reti di tipo *connectionless* su una rete ATM. Un metodo consente il *multiplexing* di più protocolli su di un singolo PVC. L'altro metodo utilizza differenti circuiti virtuali per trasportare protocolli differenti. L'implementazione della RFC 1483 da supporto a tutti e due i metodi consentendo il trasporto di *Apollo Domain*, *AppleTalk*, *Banyan VINES*, DECnet, IP, *Novell* IPX, ISO CLNS, e XNS.

Task List per la configurazione di accesso seriale ATM

La configurazione di accesso ATM su interfaccia seriale, si completa nei seguenti passaggi.

- Abilitare la *Serial Interface*
- Abilitare l'incapsulazione ATM-DXI
- Impostare il PVC ATM-DXI
- Mappare gli indirizzi di protocollo presso il PVC ATM-DXI
- Controllo (opzionale)

Abilitare la Serial Interface

Per iniziare a configurare l'interfaccia seriale per l'accesso ATM, occorre abilitare l'interfaccia seriale, eseguendo i seguenti passi a partire dal modalità *global* di configurazione.

Il primo comando è il seguente.

interface serial number

Per ogni protocollo trasportato, si assegni un indirizzo di protocollo per interfaccia.

appletalk address network.node
ip address address mask
ipx network number

I protocolli supportati sono *Apollo Domain*, *AppleTalk*, *Banyan VINES*, DECnet, IP, *Novell* IPX, ISO CLNS, e XNS.

Accesso ATM tramite interfaccia Seriale 2

Abilitare l'incapsulazione ATM-DXI

Per abilitare l'incapsulamento ATM-DXI su una seriale ***H**igh-**S**peed **S**erial **I**nterface* (**HSSI**), eseguire il seguente comando in *interface configuration mode*:

Comando

encapsulation atm-dxi

Impostare il PVC ATM-DXI

Un ATM-DXI PVC può essere definito quale elemento in grado di supportare uno o più protocolli, come descritto in RFC 1483 (singolo protocollo) e in RFC 1490 (multiprotocollo)

Per impostare ATM-DXI PVC e selezionare un metodo incapsulato, occorre eseguire il seguente comando impostabile nel *configuration mode*:

dxi pvc vpi vci [snap | nlpid | mux]

L'opzione MUX (*multiplex*) definisce il PVC per supportare un solo protocollo; ogni protocollo deve essere trasportato su un differente PVC. L'opzione **SNAP** (*SubNetwork*

Access Protocol) riguarda l'incapsulazione multiprotocollo LLC/SNAP, compatibile con RFC 1483; SNAP è l'opzione corrente di *default*. L'opzione *Network Layer Protocol IDentification* (**NLPID**) riguarda l'incapsulazione multiprotocollo, compatibile con RFC 1490; questa opzione è fornita per la compatibilità verso il basso con le impostazioni di *default* della precedente versione nel *software Cisco* IOS.

Mappare gli indirizzi di protocollo presso il PVC ATM-DXI

Questa sezione descrive come mappare gli indirizzi di protocollo all'identificatore di canale virtuale (*virtual channel identifier*, VCI) e all'identificatore di cammino virtuale (*virtual path identifier*, VPI) di un PVC che supporta traffico multiprotocollo. Il protocollo indirizza tutti i sistemi *host* fino all'altro capo del collegamento. Per mappare un indirizzo di protocollo a un ATM-DXI PVC, occorre eseguire il seguente comando nella interfaccia *configuration mode*.

dxi *map protocol protocol-address* vpi vci [*broadcast*]

Occorre ripetere questo comando per ogni protocollo che dovrà essere supportato in PVCI protocolli supportati sono *Apollo Domain*, *AppleTalk*, *Banyan VINES*, DECnet, IP, *Novell* IPX, ISO CLNS, e XNS.

Controllo

Dopo aver configurato l'interfaccia seriale per ATM, si può visualizzare lo stato dell'interfaccia, ATM-DXI PVC, o la mappa ATM-DXI. Per mostrare l'interfaccia, PVC, o le informazioni sulla mappa, occorre eseguire i seguenti comandi in EXEC *mode*.

show interfaces atm [*slot/port*]
show dxi pvc
show dxi *map*

Esempio

Il seguente esempio mostra come configurare un'interfaccia seriale per l'accesso ATM.

Nell'esempio, l'interfaccia seriale 0 è configurata per ATM-DXI con incapsulazione MUX. Poiché viene utilizzato l'incapsulamento MUX, è supportato solo un protocollo in PVC. Questo protocollo è identificato in modo esplicito da un comando di mappatura DXI, che identifica altresì l'indirizzo di protocollo del nodo remoto. Questo PVC può supportare il traffico IP di *broadcast*.

interface serial 0
ip *address* 172.21.178.48
encapsulation atm-dxi
dxi pvc 10 10 mux
dxi *map* ip 172.21.178.4 10 10 *broadcast*

Incapsulamento Frame relay

Per incapsulare in *Frame relay* a livello di interfaccia, si devono eseguire i seguenti comando in *global configuration mode*:

Comando (*configuration mode*):

interface type number
encapsulation frame-relay [ietf]

Frame relay supporta Incapsulamento di tutti i protocolli in conformità con la RFC 1490, consentendo interoperabilità tra produttori diversi.

Occorre usare la forma Internet *Engineering Task Force* (IETF) dell'incapsulazione *Frame relay* se il *router* o l'accesso del *server* è connesso a un diverso sistema tramite una rete *Frame Relay*.

Si raccomanda di 'spegnere' l'interfaccia precedente prima di cambiare il tipo di incapsulamento. Sebbene ciò non sia richiesto, porre in *shut down* l'interfaccia, assicura che la stessa sia resettata per il nuovo incapsulamento.

Esempio

Di seguito viene impostato l'incapsulamento IETF a livello di interfaccia. Il secondo esempio imposta l'incapsulamento IETF su basi per-DLCI.

encapsulation frame-relay IETF
frame-relay map ip 131.108.123.2 48 *broadcast*
frame-relay map ip 131.108.123.3 49 *broadcast*

encapsulation frame-relay
frame-relay map ip 131.108.123.2 48 *broadcast* ietf
frame-relay map ip 131.108.123.3 49 *broadcast* ietf

Impostazione del routing statico: comandi generali

ip *routing*

Abilita il *router* ad instradare pacchetti IP (processo di *forwarding*). Questo comando: è utile anche in forma negata (no ip *routing*) per cancellare completamente la precedente configurazione di *routing* e lasciare il *router* spoglio. A questo punto è possibile riabilitare il *routing* e procedere alla nuova configurazione.

ip *classless*

Nel momento in cui il *router* riceve un pacchetto per cui non ha una *route* specifica (e nemmeno la *default route*), usa la migliore supernet *route* possibile.

route statiche e di defaul

È il modo più semplice per abilitare il *routing*; non è tuttavia molto robusto in quanto tutto deve essere fatto manualmente e quindi sono estremamente frequenti gli errori (oltre alla mancanza di aggiornamento automatico da parte della rete).

ip *route* indirizzo maschera *router* [distanza]

I pacchetti destinati alle reti comprese nel *range* (indirizzo, maschera) devono essere instradati verso *router*, che deve essere (1) in una sottorete direttamente collegata a una delle interfacce, oppure (2) una porta del *router* corrente nel caso in cui l'interfaccia sia *unnumbered*. La *route* può essere sostituita da una appresa dinamicamente e

avente distanza inferiore.

ip *default-network* indirizzo

Configura una *route* di *default*. È immessa da uno *smart router* il quale normalmente conosce le *route* per qualsiasi destinazione e diffonde la *route* di *default* tramite i protocolli di *routing*. La modalità con cui questa è propagata dipende dal protocollo di *routing*: RIP annuncia 0.0.0.0 0.0.0.0, IGRP annuncia indirizzo indicandola come *route* esterna e candidata per la *route* di *default*.

Configurazione dei protocolli di routing dinamico: Comandi comuni ai protocolli

Il *routing* dinamico necessita di comandi specifici per i diversi protocolli di instradamento, ma alcuni comandi sono comuni ai vari protocolli:

router proto [ID]

Abilita il protocollo di *routing* specificato; entra in modalità di configurazione di tale protocollo; ha modalità leggermente diverse per ogni protocollo

network indirizzo_di_rete

Specifica contemporaneamente due informazioni:

- le reti (direttamente connesse al *router*) che sono nel dominio di *routing* in esame (e che verranno annunciate dal protocollo);
- le interfacce che dovranno partecipare a quel dominio di *routing* (il *router* automaticamente capisce quali sono le sue interfacce interessate dal dominio, ed abilita l'invio e le ricezione di messaggi di *updates* attraverso quelle interfacce)

La sintassi del comando è leggermente diversa in OSPF e in BGP.

passive-interface interfaccia

Inibisce l'invio di messaggi di *update* sull'interfaccia (che, ad esempio, è al bordo del dominio di *routing*). Può essere una ragione amministrativa (evitare di propagare messaggi in una specifica direzione) oppure economica (impedire la generazione di messaggi di *routing* su linee commutate quali ISDN). Questa interfaccia è comunque in grado di accettare e processare *routing update* che arrivano ad essa (inibisce l'invio ma non la ricezione)

neighbor indirizzo

Indica al *router* di inviare i messaggi all'indirizzo indirizzo specificato; è usato su reti senza capacità *broadcast* oppure per prevenire l'invio dei messaggi di aggiornamento a specifici *router* (ad esempio su LAN in congiunzione al comando *passive-interface*, per abilitare solo specifici *neighbors*, per ragioni di *policy*)

Configurazione dei protocolli di routing dinamico: Comandi specifici per i protocolli 1

Comandi specifici per RIP

router rip

Abilita il protocollo di *routing* RIP. Dal momento che questo comando non ha il parametro ID, non possono coesistere più istanze di RIP sulla stessa macchina

version 1 | 2

Abilita l'invio di messaggi secondo la versione 1 o 2 (*default* 1); nella ricezione capisce ambedue le versioni

Comandi specifici per IGRP-EIGR

router igrp *process_id - router* eigrp *process_id*

Attiva il processo di *routing*. Il *process_id* identifica il particolare processo di *routing* in esecuzione, che deve essere uguale in tutti i *router* del dominio IGRP/EIGRP in quanto l'informazione viene inclusa negli annunci. Se si è in un AS registrato è buona norma porre questo identificativo pari al numero dell'AS; nel caso si voglia impiegare contemporaneamente IGRP e EIGRP (per esempio per necessità di transizione), IGRP e EIGRP possono scambiarsi informazioni solo se *process_id* = AS

metric weights tos k1 k2 k3 k4 k5

Cambia il valore dei parametri utilizzati per il calcolo del costo per uno specifico codice *Tipe Of Service* (anche se è fortemente sconsigliato cambiarli); il significato dei termini è analogo al comando *default-metric*. I valori di *default* sono tos: 0, k1=k3= 1, k2=k4=k5= 0

no metric holddown (solo IGRP)

Disabilita l'algoritmo di *hold down* di IGRP, migliorando il tempo di convergenza a scapito di possibilità di *loop*.

Configurazione dei protocolli di routing dinamico: Comandi specifici per i protocolli 2

Comandi specifici per OSPF

router ospf *process_id*

Abilita un processo di *routing* OSPF. Il *process_id* identifica il processo di *routing* OSPF all'interno del *router* ed ha significato locale (contrariamente a IGRP/EIGRP non viene trasmesso all'esterno del *router*)

network indirizzo *wildcard* area id_area

Il protocollo OSPF prevede che le reti da annunciare vengano indicate esplicitamente con la coppia <indirizzo,*wildcard*>. Queste informazioni individuano una o più interfacce che si trovano nell'area id_area sulle quali vengono inviati e ricevuti i messaggi OSPF. La maschera è di tipo *wildcard* (come le *access list*), mentre id_area è codificato su 4 byte, ed è possibile utilizzare sia la notazione decimale che quella

decimale puntata

area id_area stub

Dichiara l'area id_area una stub area

area id_area *range* indirizzo maschera

Specifica un *address range* da annunciare all'esterno dell'area id_area, consentendo l'aggregazione di informazioni per la propagazione all'esterno dell'area id_area (se all'interno dell'area c'è almeno un'interfaccia con l'indirizzo che cade all'interno dell'*address range*, all'esterno è annunciato l'*address range* invece dei singoli indirizzi)

area id_area virtual-link ID_*router*

Crea un link virtuale con il *router* che ha ID_*router*, dove questo valore è individuabile visualizzando i *database* di OSPF; l'area id_area è comune ai due *router*

default-information originate [aways]

Abilita il *router* di annunciare una *route* di *default* all'interno del suo dominio OSPF, comportandosi da AS *Boundary Router* (lo stesso scopo può essere ottenuto con il *redistribute*). Un AS *boundary router* non annuncia necessariamente la *route* di *default*, in quanto potrebbe annunciare più *route* esterne imparate da altri protocolli. Se il *router* non ha alcuna *route* di *default*, questo comando è ininfluente, a meno che si specifichi la *keyword always*, nel qual caso viene comunque sempre immessa una *route* di *default* in quel dominio anche se il *router* non ne ha una propria

Comandi specifici per BGP

La configurazione di BGP è più complessa degli altri protocolli di *routing* e comprende i seguenti passi fondamentali:

abilitazione del *routing* BGP

configurazione dei BGP *peers*

router bgp AS

Attiva il processo di *routing* BGP nell'*Autonomous System* AS

network indirizzo [*mask netmask*]

Identifica questa rete come appartenente al dominio BGP locale e la inserisce nella propria *routing table*. Il significato è diverso dai protocolli IGP in quanto il comando *network* non definisce le interfacce sulle quali bisogna inviare gli annunci. Contrariamente ad OSPF, la *netmask* è nella forma classica

neighbor indirizzo *remote-as* AS

Dichiara come *peer (neighbor)* il *router* indirizzo dell'*Autonomous System* AS. I *neighbor* possono essere *Internal* o *External*. I *peer* di tipo *external* sono contraddistinti da un link fisico in comune, mentre quelli di tipo *internal* sono posizionati in un qualunque locazione dell'AS

aggregate-address indirizzo maschera

Se esiste almeno una *route* per una rete che rientra nel range di indirizzi (indirizzo, maschera) BGP annuncia questo range

default-information originate

Abilita la propagazione della *route* di *default* (0.0.0.0) all'interno del dominio BGP. La generazione di questa *route* non è fatta in automatico dal BGP, ma deve essere appresa da altre parti (ad esempio mediante redistribuzione).

Configurazione dei protocolli di routing dinamico: Comandi specifici per i protocolli 3

Redistribuzione

È quel processo che permette di collegare due domini di *routing* diversi scambiandosi vicendevolmente le *route* apprese in ognuno di essi. A differenza di avere un dominio unico, il processo di redistribuzione provoca un compattamento delle informazioni di *routing* in modo da rendere il processo più scalabile. In altre parole un dominio di *routing* non conoscerà completamente la topologia dell'altro dominio, ma solo delle informazioni tendenti a dire quali reti sono presenti, ma non qual è il percorso esatto che i pacchetti faranno per raggiungerle. La redistribuzione può essere uni o bi-direzionale.

Uno dei problemi di questo meccanismo è che ogni protocollo di *routing* ha un meccanismo di computo della metrica diverso dagli altri. È allora necessario dire esplicitamente al *router* la metrica con la quale gli annunci dell'altro dominio dovranno essere propagati, con il comando *default-metric*.
I principali comandi connessi alla redistribuzione sono:

redistribute protocollo [id]

Distribuisce nel dominio del *router* in questione le informazioni raccolte tramite il protocollo protocollo; è un sottocomando della modalità *router*. Il valore ID è necessario per discriminare tra più processi dello stesso protocollo (es. EIGRP).

default-metric metrics

Comando abbinato al *redistribute*, indicando che tutte le *route* apprese dall'esterno sono da ridistribuire con metrica *metrics*; ha modalità leggermente diverse per ogni protocollo.

default-metric k1 k2 k3 k4 k5 (IGRP - EIGRP)

Comando abbinato al *redistribute*, indicando che tutte le *route* apprese dall'esterno sono da redistribuire con metrica indicata. Ad esempio il comando *default*-metric 10000 100 255 1 1500 corrisponde ai termini: banda, ritardo, affidabilità, carico, MTU.

redistribute static

Redistribuisce all'interno del protocollo di *routing* in esame tutte le sue *route* statiche.

redistribute connected

Ridistribuisce le *route* che vengono create automaticamente per il fatto di avere una interfaccia in esse. Le *route* interessate da questo comando sono quelle non specificate da un esplicito comando *network*; per OSPF e IS-IS queste *route* sono ridistribuite come appartenenti all'esterno dell'AS.

Comandi di controllo

Questi comandi, contrariamente a quelli precedenti, sono attivabili dalla modalità privilegiata.

show ip route

Mostra la *routing table* del protocollo IP

clear ip route {*network [mask*] | *}

Va eseguito in modalità privilegiata, permette la cancellazione di una o più *route* che si suppongono non più valide. Questo comando non permette la cancellazione delle *route* statiche

show ip proto

Visualizza lo stato di ogni protocollo di *routing* attivo (tempistiche, parametri (es per EIGRP), redistribuzioni, ...)

show ip eigrp *interfaces* | *neighbors* | *topology* | *traffic*

Comandi di controllo del funzionamento del processo EIGRP

show ip ospf | bgp

Visualizza le informazioni generali sul processo in esame

show ip ospf *database*

Mostra il *database* dei link *state advertisement* ricevuti

sh ip ospf *neighbor*

Visualizza tutti i *router* OSPF adiacenti, indipendentemente dall'area a cui appartengono; il campo *neighbor*_ID mostra l'identificativo (*router*_ID) del *router* remoto. Nel caso in cui il *router* sia su una *Ethernet*, mostra anche chi è il *Designated Router* e il *Backup* DR, il loro indirizzo su quella rete e l'indirizzo attraverso il quale sono raggiungibili

sh ip ospf *border-routers* | *interface* | *virtual-links*

Altri comandi per la visualizzazione di aspetti specifici di OSPF

Lettura del database OSPF

OSPF offre il grande vantaggio di visualizzare un'ottima descrizione della rete. Il comando sh ip ospf *database* può però essere ostico nella sua interpretazione. Per la lettura dei risultati è allora necessario ricordare che:

- *Router Link State*: rappresenta l'elenco dei *router* presenti sull'area in esame.
- *Network Link State*: rappresenta l'elenco delle reti *broadcast* contenute nell'area in esame.
- *Summary Net Link State*: rappresenta l'elenco delle reti presenti nelle altre aree; sparisce qui la distinzione tra reti di transito e reti tradizionali.
  Ogni *entry* (del tipo specificato sopra) è composta da due informazioni, il LinkID e l'*Advertising Router*. Il significato di questo campi è variabile a seconda del tipo di *entry* ed è schematizzato in figura. Nel caso di un *Router* Link, il LinkID e l'*ADVRouter* coincidono (perché è il *router* che annuncia sé stesso).

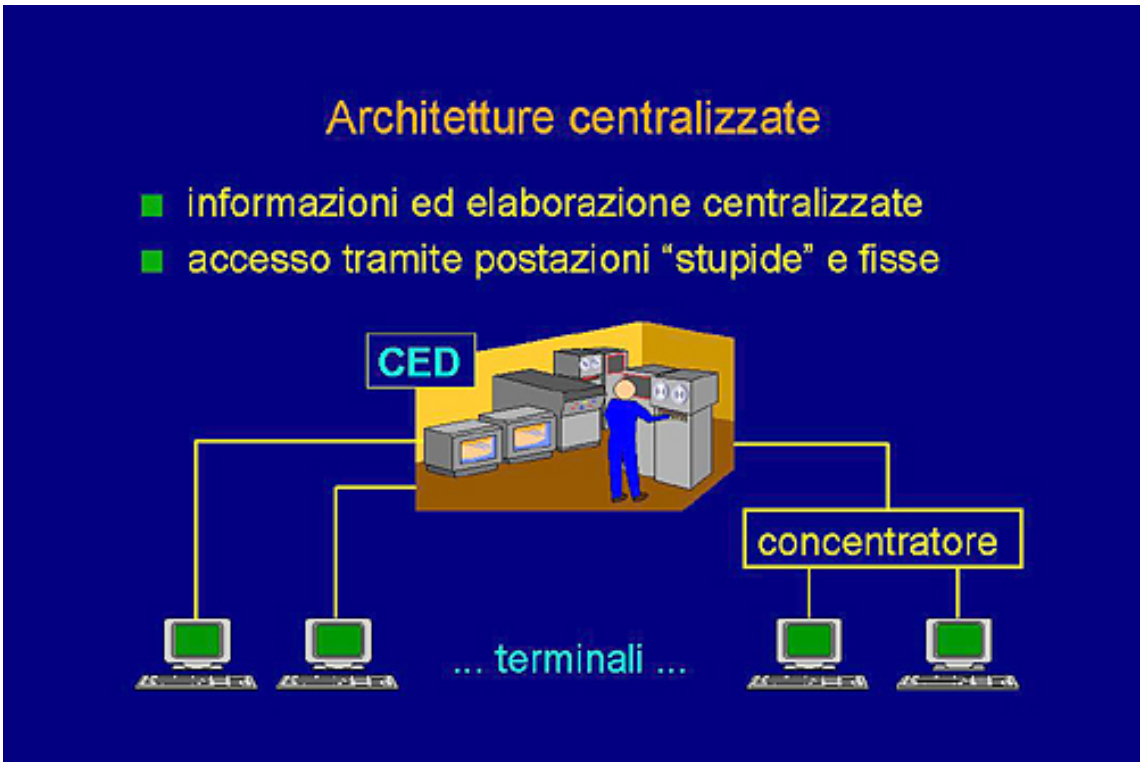
Il RouterID è solitamente dato dal più alto indirizzo configurato sulle sue interfacce, tranne nel caso in cui sia stato configurato l'indirizzo di *Loopback* che diventa automaticamente il nuovo *RouterID*. Il *database* deve essere ovviamente uguale all'interno dei vari *router* appartenenti alla stessa area. Ogni *router* può comunque avere uno o più *database*, a seconda che sia un *internal router* oppure un *router* di bordo tra più aree.

La sicurezza nelle reti

Franco Callegati
Paolo Zaffoni

8.4.1 (Spiegare i principali aspetti della sicurezza connessi alla trasmissione dei dati)

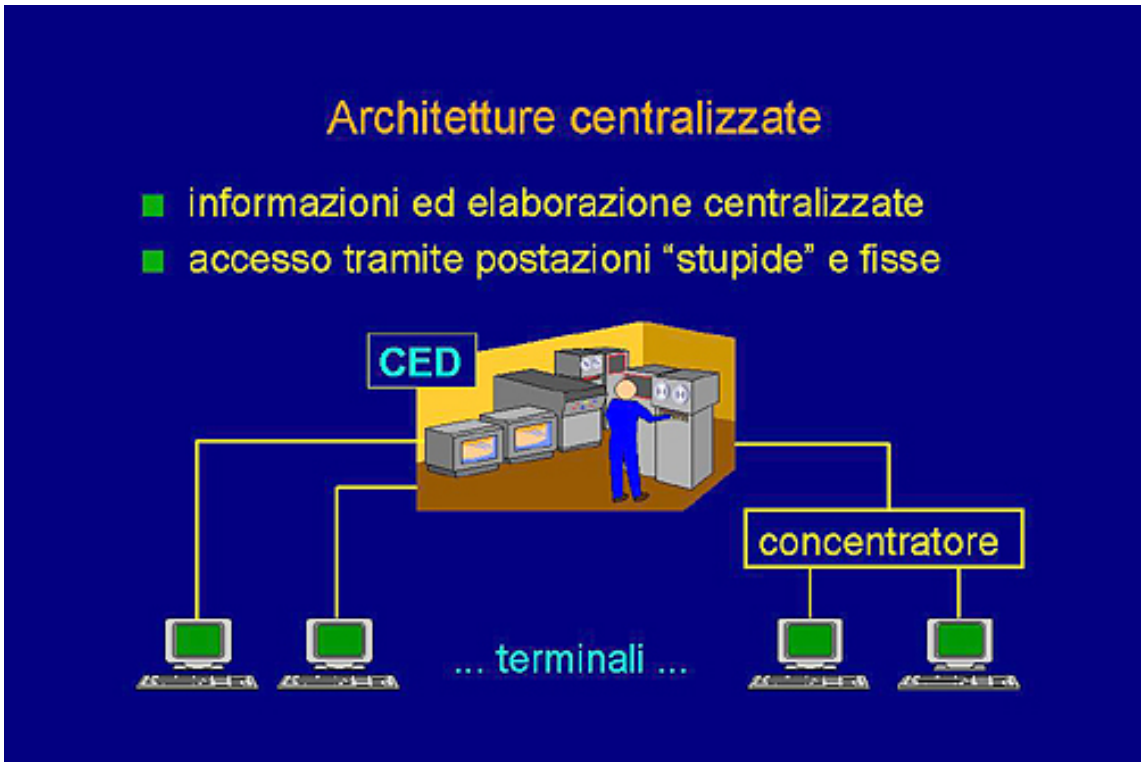
Architetture centralizzate (1)



Architetture centralizzate (1)

Benvenuti al modulo di sicurezza dei sistemi distribuiti. Io sono il Professor Antonio Lioy del Politecnico di Torino e mi occupo, come tema di ricerca ed anche di insegnamento, di sicurezza delle reti e dei sistemi informativi. In questa prima parte, noi tratteremo specificamente delle problematiche generali di sicurezza negli attuali sistemi informatici. In particolare, ci sono due grandi categorie di sistemi informatici che vengono utilizzati ampiamente da tanti enti e da tante organizzazioni. La prima architettura che andiamo a considerare è l'architettura centralizzata. In questo tipo di architetture tipicamente si ha un Centro di Elaborazione Dati (CED), che normalmente viene mantenuto in un luogo sotterraneo o protetto fisicamente, ad esempio, da porte blindate e accessi controllati, dentro cui sono conservate tutte le unità di elaborazione e tutte le unità di memorizzazione dei dati. Gli utenti accedono e sfruttano queste risorse informatiche tramite dei terminali, cioè dispositivi di *input* e *output* nei confronti del sistema di elaborazione; quindi una tastiera ed un video, non un vero e proprio *Personal Computer* o una *work-station*.

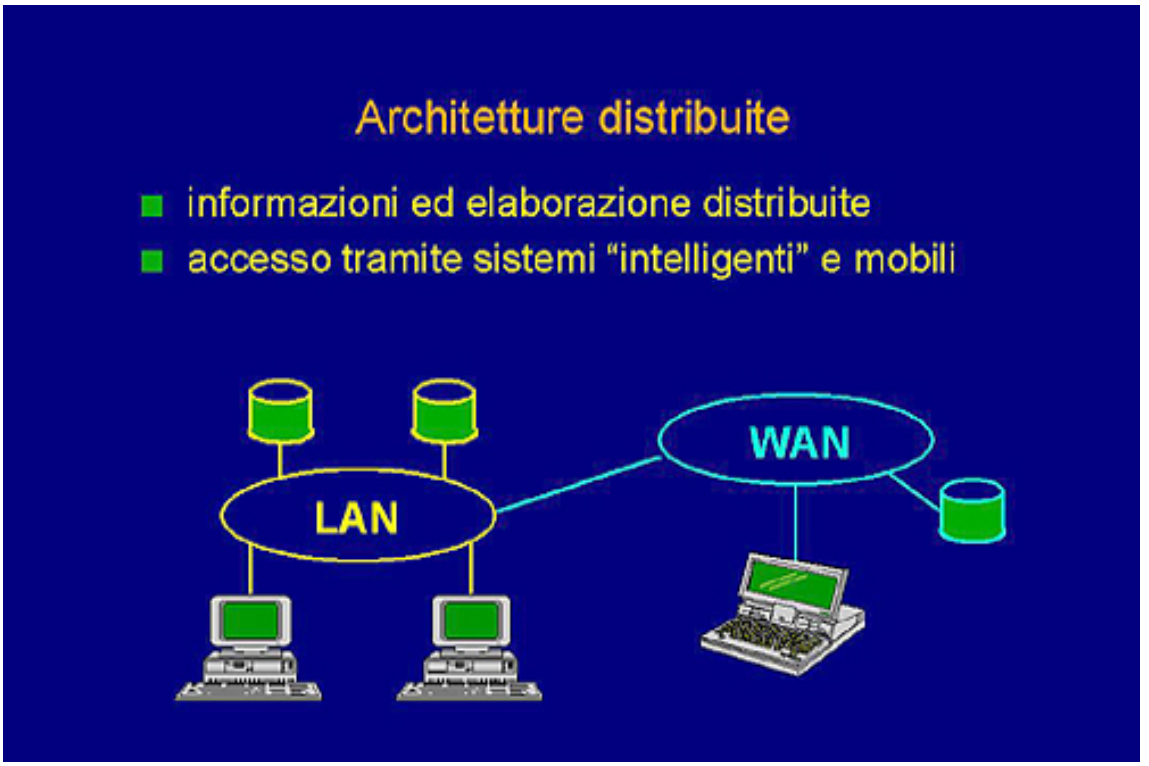
Architetture centralizzate (2)



Architetture centralizzate (2)

La sicurezza di un sistema di questo genere è in gran parte affidata alla sicurezza fisica del sistema complessivo; infatti l'accesso alle strutture di calcolo e di memorizzazione dei dati è protetto fisicamente dal locale dentro cui si trovano e gli utenti non possono in alcun modo svolgere degli attacchi attivi. Anche il tipo di collegamento che viene fatto tra la postazione di lavoro dell'utente e il sistema di elaborazione è, bene o male, fisicamente protetto, perché si tratta di un cavo che corre dalla postazione di lavoro fino al sotterraneo dove sono testati i sistemi di elaborazione. Anche nel caso in cui ci sia una sede remota, quindi non sia possibile far correre direttamente un cavo dal terminale al sistema di elaborazione, anche questa architettura è protetta fisicamente, perché tutti i terminali vengono portati, con un collegamento, su un concentratore e da questo c'è poi una linea dedicata che va al sistema di elaborazione. Riassumendo, nelle architetture di tipo centralizzato la sicurezza è di tipo fisico o, al massimo, del sistema operativo del sistema di elaborazione e gli utenti hanno a disposizione dei meri dispositivi di *input*/*output*.

Architetture distribuite



Architetture distribuite

Molto diversa è la situazione nel caso delle architetture più recenti, le cosiddette architetture distribuite. In queste architetture gran parte dei presupposti che abbiamo appena enunciato non sono più validi. Innanzitutto non è più vero che tutti i sistemi di elaborazione e di memorizzazione sono conservati in un unico locale, anzi tipicamente, come si vede qui illustrato, i sistemi sono sparsi, sia all'interno di una rete locale (LAN) ma anche su una rete geografica (*WAN*), diventa quindi difficile controllare fisicamente tutti quanti i sistemi. Inoltre, le postazioni di lavoro dell'utente non sono più delle semplici postazioni di *input*/*output*, ma sono delle postazioni di lavoro complesse, sono tipicamente dei *Personal Computer*, ossia hanno una capacità di elaborazione autonoma ed indipendente dai sistemi cui ci si collega. Questo vuol dire che l'utente del nostro sistema di elaborazione non può soltanto svolgere operazioni di *input*/*output*, ma ha anche la possibilità di attivare localmente, presso la sua stazione di lavoro, dei programmi. Questi potrebbero essere dei programmi di attacco, cioè che inficiano la sicurezza del sistema complessivo. La protezione fisica di tutti questi elementi di elaborazione non è, ovviamente, possibile ed inoltre c'è il problema dei *computer* mobili: i *computer* portatili si stanno diffondendo sempre di più e quindi diventa molto difficile riuscire ad offrire una protezione fisica, dato che questi sistemi si collegano e si scollegano alla rete in vari punti, sia all'interno della nostra rete locale, sia, nel caso di personale che stia viaggiando, da vari punti dell'Italia, se non addirittura dell'intero pianeta, tramite Internet. È quindi chiaro che, per architetture di tipo distribuito, bisogna ripensare completamente alla sicurezza, in modo tale che sia indipendente dalla locazione fisica dei sistemi di elaborazione e sia indipendente anche dalle reti a cui ci colleghiamo. È quindi sempre meno ipotizzabile il fatto che un certo nodo di elaborazione sia sempre collegato al medesimo tipo di rete.

Sicurezza: dove è il nemico? (1)

Sicurezza: dove e' il nemico? (1)

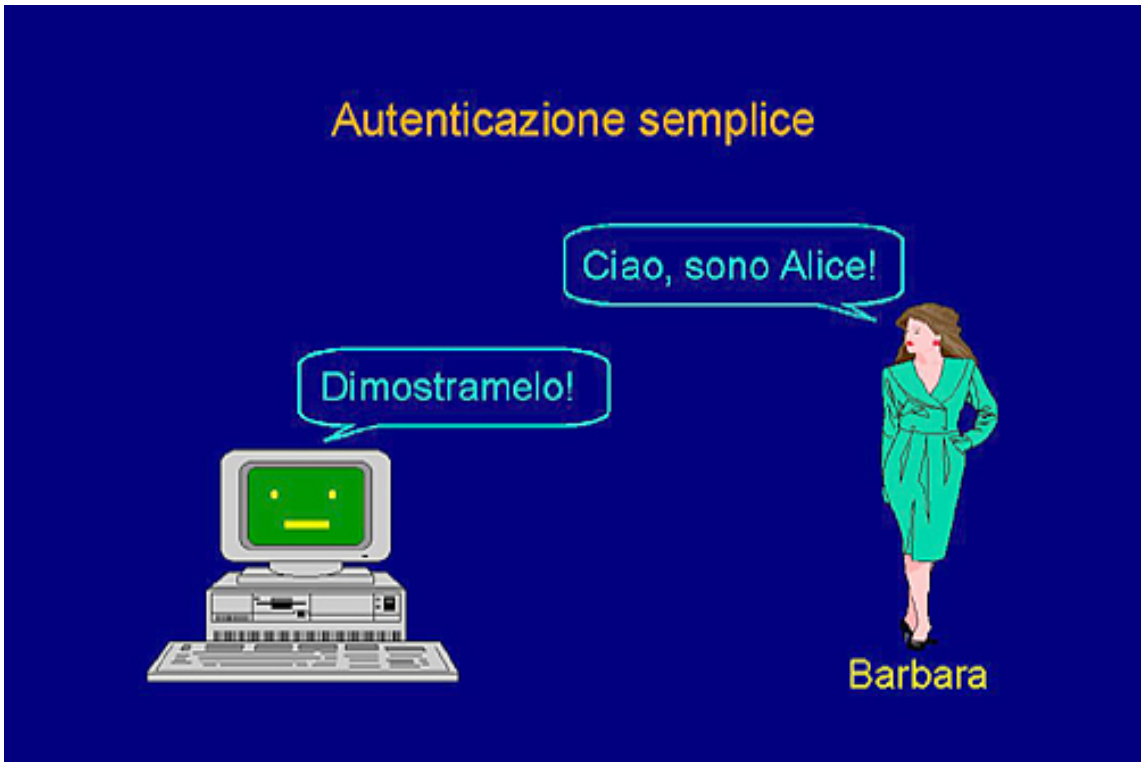
Se dobbiamo mettere in piedi un sistema di sicurezza, una delle cose più importanti è identificare dove si trova il nemico da cui vogliamo difenderci. In generale possiamo fare tre ipotesi: possiamo supporre che noi siamo i buoni ed i cattivi stiano fuori dalla nostra organizzazione. In questa ipotesi ciò che bisogna fare è cercare di proteggere la nostra rete dalle intrusioni che potrebbero provenire dall'esterno. È questo il caso tipico in cui trovano applicazione i cosiddetti *firewall*, letteralmente muri taglia-fuoco, porte antincendio, cioè evitano che l'incendio, la pericolosità insita nella rete esterna, si propaghi anche nella nostra rete interna. Purtroppo questa ipotesi, che i cattivi stiano fuori dalla nostra organizzazione, trova sempre meno corrispondenza nella realtà: le conoscenze informatiche si stanno diffondendo a tutti i livelli, ma soprattutto si stanno diffondendo sempre di più dei semplicissimi programmini di attacco che possono essere utilizzati anche da persone non esperte. Bisogna quindi considerare, almeno a livello teorico, ma molto spesso anche pratico, l'ipotesi che all'interno della nostra organizzazione ci sia qualcuno che, per vari motivi che si possono facilmente immaginare, possa volere attaccare il sistema dall'interno. Quindi, la messa in opera di un sistema di tipo *firewall* non è più sufficiente, bisogna riuscire a fare protezione della rete locale o della Intranet.

Sicurezza: dove è il nemico? (2)

Sicurezza: dove e' il nemico? (2)

Questo ci causa un problema, perché sia la rete locale sia la Intranet tipicamente sono costruite con dei sistemi che sono stati ideati per facilitare la collaborazione fra le persone ed invece la sicurezza tende ad impedire, o quantomeno a limitare, la collaborazione. Inoltre, nel caso che il nostro personale sia in viaggio, in trasferta, all'esterno della nostra rete, non si riesce più a farlo interoperare correttamente con tutti i sistemi. Da queste considerazioni deriva la conseguenza che l'unica soluzione applicabile, in generale, in maniera completamente indipendente dalla rete a cui il nostro personale è collegato, è la protezione delle applicazioni: siccome le applicazioni sono a livello più elevato dello *stack* di rete, sono le uniche ad essere completamente indipendenti dalla rete sottostante. Se noi riusciamo a proteggere le nostre applicazioni, abbiamo reso il sistema sicuro, indipendentemente dalle reti che attraversiamo e dalle postazioni di lavoro dove operiamo.

Autenticazione semplice



Autenticazione semplice

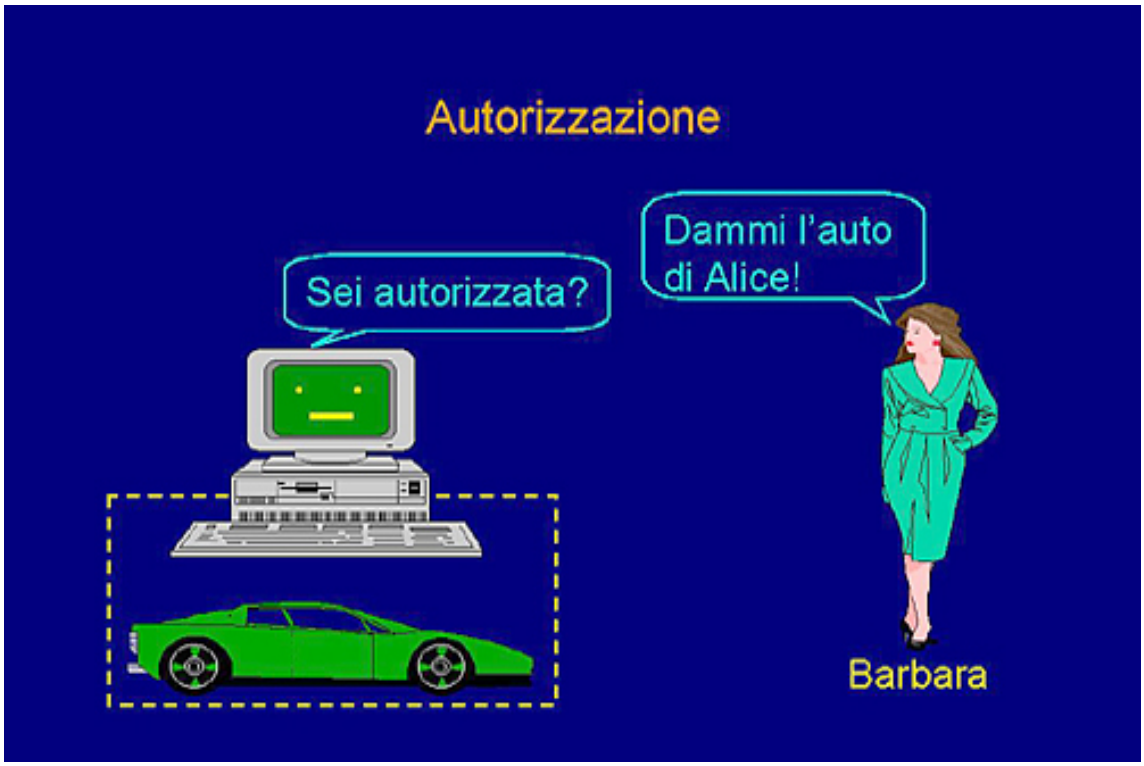
Introduciamo la terminologia di base della sicurezza informatica, il cosiddetto gergo di sicurezza, che useremo esplicitamente in questa lezione e nelle prossime. Si parla di autenticazione, ed in particolare di autenticazione semplice, ogniqualvolta un utente desidera accedere ad un sistema di elaborazione. In questo caso, noi abbiamo questa persona che, nonostante rivesta l'identità di Barbara, pretende di essere una persona diversa, pretende di essere Alice. È chiaro che un corretto sistema di autenticazione non deve credere ciecamente a quello che l'utente dice di essere, ma deve chiedere una prova formale. Per ottenere questa prova formale si useranno dei sistemi di autenticazione precisi, esatti e soprattutto non falsificabili facilmente. Questo è il tipico modo di autenticazione a cui siamo abituati: normalmente infatti quando ci presentiamo ad un sistema di elaborazione ci viene chiesto *Username* e *Password*; vedremo in seguito che questo tipico modo di autenticazione è fortemente sconsigliato, perché altamente insicuro.

Mutua autenticazione

Mutua autenticazione

Si parla anche di mutua-autenticazione, che corrisponde alla situazione mostrata in questa *slide*: non soltanto l'utente deve presentarsi nei confronti del sistema, ma l'utente potrebbe avere un ragionevole dubbio che il sistema cui lui sta facendo il collegamento, non sia quello a cui desidererebbe collegarsi. Questo perché è possibile, abbastanza facilmente, mettere in piedi una rete di calcolatori, i cosiddetti *server*-ombra, o anche *server*-fantasmi, *shadow-server*, i quali con tecniche opportune, che illustreremo in seguito, mostrano un'interfaccia simile a quella del sistema originale e quindi sarebbero in grado di fornirci dei dati sbagliati, mentre noi in assoluta buona fede crediamo di esserci collegati al sistema giusto. Questa è una caratteristica che manca normalmente nei correnti sistemi, sia di tipo centralizzato sia di tipo distribuito.

Autorizzazione



Autorizzazione

Una volta che abbiamo fatto l'autenticazione degli utenti e dei sistemi a cui si stanno collegando, bisogna decidere se questi utenti, o questi sistemi, hanno diritto a svolgere certe operazioni. In gergo informatico si parla quindi di autorizzazione per decidere se, in base ai dati di autenticazione che sono stati forniti, è lecito ottenere il controllo di un certo oggetto o attivare una certa procedura di elaborazione. In questo caso la signorina chiede di poter aprire il box elettronico, controllato dal *computer*, per prelevare questa bella automobile; il *computer* ha il ragionevole dubbio che non basta la sua identità o la sua parola per far prendere questa bella auto e farsi un giro.

Riservatezza

Virus invisibili

Un'altra proprietà di sicurezza rilevante nei sistemi informativi è la riservatezza. Con la riservatezza si intende il fatto che una comunicazione, o comunque dei dati memorizzati all'interno di un sistema di elaborazione, non possano essere visualizzati, catturati, da persone che non hanno diritto di accedere a questi dati. In questa *slide* noi abbiamo una comunicazione tra due persone che intendevano mantenerla riservata soltanto tra loro due, ma come spesso capita, una terza persona che sia in mezzo fisicamente, o dal punto di vista logico, di questa comunicazione, è sicuramente in grado di intercettarla, se questa non è stata protetta con opportuni codici di riservatezza.

Integrità (1)

Integrita' (1)

Un'altra proprietà che noi desideriamo avere all'interno di un sistema sicuro è la cosiddetta integrità dei dati. Intuitivamente è abbastanza facile da capire che cos'è l'integrità: evitare delle modifiche, ma in realtà non è questa l'unica accezione. Supponiamo di aver ricevuto un ordine di pagamento di 1.000 Euro. Questo ordine di pagamento, mentre transita in rete, potrebbe essere modificato in vario modo: ad esempio se è un assegno che sta arrivando a me, io potrei avere interesse ad aggiungerci un semplicissimo zero per trasformare la cifra da 1.000 a 10.000 Euro. Ma, nel caso in cui il pagamento sia destinato ad una persona, che mi sta cordialmente antipatica, quello che io posso fare è cancellare uno o più zeri, ad esempio trasformando l'ordine da 1.000 Euro a soltanto 100 Euro. Ovviamente ci vogliono dei codici che prevengano questo genere di modifiche; fortunatamente tale compito risulta essere in genere abbastanza facile: è più difficile, ad esempio, rilevare degli attacchi all'integrità che comportino la cancellazione del messaggio, perché se il sistema ricevente non si aspettava di ricevere questo ordine di pagamento di 1.000 Euro, il fatto che qualcuno, durante il transito in rete di questo ordine, lo cancelli, non comporta nessun allarme automatico da parte del sistema ricevente. Bisogna quindi avere dei sistemi che siano in grado di rilevare se dei messaggi sono stati cancellati. Questo tipicamente viene fatto a livello applicativo, nel senso che quando ad esempio si manda un ordine di pagamento, si aspetta una conferma che il pagamento sia stato effettivamente ricevuto, ma a livello basso, a livello di pacchetti di rete, la cancellazione di pacchetti è una cosa che può anche essere dovuta a guasti di rete e quindi i sistemi non sanno bene distinguere un attacco da un semplice guasto.

Integrità (2)

Integrita' (2)

Infine, anche nel caso che noi siamo riusciti a proteggere i nostri pacchetti da attacchi che tendano a modificarli o a cancellarli, esiste una terza categoria di attacchi all'integrità del sistema che è abbastanza pericolosa: sono gli attacchi di tipo *replay*. Anche se noi abbiamo protetto molto bene il nostro ordine di pagamento da 1.000 Euro, una persona che lavori all'interno della rete di elaborazione potrebbe semplicemente catturare quei bit e, in tempo successivo, rimetterli in circolo in rete. È una sorta di metodo sperimentale: io ho fatto un'osservazione, ho visto che quando passano quei bit in rete, di cui magari non capisco pienamente la sintassi e il significato, il mio conto corrente cresce di 1.000 Euro, allora se ho la possibilità di rimettere in gioco questi stessi bit, potrò far crescere nuovamente, per un numero di volte a piacere, il mio conto di altri 1.000 Euro. Quindi un altro tipo di attacco molto difficile da parare.

Tracciabilità e non ripudio

Tracciabilita' e non ripudio

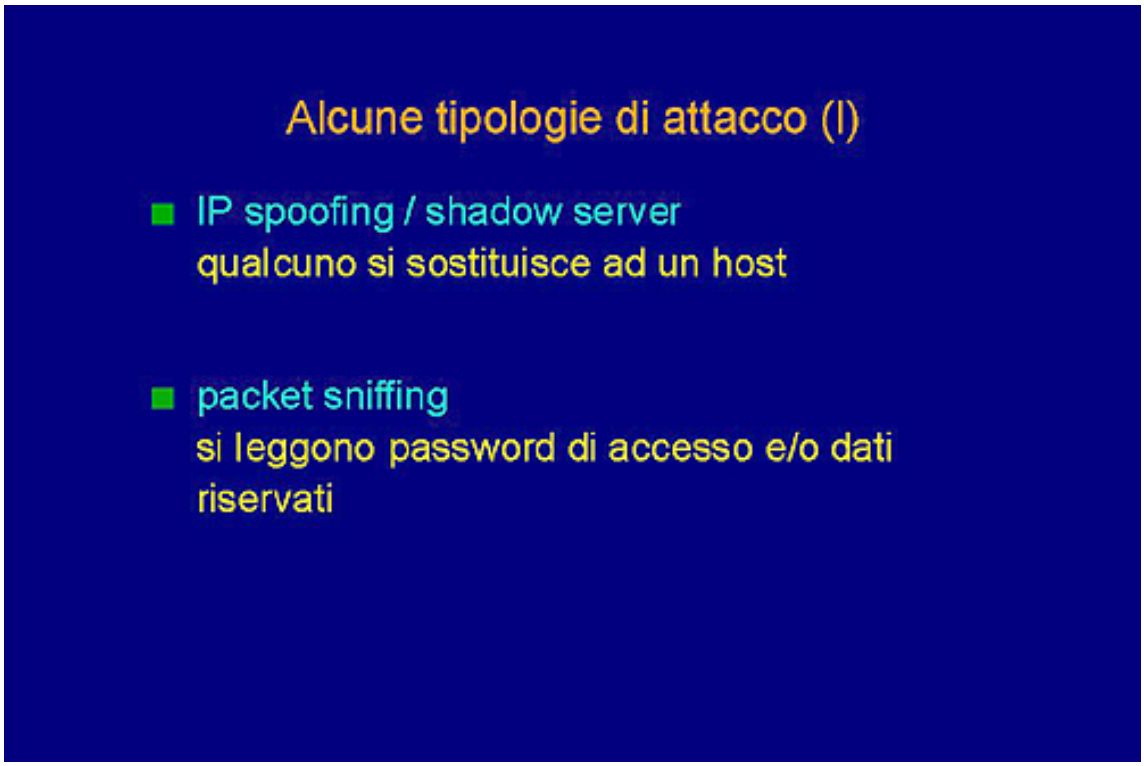
Altre proprietà di sicurezza desiderata all'interno di un sistema informativo sono: la tracciabilità e il non ripudio. Per tracciabilità si intende il fatto che quando una persona compie delle operazioni all'interno di un sistema di elaborazione, noi vorremmo poter essere in grado di seguire le sue orme, le sue tracce. Vorremmo poter essere in grado di evidenziare quali operazioni sono state compiute e da quali persone, all'interno del nostro sistema di elaborazione. Noi possiamo volerlo dimostrare sia in maniera informale, soltanto per dei nostri controlli interni, oppure vogliamo poter dimostrare le azioni che sono state compiute in modo inoppugnabile, ad esempio in modo che possa essere utilizzato anche come prova a carico o a discarico di un certo elemento, in tribunale. Quando le prove che noi adduciamo sono delle prove che sosterrebbero la verifica di un tribunale, normalmente si parla di non ripudio: ossia la persona in questione non può negare di aver svolto effettivamente un certo lavoro o una certa operazione.

Disponibilità

Disponibilita'

Una cosa spesso trascurata quando si parla di sicurezza, è il fatto che fa parte dell'ambito e della competenza del sistema di sicurezza anche garantire la disponibilità del sistema. Per disponibilità si intende che se c'è un certo sistema di elaborazione che deve svolgere un certo compito, che può anche essere molto delicato (in questo esempio vedete illustrata la situazione in cui questo sistema di elaborazione è preposto al controllo del traffico aereo, del traffico ferroviario, all'erogazione di denaro, al controllo di un impianto chimico) sicuramente il fatto che un sistema di questo genere non sia più in grado di svolgere le proprie funzioni potrebbe causare un danno anche molto rilevante.

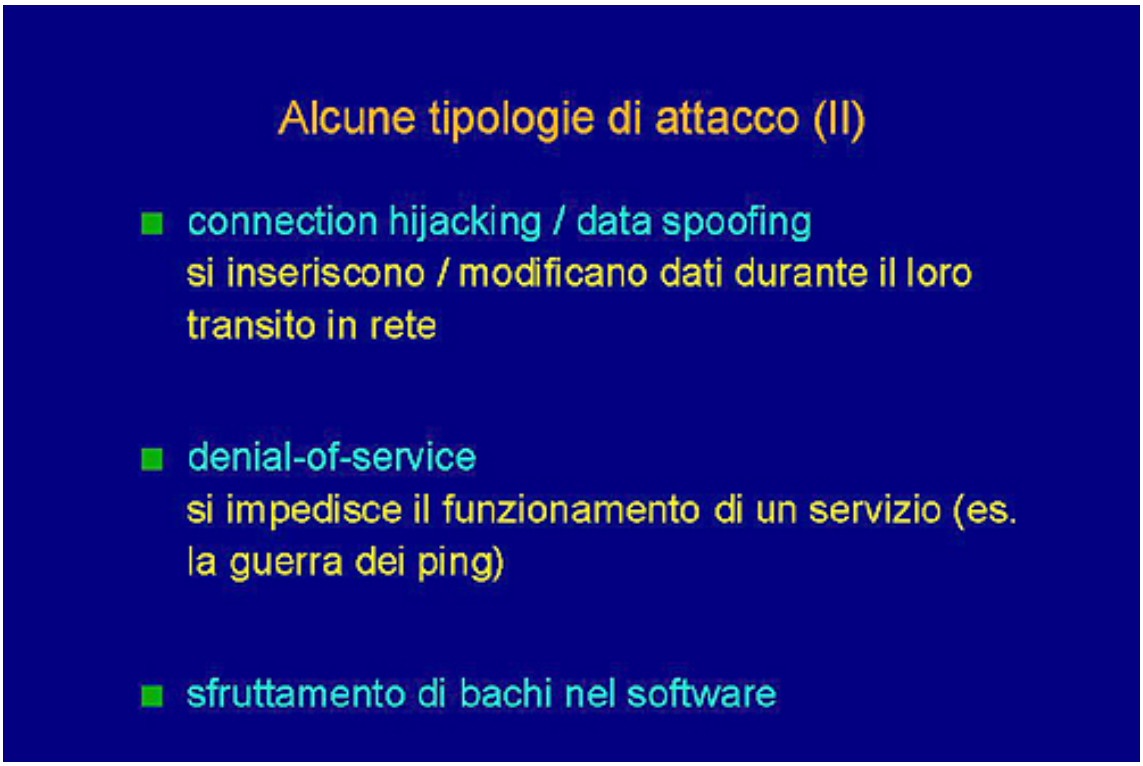
Alcune tipologie di attacco (1)



Alcune tipologie di attacco (1)

Parlando di sicurezza informatica si sentono molto spesso citare anche una serie di termini che riguardano degli attacchi specifici, che possono essere condotti contro i sistemi. In particolare, si parla di attacchi di tipo *IP spoofing* quando il nodo di elaborazione che sta conducendo l'attacco falsifica il proprio indirizzo IP, cioè il proprio indirizzo di rete, per far finta di essere un'altra macchina, tipicamente uno *shadow-server* o *server*-ombra. Si parla invece di *packet sniffing*, letteralmente annusamento dei pacchetti e quindi cattura dei pacchetti, quando un qualunque nodo di elaborazione collegato ad una rete di tipo *broadcast*, come sono in realtà gran parte delle reti locali, svolge delle operazioni di cattura dei pacchetti e quindi dei dati in essi contenuti, durante il loro transito davanti alla sua postazione di lavoro.

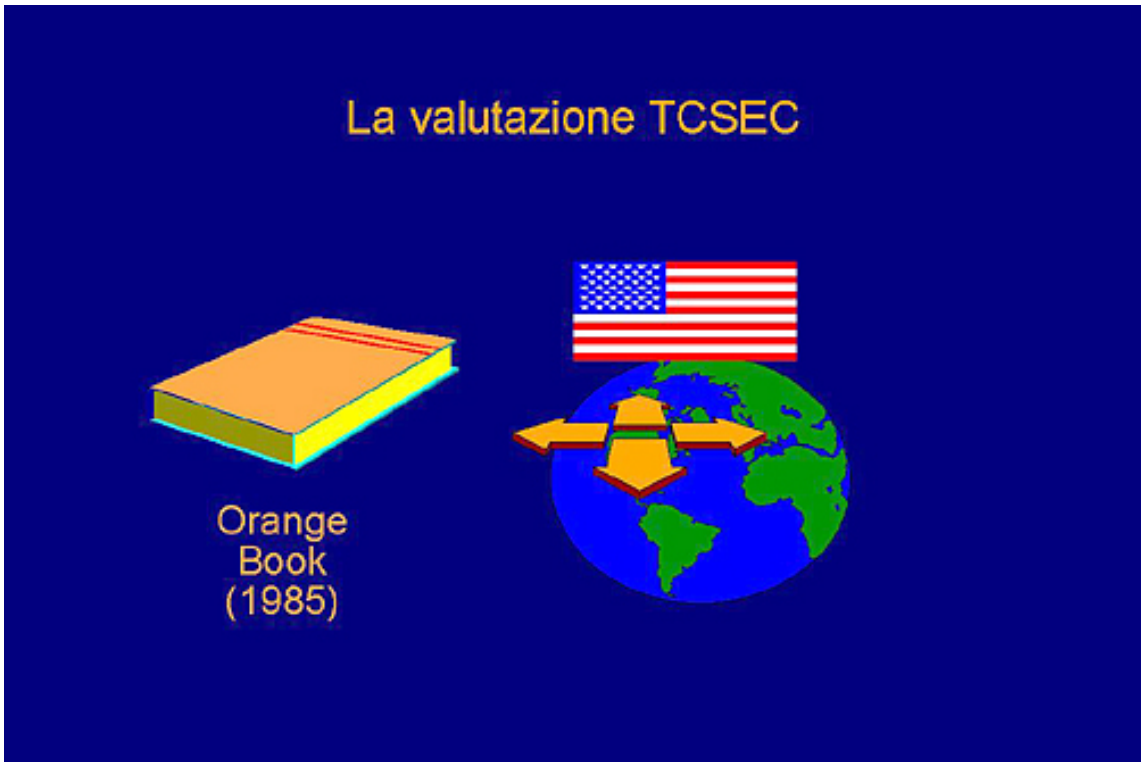
Alcune tipologie di attacco (2)



Alcune tipologie di attacco (2)

Ci sono poi degli attacchi ancora più raffinati: si parla di *connection hijacking* o di *data spoofing* quando un nodo di elaborazione non soltanto svolge un'operazione di cattura dei pacchetti che stanno transitando, ma addirittura emette dei pacchetti falsi all'interno di un collegamento già stabilito. Si parla invece di attacchi *denial-of-service* per quegli attacchi che sono mirati a togliere la disponibilità di un certo sistema informatico, tipicamente degli attacchi che portano al *crash* o al blocco di un sistema di elaborazione. Statisticamente si vede che la maggior parte degli attacchi informatici più banali vengono condotti semplicemente sfruttando bachi del *software*. Questo è un grosso problema: i normali sistemi informativi tendono ad essere sviluppati con sempre minor cura nella parte di sviluppo del *software*, questo perché c'è sempre maggior pressione per arrivare più in fretta alla nuova *release* del prodotto e i codici divengono sempre più grossi. Bisogna però ricordarsi che ogni baco *software* che è rimasto all'interno di un programma può essere sfruttato da un attaccante per conquistare o mettere in ginocchio il nostro sistema.

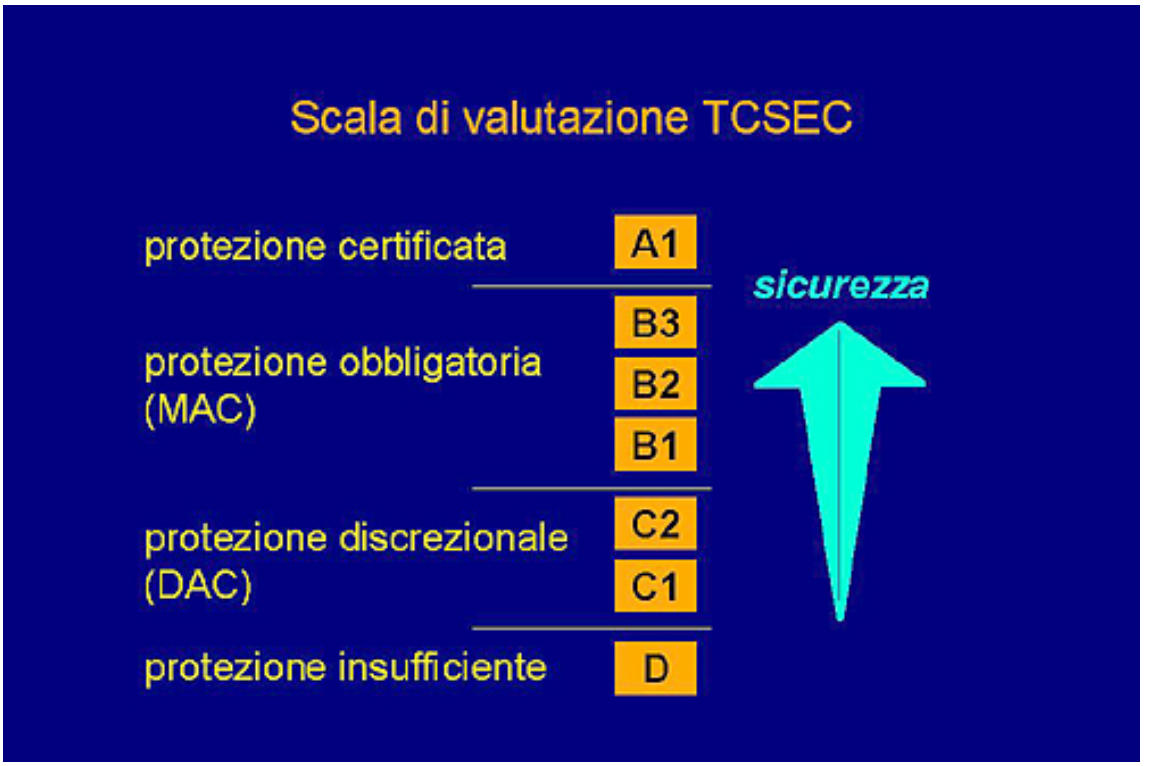
La valutazione TCSEC



La valutazione TCSEC

In generale, non è semplice riuscire a valutare la sicurezza di un sistema informativo: come ausilio per tale valutazione sono stati messi a punto dei criteri nazionali ed internazionali. Lo standard TCSEC è uno standard molto vecchio, del 1985, sviluppato dagli USA, per valutare la sicurezza dei sistemi che venivano venduti alla pubblica amministrazione americana non solo agli enti militari, ma anche agli enti civili statunitensi. Siccome questi criteri con cui valutare un sistema informatico sono stati pubblicati in un libro dalla copertina arancione, sono anche noti come *Orange Book*. Questo è stato appunto il primo tentativo di sistematicizzare la valutazione e la certificazione di sicurezza informatica e quindi dagli USA questa tecnologia si è rapidamente diffusa in tutto il mondo.

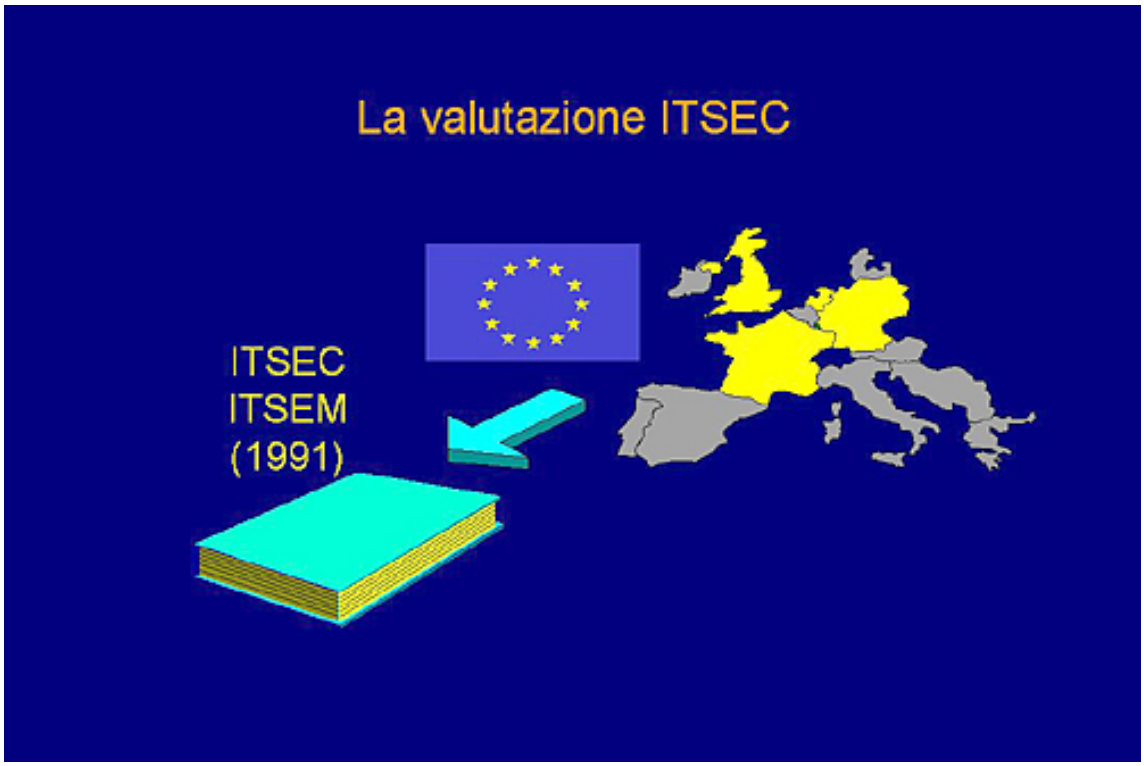
Scala di valutazione TCSEC



Scala di valutazione TCSEC

Il risultato di una valutazione di tipo TCSEC consiste in una classe, ossia il sistema che è stato valutato viene definito con uno dei seguenti tipi : D; C1, C2; B1, B2, B3; A1, a seconda del livello di sicurezza che il valutatore ha riscontrato nel sistema in esame. Come si vede dalla figura, la classe D non si nega a nessuno, è la classe in cui i sistemi hanno una protezione insufficiente. Le classi C, invece, vengono date ai sistemi dotati di protezione discrezionale o DAC (*Discretionary Access Control*), cioè il grado di protezione effettivo di questi sistemi dipende fortemente dalla loro configurazione, dalla loro manutenzione, quindi dalla operatività che il *system-manager* svolge su di essi. I sistemi nella classe B sono invece a protezione obbligatoria, vuol dire che il sistema operativo ha una serie di controlli intrinseci che non possono essere aggirati neanche dal *system-manager*; in questo senso la protezione è già cablata, insita dentro al sistema. Bisogna però prestare attenzione al fatto che più un sistema ha questi meccanismi di autodifesa, tanto più tende a limitare la normale operatività degli utenti. Quindi i sistemi di classe B sono dei sistemi che tendono ad essere di sempre più difficile utilizzo per un utente normale o medio. Infine nella classe A1 troviamo i sistemi detti a protezione certificata, ossia sistemi che sono stati progettati con tecniche formali, matematiche, informatiche, che dimostrano che il sistema è assolutamente sicuro. Come è facile intuire non esiste ad oggi nessun sistema informatico al mondo che sia mai stato certificato in classe A1, perché sembra essere un compito improponibile.

La valutazione ITSEC



La valutazione ITSEC

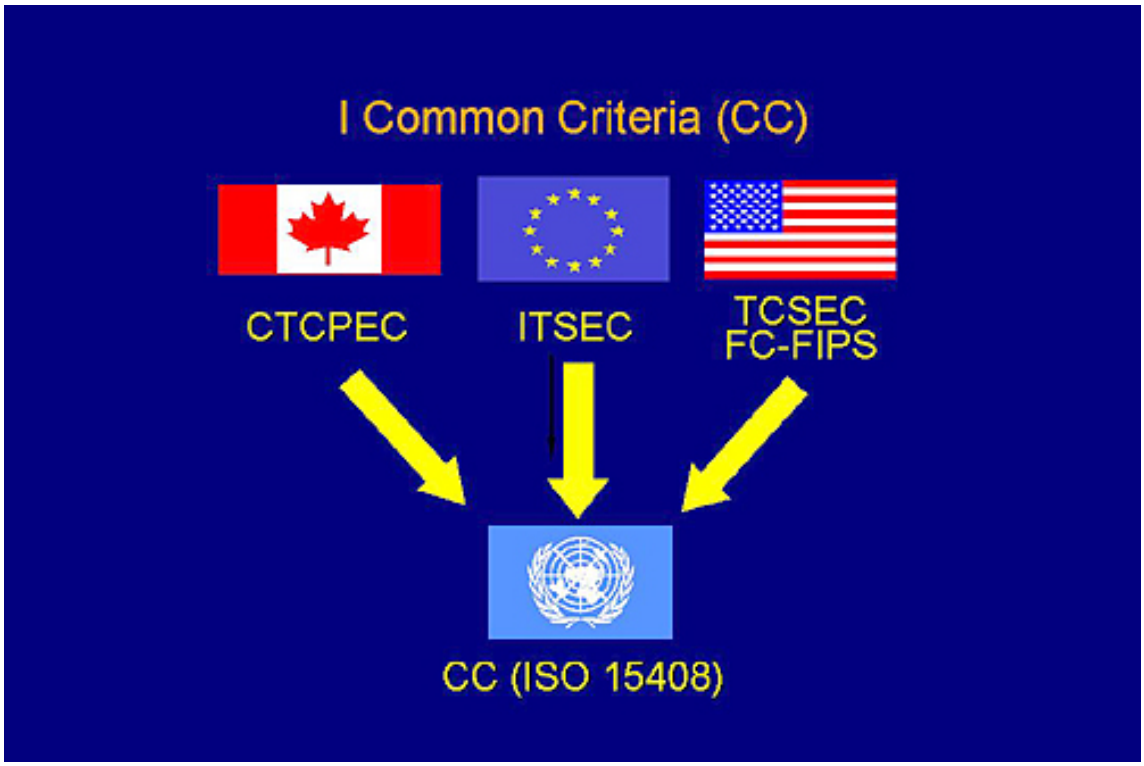
Successivamente ai criteri TCSEC, sono stati sviluppati i cosiddetti criteri di valutazione ITSEC, ideati da un gruppo di lavoro misto tra esperti tedeschi, olandesi, francesi ed inglesi, ed è poi stato adottato in gran parte da molti altri paesi appartenenti all'Unione Europea. Sono stati sviluppati nel 1991 ed in particolare uno dei loro principali vantaggi è quello di descrivere non soltanto la procedura di certificazione, ma anche la procedura con cui viene sviluppata la valutazione dei sistemi.

Scala di valutazione ITSEC

Scala di valutazione ITSEC

Il risultato di una valutazione ITSEC è dato da una coppia di parametri: la forza dei meccanismi di difesa (*strength*), che può essere *BASIC*, *MEDIUM* o *HIGH*, e il livello di correttezza dell'implementazione. Questa può essere E0, che equivale ad una correttezza bassissima o non verificata, oppure può essere un numero crescente da E1 a E6. Per comodità d'uso, a volte si fanno delle equivalenze con le classi di TCSEC, ad esempio si dice che una classe TCSEC C2 equivale ad una classe ITSEC con funzionalità equivalenti alla C2 (F-C2) e con grado di correttezza E2. Invece, ad esempio, una classe A1 corrisponderebbe ad una funzionalità B3, ma con un livello di correttezza assoluto, ossia E6.

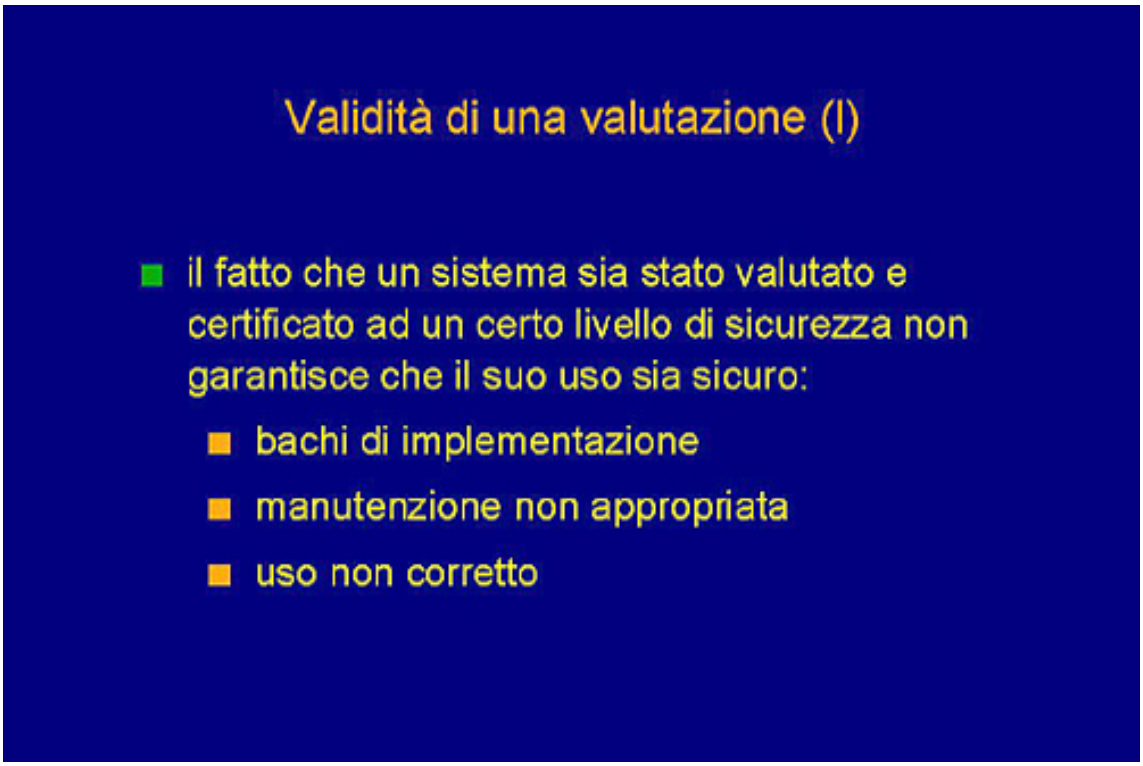
I Common Criteria (CC)



I Common Criteria (CC)

Per comporre questa diatriba tra Europa e USA per quale sia il corretto sistema di valutazione di un sistema informatico è entrata in campo ISO, l'organizzazione internazionale degli standard, il quale ha sviluppato i cosiddetti *Common Criteria*, che sono una armonizzazione dei criteri di sicurezza canadesi, europei (ITSEC) e americani, che nel frattempo, dopo aver pubblicato TCSEC, avevano sviluppato anche altri nuovi criteri, detti FC-FIPS. Questi criteri sono quelli che attualmente, in prospettiva, dovrebbero armonizzare tutte le valutazioni di sicurezza e sono già uno standard ISO con il numero 15408.

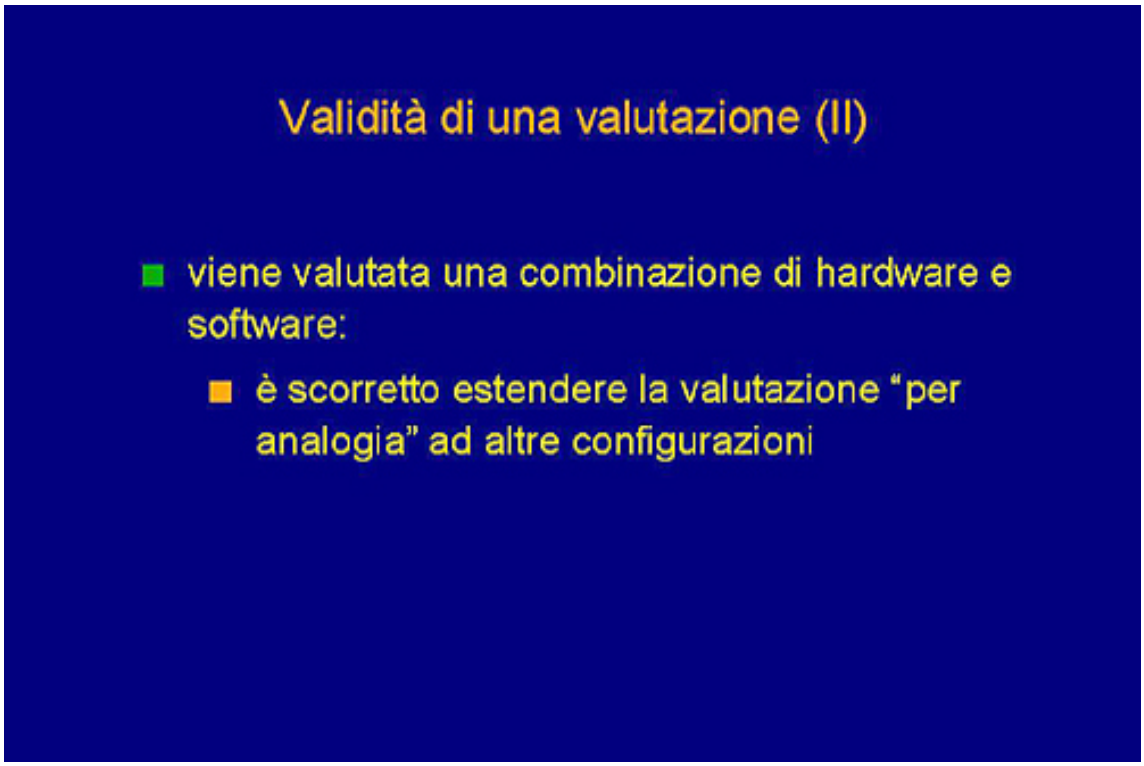
Validità di una valutazione (1)



Validita' di una valutazione (1)

Infine due parole circa la validità di una valutazione. Bisogna prestare molta attenzione al fatto che se un sistema è stato valutato con un certo livello di sicurezza, questo non significa assolutamente che il suo utilizzo sarà certamente sicuro, perché l'utilizzo del sistema informativo in questione dipenderà moltissimo da eventuali bachi ancora presenti nella sua implementazione, da procedure di manutenzione non appropriate e dall'uso non corretto da parte degli utenti o da parte del *system-manager*.

Validità di una valutazione (2)



Validita' di una valutazione (2)

Un altro errore che si commette solitamente quando si parla di valutazione dei sistemi informatici, è dimenticarsi che la valutazione si riferisce ad una ben precisa combinazione di *hardware* e *software*, è quindi scorretto in ogni modo trasferire questa valutazione su sistemi simili ma non identici. Se, ad esempio, si cambia l'*hardware* sul quale un sistema informativo è stato implementato, non è detto che la valutazione sia ancora valida, in generale occorre una nuova valutazione ogni volta che si cambia anche la più piccola parte dell'*hardware* o del *software* rispetto alla configurazione che è stata valutata.

Standard internazionali di sicurezza

Franco Callegati
Paolo Zaffoni

8.4.1 (Spiegare i principali aspetti della sicurezza connessi alla trasmissione dei dati), 8.4.2 (Descrivere gli attuali standard di crittografazione: chiavi pubbliche e private, NSA, DES, PGP)

Valutazione della sicurezza dei sistemi informatici

Nel valutare la sicurezza dei sistemi informatici, si può procedere in due diversi modi. Il primo, il **metodo sperimentale**, in genere prevede un approccio volto a dimostrare la presenza o l'assenza dei problemi sottoposti a test. Il nome del gruppo di persone che si è posto l'obiettivo di analizzare le problematiche di sicurezza della rete è *TIGER TEAMS*.

L'altro metodo, il **metodo analitico** di valutazione, prevede l'esame delle caratteristiche del sistema, dal progetto *hardware* e *software*, al fine di determinare logicamente ed analiticamente, il grado di resistenza del sistema.

Un aspetto non secondario delle valutazioni di sicurezza, è la **certificazione**, un'attestazione che la valutazione è stata condotta secondo metodologie standard. Molti sistemi di valutazione, riportano anche l'attestazione del livello di sicurezza rilevato dalla valutazione.

L'acquirente di un prodotto di sicurezza può essere certo che il prodotto che gli viene offerto fornisce la protezione richiesta attraverso la consultazione di enti certificatori.

Certificatori ufficiali 1

Coloro che definiscono i criteri e gli standard ora disponibili in tutto il mondo, sono quasi totalmente annoverabili tra le fila delle agenzie di sicurezza interna degli Stati Uniti. Così tra gli sviluppatori troviamo:

- **DoD** - ***D****epartment* ***o****f* ***D****efence*
- **NCSC** - ***N****ational* ***C****omputer* ***S****ecurity* ***C****enter*
- **NSA** - ***N****ational* ***S****ecurity* ***A****gency*
- **NIST** - ***N****ational* ***I****nstitute of* ***S****tandards and* ***T****echnology*

Nel 1977 lo *U.S. Department of Defense* (DOD) promuove la *DoD Computer Security Initiative* che coinvolge governo e privati in varie attività per fare il punto della situazione sul tema della sicurezza e per analizzare le modalità di realizzazione di meccanismi per la valutazione di sistemi sicuri.

Negli stessi anni ('73/'74) NBS (*National Bureau of Standard*) chiamato attualmente NIST (*National Institute for Standard and Technology*) propone due attività:

- definizione di standard di crittografia da adottare dalle agenzie federali che dà origine nel '77 al DES (commissionato alla IBM)
- definizione di standard per lo sviluppo e valutazione di sistemi sicuri

Nel 1979 la *Mitre Corporation* definisce un insieme di criteri per la valutazione della sicurezza di un sistema.

In questo contesto vengono presentati i seguenti set di criteri:

- **TCSEC** (***T**rusted **C**omputer **S**ecurity **E**valuation **C**riteria* - *Orange Book* del DoD)
- **CCITEC** (***C**ommon **C**riteria for **I**nformation **T**echnology **E**valuation **C**riteria*)

Certificatori ufficiali 2

Nel 1981 nasce all' interno di **NSA** (*National Security Agency*) il **CSC** (*Computer Security Center*) con l' obiettivo di continuare e potenziare l'iniziativa del DoD.

Nel 1985 nasce NCSC (*National Computer Security Center*) che unisce il CSC e tutte le agenzie federali. Obiettivi:

- essere un punto di riferimento per governo e industrie per la sicurezza di sistemi operativi con informazioni classificate
- definire i criteri per la valutazione di sicurezza di sistemi di elaborazione e sistemi di sicurezza
- incoraggiare la ricerca anche per sistemi distribuiti
- sviluppare strumenti di verifica e test per la fase di certificazione di un sistema di sicurezza
- Risultato più noto è la pubblicazione di *Orange Book*

Nel 1991 NSA e NIST avviano un progetto comune che nel '92 porta alla definizione dei nuovi criteri federali basati anche su i criteri canadesi *Canadian Tusted Computer Product Evaluation Criteria* (CTCPEC) del '89, ed i criteri europeri ITSEC.

In Europa la Germania ha istituito il CCSC (*Commercial Computer Security Center*) ed ha coinvolto le organizzazioni CCTA (*Central Computer and telecommunication Agency*) per gli ambienti industriali e commerciali CESG (*Communications-Electronics Security Group*) per gli ambienti militari. Inghilterra, Francia, Olanda collaborano con la Germania per la definizione dello standard europeo di sicurezza.

Nel 1990 CCSC pubblica ITSEC (*Information Technology Security Evaluation Criteria*) detti *White Books* (ultima edizione 1.2 a giugno 1991) e nel 1993 CCSC pubblica ITSEM (*Information Technology Security Evaluation Manual*).

TCSEC 1

Il primo standard che stabilisce i diversi livelli di sicurezza utilizzati per proteggere *hardware*, *software* ed informazioni memorizzate in un sistema è rappresentato dal ***T**rusted **C**omputer **S**ystem **E**valuation **C**riteria* redatto dal Dipartimento della Difesa degli Stati Uniti:

- *Orange Book*.

Tale denominazione deriva dal fatto che appartiene a una collana di libri ognuno dei quali ha una copertina di colore diverso.

Questi livelli descrivono differenti tipi di protezione fisica, meccanismi di autenticazione degli utenti, affidabilità del *software* del sistema operativo e delle applicazioni degli utenti. Questi standard impongono anche dei limiti sui tipi di sistemi che possono collegarsi all'*host* di cui si valuta la sicurezza.

L'*Orange Book* è rimasto inalterato da quando nel 1985 è divenuto uno standard del

Dipartimento della Difesa. Per molti anni questo libro ha rappresentato il punto di riferimento per la valutazione della sicurezza dei sistemi *mainframe* multi-utenti e dei sistemi operativi dei *minicomputer*. Altre realtà, quali *database* e reti, sono state valutate mediante estensioni interpretative dell'*Orange Book*, quali la *Trusted Database Interpretation* e la *Trusted Network Interpretation*.

TCSEC 2

Il TCSEC classifica i sistema in 4 gruppi di requisiti fondamentali:

Politica

- Politica di sicurezza - deve definire gli oggetti, i soggetti e le regole di accesso.
- *Marking* - gli oggetti devono aver associate etichette di controllo degli accessi.

Responsabilità

- Identificazione/autenticazione - i soggetti devono essere identificati.
- Resposabilità - devono essere mantenute e protette informazioni di *audit* per poter attribuire responsabilità in caso di azioni che inficiano la sicurezza del sistema.

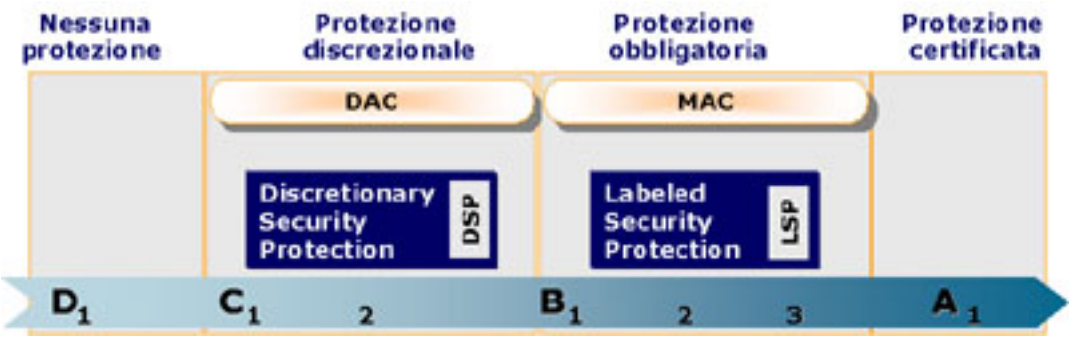
Affidabilità

- Affidabilità - i meccanismi hw e sw contenuti nel sistema devono poter essere valutati indipendentemente per verificare che i requisiti suddetti siano rispettati.
- Protezione continua - i meccanismi del sistema di sicurezza devono essere protetti da modifiche non autorizzate e manomissioni.

Qualità della documentazione

TCSEC 3

Sulla base dei requisiti (politica, responsabilità, affidabilità) raggruppa i sistemi di sicurezza in 4 classi:



D Protezione minima

C Protezione discrezionale

- C1 protezione discrezionale.
- C2 protezione degli accessi controllata.

B Protezione obbligatoria

- B1 protezione basata su etichette.
- B2 protezione strutturata.
- B3 domini di sicurezza.

A Protezione certificata

- A1 progetto certificato.

Ogni classe aggiunge requisiti rispetto alla classe precedente. **TCB** (***T****rusted* ***C****omputing* ***B****ase*) è l'insieme dei meccanismi (hw, sw e fw) che realizzano le politiche di sicurezza di un sistema o prodotto sotto valutazione.

Livello D

Il **livello D** (o D1), rappresenta la forma più lasca di sicurezza. Questo standard definisce le condizioni che si hanno quando un intero sistema non è sicuro. Non esiste alcuna protezione sull'*hardware*, il sistema operativo può essere facilmente compromesso e non sono stabilite regole per l'autenticazione degli utenti né le modalità di accesso alle informazioni del sistema. Questo livello di sicurezza fa tipicamente riferimento a sistemi operativi quali MS-DOS, MS- *Windows* e *Apple Macintosh System* 7.x.

Questi sistemi operativi non sono in grado di discriminare tra utenti differenti e non prevedono un meccanismo specifico per identificare chi sta effettivamente operando sul sistema. Inoltre, questi sistemi non impongono alcun controllo sulle informazioni cui si può avere accesso sul disco rigido.

Livello C1

Il **livello C** prevede due sottolivelli di sicurezza, **C1** e **C2**.

Il **livello C1**, conosciuto anche come livello di tipo *Discretionary Security Protection*, descrive le caratteristiche di sicurezza tipicamente presenti su un sistema *Unix*.

Sono previsti alcuni **meccanismi di protezione** per l'*hardware* in modo che non possa essere facilmente compromesso, anche se questa eventualità è ancora possibile. Gli utenti devono identificarsi sempre nei confronti del sistema mediante un *login* e una *password*; questo permette di garantire a ciascun utente i diritti di accesso ai programmi e alle informazioni del sistema. Fondamentalmente questi diritti di accesso si riassumono nei permessi (*permission*) che regolano l'accesso a *file* e cartelle. Questi meccanismi (conosciuti come *Discretionary Access Controls*) permettono al proprietario di un *file* o di una *directory* (o all'amministratore del sistema) di impedire a determinati utenti o gruppi di utenti l'accesso a specifici programmi ed informazioni.

Tuttavia, l'amministratore del sistema ha la possibilità di effettuare qualunque tipo di operazione. Di conseguenza, un amministratore poco accorto può compromettere la sicurezza dell'intero sistema senza che nessun altro utente se ne renda conto. Inoltre, molti dei compiti giornalieri di un amministratore possono essere svolti soltanto attraverso un *login* di tipo *root*.

Stante la decentralizzazione dei sistemi informatici moderni, non è improbabile la situazione per cui più di due o tre persone, all'interno della stessa organizzazione, conoscono la *password* di *root*. Questo rappresenta un problema perché risulta impossibile individuare la persona che ha operato delle modifiche sulla configurazione del sistema.

Il **Tcb di un C1** (protezione discrezionale) prevede i seguenti aspetti:

- Politica: Controllo degli accessi discrezionale. Il Tcb deve poter definire e controllare gli accessi dei singoli utenti sulle singole risorse (*file* e programmi) (meccanismi ACL).
- Responsabilità: Identificazione e Autenticazione. Il Tcb deve poter identificare gli utenti che richiedono un accesso (meccanismo delle *password*).
- Affidabilità: L'esecuzione del Tcb dovrà essere protetta. Le feature periodicamente validate, i meccanismi verificati aderenti a quanto descritto nella documentazione.
- Documentazione:
  - *User Guide* delle *feature* di sicurezza (per l' utente).
  - Manuale delle *facility trusted* (per l'amministratore del sistema).
  - Documentazione della fase di test (funzionale).
  - Documentazione di progetto (moduli, interfacce tra moduli).

Livello C2

Il **livello C2** intendeva risolvere i problemi lasciati aperti dal livello C1. Insieme a quelle tipiche di quest'ultimo livello, le caratteristiche del C2 comprendono la creazione di un ambiente ad **accesso controllato**. Questo ambiente permette di restringere ulteriormente la possibilità che gli utenti eseguano determinati comandi o accedano a *file* attraverso l'uso delle *permission* ma anche di **livelli di autenticazione**. Inoltre, questo livello di sicurezza richiede che il sistema venga monitorato (***auditing***), ovvero che venga scritto un *record* informativo per ognuno degli eventi che si verificano sul sistema stesso.

Il **meccanismo di *auditing*** viene utilizzato per tenere traccia di tutti quegli eventi che riguardano la sicurezza del sistema, come ad esempio le attività svolte dall'amministratore. Un tale meccanismo richiede anche l'uso di tecniche di autenticazione dal momento che, senza di esse non si può essere sicuri dell'identità effettiva della persona che ha eseguito un determinato comando. Lo svantaggio dell'uso di meccanismi di *auditing* risiede nel fatto che essi comportano un carico aggiuntivo e richiedono risorse computazionali.

Mediante l'uso di autorizzazioni aggiuntive, gli utenti di un sistema C2 possono garantirsi i permessi per svolgere operazioni di amministrazione del sistema senza avere per questo bisogno della *password* di *root*. Ciò permette di tenere traccia di chi ha svolto specifici compiti relativi all'amministrazione del sistema dal momento che vengono svolti da un utente e non dall'amministratore stesso.

Queste autorizzazioni aggiuntive non devono essere confuse con i permessi di tipo SGID e SUID applicabili ad un programma. Esse sono piuttosto autorizzazioni specifiche per l'esecuzione di certi comandi o per l'accesso alle tabelle del *kernel*. Ad

esempio, gli utenti che non hanno autorizzazioni d'accesso alla tabella dei processi potranno soltanto avere visibilità dei propri processi quando eseguono il comando ps.

Il **Tcb di un C2** (protezione degli accessi controllati):

- Politica - Riuso degli oggetti. Riassegnazione ad alcuni soggetti di un mezzo (settori di disco, nastro magnetico) che ha contenuto uno o più oggetti non deve avere traccia di dati relativi al precedente oggetto contenuto.
- Responsabilità - *Audit*. Il Tcb deve avere il controllo degli accessi con meccanismi di *account* sui singoli utenti, di *audit* delle attività di ciascun utente.
- Affidabilità - (come C1).
- Documentazione - (come C1).

Livello B

Livello B1

Il livello B di sicurezza comprende tre sotto-livelli. B1 (*Labeled Security Protection*) è il primo sotto-livello che supporta la sicurezza di differenti gradi, come ad esempio *secret* e *top secret*. Questo livello stabilisce che ad un oggetto (ad esempio un *file*) sottoposto ad una politica di controllo obbligato dell'accesso non possano essere modificate le *permission* neanche da parte del proprietario dell'oggetto stesso.

Il **Tcb di un B1** (protezione basata su etichette)

- Politica - Etichette. Etichettatura di soggetti e oggetti (processi, *file*, segmenti, *device*) - Controllo degli accessi obbligatorio. Basato sulle etichette.
- Responsabilità (come C2).
- Affidabilità. Verifica delle specifiche di disegno. Deve essere rimossa ogni imperfezione rilevata al momento del test.
- Documentazione. Deve fornire almeno un livello informale della politica di sicurezza (specifiche di disegno).

Livello B2

Il livello B2 (*Structured Protection*) richiede che ogni oggetto del sistema venga contrassegnato con una etichetta (*label*). I dispositivi (dischi, terminali, *tape*) possono avere uno o più livelli di sicurezza associati. B2 è il primo livello che affronta il problema delle comunicazioni tra oggetti cui sono assegnati gradi di sicurezza diversi.

Il **Tcb di un B2** (protezione strutturata)

- Politica - Le politiche di controllo degli accessi (mandatoria e discrezionale) devono essere estese a tutti i soggetti e oggetti del sistema.
- Responsabilità - Il Tcb supporterà un canale protetto (*Trusted Path*) per le fasi di *login* e autenticazione.
- Affidabilità - Il Tcb deve essere strutturato in moduli critici e non ai fini della protezione. Deve avere un'interfaccia ben definita. Il progetto e la realizzazione devono essere soggetti a test sofisticati e revisioni. Richiede

meccanismi di autenticazione e strumenti per la configurazione. Il sistema deve essere relativamente resistente a tentativi di penetrazione.
- Documentazione - Il Tcb deve essere basato su un modello ben definito e documentato. (specifiche di disegno).

Livello B3

Il livello B3 (*Security Domains*) rafforza il concetto di dominio mediante l'utilizzo di *hardware* specifico. Ad esempio viene utilizzato *hardware* appositamente predisposto alla gestione della memoria per proteggere un dominio sicuro da accessi non autorizzati. Questo livello richiede inoltre che i terminali degli utenti siano collegati al sistema mediante connessioni sicure.

Il **Tcb di un B3** (domini di sicurezza)

- Politica. (come B2).
- Responsabilità. (come B2).
- Affidabilità. Il Tcb non deve poter essere manomesso, deve poter essere analizzato e testato. Quindi il codice deve essere ben strutturato e deve usare tecniche di ingegnerizzazione del *software* in fase di progetto e realizzazione per avere minima complessità. Deve prevedere un amministratore del sistema. Deve avere un buon sistema di *auditing* e procedure di *recovery*. Deve essere altamente resistente ai tentativi di penetrazione.
- Documentazione. (come B2).

Livello A

Il livello A (*Verified Design*) è il più alto livello di sicurezza previsto dall'*Orange Book*. Questo livello comprende delle fasi di progettazione, controllo e verifica del sistema altamente sicure.

Per raggiungere un tale livello di sicurezza è necessario considerare tutte le componenti di un sistema a partire da quelle dei livelli più bassi; il progetto deve essere matematicamente verificato, si deve effettuare un'analisi dei canali di comunicazione e la distribuzione delle componenti stesse del sistema deve essere sicura. Questo significa che sia l'*hardware* che il *software* devono essere protetti durante la fase di consegna per prevenire intrusioni nel sistema globale di sicurezza.

Il **Tcb di un A1** (progetto certificato)

- si distingue da quello di un B3 dal fatto che la sua affidabilità deriva dall'aver utilizzato un modello formale delle politiche di sicurezza, una specifica formale di alto livello del progetto. Il modello deve essere estendibile.

ITSEC

ITSEC

Armonizza i criteri di valutazione della sicurezza definiti separatamente da vari paesi europei (Gran Bretagna, Francia, Germania, ...). Permette la selezione di funzioni di sicurezza in un sistema/prodotto e definisce 7 livelli di valutazione di affidabilità che

rappresentano la capacità del sistema/prodotto di realizzare le specifiche di sicurezza attraverso le funzioni suddette.

ITSEM

Manuale per la valutazione della sicurezza di prodotti e sistemi che fornisce le basi per una unificazione dei metodi di valutazione della sicurezza definiti dai vari Enti certificatori oltre che un sussidio dei concetti espressi in ITSEC.

Utilizzatore/Fornitore

L'**utilizzatore** del Sistema è la persona o l'ente proprietario del sistema e dei dati in esso elaborati. Costui dispone di una propria politica di sicurezza e deve verificare che il sistema che gli viene offerto sia in grado di soddisfare tale politica. Deve inoltre confrontare sistemi diversi per poter decidere quale risponda meglio alle sue esigenze.

Il **fornitore** del sistema è responsabile del sistema finale consegnato all'utilizzatore e deve convincere l'utilizzatore sull'adeguatezza delle funzioni fornite dal sistema. Il fornitore deve cautelarsi qualora l'utilizzatore subisca danni imputabili ad un cattivo funzionamento del sistema dal punto di vista della sicurezza.

I requisiti funzionali sono le contromisure previste attraverso le quali si esprime la sicurezza del TOE (*Target Of Evaluation*), mentre i requisiti di tipo qualitativo esprimono le modalità e l'accuratezza con cui le contromisure di sicurezza sono realizzate.

ITSEC tratta soltanto i requisiti di tipo qualitativo, ma include 10 classi predefinite di requisiti funzionali. Se non utilizzate (non è obbligatorio) suggerisce di far riferimento a 8 gruppi generici (*generic headings*).

Requisiti funzionali suggeriti

Identification and authentication
funzioni che consentono di verificare l'identità degli utenti che chiedono l'accesso a risorse controllate dal TOE.

Access control
funzioni che controllano l'accesso alle risorse (diritti di accesso e loro verifica).

Accountability
funzioni che tracciano le attività di utenti/processi con lo scopo di attribuire tali attività a chi le ha svolte.

Audit
funzioni che registrano e analizzano gli eventi che potrebbero rappresentare una minaccia.

Object reuse
funzioni che controllano il riuso delle risorse (memoria centrale o di massa).

Accuracy
funzioni che assicurano che i dati transitino attraverso i processi o passino da un

oggetto ad un altro senza subire alterazioni.

Reliability of service
funzioni che assicurano la accessibilità delle risorse a entità legittime entro tempi prefissati, individuano errori ed effettuano il *recovery*.

Data exchange (vedi ISO 7498-2)
funzioni che garantiscono la sicurezza delle informazioni trasmesse sui canali di comunicazione (autenticazione, controllo degli accessi, confidenzialità, integrità, non ripudio).

Security Target

È un documento che costituisce il riferimento per le attività di progettazione e valutazione di un TOE (un sistema o un prodotto) e descrive gli obiettivi di sicurezza in termini di

- Riservatezza.
- Integrità.
- Disponibilità.

La sua struttura è diversa se il TOE è un sistema o un prodotto. è redatto dallo Sponsor (chi richiede la valutazione del TOE).

Valutare la sicurezza di un TOE consiste nello stimare il grado di fiducia che è possibile riporre nelle funzioni di sicurezza adottate e specificate nel *Security Target* e verificare che il TOE rispetti gli obiettivi di sicurezza per cui viene realizzato.

Fasi del processo di valutazione 1

Le fasi del processo di valutazione consistono in:

- **Preparazione** - attivazione dei contatti, studio del *Security Target*.
- **Valutazione** - fase centrale.
- **Conclusione** - produzione dell' *Evaluation Technical Report* (ETR) che però non indica il livello di valutazione.
- **Certificazione** - emissione del certificato che riporta il livello di sicurezza (E1,...E6) emesso da un Ente Certificatore (in Italia ce ne sono 4).

Livelli di fiducia

- E0: Il TOE ha una sicurezza inadeguata.
- E1: Prevede un *Security Target* e una descrizione informale (linguaggio naturale) dell'architettura del TOE. I test devono dimostrare che il TOE soddisfa il suo *Security Target*.
- E2: Prevede una descrizione informale del progetto dettagliato del TOE. Deve essere valutata la prova dei test e ci deve essere un sistema di controllo della configurazione e una procedura approvata di distribuzione.
- E3: Oltre a quanto previsto per E2 devono essere valutati i disegni del codice sorgente e/o *hardware* corrispondenti ai meccanismi di sicurezza e deve essere valutata la prova dei test di tali meccanismi.
- E4: Prevede un modello formale (di tipo matematico) della politica di

sicurezza. Le funzioni di sicurezza, il progetto architetturale e quello dettagliato devono essere specificati con un sistema semiformale (basato su strumenti specifici: ad es. il *Claims Language* dell'ITSEC che utilizza il linguaggio naturale inglese per il quale sono fissate strutture di frasi molto rigide e parole chiave).
- E5: Oltre ai requisiti di E4 deve esserci una stretta corrispondenza tra il progetto dettagliato e il codice sorgente e/o i progetti dell'*hardware*.
- E6: Oltre ai requisiti di E5 le funzioni di sicurezza e il progetto architetturale devono essere specificati in modo formale consistente con il modello formale della politica di sicurezza.

Fasi del processo di valutazione 2

Un sistema è un'aggregazione di prodotti in esecuzione in un ambiente definito
La struttura secondo ITSEC/ITSEM prevede le seguenti componenti:

- Introduzione.
- Descrizione del sistema.
- Descrizione della politica di sicurezza.
- Descrizione degli obiettivi di sicurezza.
- Descrizione delle minacce.
- Descrizione delle funzioni di sicurezza.
- Descrizione dei meccanismi di sicurezza (opzionale).
- Dichiarazione della robustezza dei meccanismi richiesta.
- Dichiarazione del livello di assicurazione.

Introduzione

- Obiettivi del documento.
- Riferimenti a normative e documenti interni ed esterni.
- Definizione ed acronimi, glossario.

Descrizione del sistema

- Obiettivo del sistema.
- Descrizione e caratteristiche delle informazioni trattate.
- Apparati necessari per l'esercizio del sistema.
- Planimetria del sistema e interconnessioni.
- Misure di sicurezza fisica utilizzate.
- Relazioni con il resto del sistema (se si tratta di un sottosistema).

Descrizione della politica di sicurezza e obiettivi

Vengono definiti gli obiettivi espressi in termini di riservatezza, confidenzialità e disponibilità che riguardano:

- beni che richiedono protezione (informazioni, processi, responsabilità e ruoli degli utenti, ...).
- Risorse fisiche (singoli apparati, PC, ...).
- Risorse astratte (configurazione del sistema, processi, algoritmi, ...).
- Vengono definite le regole da seguire affinchè gli obiettivi della sicurezza possano essere raggiunti (regole di accesso alle risorse, modalità di

connessione, utilizzo di *password*, ruoli, responsabilità, profili utente).

Minacce

Individuare le azioni che possono portare alla violazione degli obiettivi di sicurezza. Le minacce possono essere:

- interne;
- esterne;
- intenzionali;
- accidentali;
- attive;
- passive;
- attività critica ==> analisi dei rischi.

Funzioni di sicurezza

- Realizzano le contromisure necessarie per soddisfare gli obiettivi di sicurezza del sistema.
- ITSEC ne suggerisce 8 gruppi (*Generic Headings*) che però non sono obbligatori.
- Per ogni funzione occorre specificare le motivazioni che hanno portato alla sua realizzazione, come è stata realizzata (caratteristiche principali) e le relazioni della funzione con l'esterno.

Meccanismi di sicurezza

Un meccanismo di sicurezza costituisce il mezzo con cui è possibile realizzare una o più funzioni di sicurezza.

- Un *Security Target* può indicare alcuni meccanismi (facoltativo).
- Occorre tener conto di quelli già presenti nei prodotti del sistema.
- è necessario analizzare le interconnessioni tra i meccanismi per garantire una migliore integrazione.
- Se non vengono specificati i meccanismi significa che si lascia completa libertà a chi realizza le funzioni.

Robustezza di un meccanismo

La robustezza di un meccanismo è definita su una scala di valori (alto medio, basso) tenendo conto delle variabili:

- tempo (minuti, giorni, mesi);
- tipo di attaccante (inesperto, competente, esperto);
- collusione (nessuna, utente autorizzato, gestore del sistema);
- apparecchiature usate (nessuna, sofisticate);
- il *Security Target* deve dichiarare la robustezza richiesta tenendo conto che quella complessiva coincide con quella del meccanismo più debole;

Livello di assicurazione

Valutare l'assicurazione di un TOE significa verificare che le misure di sicurezza realizzate:

- verifichino i requisiti definiti nel *Security Target*;

- siano in grado di contrastare efficacemente le minacce individuate e a quale livello.

L'Assicurazione si valuta in parallelo da due punti di vista: Efficacia e Correttezza. Per quanto riguarda l'efficacia questa valutazione mira a stabilire se le funzioni di sicurezza adottate sono idonee agli scopi per cui sono state scelte, indicati nel *Security Target* e se i meccanismi che le realizzano sono in grado di contrastare i possibili attacchi. Se la verifica dell'Efficacia ha esito negativo il valutatore assegnerà un livello di valutazione E0 indipendentemente dal risultato della valutazione della Correttezza. Infatti se le funzioni non sono idonee, non ha senso valutare se i corrispondenti meccanismi sono stati realizzati correttamente.

La valutazione della correttezza mira a stabilire se le funzioni di sicurezza ed i corrispondenti meccanismi sono stati realizzati correttamente (con la dovuta accuratezza e coerentemente con quanto specificato nel *Security Target*) e prende in esame separatamente il Processo costruttivo e gli Aspetti operativi del TOE.

Confronto

Sia ITSEC che TCSEC forniscono i criteri base per la valutazione della sicurezza da effettuare a carico di organizzazioni di certificazione e costituiscono per gli utenti una guida per la comprensione delle caratteristiche di sicurezza dei sistemi da acquisire.

Orange Book

- Le classi di sicurezza inglobano sia requisiti funzionali che di affidabilità.
- È orientato alla valutazione di prodotti.
- Non specifica, se non marginalmente le azioni che il valutatore deve eseguire.
- Non richiede documentazione specifica a chi pretende la valutazione.

ITSEC

- Separa i requisiti funzionali da quelli di affidabilità.
- È più flessibile poiché si adatta sia alla valutazione di sistemi che di prodotti.
- Specifica le azioni che il valutatore deve eseguire.
- Per ogni fase del processo di valutazione specifica la documentazione che lo Sponsor (chi richiede la valutazione) deve fornire, il suo contenuto e le prove che devono essere fornite per dimostrare che il TOE soddisfa i requisiti richiesti.

Common Criteria

I ***C**ommon **C**riteria* (CC - secondo la terminologia ISO IS 15408 - *Information technology - Security techniques - Evaluation criteria for IT security*) rappresentano il risultato di uno sforzo, iniziato nel 1993 e destinato a superare i limiti dei precedenti standard e a sviluppare una comune metodologia per la valutazione della sicurezza nel

mondo dell'Informatica applicabile in campo internazionale.

I rappresentanti degli Stati Uniti, Canada, Francia, Germania, Olanda e Regno Unito, in collaborazione con l'ISO (*International Standard Organization*), si sono accordati per lo sviluppo di uno standard internazionale di valutazione della sicurezza in ambito informatico con l'obiettivo di rilasciare dei nuovi criteri che fornissero utili risposte all'esigenza di standardizzazione di un mercato informatico sempre più globale. Tali nuovi criteri dovevano inoltre permettere il reciproco riconoscimento della valutazione dei prodotti di sicurezza.

- Nel 1996 è stata rilasciata la versione 1.0 dei *Common Criteria*.
- Nel 1998 si è avuto il rilascio della versione 2.0 , (DIS 15408) divenuto standard con l'approvazione dell'ISO, mentre si è in attesa della prossima versione 2.1.
- Nel gennaio 1999 è stata rilasciata una versione *draft* (v 0.6) della *Common Evaluation Methodology* avente lo scopo di armonizzare le modalità di valutazione da parte degli enti valutatori. Tale metodologia è alla base per il reciproco riconoscimento.

Documentazione

Il documento relativo alla seconda versione finale, si compone di tre parti per un totale di più di 600 pagine. L'elenco che segue, riepiloga le tre parti:

- Parte 1: Introduzione e descrizione del modello generale.
- Parte 2: Requisiti delle funzioni di sicurezza.
- Parte 3: Requisiti e livelli dell'affidabilità della sicurezza.

Il documento, nonostante il linguaggio comprensibile a chi è prossimo all'informatica ed in genere alla sicurezza, diventa complesso a causa del diffuso utilizzo di sigle ed abbreviazioni. Sarà opportuno munirsi di un glossario, di una buona dose di pazienza e di tanto tempo a disposizione per studiarlo.

La prima parte [Introduzione e descrizione del modello generale] fornisce:

- all'utente una conoscenza generale;
- allo sviluppatore una conoscenza generale per la formulazione di requisiti e specifiche di un TOE;
- al certificatore una conoscenza generale e guida per la struttura dei PP e ST.

La seconda parte [Requisiti delle funzioni di sicurezza] fornisce:

- all'utente una guida per la formulazione dei requisiti per funzioni di sicurezza;
- allo sviluppatore dei riferimenti per interpretare i requisiti di sicurezza e definire le specifiche funzionali di un TOE;
- al certificatore una formulazione delle dichiarazioni obbligatorie dei criteri di valutazione per valutare se il TOE rispetta le funzioni di sicurezza dichiarate.

La terza parte [Requisiti e livelli dell'affidabilità della sicurezza]:

- all'utente una guida per determinare o richiedere il livello di affidabilità di

sicurezza;
- allo sviluppatore dei riferimenti per interpretare i requisiti di affidabilità e definire l'approccio di affidabilità di un TOE;
- al certificatore una formulazione delle dichiarazioni obbligatorie dei criteri di valutazione per valutare l'affidabilità del TOE, dei PP o ST.

Altra documentazione

I protection profile

Alcuni documenti integrano la documentazione dei CC; si tratta dei ***P**rotection **P**rofile* o **PP** che traducono i principi e le guide contenute nei *Common Criteria* in riferimenti a prodotti o sistemi. Tra i PP si ricordano alcuni tra i più interessanti:

- *Application-Level Firewall for Low-Risk environment* (v. 2.0).
- *Traffic-Filter Firewall for Low-Risk environment* (v. 2.0).
- *Commercial Security* 1 (v. 1.0) - Ambiente semplice con sicurezza a livello base.
- *Commercial Security* 3 (v. 1.0) - Ambiente multi-*user* con *database* e necessità di un livello di sicurezza selettivo.

Metodologia di valutazione

La documentazione disponibile si compone altresì della *draft* rilasciata della ***C**ommon **E**valuation **M**ethodology* (v. 0.6) (**CEM**), articolata in 3 parti:

- Parte 1: Introduzione e modello generale, terminologia e principi di valutazione.
- Parte 2: Metodologia di valutazione, *Protection Profile & Security Target*, Livelli e componenti di affidabilità.
- Parte 3: Ampliamento della metodologia.

Tale documento pone l'attenzione sull'attività degli enti che sono preposti alla valutazione della sicurezza dei prodotti/sistemi informatici garantendo che il loro operato sia congruente con i requisiti dei CC stessi. Si presenta come uno strumento che dovrebbe garantire la consistenza dell'applicazione dei principi contenuti nei CC in caso di valutazioni ripetute nel tempo e secondo schemi diversi.

Modello generale

I *Common Criteria* contengono essenzialmente i principi tecnici fondamentali - di validità generale, chiari e flessibili - per descrivere i requisiti di sicurezza per i prodotti o sistemi informatici.

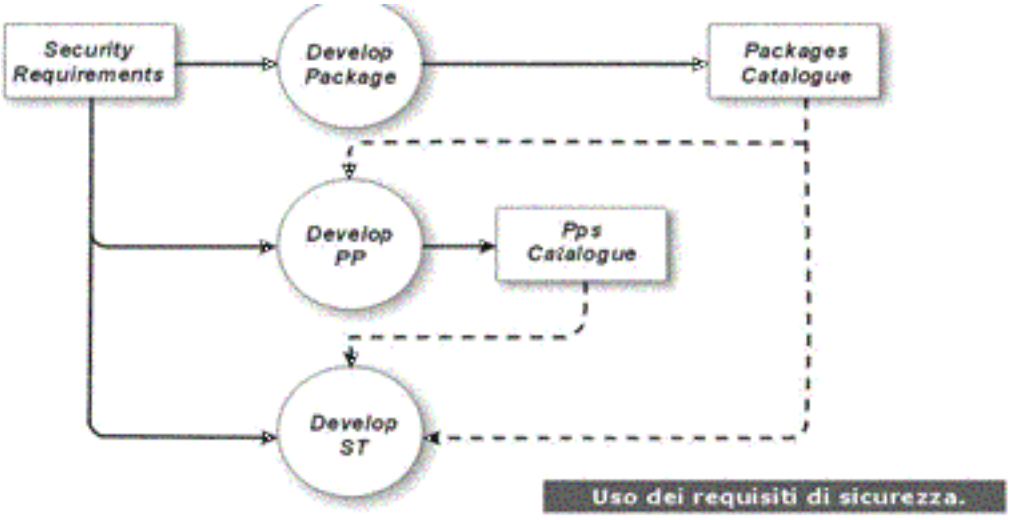
Tali requisiti sono descritti in modo organico e strutturato per due tipologie di situazioni:

- ***P**rotection **P**rofile* (**PP**) - Si riferiscono a famiglie o categorie di prodotti e ambienti generici senza riferimenti a specifici prodotti o sistemi e sono un insieme organico di obiettivi e requisiti di sicurezza associabili a categorie di prodotti o sistemi informatici che soddisfano le necessità di sicurezza degli utenti.
  I PP non devono fare riferimento a specifici prodotti realmente realizzati.

Alcuni esempi per chiarire meglio:

- *Protection Profile* per i *firewall*, per un sistema tipo utilizzato in ambiente commerciale, per sistemi multi-*user* basati su sistemi operativi di normale commercio o per un sistema di controllo accessi basato su regole predefinite.

- ***S**ecurity **T**arget* (**ST**) - Si riferisce ad uno specifico prodotto o sistema di cui si conoscono le specifiche di sicurezza, ed è un insieme organico di requisiti e specifiche di sicurezza associate ad uno specifico prodotto o sistema informatico a sua volta chiamato ***T**arget **O**f **E**valuation* (**TOE**) e che è oggetto di valutazione. Ad esempio:
  - *Security Target* per Oracle v7,
  - *Security Target* per il *firewall* XYZ, ecc.



Uso dei requisiti di sicurezza.

Tutti i requisiti di sicurezza (specifiche, descrizione, collegamenti, interdipendenze, ecc.) che si possono comporre nei PP e ST sono contenuti in un **Catalogo dei requisiti funzionali della sicurezza** [*Security Requirements*].

Allo stesso tempo i *Common Criteria* contengono i principi fondamentali per valutare i dispositivi di sicurezza dei prodotti o sistemi informatici Per ottenere ciò ci si avvale di un **Catalogo dei requisiti di affidabilità** [*Assurance Requirements*] strutturato in sette livelli di valutazione della affidabilità (EAL).

Tra i concetti chiave dei *Common Criteria*, vi sono quindi le tipologie dei requisiti:

- **Requisiti funzionali** - fondamentali per definire i comportamenti in materia di sicurezza dei prodotti e sistemi informatici. I requisiti effettivamente implementati diventano così funzioni di sicurezza.
- **Requisiti di affidabilità** - fondamentali per stabilire la fiducia che si può riporre nelle funzioni di sicurezza sia in termini di correttezza di implementazione sia in termini di efficacia di soddisfare gli obiettivi propri delle stesse funzioni di sicurezza.

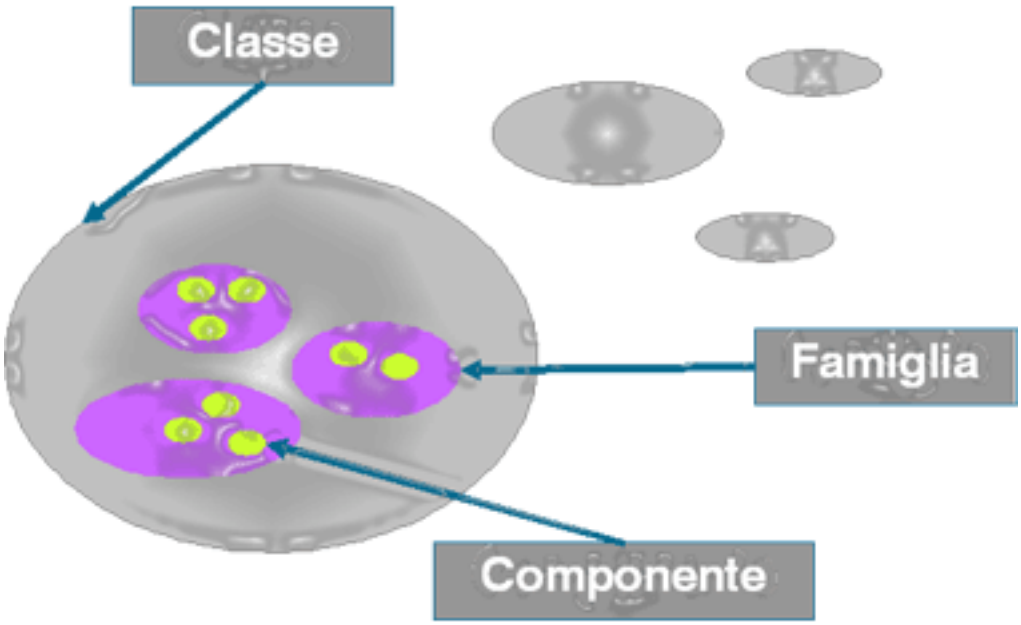
Un ST può includere uno o più *Protection Profile*.

Target Of Evaluation (TOE)

Il TOE è il prodotto o il sistema informatico oggetto della valutazione. Al TOE sono anche associati altri due elementi:

- La TOE *Security Policy* (TSP) - cioè l'insieme di regole che governano le modalità con cui i beni (*assets*) informatici sono gestiti, protetti e distribuiti all'interno del prodotto o sistema che è oggetto di valutazione.
- Le TOE *Security Functions* (TSF) - Sono considerate funzioni di sicurezza (TSF) tutte quelle parti del prodotto o sistema informatico oggetto di valutazione dalle quali dipende la garanzia della corretta esecuzione delle Politiche di Sicurezza (TSP).

Gerarchia



I componenti dei requisiti e delle funzioni di sicurezza e di affidabilità sono definiti e classificati secondo una gerarchia che prevede Classi, Famiglie e Componenti.

Le **Classi** sono un insieme organico di Famiglie che perseguono uno scopo comune. (per esempio: La classe *User Data Protection* o la Classe *Audit*).

Le **Famiglie** sono un insieme organico di Componenti che perseguono un obiettivo di sicurezza comune, ma che possono differire nel rigore o nell'intensità. (Per esempio: la famiglia *Access Contro Policy* raggruppa tutte le componenti che svolgono questa funzione).

Le **Componenti** sono l'insieme minimo e non divisibile che può essere utilizzato ed inserito nei PP o ST. (Per esempio: *Access Control*.1 raggruppa le funzioni di controllo accessi di livello 1).

Questa impostazione dovrebbe garantire flessibilità pur suggerendo di muoversi all'interno di requisiti predefiniti.

Classi funzionali dei requisiti di sicurezza

Ogni classe è individuata da una sigla di tre lettere che è ripetuta su tutte le famiglie e componenti che appartengono alla stessa classe.

FAU - La classe *Security Auditing* comprende il riconoscimento, la registrazione, la conservazione e l'analisi delle informazioni connesse con le più significative attività che coinvolgono la sicurezza. Le registrazioni così ottenute possono essere esaminate per determinare quali attività di sicurezza sono avvenute e chi le ha attivate.

FCO - *Communications* - Questa classe è formata da due famiglie col compito di assicurare l'identità delle parti che sono coinvolte nello scambio di dati. Queste famiglie hanno il compito di garantire l'identità di chi origina le informazioni trasmesse (*proof of origin*) e di garantire la identità di chi riceve le informazioni trasmesse (*proof of receipt*), assicurano inoltre che chi origina il messaggio non possa negare di averlo spedito ed il ricevente non possa negare di averlo ricevuto.

FCS - Questa classe fornisce le funzionalità crittografiche nel caso ciò sia richiesto, per soddisfare più severi obiettivi di sicurezza. Fanno parte di questa classe molte funzioni tra cui. Identificazione ed autenticazione, non-rigetto, percorsi e canali fidati, separazione dei dati, ecc.

FDP - Quattro sono le famiglie di questa classe che definisce i requisiti di protezione dei dati utente. Tali famiglie indirizzano la protezione dei dati utente all'interno del TOE, durante l'importazione e l'esportazione, nella fase di memorizzazione e ne gestiscono gli attributi di sicurezza.

FIA - Identificazione ed Autenticazione - Le famiglie appartenenti a questa classe indirizzano i requisiti connessi con le funzioni che stabiliscono e verificano l'identità degli utenti. Tale funzione è richiesta per garantire che ogni utente sia associato con un appropriato profilo di sicurezza (attributi di sicurezza - es. identità, gruppo di appartenenza, regole di riferimento, livello di riservatezza, ecc.). Una identificazione non ambigua degli utenti autorizzati ed una corretta associazione dei relativi attributi di sicurezza con quelli dei dati e oggetti informatici è un elemento critico per garantire che le politiche di sicurezza volute siano rispettate.

Le famiglie appartenenti a questa classe permettono di determinare e verificare l'identità degli utenti, determinarne la specifica autorità per interagire con il TOE secondo specifici profili di sicurezza.

FMT - *Security Management* - Questa classe è preposta a specificare le regole di gestione delle molteplici funzioni di sicurezza presenti nel TOE, inclusi gli attributi di sicurezza ed i dati essenziali al funzionamento delle stesse funzioni. Sono specificate differenti regole di gestione, di interrelazioni tra le funzioni e le aree di competenza. Questa classe si ripropone molteplici obiettivi:

- Gestione dei dati relativi al funzionamento delle stesse funzioni di sicurezza.
- Gestione degli attributi di sicurezza. Esempio: liste di accesso, liste delle potenzialità.
- Gestione delle funzioni di sicurezza. Esempio: selezione delle funzioni attivabili, regole o condizioni di funzionamento.
- Definizione delle regole di sicurezza.

FPR - Questa classe contiene i requisiti per la *Privacy*, intesa come protezione per ogni

utente contro la possibilità che un altro utente possa individuarne l'identità e farne un uso improprio. Le famiglie a disposizione sono:

- *Anonimity*: garantisce che un utente possa utilizzare una risorsa od un servizio certo che la propria identità non venga rivelata ad altri utenti.
- *Pseudonymity*: come nel caso precedente con in più la possibilità però di rendere conto (*accountable*) delle attività eseguite.
- *Unlinkability*: garantisce che un utente possa accedere più volte alle risorse ed ai servizi senza che altri utenti possano ricostruire questi passaggi.
- *Unobservability*: garantisce che un utente possa utilizzare una risorsa od un servizio senza che altri utenti possano osservare quale servizio o risorsa egli stia utilizzando.

FPT - Questa classe comprende famiglie di requisiti funzionali che fanno riferimento all'integrità e alla gestione dei meccanismi che compongono le funzioni di sicurezza del TOE e all'integrità dei dati delle stesse funzioni di sicurezza. Le famiglie di questa classe potrebbero apparire come una duplicazione della classe FDP (protezione dei dati utente) ed utilizzare anche gli stessi meccanismi.

Mentre le funzioni incluse nella classe FDP si occupano della sicurezza dei dati utente, le famiglie incluse nella classe FTP si occupano della protezione dei dati che sono essenziali al funzionamento delle stesse funzioni di sicurezza del sistema o prodotto oggetto di valutazione (TSF).

FRU - Questa classe fornisce tre famiglie che supportano la disponibilità delle risorse necessarie per il funzionamento del TOE, quali per esempio le capacità elaborative e/o di memorizzazione.
La famiglia *Fault Tolerance* fornisce la protezione contro la indisponibilità delle risorse elaborative per effetto di inconvenienti o guasti al TOE stesso.

La famiglia chiamata *Priority of Service* assicura che le risorse siano allocate alle attività più critiche o importanti e non vengano monopolizzate da attività a bassa priorità.
La famiglia *Resource Allocation* permette di porre dei limiti all'utilizzo delle risorse per evitare che un utente le monopolizzi.

FTA - Questa classe contiene le famiglie che disciplinano l'accesso allo stesso TOE definendo i requisiti per controllare l'esecuzione delle sessioni utente.

Le famiglie a disposizione sono:

- Inizio delle sessioni.
- Limitazioni nelle sessioni multiple concomitanti.
- Limitazioni negli scopi degli attributi di sicurezza utilizzabili.
- Blocco delle sessioni.
- Storia degli accessi.
- Simboli (*banners*) degli accessi.

FTP - Le famiglie comprese in questa classe forniscono i requisiti di sicurezza per un percorso fidato (*trusted*) di comunicazione tra gli utenti e le stesse funzioni di sicurezza (TSF) e per un canale fidato di comunicazione tra le TSF e gli altri prodotti informatici. Il percorso di comunicazione deve garantire che l'utente sia in comunicazione con le corrette TSF e che le TSF siano in comunicazione col corretto utente.

Classi dei requisiti di affidabilità

Le classi di affidabilità sono anch'esse codificate con una sigla di tre lettere che si ripete sulle famiglie che compongono la classe.

Di seguito una breve descrizione delle classi:

ACM - La gestione della configurazione aiuta a garantire che sia preservata la integrità del TOE richiedendo che esista adeguata disciplina e controllo dei processi di messa a punto e modifica del TOE e dei dati ad esso correlati. Tale disciplina deve prevenire che il TOE venga modificato, che ci siano aggiunte o cancellazioni di sue parti senza la dovuta autorizzazione. Inoltre il TOE deve essere fornito di appropriata documentazione sia per la sua distribuzione sia per la valutazione.

ADO - Questa classe definisce i requisiti per le misure, le procedure e gli standard che si riferiscono alla fasi di spedizione, installazione e di utilizzo sicuro del TOE garantendo che le protezioni di sicurezza offerte non siano compromesse durante le fasi di trasferimento, installazione, start-up e funzionamento.

ADV - Definisce i requisiti per la graduale messa a punto, nella fase di sviluppo, delle funzioni di sicurezza del TOE partendo dalle specifiche generali sino alla effettiva implementazione.

ADG - definisce i requisiti finalizzati alla comprensibilità, copertura e completezza della documentazione operativa fornita dallo sviluppatore. Questa documentazione, che è divisa in due categorie - una per gli utenti ed una per l'amministratore - è uno dei fattori chiave per la sicurezza dell'operatività del TOE.

ALC - definisce i requisiti per l'affidabilità attraverso l'adozione di un ben definito modello di ciclo di vita per tutti i passi dello sviluppo del TOE, inclusa la politica e le procedure per rimediare ai difetti, il corretto uso dei *tool*, le tecniche e le misure di sicurezza utilizzate per proteggere l'ambiente di sicurezza.

ATE - stabilisce i requisiti per i test che dimostrano come le funzioni di sicurezza soddisfino i requisiti funzionali del TOE. Vi fanno parte test di copertura, di profondità e funzionali, così come modalità di test indipendenti.

AVA - definisce i requisiti diretti alla identificazione dei punti deboli sfruttabili. In particolare quei punti deboli generati nel TOE nelle fasi di costruzione, di utilizzo, di uso improprio o configurazione non corretta.

APE - L'obiettivo della valutazione di un *Protection Profile* (PP) è di dimostrare che il PP è completo, consistente e tecnicamente corretto. Un PP già valutato può essere utilizzato come base per lo sviluppo di *Security Target* (ST). I PP già valutati possono essere inclusi in appositi registri a disposizione degli utenti.

ASE - L'obiettivo della valutazione di un *Security Target* (ST) è di dimostrare che il ST è completo, consistente, tecnicamente corretto e pertanto utilizzabile per essere utilizzato come base per la valutazione di un corrispondente TOE.

AMA - fornisce i requisiti che devono essere applicati dopo che un TOE sia stato certificato secondo i *Common Criteria*. Questi requisiti si ripropongono di garantire che il TOE, dopo la valutazione, continui a soddisfare i propri *Security Target* malgrado

possibili variazioni allo stesso TOE ed all'ambiente in cui è posto. Come cambiamenti vengono considerati la scoperta di nuove minacce o punti deboli e la correzione di errori di codifica.

Evaluation Assurance Levels

I livelli di valutazione dei CC sono 7 e vengono definiti con la sigla **EAL** (***E**valuation **A**ssurance **L**evels*). I livelli sono stati definiti in modo da essere (grossolanamente) confrontabili con gli equivalenti livelli dei TCSEC e ITSEC. La tabella riporta questa corrispondenza.

Livello	Nome	TCSEC	ITSEC
EAL1	*Functionally Tested*		
EAL2	*Structurally Tested*	C1 - *Discretionary security protection*	E1: *Informal architectural design*
EAL3	*Methodically Tested & Checked*	C2 - *Controlled access protection*	E2: E1 + *informal detailed design & test documentation*
EAL4	*Methodically Designed, Tested & Reviewed*	B1 - *Labeled security protection*	E3: E2 + *Source code or hardware drawing & evidence of testing*
EAL5	*Semiformally Designed & Tested*	B2 - *Structured protection*	E4: E3 + *Semiformal architectural design & formal model of security policy*
EAL6	*Semiformally Verified Designed & Tested*	B3 - *Security domains*	E5: E4 + *Correspondence between detailed design & source code*
EAL7	*Formally Verified Designed & Tested*	A1- *Verified design*	E6: E5 + *Formal description & detailed architectural design*

Di seguito una breve descrizione dei livelli di affidabilità:

Il livello **EAL1** è applicabile quando è richiesta una certa fiducia nella correttezza delle operazioni, ma le minacce alla sicurezza non appaiono serie. Ciò potrebbe avere senso dove viene richiesta una generica affidabilità della sicurezza per dimostrare che un minimo di attenzione sia stata posta nella protezione dei dati. EAL1 fornisce una valutazione del TOE così come viene reso disponibile ai clienti, inclusi i risultati di test indipendenti e un esame della documentazione di guida normalmente fornita. Si presuppone che la valutazione EAL1 si possa condurre con successo anche senza il coinvolgimento degli sviluppatori del TOE e con non troppa spesa. Una valutazione a questo livello dovrebbe fornire la prova che TOE funzioni così come descritto nella documentazione e che fornisce sufficiente protezione contro le minacce identificate.

Il livello **EAL2** richiede la cooperazione degli sviluppatori del TOE in termini di informazioni sui processi di spedizione e di *design* e risultati dei test, ma non dovrebbe richiedere agli sviluppatori uno sforzo più ampio di quello richiesto nella normale messa a punto di un buon prodotto. In altri termini non ci dovrebbero essere sensibili aggravi di costo e di tempi. EAL2 è applicabile in quelle situazioni dove gli sviluppatori o gli utenti richiedono un basso o moderato livello di affidabilità della sicurezza pur in mancanza di una completa documentazione di sviluppo. Tale circostanza si potrebbe avere nel caso di sistemi proprietari o nel caso non sia facile disporre della necessaria documentazione di sviluppo.

Il livello **EAL3** permette ad uno sviluppatore coscienzioso di raggiungere il massimo di affidabilità da una appropriata progettazione della sicurezza in fase di *design* senza comunque alterare in modo sostanziale le esistenti corrette procedure di sviluppo.
EAL3 è applicabile in quelle situazioni dove gli sviluppatori o gli utenti richiedono un moderato livello di affidabilità della di sicurezza senza, per condurre la valutazione, dover effettuare un sostanziale re-*engineering* del TOE stesso.

Il livello **EAL4** permette ad uno sviluppatore di raggiungere il massimo di affidabilità da una appropriata progettazione della sicurezza basata su un buon processo di sviluppo tra quelli disponibili in commercio che, basato sul rigore, non richieda specialisti con elevati *skill* o particolari conoscenze in materia di tecnologie di sicurezza. EAL4 è il livello più alto di affidabilità che probabilmente si potrà economicamente ottenere aggiustando prodotti già esistenti.

Il livello **EAL5** permette ad uno sviluppatore di raggiungere il massimo di affidabilità da una appropriata progettazione della sicurezza basata su un rigoroso processo di sviluppo tra quelli disponibili in commercio che preveda un utilizzo moderato di specialisti in tecniche di architettura di sicurezza. Tali TOE saranno probabilmente progettati e sviluppati col preciso intento di raggiungere il livello EAL5. è altamente probabile che per il fatto di dover soddisfare i requisiti di questo livello ci siano dei costi aggiuntivi che non dovrebbero comunque essere consistenti nel caso si utilizzino processi rigorosi di sviluppo senza ricorrere a specializzate tecnologie.
EAL5 è pertanto applicabile in quelle situazioni dove gli sviluppatori o gli utenti richiedano un elevato livello di affidabilità della sicurezza in uno sviluppo pianificato e condotto in modo rigoroso evitando di subire elevati e non ragionevoli costi a seguito dell'utilizzo di tecniche specialistiche di *engineering* di sicurezza.

Il livello **EAL6** permette agli sviluppatori di raggiungere un alto livello di affidabilità con l'utilizzo di specifiche tecnologie di sicurezza in rigorosi ambienti di sviluppo per produrre TOE di prima qualità nel caso si debbano proteggere beni informatici di alto valore contro rischi notevoli.
EAL6 è pertanto applicabile nei casi in cui si debbano sviluppare dei TOE per utilizzi in situazioni ad alto rischio dove il valore dei beni protetti giustifichi notevoli costi addizionali.

Il livello **EAL7** è applicabile allo sviluppo della sicurezza dei TOE per applicazioni in situazioni di estremo rischio e dove l'alto valore delle risorse da proteggere giustificherebbe gli elevati costi. L'applicazione pratica dell'EAL7 è attualmente limitata a quei TOE con funzionalità di sicurezza fortemente focalizzate che siano riconducibili ad un'analisi formale estesa.

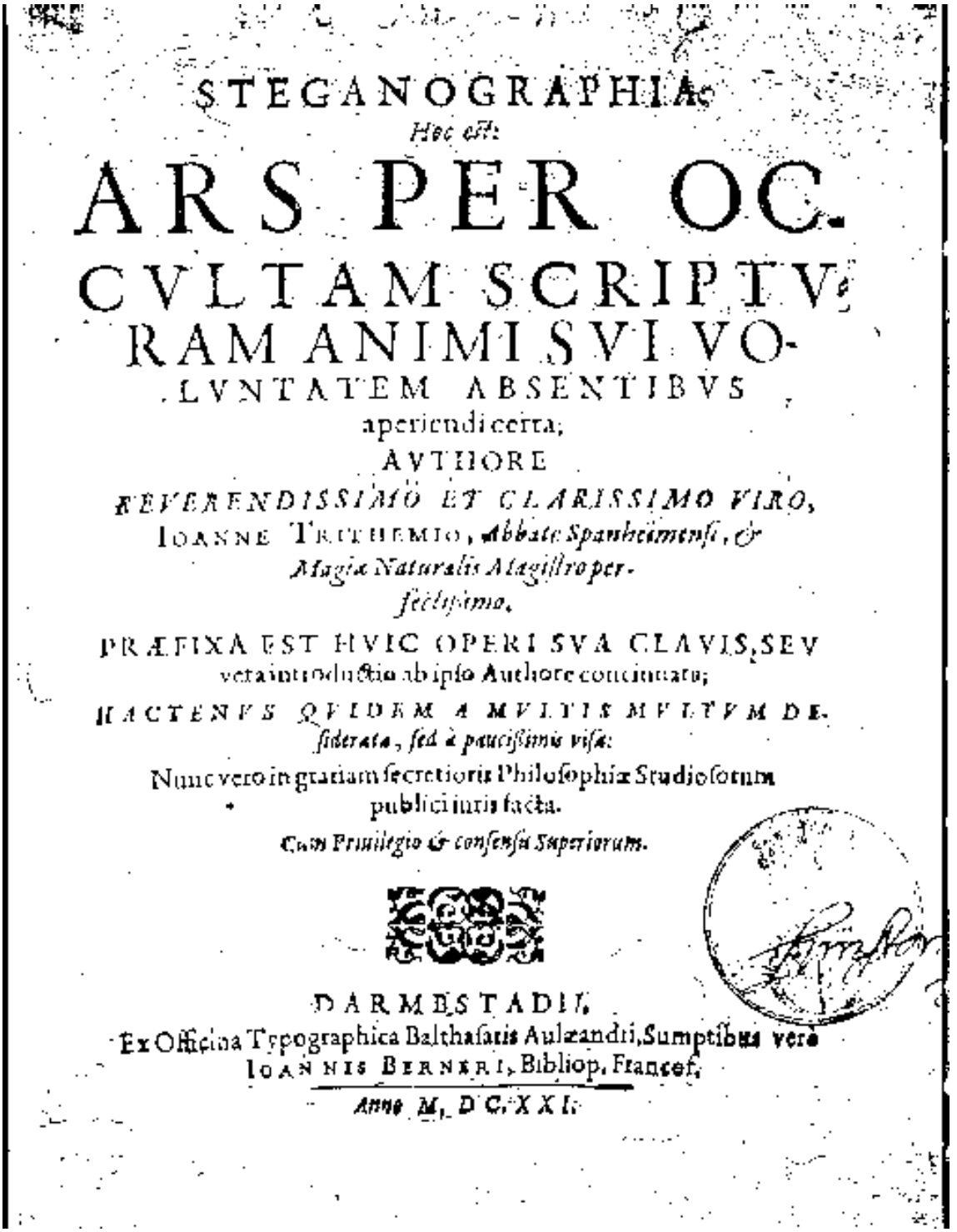
Tecniche di crittografia

Franco Callegati
Paolo Zaffoni

8.4.1 (Spiegare i principali aspetti della sicurezza connessi alla trasmissione dei dati), 8.4.2 (Descrivere gli attuali standard di crittografazione: chiavi pubbliche e private, NSA, DES, PGP)

Crittografia e firma digitale

La parola crittografia ha origine greca e significa nascosto. Un'altra parola correlata è **steganografia** che significa scrittura nascosta. Un esempio legato all'antichità è di scrivere messaggi segreti non sull'argilla che ricopriva le tavolette, ma sulle stesse tavolette che venivano poi ricoperte d'argilla e sembravano non usate. Della steganografia l'ababe Tritemio (1500 d.C.) è forse uno dei più noti autori.



STEGANOGRAPHIAE
Hoc eſt:
ARS PER OC.
CVLTAM SCRIPTV
RAM ANIMI SVI VO-
LVNTATEM ABSENTIBVS
aperiendi certa;
AVTHORE
REVERENDISSIMO ET CLARISSIMO VIRO,
IOANNE TRITHEMIO, *Abbate Spanheimenſi, &*
Magiæ Naturalis Magiſtro per-
feciſſimo.
PRÆFIXA EST HVIC OPERI SVA CLAVIS, SEV
vera introductio ab ipſo Authore concinnata;
HACTENVS QVIDEM A MVLTIS MVLTVM DE-
ſiderata, ſed à paucisſimis viſa:
Nunc vero in gratiam ſecretioris Philoſophiæ Studioſorum
publici iuris facta.
Cum Priuilegio & conſenſu Superiorum.

DARMESTADII,
Ex Officina Typographica Balthaſaris Aulæandri, Sumptibus vero
IOANNIS BERNERI, Bibliop. Francof.
Anno M. DC. XXI.

Quindi la trasformazione del messaggio al fine di renderne incomprensibile il significato, è stato, è e sarà lo stratagemma tramite il quale raggiungere uno degli obbiettivi sella sicurezza: la **confidenzialità**.

In particolare definiamo:

- **Crittologia**: disciplina che tratta delle scritture segrete, dei documenti in cifra.
- **Crittografia**: insieme delle tecniche che consentono di realizzare la cifratura di un testo e la decifratura di un crittogramma.
- **Crittoanalisi**: disciplina che studia come forzare i cifrari.

Sicurezza dell'informazione

Il concetto di informazione viene associato ad una quantità ben definita. Per introdurre la crittografia, è necessario comprendere preliminarmente le caratteristiche relative alla sicurezza dell'informazione.

La sicurezza dell'informazione si manifesta in molti modi a seconda delle situazioni e dei requisiti. In ogni caso, le parti di una transazione devono essere rassicurate che determinati scopi della *information security* siano raggiunti. Abbiamo elencato i principali obbiettivi nella tabella seguente.

privacy o confidenzialità	mantenere segrete le informazioni a tutti tranne ai coloro autorizzati a vederle.
integrità dati	garantire che le informazioni non siano alterate tramite mezzi non autorizzati sconosciuti.
autenticazione o identificazione di entità	verifica dell'identità di una entità (es., persona, *computer*, carta di credito, ...).
autenticazione dei messaggi	verifica della sorgente delle informazioni.
firma	un mezzo per collegare inscindibilmente l'informazione ad una entità.
autorizzazione	convenienza, ad un'altra entità, di poter eseguire una operazione.
validazione	un mezzo per fornire linee temporali di autorizzazione all'utilizzo o manipolazione di informazioni o risorse.
controllo d'accesso	restringere l'accesso alle risorse per le entità privilegiate.
certificazione	certificazione di informazione tramite entità di fiducia.
timestamping	registrazione dell'istante di creazione o esistenza dell'informazione.
witnessing	verificare la creazione o l'esistenza di informazioni tramite una entità differente dal creatore della stessa.
receipt	riscontro che l'informazione è stata ricevuta.
confirmation	riscontro che il servizio è stato fornito.
ownership	un mezzo per fornire ad una entità il diritto legale di utilizzare o trasferire risorse ad altri.
anonimato	nascondere l'identità di una entità coinvolta in un processo.
non ripudio	prevenire la negazione di una precedente azione.
revoca	revoca di un certificato o autorizzazione.

Nel secolo appena trascorso, sono stati sviluppati insiemi di protocolli e meccanismi molto elaborati ai fini della sicurezza delle informazione quando questa era veicolata su documenti fisici.

Spesso l'obiettivo della *information security* non può essere garantito solo da algoritmi matematici e protocolli, ma richiede tecniche procedurali e legali per risaltare il risultato desiderato. La sicurezza fisica del contenitore dell'informazione è, per necessità pratica, limitata e quindi a, in alcuni casi, la sicurezza viene procurata attraverso il documento che ospita l'informazione e non dall'informazione stessa. Basti pensare alla

carta-moneta, che richiede inchiostri speciali e materiali per prevenire la contraffazione.

Obiettivi

Prima di iniziare la reale trattazione dell'argomento crittografia, sarà utile puntualizzare quali siano gli scopi e gli obbiettvi da raggiungere con tale strumenti.

La figura evidenzia gli obiettivi direttamente raggiungibili con gli strumenti crittografici, e dai quali trarre spunto per ottenere la realizzazione degli scopi di protezione dei sistemi, quindi delle applicazioni e dei sistemi.

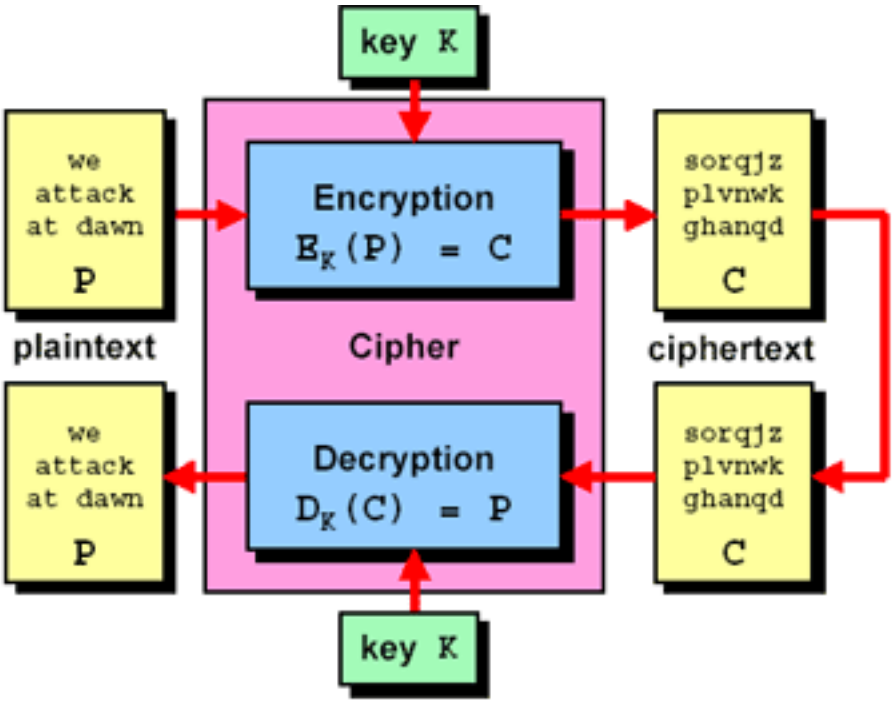
Terminologia

Presentiamo una serie di terminologie attuali rispetto alle tecniche di cifratura:

- Un messaggio è un ***plaintext*** (o ***cleartext***), cioè testo in chiaro. Il processo di alterazione di un messaggio in un qualche modo al fine di nasconderlo, in sostanza è definito cifratura (*encryption*).
- Un messaggio è cifrato o ***ciphertext***. Il processo che restituisce li *plaintext* viene chiamato decifratura (***decryption***).

E riguardo algoritmi e chiavi:

- un algoritmo crittografico (***cipher***), è la funzione matematica utilizzata per la cifratura e decifratura(***encryption*** e ***decryption***).
- La sicurezza di un algoritmo di cifratura moderno è basata sulla chiave segreta. Questa chiave può essere scelta in un insieme di valori. Il campo dei possibili valori delle chiavi viene definito spazio delle chiavi (***keyspace***).
- Sia la cifratura che la decifratura sono operazioni che dipendono dal valore della chiave K e ciò è denotato dal fatto che K è espresso nella funzione EK(P) = C e DK(C) = P.



Considerazioni storiche 1

A volte il confine tra **lecito ed illecito** risulta essere molto sottile. Ci si potrà trovare di fronte alla necessità di scambiare informazioni confidenziali con altri o anche semplicemente di conservare dati in possesso in modo che solo noi si possa accedervi.

Il problema per chi utilizza la crittografia convenzionale è che di fronte all'esigenza di scambiare informazioni con altri in un canale definibile insicuro (come è Internet), non si hanno a disposizione altri canali sicuri (potrebbe essere ad esempio un appuntamento di persona in un luogo riservato) per comunicare quale è la parola chiave o altre informazioni che permettano l'accesso al *file* cifrato.

Grazie a un metodo introdotto pubblicamente (prima era sviluppato soltanto su commissione da società specializzate) da *Philip Zimmermann*, tale inconveniente è stato ovviato.

Zimmermann ha avuto la brillante idea di sviluppare un programma gratuito che permettesse a tutti di utilizzare un metodo nuovo unito ad una elevata sicurezza che la lunghezza delle chiavi offre, usando la crittografia a **chiave pubblica**. è questo una dei fattori che hanno contributo alla diffusione così capillare del noto programma ***P**retty **G**ood **P**rivacy*, chiamato comunemente **PGP**.

Il punto cruciale su cui si basa il pensiero di *Zimmermann* è il seguente:

premesso che la privacy è un diritto dell'uomo che sta alla base delle società

tecnologicamente evolute e deve essere tutelato, accade sempre più frequentemente con l'introduzione di nuove tecnologie che tale diritto non venga riconosciuto; è indispensabile quindi utilizzare un metodo per garantire la riservatezza, sia che si utilizzi la crittografia per spedire gli auguri di buon anno, piuttosto che un manuale su come costruire una bomba atomica e PGP è un programma che permette di farlo.

Criminalizzare l'utilizzo di Internet abbinato alla crittografia perchè non controllabile sarebbe come vietare di utilizzare il telefono come mezzo per organizzare attività illecite; sta all'utente in ogni caso deciderne l'uso.

Considerazioni storiche 2

La crittografia come modifica volontaria del testo esisteva già al tempo degli egiziani nel 1900 a.C. (tomba del faraone Knumotete II).

AVANZARE TRA DUE GIORNI
ALL ALBA VERSO IL FIUME

ADRARI ZAOLEF NRILVL ATGAAIEVEEIBOMARUNLSU

Gli Spartani per cifrare un messaggio segreto di tipo militare usavano, 2500 anni fa, una striscia di papiro avvolta a spirale attorno ad un bastone (che costituirà la chiave di decodifica). Una volta scritto il messaggio in verticale sul papiro questo veniva consegnato al destinatario che, con un bastone dello stesso diametro poteva leggere il messaggio in chiaro. Questo metodo è di **trasposizione** perchè il messagggio è in chiaro ma l'ordine delle lettere è da scoprire.

Se ne trovano altre tracce a partire da alcuni scritti storici riguardanti Giulio Cesare (*Vite dei dodici Cesari* di Svetonio) che in luogo di ogni lettera scriveva quella situata tre posizioni oltre nell'alfabeto: **sostituzione**. Una crittografia così semplice si decifra facilmente: si calcola la frequenza delle lettere usate e si confronta con quella delle lettere nella lingua che si suppone usata nel testo in oggetto. Se il testo non è molto breve, si riconoscono subito le equivalenze di tre o quattro lettere più frequenti. Poi si fanno ipotesi e controlli successivi e presto si arriva alla conclusione. Le frequenze delle lettere variano da un testo all'altro, ma si spostano di poco nell'ordine decrescente.

Esempio: pippo con chiave 3 diventa snssr.

Questo metodo è facilmente attaccabile perchè basta confrontare la frequenza delle lettere nella lingua italiana con la frequenza dei simboli usati nel messaggio cifrato. Bisogna inoltre considerare che le chiavi possibili sono solo 26, quindi con un *brute force* si potrebbe scovare la chiave. Data la bassa complessità del metodo usato da Cesare è chiaro che non fosse infallibile, ma dati i risultati militari è stato efficace! Il metodo di Cesare ha ispirato un sistema usato ancora oggi, il ROT-13 dove la chiave è appunto 13, quindi A->N, B->O, etc.

Il metodo che usavano gli ebrei è detto ATBASH. La sostituzione avviene utilizzando questa tabella dove le lettere della seconda riga sono scritte in ordine decrescente:

a b c d e f g h i j k l m n o p q r s t u v w x y z
z y x w v u t s r q p o n m l k j i h g f e d c b a
Messaggio: Il Libro di Geremia
Testo Cifrato: Ro Oryil wr Tvivnrz

Lo storico greco Polibio sviluppò una tecnica di codifica legando le lettere a una coppia di numeri che ne indicava la posizione in una tabella. La coppia di numeri era comunicata nella notte attraverso delle torce. Ecco un esempio di tabella:

	1	2	3	4	5
1	a	b	c	d	e
2	f	g	h	ij	k
3	l	m	n	o	p
4	q	r	s	t	u
5	v	w	x	y	z

pippo diventa (3,5) (2,4) (3,5) (3,5) (3,4)

Se la disposizione delle lettere nella tabella non seguono l'ordine alfabetico si capisce la difficoltà di trovare la chiave che in questo caso è la tabella.

L'imperatore romano Augusto usava invece un altro interessante metodo di sostituzione usando come chiave un'altra parola o frase. La chiave e il testo avevano un corrispettivo numerico, il testo cifrato risultava una sfilza di numeri ottenuti come somma fra testo e chiave. Se la somma (valore cifrato) eccede 21 si ricomincia dalla a; ciò equivale a fare una somma modulo 21. La decifrazione è una semplice sottrazione. La crittoanalisi di questo metodo non beneficia della frequenza delle lettere della lingua usata. Questo metodo può essere attaccato con un *brute force* utilizzando un dizionario di parole. Ovviamente ci si può difendere non usando come chiave parole di senso compiuto, ma un insieme di lettere generate in maniera casuale.

Considerazioni storiche 3

Dopo il fiorire dell'arte della cifratura in Medio Oriente, nel rinascimento si torna in Occidente per dare supporto al rifiorire degli scambi commerciali internazionali e alle innumerevoli tresche politiche, soprattutto in Italia. Lo sforzo propinato nel cifrare i documenti, era bilanciato da quello per intercettare e ricostruire i messaggi di terzi.

Possiamo enumerare il primo trattato noto sulla crittologia di Cicco Simonetta, presso la cancelleria degli Sforza nel XV secolo. Sempre in tale epoca, **Leon Battista Alberti**,

architetto letterato e crittologo dilettante, scrisse un trattato dal nome *De Cifris* nel quale introdusse il suo **sistema polialfabetico** che per tre secoli, seppur attribuito ad altri autori, costituì il basamento dei sistemi crittografici. Il suo sistema usa diversi alfabeti spostati rispetto a quello del testo originario saltando dall'uno all'altro ogni due o tre parole. L'alfabeto da usare viene indicato da una lettera in chiaro che, da quel punto in poi, mittente e destinatario impostano in modo da fissare una certa posizione mutua fra due cerchi concentrici ruotanti, che riportano l'intero alfabeto. Lo strumento è costituito da due dischi che ruotano l'uno sull'altro. Sul disco esterno sono riportati numeri e lettere normali, su quello interno i relativi segni cifrati. Per creare una chiave, si ruota il disco e si fa corrispondere una lettera M e un'altra lettera prestabilita.

Altro esempio di cifratura con il sistema del polialfabeto è quello inventato da un tedesco contemporaneo dell'Alberti: **Johannes Trithemius**. Il sistema fa uso di una tabella che si chiama *tabula recta*, formata da 26 righe (tante quante le lettere dell'alfabeto inglese) riportanti ognuna un alfabeto scalato di una posizione rispetto a quello precedente. La tabella si usa così: la prima lettera da cifrare rimane la stessa, la seconda si cifra con il secondo alfabeto, la terza lettera userà il terzo alfabeto e così via fino a ricominciare dal primo alfabeto dopo la ventiseiesima lettera. Per rendere difficile il lavoro dei crittoanalisti si può usare un alfabeto disordinato o, meglio ancora (è più facile da ricordare e comunicare al destinatario del messaggio) una frase chiave.

Nel 1553, un altro famoso interprete di questa arte era **Giovan Battista Bellaso**, che pubblicò una serie di cifrari, uno dei quali ripreso all'inizio del secolo odierno per l'uso con le telescriventi. Il più noto sistema di Belaso prevede di usare una parola anziché una lettera; questa parola va scritta al di sopra del testo da crittografare. Ogni singola lettera della chiave determina un alfabeto circolare che inizia da quella stessa lettera. Se quindi la prima lettera sarà una V, l'alfabeto corrispondente sarà Vwxyzabcd.... Per cifrare quindi si dovrà vedere quale posizione occupa la lettera nell'alfabeto normale (es. una E occupa il quinto posto) e sommarla nell'alfabeto circolare creato. Questa sarà la lettera da scrivere nel messaggio cifrato.

Se ogni volta si cambia la chiave generandola casualmente, stiamo usando un metodo chiamato *One-Time Pad*, che è difficile da superare senza conoscere la chiave e generando in maniera davvero casuale la chiave (e non è facile perchè si può ottenere solo osservando fenomeni casuali naturali). Occorre notare che non bisogna usare questo metodo per due diversi messaggi con la stessa chiave perchè la differenza tra testo cifrato e testo in chiaro è uguale e, in unione a un *brute force*, aiuta di parecchio la crittoanalisi di questo metodo.

Considerazioni storiche 4

L'uso della crittografia continua intensificandosi sempre di più e migliorandosi con il tempo fino ad avere importanza tale da cambiare il corso della storia durante le due guerre mondiali quando appaiono le prime macchine elettriche per cifrare i messaggi ma, soprattutto, per la crittoanalisi.

I tedeschi usarono per tutta la seconda guerra mondiale una macchia chiamata **Enigma** che avrebbe dovuto cifrare i messaggi in maniera sicura. Così non successe, perchè inglesi e polacchi unendo le loro forze furono in grado di decifrare quasi tutti i messaggi intercettati. L'Enigma era una macchina elettromeccanica con contatti, lampadine, rotori e una tastiera. Ogni lettera veniva cifrata con un alfabeto diverso

dando luogo ad un numero così elevato di combinazioni da rendere la decodifica teoricamente impossibile per l'epoca. Ma ciò non fermò gli inglesi che trassero grande beneficio dai messaggi decodificati.

	V.24/V.28	V.35	V36
meccaniche	ISO 2110 o 4902	ISO 2593	ISO 4902
elettriche	V.28	V.10 eV.11.	V.10 e V.11
funzionali	V.24	V.24	V.24
procedurali	V.24, V.25 o V.25 bis		

	X.20	X.20bis	X.21	X.21 bis
meccaniche	ISO 4903	ISO 2110	ISO 4903	ISO 2110
elettriche	X.26 o X.27	V.28	X.26 o X.27	V.28
funzionali	X.24	V.24	X.24	V.24
procedurali	X.20	X.20bis	X.21	X.21 bis

Algoritmi semplici e chiave lunghe ...

Per anni la regola generale è stata di creare algoritmi semplici e di impiegare chiavi molto lunghe per rendere difficile la vita al crittoanalista.

... oppure viceversa

Oggi l'orientamento è opposto vista la potenza di calcolo di cui si può disporre per fare un *brute force*, quindi si creano algoritmi complicatissimi da decifrare in modo che anche se il nostro avversario avesse parecchio materiale su cui condurre un'analisi, gli sarebbe pressocchè inutile.

Oggi la crittografia è utilizzata per il commercio elettronico, l'*autenticazione*, la *riservatezza* delle informazioni, l'*integrità* etc. Uno dei presupposti fondamentali è che si suppone che il crittoanalista di turno conosca in generale il nostro metodo di crittografia, questo perchè sarebbe davvero un disastro cambiare metodo di crittografia ogni qual volta si ha il sospetto che qualcuno sia riuscito a infrangerlo. Da questo presupposto segue che i metodi basano la loro forza sulle chiavi di codifica e decodifica.

Se la chiave è la stessa sia per la codifica che per la decodifica ricadiamo nel caso delle crittografia classica: questi sono i metodi a **chiave simmetrica** o segreta.

Gli algoritmi a **chiave asimmetrica** o pubblica (che risalgono agli anni '70) utilizzano coppie di chiavi complementari. Una delle due chiavi è pubblica e conosciuta da tutti. Le chiavi vanno distribuite a coppie e quindi solo una chiave può decifrare il messaggio generato utilizzando la chiave a lei complementare. In questo modo possiamo trasmettere tranquillamente la nostra chiave pubblica senza paura che venga intercettata. In ogni modo l'argomento verrà approfondito nell'apposito **paragrafo**.

Spesso ci si orienta per metodi ibridi simmetrici-asimmetrici (come succede nel famoso PGP) perchè il solo metodo asimmetrico non è efficiente (è lento!) per grandi moli di dati e, se dovessimo inviare lo stesso messaggio a più persone, dovremmo cifrarlo ogni volta con la giusta chiave.

Cifratura simmetrica

La sicurezza di un crittosistema, o cifrario, deve dipendere solo dalla segretezza della chiave e non dalla segretezza dell'algoritmo usato.

Jean Guillaume Hubert Victor Francois Alexandre Auguste Kerckhoffs von Nieuwenhof (1835-1903), filologo olandese, La Crittographie Militaire [1883]

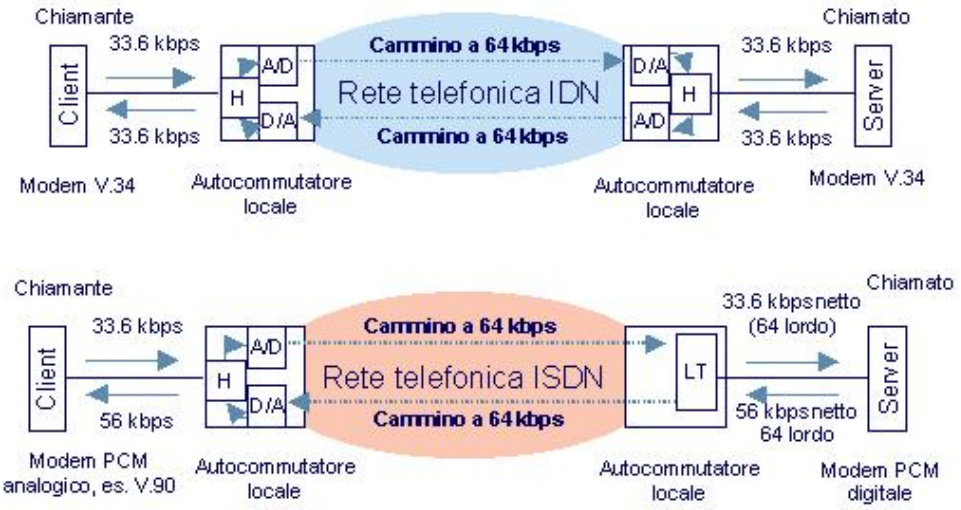
Il crittoanalista conosce sempre il crittosistema che è stato usato e gli algoritmi che il sistema prevede. Solo la chiave è segreta.

La crittografia moderna si incentra prevalentemente su algoritmi basati su sostituzioni, scambi fatti in relazione a tabelle. Tuttavia, durante il passaggio dagli scambi monoalfabetici alla crittografia moderna, è stata ampliamente utilizzata (e lo è ancora) la crittografia mediante l'utilizzo di cifrari polialfabetici. Anche questi tramontarono quando si comprese che, analizzando le statistiche delle ripetizioni di caratteri nel testo ed in seguito a vari tentativi, si poteva poco a poco ottenere la chiave segreta procedendo per tentativi.

Questa tipologia di crittoanalisi ha senso se le chiavi rimangono di lunghezza inferiore al testo da cifrare (diversi decenni fa era inutile e difficile di trasportare chiavi di lunghezza superiore o uguale al testo, ora le cose sono cambiate). Attualmente possono essere ancora usati gli algoritimi polialfabetici se dipendenti da una chiave segreta di lunghezza infinita (matematicamente e intuitivamente è dimostrata la loro inattaccabilità); una **chiave di lunghezza infinita**, magari generate in modo pseudo-casuale in modo dipendente da una *password*, non serve se più lunga del testo da cifrare.

Chiave simmetrica

La cifratura a chiave simmetrica, richiede che mittente e destinatario di una determinata comunicazione utilizzino la stessa parola chiave per decifrare il messaggio in oggetto. Algoritmo e chiave sono le due componenti principali di ogni sistema di crittografia, componenti che permettono il passaggio dal messaggio in chiaro al messaggio cifrato e viceversa. Esistono due tipi di 'crittosistemi' che si basano su chiavi o codici fondamentalmente diversi tra loro. Vengono definiti crittosistemi a **chiave segreta o simmetrica** (**secret-key**) e **chiave pubblica** (***public-key***).



Nel primo sistema viene usata una sola chiave, detta appunto segreta, utilizzata come parametro di una funzione univoca e invertibile permettendo così di elaborare il testo del messaggio da trasmettere rendendolo incomprensibile agli intercettatori. Essendo la funzione invertibile, il destinatario dovrà soltanto elaborare nuovamente il crittogramma richiamando l'inversa della funzione di cifratura avente come parametro la stessa chiave utilizzata dal trasmettitore del messaggio.

La tecnica si basa sulla capacità del mittente e del destinatario di mantenere segreto il codice di cifratura. Tale metodo, noto da secoli, è definito crittografia simmetrica mentre la crittografia a chiave pubblica, relativamente recente in quanto risale agli anni '70, viene anche definita crittografia asimmetrica. Quest'ultima a differenza della precedente, utilizza due chiavi distinte: una per cifrare il messaggio e l'altra per decifrarlo.

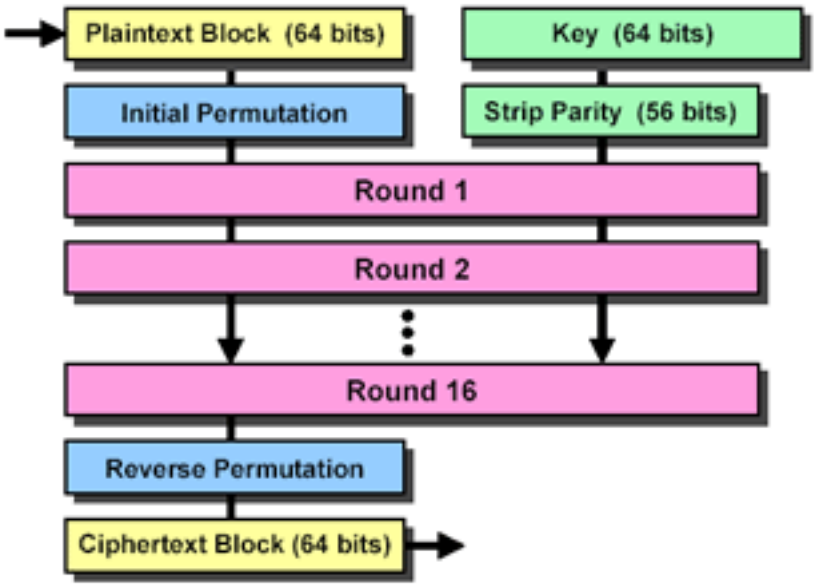
Con il sistema a chiave simmetrica il mittente e il destinatario devono raggiungere un accordo sulla scelta della chiave.
A rendere più complessa l'operazione è il fattore distanza, tipicamente tale da impedire lo scambio di persona della chiave. Come potranno allora scambiarsi la chiave?

Il problema esposto non risulta di semplice soluzione, poiché implica il coinvolgimento di una terza parte che ha il compito di distribuire la chiave accordata. Quest'ultima soluzione diviene presto improponibile nel momento in cui a fare uso di tale metodo è una struttura dalle dimensioni notevoli. Allora si sostituiscono le consegne con terze parti con trasmissioni cifrate con una chiave principale contenenti la chiave da trasmettere (chiave di sessione). Ancora possiamo dire che non vi è sufficiente sicurezza con questo metodo.

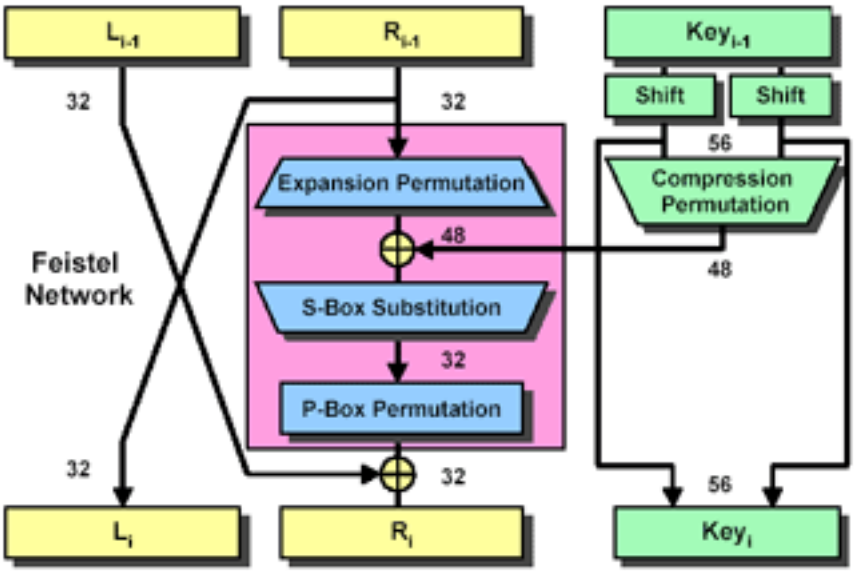
DES, Data Encrytion Standard

Standard federale ancora oggi ufficiale (nella versione triplo-des) per gli USA, è nato nel 1977 per implementazioni per lo più *hardware* come derivazione di *Lucifer*, un algoritmo di IBM nato nel '70, su insistenza del *National Bureau of Standard* per difendere dati riservati ma non segreti militari.

Il DES brevettato nel 1976 da IBM è *royalty-free* dal 1993. Il DES è un codice cifrato a blocchi. Si dice che un codice è cifrato a blocchi quando si applica un codice cifrato a un bit, *byte*, parola o gruppi di parole alla volta. Il blocco che si usa per crittografare è di 64 bits (8 sottoblocchi da 8 bit). Dato che l'ultimo bit di ogni sottoblocco è di controllo, i bit utili sono 56.



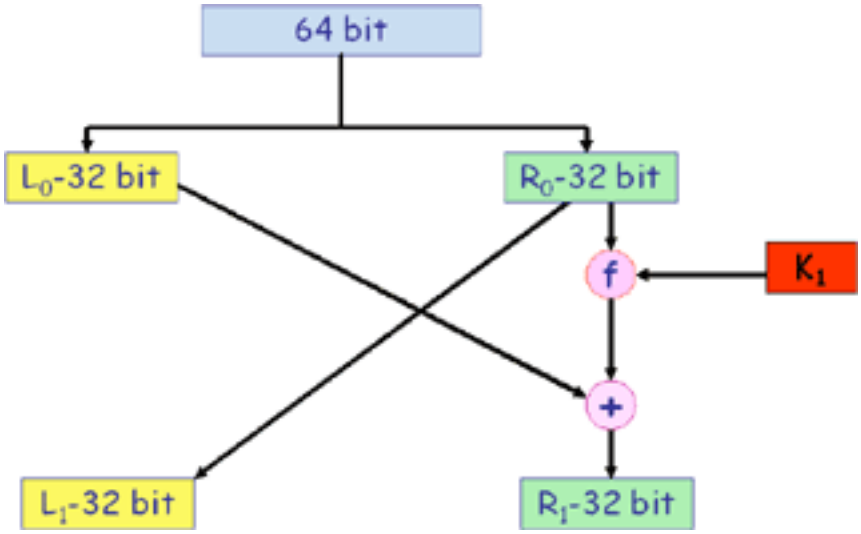
Per cifrare il testo si divide in blocchi da 64 bit che sono cifrati in successione. Se un messaggio non riempie i 64 bit si può completare in diversi modi: si possono aggiungere zeri, si possono aggiungere bit random specificando nell'ultimo quanti se ne aggiungono, etc.



Il nucleo di ogni ciclo DES è la *Feistel network*, così chiamata dal nome dello scienziato della IBM *Horst Feistel*. I 64 bit di *plaintext* vengono divisi (***split***) in destra (**R**) e sinistra (**L**) di 32 bit ciascuno. La parte destra elabora *output* del lato sinistro alla fine del ciclo,

ed il lato sinistro entra in una *black box* ove prima viene espanso a 48 bit da una permutazione di espansione e successivamente passato in uno XOR con una chiave da 48 bit.

La somma risultante entra in una matrice di 8 ***S-box*** con 6 linee di *input* e 4 di *output* ognuna, producendo un risultato di 32 bit che verrà permutato dalla ***P-box***. Il risultato della *black box* passa per uno XOR con la metà sinistra dell'*input* iniziale e diviene la metà destra per il seguente ciclo.



Ogniuno dei 16 cicli DES ha una chiave da 48 bit, derivata dallo scorrimento (***shift***) e dalla permutazione dell'intera chiave da 56 bit ciclo per ciclo.

Il DES è molto usato in ambito commerciale perchè, i numerosi passaggi che comprende, sono relativamente semplici (XOR, sostituzioni e permutazioni).

Occorre ricordare che il DES cambia solo la chiave; questo porta vantaggi economici immediati, ma appena verrà scoperto il modo per forzarlo (senza *bruteforce*) occorrerà cambiare radicalmente tutto. Un altro difetto fondamentale è lo spazio limitato delle chiavi pari a 2^56. Per ovviare al problema si tenta di allungare le chiavi o di applicare più volte il DES (triplo-DES o TDES).

Il progetto originale dell' IMB per il DES prevedeva una chiave più lunga dei 56 bits usati di *default*. Probabilmente il progetto originario fu influenzato dall'NSA che impose all'IBM una chiave sicura, ma comunque alla portata dei loro mezzi.

Modalità di utilizzo

Ovviamente non è sufficiente fornire un algoritmo di cifratura e decifratura, ma è anche necessario fornirne le modalità di utilizzo. Questo si rende necessario in quanto DES è un algoritmo di cifratura che utilizza blocchi di 64 bit e una chiave a 56 bit. In generale si dovranno cifrare messaggi di lunghezza arbitraria.

Il comitato ANSI, nel documento ANSI X3.106-1983 (*Modes of Use*) ha stabilito per DES 4 modi di utilizzo:

- Due di tipo A Blocchi;
- Due di tipo A Flusso.

Electronic Code Book [Block Mode]

In questa modalità di utilizzo ogni messaggio più lungo di 64 bit viene spezzato in blocchi di 64 bit che poi vengono cifrati.
Può accadere che l'ultimo blocco di bit sia più breve di 64 bit: quando questo avviene su questo blocco si applica il *Padding* ossia si completa il blocco introducendo zeri fino a raggiungere la lunghezza corretta.

Questo modo di utilizzo presenta, ovviamente, debolezze:

- Ripetizioni nel messaggio in chiaro possono riflettersi nel testo cifrato:
  - Se allineate con i blocchi del messaggio.
  - Con particolari tipi di dati - ad esempio la grafica.
  - Se il messaggio risulta essere molto costante si presta alla crittoanalisi statistica.
- Una debolezza è data dalla indipendenza reciproca dei blocchi di messaggio cifrato.

Cipher Block Chaining (CBC)[Block Mode]

Questo metodo utilizza il risultato di uno step di cifratura per modificare l'*input* del successivo step. Ciò implica un vantaggio ed uno svantaggio.

Il vantaggio è rappresentato dal fatto che ogni blocco di cifratura è dipendente da tutti quelli che lo precedono. Lo svantaggio è rappresentato dal così detto effetto valanga - *avalanche effect* - ossia la modica di un bit in un blocco si ripercuote su tutti i blocchi a questo susseguenti.
Per cifrare il primo blocco si utilizza un Vettore di Inizializzazione (IV) con il quale si cifra il primo blocco. Ovviamente il vettore di inizializzazione deve essere conosciuto sia da chi cifra che da chi decifra. Quindi il problema è, nel caso di messaggi che viaggiano su rete, l'inviare il vettore di inizializzazione. Ovviamente anche mediante questo modo di utilizzo può essere necessario espandere la dimensione dell'ultimo blocco: il metodo da utilizzarsi è esattamente il medesimo visto per l'utilizzo *Electronic Code Book*.

Cipher FeedBack (CFB) [Stream Mode]

Nel caso in cui si debba operare su dati bit o *byte oriented*, ossia su flussi di dati, tipicamente *file*, è necessario cambiare approccio di cifratura.
L'approccio di cifratura realizzato dal CFB è il seguente:

- Si cifra il Vettore di Inizializzazione IV.
- Del IV cifrato si scartano i 56 bit meno significativi.
- Con il *byte* più significativo dell'IV cifrato si cifra il primo *byte* del *PlainText*.
- Si opera uno *shift* a sinistra di IV di un *byte*.
- Nello spazio così creato si inserisce il primo *byte* cifrato del *PlainText*.
- Si tratta il vettore così ottenuto come se fosse un nuovo vettore di inizializzazione e si reitera il processo fino ad esaurimento del flusso dati.

La decifratura è molto simile a quella utilizzata nel DES Classico. Per decifrare è sufficiente infatti invertire una parte del processo, avendo cura di utilizzare il DES

sempre in modalità cifratura.

Output FeedBack (OFB) [Stream Mode]

Questo tipo di implementazione è superficialmente simile alla implementazione CFB. La differenza risiede nel fatto che il *feedback* non è il risultato della cifratura del *PlainText*, ma è il risultato della cifratura dell'IV. In questo modo si ottengono due cose:

- Indipendenza della cifratura dal messaggio - non sono più possibili attacchi crittoanalitici statistici.
- Si riesce ad eliminare del tutto l'effetto valanga.

Crittoanalisi del DES

A causa della crescente attenzione e dello sviluppo in tema di crittoanalisi, il DES ed il 3DES sono stati abbandonati. Le tecniche di crittoanalisi possono essere raggruppate come segue:

Ciphertext-Only Attack

- L'attaccante conosce il cifrato (*Ciphertext*) di diversi messaggi cifrati con la stessa chiave o con diverse chiavi.
- Recupera il testo chiaro (*plaintext*) di più messaggi o meglio cerca di dedurne la chiave (o le chiavi).

Known-Plaintext Attack

- Si conosce la coppia *ciphertext / plaintext* di diversi messaggi.
- Viene dedotta la chiave o un metodo per decifrare messaggi futuri.

Chosen-Plaintext Attack

- L'attaccante può scegliere il *plaintext* che è stato cifrato del quale si ha maggior possibilità di rintracciare informazioni sulla chiave.

Adaptive Chosen-Plaintext Attack

- L'attaccante può scegliere una serie di *plaintext*, basando la scelta sul risultato di precedenti cifrature (*differential cryptanalysis*).

Differential Cryptanalysis

È stata introdotta nel 1990 da Eli Biham e Adi Shamir, che la usarono per dimostrare che per determinate classi di algoritmi crittografici esiste un *adaptive chosen plaintext attack* che era più efficiente degli attacchi *brute force*. Anche se potenzialmente vulnerabile, il *Data Encryption Standard* (DES) si è dimostrato sorprendentemente resistente alla *differential cryptanalysis*. Perchè le *S-boxes* contengono esattamente valori ottimali da rendere un *differential attack* più difficile possibile? Perchè il DES utilizza esattamente 16 cicli, il minimo richiesto per rendere l'efficienza della *differential cryptanalysis* simile a quella di un attacco *brute force*? Risposta: Perchè nel tardo 1970 gli sviluppatori della IBM conoscevano già la crittoanalisi differenziale!

Don Coppersmith, della IBM, nel 1992 scrisse:

The design took advantage of certain cryptanalytic techiques, most prominently the technique of differential cryptanalysis, which were not known in the published literature.

After discussions with the National Security Agency (NSA), it was decided that disclosure of the design considerations would reveal the technique of differential cryptanalysis, a powerful technique that can be used against many ciphers. This in turn would weaken the competitive advantage the United States enjoyed over other countries in the field of cryptography.

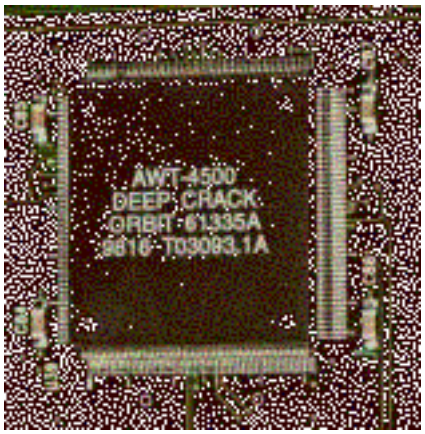
Breaking DES

Attack Method	Data Complexity		Storage Complexity	Processing Complexity
	Known	Chosen		
Exhaustive Precomputation	—	1	2^{56}	1 (table lookup)
Exhaustive Search	1	—	negligible	2^{55}
Linear Cryptanalysis	2^{43}	—	for texts	2^{43}
Differential Cryptanalysis	—	2^{47}	for texts	2^{47}
Differential Cryptanalysis	2^{55}	—	for texts	2^{55}

Considerando la tabella, (che proviene p. 259 del testo di ALFRED MENEZES, PAUL VAN OORSCHOT, e SCOTT VANSTONE, *Handbook of Applied Cryptography*, *CRC Press*, 1996), risulta considerevolmente facile eseguire una ricerca esaustiva (*exhaustive search*) con una coppia nota *plaintext-ciphertext* e 2^55 operazioni DES. Inoltre si può tentare una crittoanalisi lineare (*linear cryptanalysis*) che richiede 2^43 coppie note *plaintext-ciphertext*. La crittoanalisi differenziale ha minato la forza degli algoritmi a blocchi, ed un attacco che ha percentuali di successo dello 0,01% è potenzialmente devastante.

Nel 1993 *Michael Wiener* aggiornò il sistema a ricerca esaustiva con le più attuali tecnologie. il risultato fu una macchina da un milione di dollari con 57,000 *chip* DES ed una architettura *pipelined*. *Wiener* stimò la soluzione del problema DES in 3 ore e mezza. Il seguente fervore fu d'obbligo.

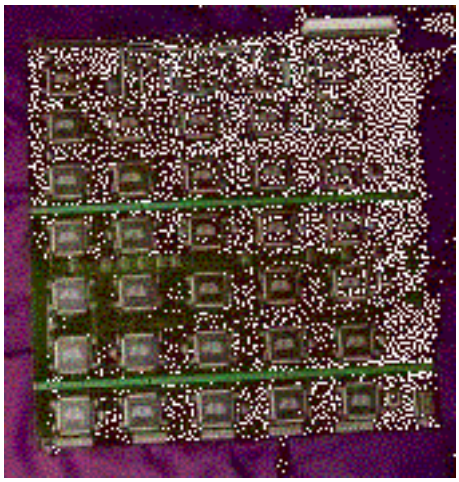
Nel Luglio 1998, utilizzando un calcolatore appositamente sviluppato, la ***Electronic Frontier Foundation*** costruì il **DES *Cracker***. Il suo costo fu di $250,000 e si impiegò meno di un anno per realizzarlo; il DES *Cracker* riuscì a decifrare il DES in 56 ore. In realtà la chiave fu trovata dopo solo un quarto dello spazio delle chiavi, mentre ci si aspettava di doverne attendere almeno metà. La costruzione del DES *Cracker* ha richiesto l'utilizzo di 1,536 *chip*, i quali sono in grado di cercare 88 miliardi di chiavi al secondo.



Se la **EFF** avesse investito altri $250,000 collegando le due macchine in parallelo si disporrebbe di un DES *Double-Cracker* che impiegherebbe la metà del tempo.

Oggi, disponendo di uno spazio delle chiavi DES pari a 2^56, cioè circa 7,2056 *10^16, si può stimare che un PC, con *clock* a 500 Mhz in grado di provare una chiave per ciclo di *clock* impieghi (ipotesi per assurdo):

in 144115188 secondi (ossia 834 giorni, che equivalgono a 2 anni e 3 mesi) si possono provare 2^55 (3,6 *10^16 chiavi).



È anche interessante osservare quale tipo di attività porta avanti il progetto ***Distributed.net***, degno di nota il loro *Moo client*.

Triplo DES

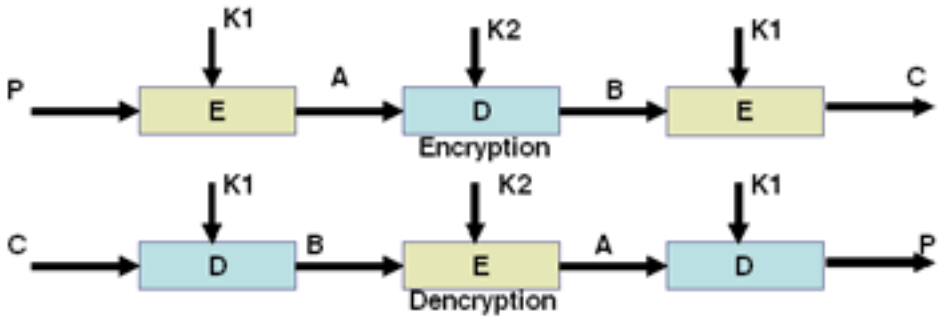
La brevità della chiave consente attacchi esaustivi di ricerca. Il problema quindi risiede nel riuscire a trovare un metodo che permetta di aumentare la lunghezza della chiave, migliorare la sicurezza dell'algoritmo ed al tempo stesso non aumentare troppo la complessità computazionale dello stesso.

Il primo metodo che può venire in mente è quello di cifrare più volte il medesimo messaggio con chiavi differenti. Ciò che si ottiene è un cifrario a singolo passo dove la

chiave, però, è lunga doppia o tripla rispetto il cifrario originale.

TRIPLE-DES (3DES)

Per ovviare alle debolezze della cifratura DES reiterata due volte, si è pensato di applicare la medesima strategia, ma di cifrare tre volte con tre chiavi differenti.



Utilizzando 3 chiavi da 56 bit, si ottiene una chiave da 168 bit, difficile da identificare. Per questo motivo quasi tutte le implementazioni di 3DES utilizzano K1 = K3. Altra interessante caratteristica presentata da quasi tutte le implementazioni di 3DES è conosciuta con il nome di schema **EDE** ***E****ncryption* - ***D****ecryption* - ***E****ncryption*. In sostanza in questa implementazione durante la seconda fase, DES non viene utilizzato in cifratura, ma in decifratura. L'effetto collaterale ottenuto mediante questa implementazione è che utilizzando K1 = K3 = K2 si ottiene l'effetto di cifratura che si otterrebbe con un DES classico. La necessità di una simile implementazione è data dalla volontà di mantenere la compatibilità con applicazioni che utilizzano il DES classico invece che il più robusto 3DES.

Altri algoritmi, AES

Name of Algorithm	*Block Size*	*Key Size*
DES (*Data Encryption Standard*, IBM)	64	56
3DES (*Triple* DES)	64	168
IDEA (Lai / Massey, ETH Zurigo)	64	128
RC2 (*Ron Rivest*, RSA)	64	40...1024
CAST (Canada)	64	128
Blowfish (*Bruce Schneier*)	64	128 ... 448
Skipjack (NSA, *clipper chip*)	64	80
RC5 (*Ron Rivest*, RSA)	64...256	64...256

Considerando l'età ed i manifesti difetti di robustezza del DES, diverse sono le soluzioni che nel tempo, fino ad oggi si alternano ad esso. Il già menzionato *Triple* DES con chiave da 168 bit è attualmente un *Federal Information Processing Standard* FIPS 46-3 (rinnovato nell Ottobre 1999).

Ad oggi è terminato il processo ***Evaluation of an Advanced Encryption Standard*** che ha dato risultato (visibile nel sito **http://www.nist.gov/aes**). Il *National Institute of*

Standards and Technology (NIST, *U.S. Department of Commerce*) ha avviato un contesto pubblico nel 1997. Cinque dei finalisti hanno ottenuto una menzione d'onore, ma la palma d'oro è spettata al sistema **Rijndael** in Ottobre 2000.

Il nuovo AES ha i seguenti requisiti:

- AES sarà definito pubblicamente.
- AES sarà di tipo *symmetric block cipher*.
- AES sarà implementabile sia tramite *hardware* sia tramite *software*.
- AES sarà progettato in modo tale che la lunghezza della chiave possa essere incrementata come necessario.
- AES avrà blocchi di dimensione n = 128 bit, *key size* k = 128, 192, 256 bit.

Conclusioni

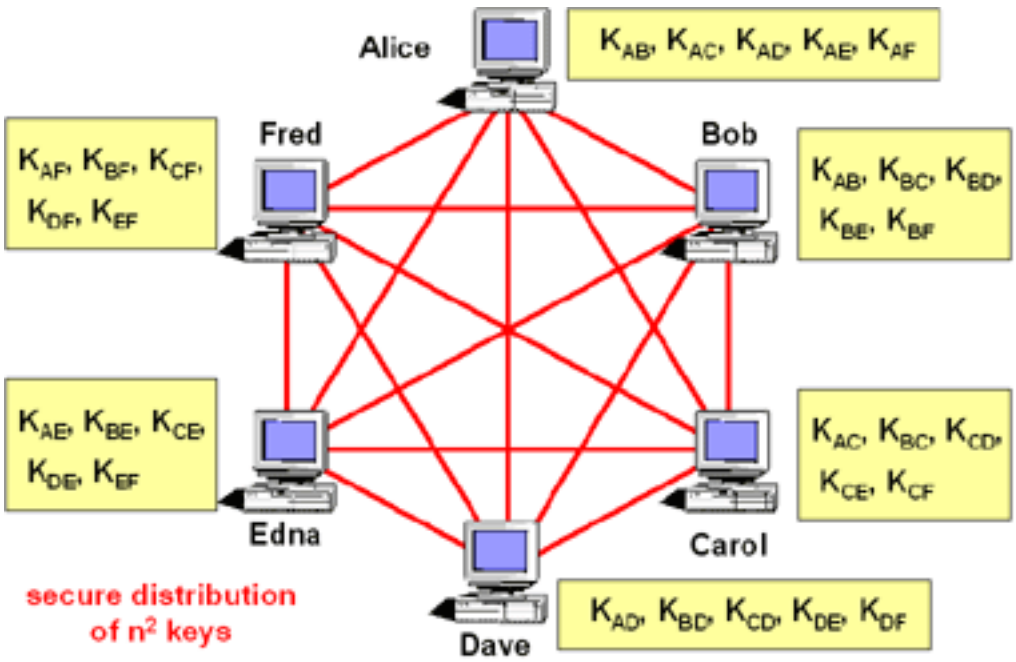
In conclusione va brevemente detto che i sistemi simmetrici lasciano aperti due grandi problemi:

- Metodo per scambio di chiavi assente (OOB).
- Numero di chiavi troppo grande per gestire dei segreti condivisi tra singoli utenti [n(n-1)]/2.

Necessita una soluzione...

Problema della distribuzione delle chiavi

Nelle reti magliate densamente popolate, dove molte parti potrebbero voler comunicare con altre, il numero di chiavi segrete richieste quando si usano algoritmi a chiave simmetrica aumenta quadraticamente con il numero dei partecipanti n ed è dato da [n(n-1)]/2, dal momento che occorre una chiave per ciascuna coppia di terminali.



Si prenda ad esempio una rete di comunicazione *broadband* con 100 nodi completamente magliati ove ogni chiave sia cambiata ad ogni sessione ogni ora. Come risultato avremo la necessità di distribuire 240000 chiavi in tutta sicurezza (OOB) ogni giorno.

La scala di distribuzione delle chiavi segrete peggiora pesantemente la complessità del sistema con un lieve incremento di partecipanti. Si cerca quindi una soluzione al problema della distribuzione delle chiavi tramite canali su connessioni sicure. Una soluzione efficiente è rappresentata dal concetto di ***Public Key Cryptosystem***.

Public Key Distribution System

In un *Public Key Cryptosystem* l'idea applicata è la stessa utilizzata dai cifrari simmetrici, cioè quella di rendere non leggibile un messaggio mediante la sua cifratura a mezzo di una chiave. La differenza risiede nel fatto che nei cifrari asimmetrici ogni utente possiede una coppia di chiavi. Queste sono note con il nome di chiave pubblica e chiave privata.

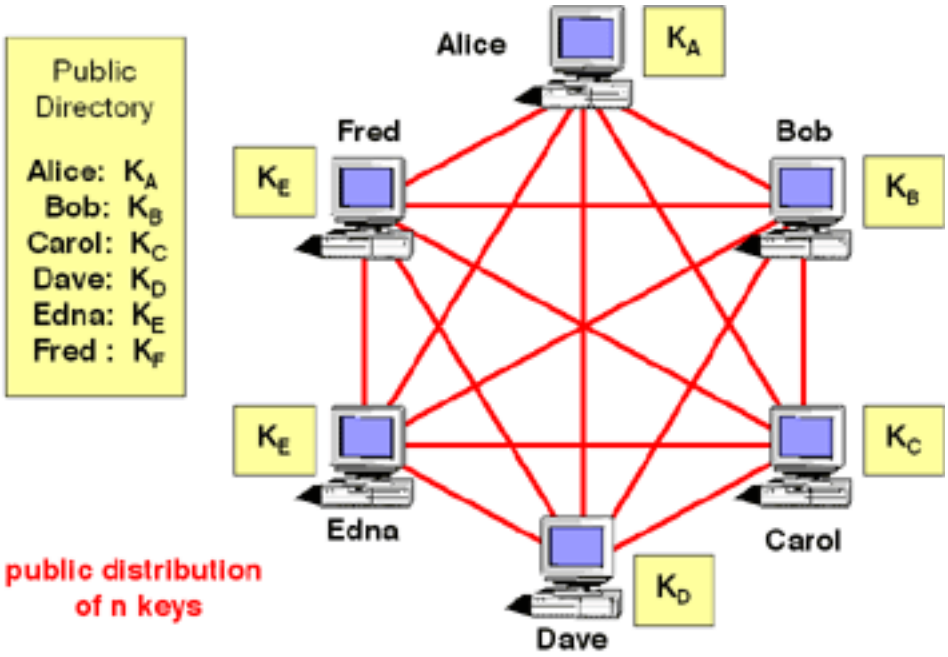
Occorre precisare alcuni requisiti sulle chiavi di un ipotetico sistema asimmetrico:

- Le chiavi sono tra loro indipendenti. Sceglie l'utente quale rendere pubblica e quale mantenere privata.
- Le due chiavi possono essere utilizzate indifferentemente per cifrare o decifrare.
- Entrando in possesso di una chiave, tipicamente quella pubblica, non è possibile in alcun modo risalire all'altra chiave.
- Ogni utente può possedere più di una coppia di chiavi, destinate agli usi più disparati; tali chiavi possono essere disponibili o pubblicate in *public directory* (es. LDAP o HTTP *server*).
- La generazione della coppia di chiavi avviene contemporaneamente.

L'utilizzo di questo tipo di cifrari introduce sostanziali vantaggi:

- Migliora il metodo di gestione delle chiavi.
- A mezzo dei cifrari asimmetrici è possibile realizzare il concetto di Firma Elettronica.

Inoltre, i cifrari asimmetrici permettono di realizzare non solo il concetto di Riservatezza del messaggio, ma anche il concetto di Autenticazione della fonte.



Se Alice vuole mandare un messaggio cifrato a Bob, Alice cifra il suo messaggio con la chiave pubblica di Bob recuperata da una *public directory* e lo invia a Bob. Poichè Bob è l'unico in possesso della chiave privata collimante, solo lui potrà decifrare il messaggio a lui destinato.

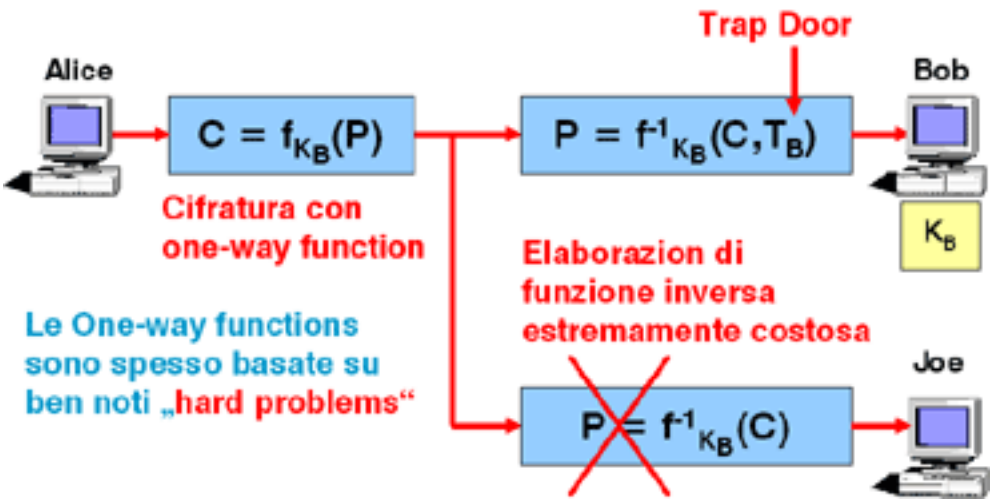
Dato che è necessario disporre solo della chiave pubblica del destinatario, con n utenti serviranno n chiavi distinte. Imponendo l'assioma che ogni utente genererà una propria coppia di chiave pubblica/privata locale, non serviranno canali sicuri per la distribuzione della chiave.

Principi di base della crittografia a chiave pubblica

Il concetto di crittografia a chiave pubblica è stato inventato quasi contemporaneamente da *Whitfield Diffie*, *Martin Hellman* e *Ralph Merkle*. I primi due ricercatori pubblicarono la loro invenzione nel 1976 e disposero di tutta la fama, *Ralph Merkle* ebbe la sfortuna che la stampa del suo lavoro subì il ritardo di oltre un anno e non fu quindi pubblicata prima del 1978. Oggi è generalmente riconosciuto che i tre scienziati sono padri della crittografia a chiave pubblica.

Recentemente è stato reso noto che già nel 1970, *James Ellis*, che al tempo lavorava per il governo Inglese come membro del *Communications-Electronics Security Group* (CESG), formulò l'idea di un *Public Key Cryptosystem*. Sfortunatamente, il governo non consentì la pubblicazione dei risultati della ricerca per ragioni di sicurezza.

Tutti i sistemi di crittografia a chiave pubblica sono basati sulla nozione di ***one-way function***, che, in funzione della chiave pubblica, converte il testo in chiaro in messaggio cifrato, utilizzando una relativamente piccola quantità di potenza computazionale ma la quale **funzione inversa** è estremamente costosa in termini di risorse di calcolo, al punto da rendere impossibile per un attaccante di ricavare il testo chiaro originale dal cifrato trasmesso in tempi ragionevoli.



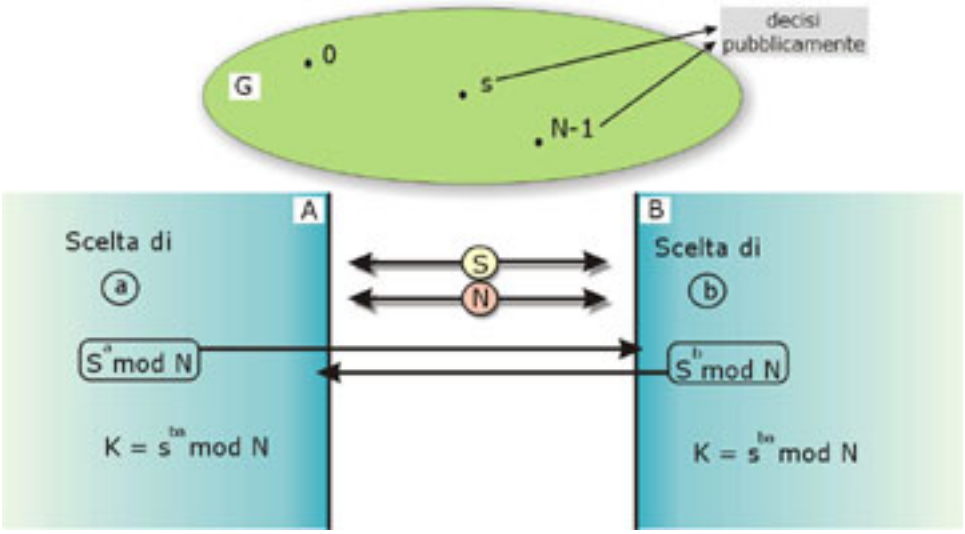
Un'altra nozione utilizzata nei sistemi a chiave pubblica è quello di una ***trap door*** che ogni funzione *one-way* possiede è che può essere attivata solo dal legittimo proprietario tramite la chiave privata. Utilizzando la *trap-door*, la decifrazione si semplifica.

Molti sistemi a chiave pubblica sono basati sulla nozione di ***known hard problem*** (problemi noti e di difficile soluzione) come ad esempio la fattorizzazione di numeri grandi nei loro fattori primi (RSA) o il prelievo di logaritmi discreti su di un campo finito (*Diffie-Hellman*).

Diffie Hellman

Il sistema di *Diffie* ed *Hellman* consiste nello scambio confidenziale di un messaggio, e viene definito: ***Key-Exchange Algorithm***. Ecco i passi necessari, per eseguire lo scambio:

- Alice e Bob scelgono pubblicamente un insieme di interi G=[0,N-1] ed un elemento s dello stesso insieme.
- Alice sceglie in modo casuale un elemento a di G, calcola s^a mod N e lo invia a Bob.
- Bob sceglie in modo casuale un elemento b di G, calcola s^b mod N e lo invia ad Alice.
- Alice, calcolato, come di seguito spiegato, s^b, calcola K= (s^b)a mod N.
- Bob, calcolato, come di seguito spiegato, s^a, calcola K= (s^a)b mod N.



Nella figura la lettera A sostituisce il nome proprio Alice, mentre la lettera B sostituisce il nome Bob.

RSA

Gli inventori *Ron **R**ivest*, Adi **S**hamir e *Leonard **A**dleman* utilizzano come *one-way function* la funzione esponenziale y = f(x) = x^e mod n che può essere elaborata con sforzo ragionevole. La sua inversa x = f^-1(y) è invece estremamente difficoltosa da elaborare.

Il sistema con algoritmo a chiave pubblica RSA è basato sul problema noto e difficile della fattorizzazione di numeri grandi nei loro fattori primi che è stata studiata per molti secoli.

La sfida RSA-155 basata su un numero da 512 bit (155 numeri decimali) ha visto impegnati 301 elaboratori in rete (300 fra *workstation* e *pc pentium*, 1 *Cray supercomputer*) ed ha portato alla soluzione della fattorizzazione in un tempo di 7 mesi.

109417386415705274218097073220403576120
037329454492059909138421314763499842889
347847179972578912673324976257528997818
337970765372440271467435315933544338 97
? =
102639592829741105772054196573991675900

716567808038066803341933521790711307779
*
106603488380168454820927220360012878679
207958575989291522270608237193062808643

I passi previsti per eseguire RSA sono i seguenti:

- Step 1: Scegliere casualmente due numeri primi grandi *p* e *q* - Per una maggior sicurezza, scegliere *p* e *q* di lunghezza circa uguale, esempio: 512-1024 bit ciascuno.
- Step 2: Calcolare il prodotto ***n*** = *p*q*.
- Step 3: Scegliere un intero a caso ***e*** < *(p-1)(q-1)* (i numeri ***e*** e *(p-1)(q-1)* devono essere primi fra loro, cioé non devono condividere fattori primi).
- Step 4: Calcolare l'inverso unico ***d***= *e^-1 mod (p-1)(q-1)* (l'equazione *d*e mod (p-1)(q-1) = 1* può essere risolta usando l'algoritmo Euclideo).

RSA esempio 1

Si prenda ad esempio p = 3 e q = 11:

n = p*q = 33

(p-1)*(q-1) = 2 * 10 = 2 * 2 * 5 = 20

l'esponente **e** deve essere relativamente primo a (p-1)*(q-1), non deve quindi contenere i fattori 2 e 5. Ecco le possibili scelte di **e** e **d**.

e	d	(e*d)	(e*d) mod 20
3	7	21	1
7	3	21	1
9	9	81	1
11	11	121	1
13	17	221	1
17	13	221	1
19	19	361	1

Public Key: modulo n ed esponente pubblico e; pubblicare n ed e in una *directory* pubblica, in modo che chiunque voglia mandare un messaggio confidenziale a noi possa venire in possesso di n ed e.

Private Key: modulo n ed esponente privato d; l'esponente privato d è segreto. Deve essere protetto sia memorizzandolo in una *smart card* a prova di intrusione o quando memorizzata in un disco cifrata con algoritmo simmetrico ed una frase di propria scelta. I numeri primi p e q utilizzati per la generazione delle chiavi possono anche essere cancellati dopo aver fatto tale operazione.

RSA esempio 2

Cifratura di un blocco di testo x:

$$y = x^e \bmod n$$

Il mittente usa la chiave pubblica del destinatario per cifrare x < n.

Decifratura del blocco y:

$$x = y^d \bmod n$$

Il destinatario utilizza la chiave privata per recuperare il blocco in chiaro x.

Senza prova:

$$y^d = (x^e)^d = x^{e \cdot d} = x^{m \cdot (p-1) \cdot (q-1) + 1} = x^1 = x \pmod{n}$$

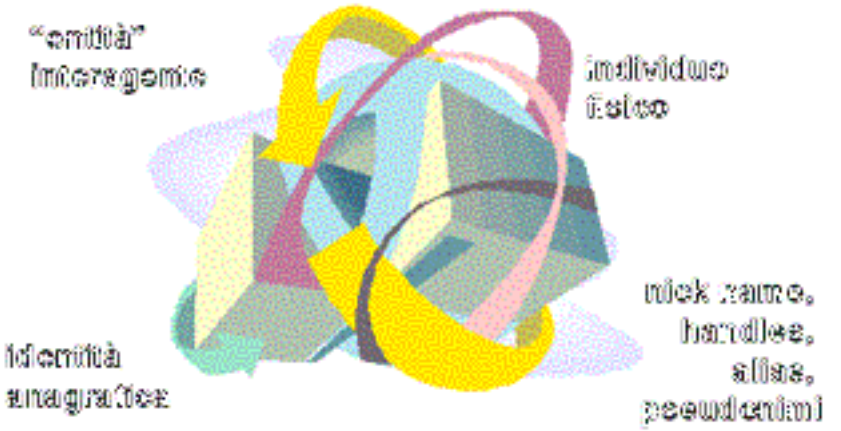
La cifratura e la decifratura sono operazioni simmetriche e l'ordine con il quale si calcolano gli esponenti pubblici e e privati d può essere cambiato.

L'esempio seguente, mostra in dettaglio i passaggi matematici:

Firma digitale

Le interazioni in rete avvengono tra entità. La difficoltà di risalire con certezza al collegamento biunivoco tra **entità interagente e individuo fisico** che trasferisce una determinata entità costituisce sempre più un problema da un punto di vista giuridico: nonostante in rete possano essere compiuti reati, truffe e illeciti di vario tipo, la responsabilità legale di questi atti non è facilmente attribuibile, in quanto è molto difficile accertare quale persona fisica stia effettivamente operando. Tali problemi sono

peculiari della giurisdizione statale (la particolare giurisdizione sviluppata spontaneamente in rete pare non preoccuparsene troppo). è solo in un'ottica repressiva e di controllo che, partendo da questa considerazione, si può arrivare a concludere che la comunicazione in rete è sempre anonima. Ciò che manca in rete non è il nome delle persone, ma semplicemente l'identità anagrafica.



A sostituzione del nome anagrafico, assegnato per legge e immodificabile, in rete prolifera una quantità enorme di altri nomi, *nicks*, *handles*, alias, pseudonimi. L'importanza di questi nomi non è minore di quella del proprio nome anagrafico: è attraverso il loro riconoscimento pubblico che in rete è possibile costruire relazioni sociali significative, relazioni che potranno essere trasferite anche al di fuori della rete.

Se esiste dunque una modalità caratteristica dell'interazione in rete rispetto ai nomi e alle identità individuali, questa non è data principalmente dall'anonimato ma piuttosto dallo **pseudonimato**. Lo pseudonimato comporta un processo di costruzione dell'identità e un suo riconoscimento sociale che perdurano nel tempo ma sono anche mutevoli e continuamente in divenire, mai acquisiti definitivamente; patrimonio fondamentale dello pseudonimo (sia esso corrispondente o no a un nome anagrafico) è la reputazione che riesce a guadagnare attraverso la sua vita in rete o quella che eredita da un'eventuale ragnatela di relazioni sociali avviate in precedenza *off-line*.

La perfetta realizzazione dello pseudonimato si scontra però con gli stessi problemi cui si è accennato a proposito dell'identità anagrafica: via rete è possibile modificare non solo il nome-numero di serie definito dallo Stato, ma anche lo stesso pseudonimo scelto. è possibile scrivere firmandosi con uno pseudonimo altrui. Questa possibilità realizza spesso un'utile opera di decostruzione dei propri pregiudizi. Nonostante questo, ci sono casi in cui, magari a causa della natura molto specifica e concreta della comunicazione, è assolutamente necessario essere certi dell'autore di un dato messaggio. Non tanto essere certi del suo numero di serie statale, quanto piuttosto del fatto che egli è effettivamente la stessa entità (individuale o collettiva) con cui si è comunicato in precedenza, via rete o anche in carne ed ossa. In altre parole, è necessario essere certi del suo pseudonimo.

è in una situazione come questa che la crittografia a chiave pubblica viene in aiuto. Ribaltando l'impiego delle chiavi pubbliche e private, è possibile porre una firma digitale crittografica sui messaggi che immettiamo in rete. La chiave segreta del mittente può infatti essere usata (oltre che per decifrare i messaggi ricevuti) anche per generare una

firma da apporre nei messaggi che si spediscono. La firma digitale del messaggio può poi essere verificata dal destinatario (o da chiunque altro) utilizzando la chiave pubblica del mittente.

Questo serve a garantire che il mittente è colui che davvero ha scritto il messaggio e che il messaggio non è stato successivamente manipolato da nessun altro, poiché solo il mittente possiede la chiave segreta per poter firmare. è tecnicamente impossibile falsificare o modificare un messaggio autenticato senza invalidarne la firma e lo stesso mittente non può più revocare la firma una volta apposta.

I rischi della firma digitale

Una delle applicazioni più utili della firma digitale, a parte l'autenticazione dei messaggi veri e propri, riguarda la **conferma delle chiavi pubbliche di terze persone**. La crittografia a chiave pubblica infatti lascia scoperto un possibile punto debole. Nel momento in cui si vuole comunicare con l'utente A, serve la sua chiave pubblica. Il modo migliore per ottenerla è direttamente dalle sue mani. Talvolta questo non è possibile e si è costretti a farla inviare attraverso la rete. Il sistema a chiave pubblica risolve ogni problema rispetto a un'eventuale intercettazione della chiave lungo il tragitto, ma presta il fianco alla possibilità che l'utente B, conoscendo la volontà di comunicare con l'utente A, si spacci per lui e spedisca una chiave pubblica contraffatta. Se si cade nel tranello e si utilizza quella chiave, i successivi messaggi saranno leggibili non dall'utente A, bensì dall'utente B, titolare della vera corrispondente chiave segreta. L'utente B potrà poi perfezionare il suo inganno rispedendo a sua volta tutti i messaggi all'utente A, che in questo modo non si accorgerà nemmeno dell'esistenza di una tappa in più lungo la strada.

Questo problema (chiamato **problema dell'uomo nel mezzo**) è assolutamente concreto e reale. La soluzione sta nel chiedere e ottenere che ogni nuova chiave pubblica sia firmata da qualcuno che si conosce e di cui si dispone già con certezza della rispettiva chiave pubblica. Se si viene costretti a ottenere la chiave dell'utente via rete, si avrà cura di verificare che essa sia firmata da un utente di cui si ha fiducia, con cui si hanno contatti quotidiani, a patto di possederne con certezza la vera chiave pubblica. Con questa chiave si potrà verificare la firma che l'uomo di mezzo ha apposto sulla chiave pubblica dell'utente A, cioè verificare che l'utente di cui ci si fida garantisce che la chiave pubblica che è appena arrivata è effettivamente la chiave dell'utente A. L'utente B, da solo, non sarebbe mai in grado di inviare una chiave firmata dall'uomo di mezzo, spacciandola per la chiave dell'utente A.

A questo punto è evidente che la certificazione delle chiavi può diventare rapidamente molto complessa consentendo di estendere la rete di contatti a partire da un unico punto iniziale sicuro e comprendendo anche entità che non si incontreranno mai di persona. Con questa caratteristica il cerchio viene chiuso e diventa veramente possibile stabilire un'infrastruttura comunicativa priva di contatti fisici che sia doppiamente sicura, sia dal punto di vista della possibilità di leggere il contenuto della comunicazione, sia da quello di poterne garantire la provenienza.

In conclusione si potrà dire che la firma digitale è una informazione che viene aggiunta ad un documento informatico al fine di garantirne integrità e provenienza. La principale differenza tra firma autografa e firma digitale sta nel fatto che la prima è direttamente riconducibile all'identità di colui che la appone, poiché la calligrafia è un elemento identificativo della persona, mentre la seconda non possiede questa proprietà. Per

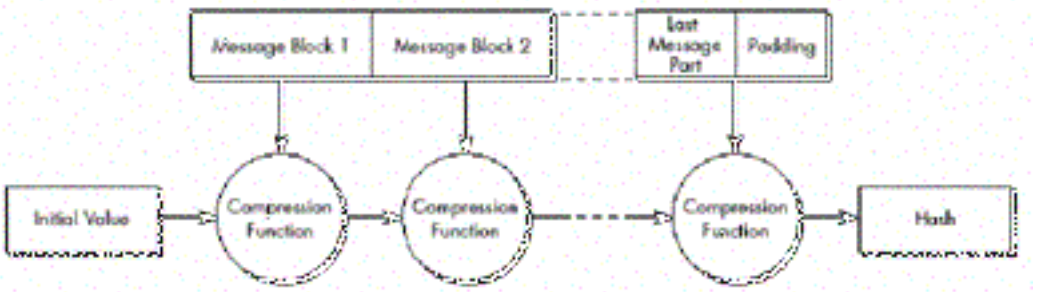
coprire questa deficienza si ricorre all'autorità di certificazione, il cui compito è quello stabilire, garantire e pubblicare l'assoc+iazione tra firma digitale e soggetto sottoscrittore.

Per contro, mentre l'associazione tra testo di un documento e la firma autografa è ottenuta esclusivamente attraverso il supporto cartaceo, la firma digitale è intrinsecamente legata al testo a cui è apposta, tanto che i due oggetti possono essere fisicamente separati senza che per questo venga meno il legame esistente tra loro. Conseguenza di ciò è l'unicità della firma digitale, nel senso che a testi diversi corrispondono firme diverse; in tal modo, nonostante la sua perfetta replicabilità, è impossibile trasferire la firma digitale da un documento ad un altro.

Processo di firma digitale

La firma elettronica, oltre ad avere la possibilità di autenticare i messaggi ai quali è associata, offre la possibilità di identificare univocamente l'utente firmatario. In sostanza ha applicazioni molto simili a quelle della firma autografa. Ciò è possibile in quanto si utilizza un cifrario asimmetrico del quale si usa la chiave privata per firmare e quella pubblica per autenticare.

Al testo da firmare viene applicata una **funzione di *hash*** appositamente studiata che produce una stringa binaria di lunghezza costante e piccola, normalmente 128 o 160 bit. La funzione di *hash* assicura l'unicità di tale stringa, nel senso che a due testi diversi non corrisponde la medesima impronta. Sono disponibili diversi algoritmi di generazione, quali, ad esempio, **MD2**, **MD4** e **MD5**, originariamente progettati per operare in combinazione con RSA ma utilizzabili con qualsiasi cifrario. Sono disponibili anche algoritmi di *hash* per i quali è in corso la standardizzazione ufficiale da parte organismi internazionali; ne sono un esempio il RIPEMD a 128 e 160 bit ed il ***S**ecure **H**ash **A**lgorithm* (**SHA-1**).



Affinché una funzione di *hash* operi correttamente è necessario procedere come segue:

- Suddividere il messaggio in blocchi di lunghezza congruente con il *digest* prodotto dalla funzione di *hash* prescelta.
- Sottoporre ogni blocco del messaggio all'operazione di *hashing*.
- Reiterare il procedimento.

Un ***M**essage **D**igest* rappresenta il riassunto di un messaggio. Il suo scopo non è quello di garantire la *privacy*, ma l'**integrità**. Grazie ad esso infatti si può verificare che il contenuto del messaggio non sia stato alterato nel suo tragitto. Lo si applica anche in

situazioni che richiedono velocità di calcolo e dove il corpo del messaggio in questione sia di dimensioni troppo elevate.

L'utilità dell'impronta è duplice, in primo luogo consente di evitare che per la generazione della firma sia necessario applicare l'algoritmo di cifratura, che è intrinsecamente inefficiente, all'intero testo che può essere molto lungo. Inoltre consente l'autenticazione, da parte di una terza parte fidata, della sottoscrizione di un documento senza che questa venga a conoscenza del suo contenuto. Una tipica situazione in cui si sfruttano tali caratteristiche dell'impronta è la marcatura temporale che verrà discussa più avanti.

Generazione della firma digitale

La generazione della firma consiste semplicemente nella cifratura dell'impronta digitale generata il precedenza mediante la chiave segreta Ks. In questo modo la firma risulta legata da un lato al soggetto sottoscrittore (attraverso la chiave segreta usata per la generazione) e dall'altro al testo sottoscritto (per il tramite dell'impronta).

In realtà l'operazione di cifratura viene effettuata, anziché sulla sola impronta, su una struttura di dati che la contiene insieme con altre informazioni utili, quali ad esempio l'indicazione della funzione *hash* usata per la sua generazione. Sebbene tali informazioni possano essere fornite separatamente rispetto alla firma, la loro inclusione nell'operazione di codifica ne garantisce l'autenticità.

Apposizione della firma

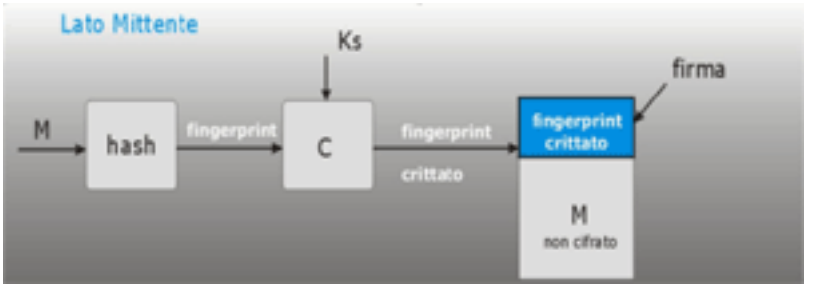
La firma digitale generata al passo precedente viene aggiunta in una posizione predefinita, normalmente alla fine del testo del documento. Normalmente, insieme con la firma vera e propria, viene allegato al documento anche il valore dell'impronta digitale ed eventualmente il certificato da cui è possibile recuperare il valore della chiave pubblica. è evidente che essendo il legame tra firma e documento stabilito attraverso l'impronta, di natura puramente logica, la firma stessa e le informazioni aggiuntive eventualmente ad essa associate possono essere registrate e gestite in modo del tutto separato rispetto al testo sottoscritto; in particolare possono trovarsi su supporti e sistemi di elaborazione del tutto indipendenti tra loro.

Attualmente le tecniche di firma elettronica sono realizzate mediante:

- RSA in unione con MD5 / SHA-1;
- DSA in unione con SHA-1.

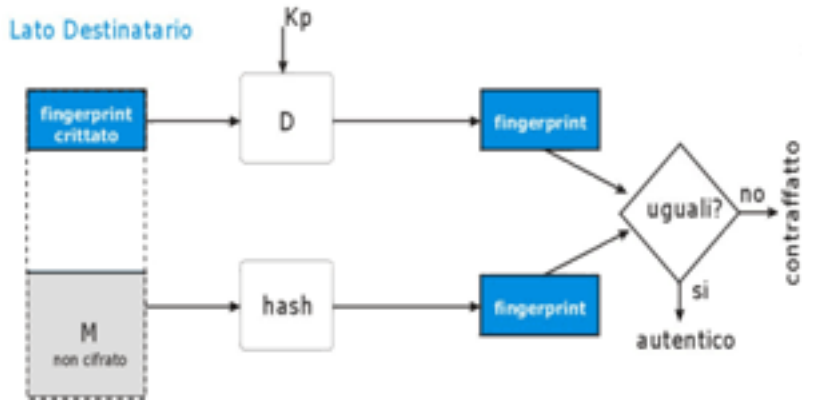
Firma RSA

L'uso dello schema RSA per generare firme elettroniche si basa semplicemente sull'inversione del ruolo delle chiavi rispetto a quello assegnato nell'assicurare la riservatezza; la principale differenza tra le due applicazioni sta nel fatto che per la firma si evita di applicare l'operazione di codifica all'intero testo. Ciò è particolarmente conveniente vista la complessità e la lentezza delle operazioni coinvolte.



- Si applica una funzione di *hash* (SHA-1) al messaggio.
- Il *digest* così ottenuto viene cifrato mediante RSA.
- Il *file* di firma ottenuto è accodato al messaggio oppure inviato parallelamente.

In pratica il testo da firmare viene compresso in una sorta di riassunto, che viene spesso riferito come impronta, mediante una opportuna funzione di *hash* progettata in modo da rendere trascurabile la probabilità che da testi diversi si possa ottenere il medesimo valore. La dimensione del riassunto è fissa ed è molto più piccola di quella del messaggio originale; essa è dell'ordine del centinaio di bit, in modo da rendere estremamente più rapida la generazione della firma effettuata a partire dall'impronta anziché dal testo.



- Si applica SHA - 1 al messaggio, ottenendo un *digest*.
- Si decifra il *digest* allegato al messaggio tramite la chiave pubblica del mittente.
- Si confrontano i *digest*: se corrispondono si ha autenticazione.

Possibili debolezze della firma

La principale osservazione sulla firma elettronica riguarda la conoscenza della chiave pubblica del mittente. Ossia il ricevente il messaggio deve conoscere in modo certo la chiave pubblica del mittente. Altrimenti sarebbe possibile generare una coppia di chiavi facendo credere che queste appartengano a qualcun altro di cui il destinatario conosce l'identità. A questo punto è sufficiente intercettare i messaggi inviati dal mittente, sostituirli, firmarli con la falsa chiave ed inviarli al destinatario che considererà autentici dei messaggi falsi.

Importanza della protezione Birthday Attack

In un contesto di firma elettronica è molto importante proteggere i *Birthday Attack*, quindi utilizzare una funzione di *hash* resistente alle collisioni.Supponiamo di procedere in questo modo:

- Si generano una quantità di varianti di messaggi che il potenziale mittente si sente pronto a firmare.
- Parallelamente si generano una quantità di messaggi alterati, ma che producono i medesimi risultati di *Hash* (collisioni).
- A questo punto è sufficiente staccare la firma dal messaggio originale ed attaccarla al messaggio alterato. Essendo questi messaggi una collisione per la funzione di *hash* risulteranno indistinguibili per il destinatario.

Firma DSS

Il metodo di firma **DSS** (***D**igital **S**ignature **S**tandard*) ed il relativo algoritmo **DSA** (***D**igital **S**egnature **A**lgorithm*) sono metodi di firma alternativi allo standard RSA, e sono una variante degli algoritmi di firma e cifratura Schnorr (brevettato con scadenza nel 2008) e ElGamel.

Pubblicato dal **NIST** (***N**ational **I**nstitute of **S**tandard and **T**echnology*) il 30 agosto 1991, è sostanzialmente una funzione di *hash* H che ha come unico argomento il messaggio; in tal modo il suo valore non dipende dalla chiave di cifratura. Il suo attuale nome è **FIPS 186**, poiché una pubblicazione del ***F**ederal **I**nformation **P**rocessing **S**tandard* ne descrive il DSA. Nel documento, l'algoritmo chiave della codifica è il ***S**ecure **H**ash **S**tandard* (**SHS**).

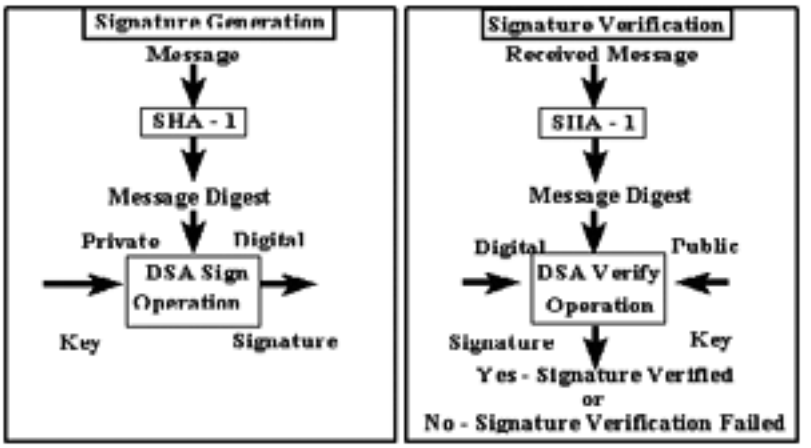
La sicurezza dell'algoritmo è basata sulla difficoltà di calcolare il logaritmo discreto in un gruppo finito. In sostanza occorre calcolare:

a^x mod z (che è considerato computazionalmente facile), essendo

x tale che a^x = b mod z (che è considerato computazionalmente difficile).

Ricordiamo, invece, che la forza di RSA è basata sulla difficoltà di fattorizzare in primi. è possibile dimostrare che il calcolo di un logaritmo discreto in un gruppo finito equivale dal punto di vista computazionale a fattorizzare un numero ottenuto come prodotto di due primi.

Il processo di firma è graficamente rappresentabile con la seguente figura:



Certificati

Il processo di firma digitale richiede che l'utente effettui una serie di azioni preliminari necessarie alla predisposizione delle chiavi utilizzate dal sistema di crittografia su cui il meccanismo di firma si basa; in particolare occorre effettuare le seguenti operazioni:

- registrazione dell'utente presso un'autorità di certificazione;
- generazione di una coppia di chiavi Ks e Kp;
- certificazione della chiave pubblica Kp;
- registrazione della chiave pubblica Kp.

Per motivi di confidenzialità, la chiave privata non dovrà mai essere inviata con mezzi di comunicazione non sicuri. Quindi non si avrà la certificazione della chiave privata (DPCM 08 febbraio 1999 - Art. 28 - Generazione dei certificati).

Un certificato è costituito dalla coppia [nome utente, chiave pubblica utente] certificata, ossia firmata da una autorità di certificazione. Le autorità di certificazione possono

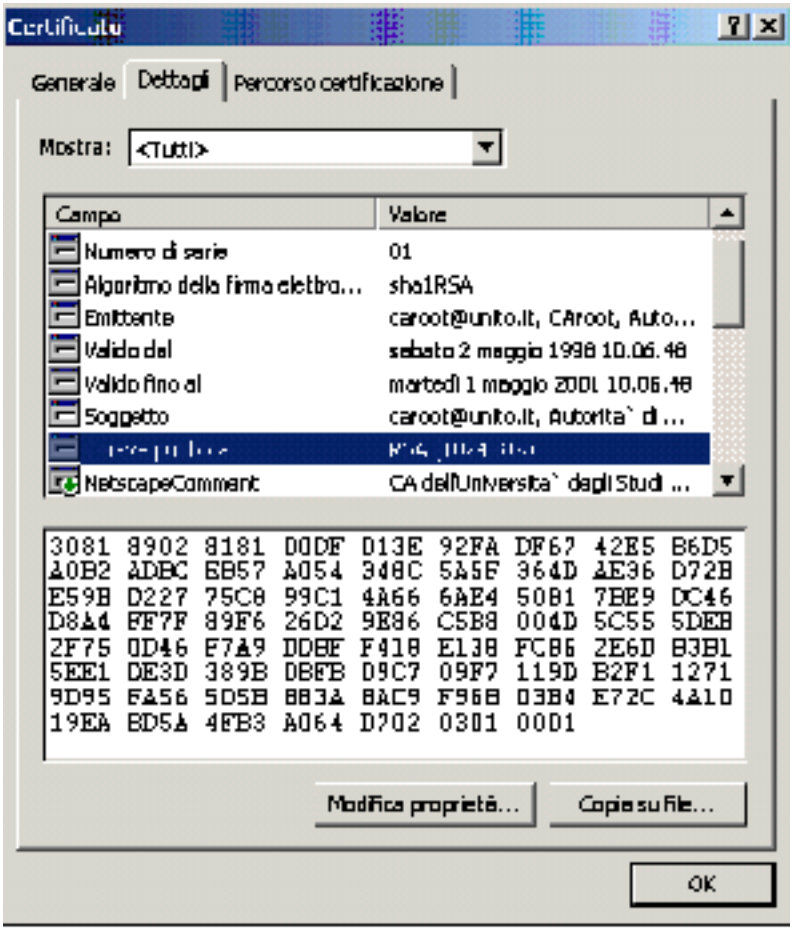
essere di due tipi:

- autorità di certificazione create da una realtà, aziendale o amministrativa, il cui compito è quello di certificare ed autenticare lo scambio di informazioni all'interno della rete dell'organizzazione;
- autorità di certificazione il cui scopo è quello di certificare realtà commerciali che si occupano di *e-commerce*.

Si può ottenere un elenco aggiornato delle autorità di certificazione legalmente riconosciute presso il sito **AIPA**. La richiesta di un certificato è una operazione tradizionale: infatti ci si deve recare dall'autorita di certificazione personalmente e muniti di un documento di identità, in questo modo l'autorità potrà essere sicura dell'identità di colui il quale richiede il certificato. In seguito a questa operazione e successivamente nel tempo, l'utente riceverà dall'autorità il proprio certificato il quale contiene:

- I dati identificativi dell'utente.
- La chiave pubblica dell'utente.
- La firma dell'autorità di certificazione.

Da questo momento l'utente sarà in grado di utilizzare il certificato per gli scopi previsti. Notiamo che un certificato non può essere manomesso: infatti ogni certificato reca con sé un *fingerprint* univoco il quale altro non è che un *digest* ricavato dal messaggio stesso. Ovviamente essendo un *digest* ottenuto con una funzione di *hash* ad una via e *collision resistant* difficile è che due certificati differenti abbiano il medesimo *fingerprint*. Di solito il *fingerprint* è una caratteristica del certificato al quale si cerca di dare la maggiore pubblicità possibile in modo che sia facilmente verificabile.



Autorità di certificazione

Una autorità di certificazione è un organismo che si occupa di verificare ed assicurare la corrispondenza chiave pubblica - utente. Inizialmente, si era pensato di organizzare le autorità in maniera gerarchica, per alcuni motivi:

- Tramite una gerarchia è possibile creare autorità locali che vengono certificate da autorità di livello superiore.
- Tramite una gerarchia è possibile generare CA indipendenti e specializzate.
- è possibile, inoltre, fornire maggiore sicurezza all'autorità stessa. Infatti se per qualche motivo dovesse venire meno l'affidabilità di una autorità intermedia, è possibile sostituire l'entità intermedia con una nuova autorità senza dovere invalidare i certificati emessi dall'autorità di più alto livello.
- Tramite una gerarchia si può garantire interoperabilità ai certificati: è quindi possibile considerare attendibile un certificato emesso, ad esempi, alle isole Hawaii.

La gerarchia delle autorità produce un effetto immediato: le autorità al di sotto della *root* ereditano le proprietà di quest'ultima, ossia se la *root* è abilitata a certificare la posta ed i siti web, ma non le applicazioni anche i certificatori che da questa dipendono avranno

le medesime caratteristiche.

Il meccanismo di gerarchia non ha però avuto successo. L'insuccesso è dovuto sostanzialmente al fatto che le CA corporate non hanno bisogno di essere inserite in una gerarchia in quanto il loro scopo è quello di fornire sicurezza all'interno dell'organizzazione alla quale appartengono. A questo è necessario aggiungere che alcune CA corporate, in particolare quelle bancaria, mal tollerano il fatto di avere un'autorità a loro superiore che, di fatto, le costringerebbe a considerare attendibili certificati emessi da altri.

Certificate Revocation List

Il ***C****ertificate* ***R****evocation* ***L****ist* (**CRL**) è l'elenco dei certificati che sono stati revocati dall'autorità che li ha emessi. I motivi di revoca possono essere diversi: ad esempio per qualche motivo viene violata la chiave privata dell'utente e diventa quindi possibile falsificarne la firma, oppure vengono meno i motivi per cui l'utente ha necessità di mantenere un certificato. Si potrebbe anche lasciare scadere il certificato: la situazione che si verificherebbe tra il momento della violazione e quello della scadenza sarebbe ingestibile.

Sicurezza nelle applicazioni client-server

Franco Callegati
Paolo Zaffoni

8.4.1 (Spiegare i principali aspetti della sicurezza connessi alla trasmissione dei dati), 8.4.2 (Descrivere gli attuali standard di crittografazione: chiavi pubbliche e private, NSA, DES, PGP)

Servizi sicuri

Parlando di servizi sicuri, tipicamente ci si riferisce a servizi che forniscono due tipi di garanzie:

- il servizio non può essere utilizzato in nessun modo se non per le operazioni previste.
- non è possibile leggere e/o falsificare le transazioni che avvengo attraverso il servizio.

Tali garanzie non implicano che si possano eseguire transazioni con il servizio continuando ad essere al sicuro. Per esempio, si potrebbe utilizzare un HTTP (*HyperText Transfer Protocol*) sicuro per effettuare il *download* di un *file*, ed essere sicuri che si stia effettivamente effettuando il *download* del *file* a cui si è interessati, e che nessuno lo stia modificando nel transito. Ma non si possono avere garanzie che il *file* non contenga dei virus o programmi dannosi.

È possibile anche utilizzare servizi insicuri in modo sicuro, ma ciò richiede maggiore cautela. Ad esempio, la posta elettronica attraverso il protocollo SMTP (*Simple Mail Transfer Protocol*) è un classico esempio di un servizio insicuro.

Tutte le volte che si valuta la sicurezza di un servizio, bisogna contestualizzare le valutazioni al proprio ambiente e tener conto delle proprie configurazioni; non è interessante la sicurezza in astratto di un servizio.

Il World Wide Web

Il *World Wide Web* è divenuto così popolare che molte persone pensano che sia Internet stesso. Se non si appartiene al *Web*, non si è nessuno. Sfortunatamente, sebbene il *Web* si basi principalmente su un singolo protocollo (HTTP), i siti *Web* utilizzano spesso una varietà di protocolli.

Molte persone confondono le funzioni e le origini del *Web* con i *browser* (*Netscape, Microsoft Internet Explorer*), con un protocollo (HTTP) e con il linguaggio di pubblicazione dei documenti sui *server* Internet (HTML). Riteniamo importante quindi darne una sufficiente descrizione:

- Il *Web* consiste nell'insieme dei *server* HTTP in Internet.
- Il protocollo HTTP costituisce il protocollo principale su cui si basa il *Web*; consente agli utenti l'accesso ai documenti che sono resi disponibili dai *server Web*. Tali documenti possono assumere diversi formati (testo, immagini, audio, video, ecc.), ma il formato usato per fornire il collegamento tra essi è il linguaggio HTML (*HyperText Markup Language*).
- Il linguaggio HTML costituisce uno standard per la realizzazione di pagine *Web*. Fornisce delle funzionalità base per la formattazione dei documenti e

per la definzione di link ipertestuali ad altre pagine e/o ad altri *server*.

- *Netscape Navigator* e *Microsoft Internet Explorer*, comunementi noti come *Netscape* ed *Explorer*, (anche altri *browser* sono discretamente diffusi: Lynx, Opera, Slurp, Go!Zilla e perlWWW) sono prodotti commerciali che realizzano il lato *client* dell'applicazione *Web*. Un *Web client*, chiamato anche *Web browser*, consente la lettura di documenti attraverso il protocollo HTTP oppure attraverso altri protocolli.

Sicurezza dei Web client

I *Web browser* forniscono una interfaccia grafica per un gran numero di risorse Internet. Le informazioni e i servizi che non erano disponibili o che erano accessibili solo agli esperti di informatica diventano grazie ai *browser* facilmente accessibili.

Sfortunatamente è difficile rendere sicuri i *Web browser* ed i *Web server*. L'utilità del *Web* è in larga parte basata sulla sua flessibilità, ma tale flessibilità rende difficoltosi i controlli. Si pensi, ad esempio, quanto sia più facile trasferire ed eseguire un programma attraverso un *Web browser* rispetto al servizio FTP, ma si consideri anche la possibilità di trasferire ed eseguire un programma malizioso. I *Web browser* dipendono da programmi esterni, genericamente chiamati *viewer* (sono chiamati "visualizzatori" anche quando eseguono un brano sonoro invece di mostrare un'immagine), per gestire i tipi di *file* che non risultano decodificabili dal *browser*. Genericamente interpretano correttamente i principali tipi di *file* (HTML, il normale testo e le immagini in formato JPEG e GIF). *Netscape* ed *Explorer* attualmente supportano un meccanismo (progettato per rimpiazzare i *viewer* esterni) che consente a terze parti di produrre dei *plug-in* che possono essere scaricati per costituire una estensione del *Web browser*. Si deve fare molta attenzione a quali *viewer* vengono configurati e a quali *plug-in* vengono scaricati.

La maggior parte dei *Web browser* comprende uno o più linguaggi (*Java*, *Javascript* o *ActiveX*) che consentono di estendere le loro caratteristiche e le loro funzionalità. Questi linguaggi rendono i *Web browser* più potenti e più flessibili, ma introducono anche nuovi problemi. Mentre l'HTML è principalmente un linguaggio per la formattazione dei documenti, i linguaggi di estensione forniscono capacità di elaborazione locale delle informazioni, al pari di un linguaggio di programmazione. Tradizionalmente, quando si acquista un nuovo programma si sa da dove proviene e chi lo ha realizzato. Se si decide di copiare un programma da un sito Internet, non si hanno le stesse informazioni.

I progettisti di *Javascript*, *VBscript*, *Java* ed *ActiveX* hanno proposto diversi approcci per la soluzione a questo problema. Per quanto riguarda *Javascript* e *VBscript* si suppone semplicemente che non possano eseguire azioni dannose; tali linguaggi infatti non hanno, ad esempio, comandi per scrivere sul disco. *Java* usa un approccio chiamato *sandbox*. *Java* a differenza dei precedenti linguaggi contiene dei comandi che potrebbero essere dannosi, ma l'intrerprete *Java* blocca un programma non fidato ogniqualvolta tenta un'azione dannosa. Sfortunatamente, ci sono stati problemi di implementazione con *Java* e sono stati trovati diversi modi per eseguire delle operazioni che si credevano impossibili.

La tecnologia *ActiveX* invece di provare a limitare le possibilità di un programma, cerca di associare ad un programma le informazioni che ci consentono di individuare da dove e da chi proviene in modo da valutarne l'affidabilità. Tutto ciò è realizzato attraverso il

meccanismo della firma digitale; prima dell'esecuzione di un programma *ActiveX* un *Web browser* mostra le informazioni relative alla firma digitale dell'autore del programma e l'utente può decidere se eseguirlo o meno.

Sicurezza dei Web server

L'attivazione di un *server Web* implica che tutta la comunità di utilizzatori che possono accedervi possano inviare dei comandi. Anche se il *Web server* è configurato per fornire solamente *file* HTML e quindi i comandi sono abbastanza limitati possono comunque verificarsi problemi di sicurezza. Ad esempio, molte persone ritengono che l'utente non possano vedere i *file* del *server* a meno che non esistano link espliciti ad essi; tale assunzione è generalmente falsa. È più corretto ritenere che se un *Web server* è in grado di leggere un *file*, è anche in grado di fornirlo ad un utente remoto. I *file* che non dovrebbero essere pubblicamente accessibili dovrebbero almeno essere protetti con dei permessi a livello di *file system*, e dovrebbero, se possibile, essere posti al di fuori dell'area di disco accessibile al *Web server*.

Molti *Web server*, inoltre, forniscono altri servizi rispetto alla semplice gestione di *file* HTML. Ad esempio, alcuni *Web server* forniscono dei servizi amministrativi che consentono all'amministratore del *Web server* di configurarlo attraverso un *Web browser* da remoto, senza necessità di lavorare sulla macchina *server*. Se l'amministratore del *Web server* può raggiungere il *server* attraverso un *browser*, chiunque può farlo; bisogna quindi essere certi che la configurazione iniziale del *server* sia impostata in un ambiente sicuro.

I *Web server* possono anche invocare l'esecuzione di programmi esterni in diversi modi. Tali programmi sono molto facili da scrivere ma molto difficile da rendere sicuri, poiché possono ricevere comandi arbitrari da utenti esterni. Il *Web server* non fornisce alcuna protezione significativa per tali programmi.

Protocolli HTTP sicuri

Allo stato attuale 2 protocolli forniscono *privacy* del contenuto all'HTTP usando meccanismi di cifratura e di autenticazione forte. Quello che comunemente viene adottato si chiama HTTPS ed è utilizzato inserendo nell'URL la parola chiave https. L'altro, per lo più sconosciuto, si chiama *Secure* HTTP ed utilizzato inserendo nell'URL la parola chiave shttp.

L'obiettivo del protocollo HTTPS è quello di proteggere il canale di comunicazione quando si ricevono o si spediscono dati. Attualmente HTTPS utilizza il protocollo SSL per ottenere tale obiettivo.

L'obiettivo del protocollo *Secure* HTTP è quello di proteggere i singoli oggetti che vengono scambiati piuttosto che il canale di comunicazione. Questo consente, ad esempio, che alcune pagine su un *Web server* possano essere associate ad una firma digitale e che un *Web client* possa verificare la firma al momento del *download* di tali pagine.

L'uso del *Secure HTTP* potrebbe avvantaggiare significativamente i consumatori nel mondo del commercio elettronico, infatti l'identità del consumatore e quella del venditore sono associate in maniera inscindibile agli oggetti che fanno parte di una

transazione Secure HTTP, mentre nel caso del protocollo HTTPS l'identità del consumatore e quella del venditore sono associate al canale di comunicazione.

La posta elettronica

La posta elettronica e le *news* forniscono agli utenti un modo per scambiarsi informazioni senza la necessità di risposte interattive o immediate.

La posta elettronica è uno dei più popolari servizi di Internet. Solitamente è un servizio a basso rischio (ma non esente da rischi). Modificare la posta elettronica è banale, e le modifiche facilitano due differenti tipi di attacco:

- attacchi contro la propria reputazione;
- attacchi di manipolazione sociale (ad esempio, ad un utente viene inviato un messaggio da parte di un sedicente amministratore che lo invita ad impostare la propria *password* in un certo modo).

Accettare messaggi di posta elettronica significa consumare tempo di CPU e spazio di memoria e sul disco; anche su ciò si basano gli attacchi di tipo DoS (*Denial of Service*). In particolare, con gli attuali sistemi di posta elettronica multimediali, alcuni utenti possono spedire messaggi di posta elettronica contenenti dei programmi che possono essere eseguiti. All'interno di tali programmi possono celarsi dei *Trojan* (Cavalli di Troia).

Sebbene molti utenti si preoccupano degli attacchi diretti, in pratica, i problemi più comuni con la posta elettronica sono dovuti agli attacchi di *flooding* (incluse le cosiddette catene di Sant'Antonio) ed alle persone che inviano dati riservati fidandosi della confidenzialità del servizio. Se gli utenti sono informati adeguatamente ed il servizio di posta elettronica è isolato da altri servizi, in modo tale che gli attacchi DoS possano provocare il minor numero di danni, il servizio stesso è ragionevolmente sicuro.

Il protocollo SMTP (*Simple Mail Transfer Protocol*) è il protocollo Internet standard per spedire e ricevere posta elettronica. La posta che viaggia attraverso i *server* Internet viene gestita principalmente attraverso il protocollo SMTP. Il protocollo SMTP in sè non è un problema, i *server* SMTP invece possono esserlo. Un programma che consegna la posta agli utenti spesso deve essere in grado di accedere alle risorse dei singoli utenti.

Il più comune *server* SMTP in *Unix* è *Sendmail*. *Sendmail* è stato attaccato con successo in diversi modi, (si pensi, ad esempio al caso dell'Internet *worm*), scoraggia gli utenti ad usarlo. Comunque, molti *server* di posta che sono stati sviluppati per superare i problemi di *Sendmail* non sono certo migliori di *Sendmail*. L'evidenza suggerisce che subiscono meno attacchi perché sono meno popolari e non perché siano più sicuri o meno vulnerabili.

Il più comune *server* SMTP nei sistemi operativi della *Microsoft* è *Microsoft Exchange*; anche *Exchange* è stato più volte attaccato con successo.

Il protocollo SMTP viene utilizzato per scambiare messaggi di posta elettronica tra i *server*, e per trasferire la posta da un *client* al *server*. Per leggere la posta dalla propria casella ospitata da un *server*, gli utenti impiegano un protocollo diverso: i più utilizzati a questo scopo sono POP (*Post Office Protocol*) ed IMAP (*Internet Message Access*

Protocol). *Mircosoft Exchange* e *Lotus Notes* utilizzano dei protocolli proprietari che forniscono caratteristiche aggiuntive.

I protocolli POP ed IMAP hanno le medesime implicazioni per quel che riguarda la sicurezza; entrambi trasferiscono i dati relativi all'autenticazione degli utenti ed al contenuto dei messaggi senza cifrarli, consentendo agli attaccanti di leggere la posta e di ottenere le credenziali degli utenti.

Usenet news

I *newsgroup* equivalgono a bacheche su cui si affiggono gli annunci e sono stati progettati per realizzare comunicazioni molti a molti. Anche le *mailing list* supportano le comunicazioni molti a molti, ma lo fanno meno efficientemente.

I rischi delle *news* sono molto simili a quelli della posta elettronica:

- gli utenti possono fidarsi inavvertitamente delle informazioni ricevute;
- gli utenti possono rivelare informazioni riservate;
- il proprio *server* delle *news* può subire attacchi di tipo DoS.

Poiché le *news* sono raramente un servizio essenziale, gli attacchi di tipo DoS contro un singolo *server* sono solitamente ignorati. I rischi di sicurezza delle *news* sono quindi abbastanza insignificanti.

Attualmente molti *Web server* consentono agli utenti di accedere al servizio delle *news* attraverso il protocollo HTTP. Questa soluzione non è molto efficiente se un numero elevato di utenti leggono le *news*.

Il protocollo NNTP (*Network News Transfer Protocol*) viene utilizzato per trasferire le *news* attraverso Internet. Nel configurare un proprio *server* delle *news*, si dovrà determinare il modo più sicuro per far giungere le *news* presso i sistemi interni in modo tale che il protocollo NNTP non possa essere utilizzato per attaccare i sistemi stessi.

Il trasferimento, la stampa e la condivisione dei file

La posta elettronica può essere utilizzata per trasferire dati da un sito ad un altro, ma è stata progettata per piccoli *file* in forma testuale.

Anche se gli attuali sistemi di posta elettronica includono delle caratteristiche che consentono di trasferire ingombranti *file* in formato binario, suddividendoli in più parti codificate opportunamente dal mittente, e poi decodificati e riassemblati dal ricevente. Sfortunatamente, tali operazioni sono complesse e possono provocare facilmente degli errori. Inoltre, gli utenti sono interessati a cercare i *file* senza attendere che qualcuno spedisca loro ciò di cui necessitano. Per questi motivi, anche quando la posta elettronica è disponibile, è utile avere un metodo progettato per trasferire *file* a richiesta.

A volte, più che un trasferimento di *file* tra macchine, può essere interessante ed utile disporre di una singola copia del *file* e renderla accessibile a una serie di *client*. In questo caso si parla di condivisione. I protocolli per la condivisione dei *file* possono anche essere utilizzati come protocolli per il trasferimento dei *file*, ma principalmente consentono di usare un *file* come se fosse locale. Solitamente, la condivisione dei *file*

risulta più conveniente per gli utenti, ma poiché offre maggiori funzionalità, è meno efficiente, meno robusta e meno sicura.

La stampa dei *file* è spesso basata sui protocolli per la condivisione o per il trasferimento dei *file*.

Il trasferimento dei file

Il protocollo FTP (*File Transfer Protocol*) è il protocollo Internet standard per i trasferimenti dei *file*. Molti *Web browser* supportano sia FTP che HTTP e usano automaticamente il protocollo FTP per accedere alle locazioni i cui nomi iniziano con ftp://[utente:*password*@]macchina.dominio/*file*.; in tal modo molti utenti usano il protocollo FTP senza neanche accorgersene. In teoria, consentire ai propri utenti il *download* di *file* non implica un aumento dei rischi rispetto all'uso della posta elettronica; infatti, alcuni siti offrono servizi che consentono agli utenti di accedere all'FTP attraverso la posta elettronica.

I principali problemi in molti siti sono relativi al fatto che gli utenti possono scaricare *software* contenente dei *Trojan*. Sebbene questo possa capitare, attualmente le maggiori preoccupazioni sono relative all'installazione di giochi per il *computer*, all'uso di *software* pirata o allo scambio di immagini pornografiche. Anche se questi non sono problemi direttamente collegati alla sicurezza, essi provocano una serie di altri problemi (incluso il consumo di tempo e di spazio su disco e l'introduzione di problemi di natura legale). Si consideri inoltre la possibilità di acquisire virus; i virus infatti potrebbero essere nascosti all'interno dei *file* che gli utenti copiano dal sito.

Seguendo le seguenti semplici regole si può essere sicuri che il traffico FTP in ingresso sia ragionevolmente sicuro:

- Gli utenti vengono educati a diffidare di qualunque *software* che possa essere scaricato attraverso l'FTP.
- Gli utenti vengono informati delle politiche relative al materiale sessuale e all'uso delle risorse dell'organizzazione a cui appartengono.

I servizi FTP anonimi sono un meccanismo estremamente popolare per consentire agli utenti l'accesso remoto ai *file* senza fornire loro la possibilità di avere un accesso completo alla propria macchina. Se si esegue un FTP *server* si può consentire agli utenti di reperire i *file* che sono stati collocati in un'area pubblica del proprio sistema senza consentire loro di accedere a qualunque risorsa del proprio sistema. L'area relativa al proprio *server* FTP anonimo può contenere gli archivi pubblici della propria organizzazione (articoli, *software*, immagini grafiche e informazioni di qualunque altro genere).

Per ottenere l'accesso ai *file* che sono stati resi disponibili, gli utenti effettuano il *log in* nel sistema usando il servizio FTP e servendosi di un *login name* speciale (solitamente *anonymous* o ftp). Molti siti richiedono agli utenti che inseriscano per cortesia il proprio indirizzo di posta elettronica, in risposta al *prompt* della *password*, in modo tale il sito possa tracciare chi sta usando il servizio FTP anonimo, ma questo requisito può essere gestito raramente (soprattutto perchè non c'è un modo per verificare la validità di un indirizzo di posta elettronica).

Installando un servizio FTP anonimo, bisogna essere sicuri che gli utenti che lo usano non possano avere accesso ad altre aree o a *file* del sistema, e che non possano usare

il servizio FTP per ottenere un accesso a livello di *shell* nel sistema stesso.

La condivisione dei file

Sono disponibili diversi protocolli per la condivisione dei *file*, che consentono ai *computer* di usare *file* che sono fisicamente collocati su dischi appartenenti fisicamente ad altri *computer*. Tutto questo è molto vantaggioso, perchè consente agli utenti di usare i *file* remoti senza l'*overhead* di trasferirli avanti e indietro e di cercare di mantenere le versioni sincronizzate. In ogni caso, la condivisione dei *file* è più complessa da implementare rispetto al trasferimento dei *file*. I protocolli per la condivisione dei *file* devono fornire trasparenza (il *file* sembra essere locale, e non ci si rende conto della condivisione) e completezza (si deve poter fare sul *file* remoto tutto ciò che si può fare su un *file* locale). Queste caratteristiche rendono vantaggiosa la condivisione dei *file*, ma la necessità di trasparenza pone dei limiti alla sicurezza, e la necessità di fornire completezza rende i protocolli complessi da implementare. Una maggiore complessità conduce inevitabilmente ad una maggiore vulnerabilità.

I protocolli per la condivisione dei *file* più comunemente utilizzati sono in ambito *Unix* il protocollo NFS (*Network file System*), in ambito *Microsoft* il protocollo CIFS (*Common Internet file System*) e in ambito *Macintosh* il protocollo *AppleShare*. Il protocollo CIFS fa parte di una famiglia di protocolli tra i quali SMB (*Server Message Block*), NetBIOS/NetBEUI e *LanManager*. Sono simili fra loro, in larga parte intercambiabili, e presentano gli stessi problemi per quel che riguarda la sicurezza.

Il protocollo NFS è stato progettato per essere utilizzato in area locale e assume che ci siano bassi tempi di risposta, elevata affidabilità, sincronizzazione temporale ed un elevato grado di fiducia tra le macchine. Se non si configura propriamente il protocollo NFS, un attaccante può essere in grado di accedere facilmente al *file system*. Con il protocollo NFS le macchine client hanno la possibilità di leggere e cambiare i *file* memorizzati in un *server* senza avere la necessità di effettuare il *login* sul *server* o inserire una *password*. Poichè il protocollo NSF non effettua il *logging* delle transazioni, potrebbe accadere che non si scopra mai che qualcuno ha accesso completo ai propri *file*. Il protocollo NSF fornisce un modo che consente di controllare quali macchine abbiano accesso ai propri *file*. Un *file* chiamato /etc/exports ci consente di specificare a quali *file system* si può accedere e quali macchine possano accedervi. Se si lascia un *file system* al di fuori del *file* /etc/exports, nessuna macchina potrà accedervi. Se lo stesso *file* si inserisce in /etc/exports, ma non si specifica quali macchine possono accedervi allora si consente a qualunque macchina di accedervi. Il protocollo NFS ha meccanismi di autenticazione dei *client* molto deboli, e un attaccante può essere in grado di convincere un *server* NFS che una richiesta proviene da uno dei *client* elencati nel *file* /etc/exports. Ci sono anche delle situazioni in cui un attaccante può effettuare l'*hijacking* di accessi NFS esistenti.

Entrambi i protocolli CIFS e *AppleShare* fanno affidamento su meccanismi di autenticazione a livello di utente, invece che a livello di *host*. Tale approccio costituisce un miglioramento per quel che riguarda la sicurezza. *AppleShare* tuttavia non è in grado di supportare dei metodi flessibili per l'autenticazione degli utenti. Si è costretti ad utilizzare *password* riusabili, il che significa che un attaccante può semplicemente limitarsi a catturare le *password*. Il protocollo CIFS fornisce buoni meccanismi di autenticazione e protezione nelle sue versioni recenti. Comunque, le caratteristiche di compatibilità all'indietro del protocollo CIFS aumentano la sua vulnerabilità. Inoltre, allo

stato attuale il protocollo CIFS fornisce una famiglia completamente nuova di servizi, alcuni dei quali più vulnerabili dei servizi relativi alla condivisione dei *file*.

Alcuni protocolli per la condivisione dei *file* sono stati progettati per essere utilizzati su reti quali Internet; ad esempio, il protocollo AFS (*Andrew file System*) usa Kerberos per l'autenticazione ed opzionalmente la cifratura. I protocolli NFS, CIFS ed *AppleShare* sono tutti parte di popolari sistemi operativi, mentre AFS è un prodotto di terze parti. A causa di questo e poichè AFS e Kerberos richiedono un'esperienza tecnica significativa per essere installati e manutenuti, il protocollo AFS non viene molto utilizzato.

La stampa dei file

Quasi tutti i sistemi operativi oggi forniscono la possibilità di stampare da remoto, attraverso lp o lpr sulle macchine *Unix*, attraverso il sistema di stampa SMB sulle macchine *Windows* oppure mediante i servizi di stampa *AppleTalk* sulle macchine *Macintosh*. La stampa da remoto consente ad un *computer* di stampare su una stampante che è fisicamente connessa ad un altro *computer*, oppure direttamente alla rete. Tale prestazione è molto vantaggiosa in una rete locale, ma tutte le opzioni di stampa da remoto sono non sicure e risultano inefficienti quando vengono utilizzate in reti geografiche. Se si ha la necessità di stampare su un sito attraverso Internet o consentire ad un altro sito di utilizzare la propria stampante, è possibile impostare uno speciale alias di posta elettronica che stampi il messaggio di posta elettronica non appena riceva il messaggio.

Sicurezza degli accessi remoti

In molti casi, anche per ragioni di performance è necessario che l'elaborazione dei dati avvenga in modalità *server-side*.

In origine, i programmi che fornivano un accesso remoto a *server/mainframe* consentivano agli utenti di utilizzare un sistema remoto come se il proprio *computer* fosse un terminale locale del sistema. Attualmente troviamo sistemi di elaborazione che supportano l'accesso remoto senza richiedere funzionalità di terminale ai *client*.

Il protocollo *Telnet* è il protocollo standard per l'accesso remoto in Internet con modalità di emulazione di terminale alfanumerico. Il protocollo *Telnet* in principio è stato considerato come un servizio abbastanza sicuro poichè richiede agli utenti di autenticarsi. Sfortunatamente, *Telnet* spedisce tutte le proprie informazioni in chiaro, il che lo rende estremamente vulnerabile ad attacchi di tipo *sniffing* e *hijacking*. *Telnet* è sicuro solamente se la macchina remota e tutte le reti tra la macchina remota e le altre sono sicure. Ciò significa che *Telnet* non è sicuro attraverso Internet.

Per risolvere i problemi del protocollo *Telnet* ci sono due modi. Il primo prevede di utilizzare in alternativa un programma che faccia uso della cifratura; lo standard Internet comunemente accettato in questo caso è SSH (*Secure SHell*) che fornisce una varietà di servizi di accesso remoto cifrati. Il secondo prevede l'attivazione di una connessione di rete cifrata (cioè una VPN, *Virtual Private Network*) ed eseguire *Telnet* utilizzando tale connessione.

Altri programmi come *rlogin, rsh* sono utilizzati per fornire accessi remoti senza necessità di autenticarsi nuovamente.

Interfacce grafiche remote per sistemi operativi Microsoft

Spesso l'utente preferisce che l'accesso ad una macchina remota avvenga mediante un'interfaccia grafica, piuttosto che a linea di caratteri.

Microsoft fornisce un'interfaccia grafica remota come parte dei *server Windows* 2000 in un *package* chiamato *Terminal Services*. (Esiste anche una versione di *Windows* NT 4.0 chiamata *Terminal Server*). Sia i *Terminal Services* di Windows 2000 che *Terminal Server* di Windows NT 4.0 impiegano un protocollo sviluppato dalla *Microsoft* chiamato RDP (*Remote Desktop Protocol*) per le comunicazioni tra *client* e *server* (impiega connessioni TCP con *port* 3389 sul *server*).

Diversi protocolli proprietari vengono utilizzati per realizzare interfacce grafiche remote per *Windows*, tra questi il più potente e il più utilizzato è il protocollo ICA (*Indipendent Computing Architecture*) sviluppato da *Citrix*. Il protocollo ICA è stato adottato da diversi produttori.

Molti programmi quali *LapLink, RemotelyPossible* e *PcANYWHERE* rendono possibile l'accesso remoto attraverso i protocolli TCP/IP. Anche BO2K (*Back Orifice* 2000) è un programma gratuito che fornisce accesso remoto e può essere considerato uno dei *tool* più potenti che sia mai stato realizzato per l'amministrazione remota, sebbene venga spesso citato nell'ambito di attacchi e intrusioni a sistemi informatici.

I Window System di rete

Molte macchine *Unix* attualmente forniscono dei *Window System* basati sul sistema X11. I *server* X11 sono anche disponibili come applicazioni di terze parti per molti altri sistemi operativi, incluse tutte le versioni di *Windows* e molte versioni di MacOS. I *client* X11 sono abbastanza rari ma anch'essi sono disponibili per *Windows* NT. L'accesso alla rete è un'importante caratteristica del sistema X11.

I *server* X11 sono bersagli appetibili per gli attaccanti. Un attaccante che guadagna l'accesso ad un *server* X11 può causare i seguenti danni:

- Catturare il *dump* dello schermo, in questo modo si riesce a leggere qualunque cosa sia visualizzata sullo schermo dell'utente.
- Leggere i tasti premuti dall'utente.
- Emulare la pressione dei tasti da parte dell'utente.

In passato, il sistema X11 usava principalmente come strumento di autenticazione l'indirizzo da cui provenivano le richieste di connessione. Oggi molti *server* X11 implementano strumenti di autenticazione più sicuri. Ad ogni modo, il sistema X11, come il protocollo *Telnet* è ancora vulnerabile ad attacchi di tipo *hijacking* e *sniffing*.

Sicurezza telematica - tecniche di attacco

Franco Callegati
Paolo Zaffoni

8.4.1 (Spiegare i principali aspetti della sicurezza connessi alla trasmissione dei dati), 8.4.3 (Descrivere le funzioni e le caratteristiche di un firewall)

Tecniche di attacco

Il più comune attacco contro una rete o una organizzazione è basato sulle intrusioni. Attraverso le intrusioni, gli attaccanti possono utilizzare i *computer* che non appartengono loro.

Gli attaccanti hanno a disposizione diverse tecniche per penetrare all'interno di un *computer*. Tali tecniche variano da attacchi di tipo *Social Engineering* (si conosce il nome di un certo utente e si telefona all'amministratore della rete dicendo che è necessario cambiare la *password* per l'utente che si finge di impersonare) ad attacchi basati sui tentativi di indovinare *username* e *password* di un certo utente.

Denial of Service

Un attacco di tipo *Denial of Service* è un attacco che impedisce ad una organizzazione di usare i servizi della propria rete. Sebbene molti casi di sabotaggio elettronico implichino la reale distruzione dei componenti di elaborazione o lo *shutdown* dei sistemi di elaborazione, molto spesso tali attacchi possono essere ricondotti ad azioni di *flooding* ("inondazione": un attaccante spedisce ad un sistema o ad una rete una lunga sequenza di messaggi in modo da occupare interamente o quasi la CPU ed altre risorse del sistema). Negli attacchi di tipo *flooding* infatti, il sistema spende la maggior parte del tempo a rispondere ai messaggi.

Mentre il *flooding* è il modo più semplice più comune per realizzare un attacco DoS, un metodo più intelligente potrebbe disabilitare i servizi, reindirizzarli o rimpiazzarli.

In un certo senso è quasi impossibile evitare attacchi DoS. Molto spesso, il rischio di attacchi DoS è inevitabile. Gli attacchi di tipo *flooding* sono considerati poco interessanti per gli attaccanti, perché risultano troppo semplici.

Furto di informazioni

Alcuni tipi di attacchi consentono ad un attaccante di ottenere delle informazioni anche senza dover utilizzare direttamente un *computer* dell'organizzazione che intende attaccare. Solitamente tali attacchi sfruttano dei servizi Internet che sono stati configurati per fornire informazioni, inducendoli a fornire più informazioni di quelle previste oppure a fornirle alle persone sbagliate.

I furti di informazioni possono essere di tipo attivo o di tipo passivo; nel caso attivo, un attaccante può ottenere le informazioni effettuando delle *query* ad un *server*; nel caso passivo, un attaccante può ottenere le informazioni catturando il traffico che interessa un segmento di rete da lui controllato.

La cattura di informazioni può avvenire ad esempio catturando i dati di *username* e *password* che transitano su un segmento di rete al quale l'attaccante è collegato ed ha

attivato un programma di *sniffing*.

Ci sono diversi tipi di protezione contro il furto di informazioni. Un *firewall* propriamente configurato può aiutare a proteggere le informazioni che si intendono fornire all'esterno.

IP spoofing 1

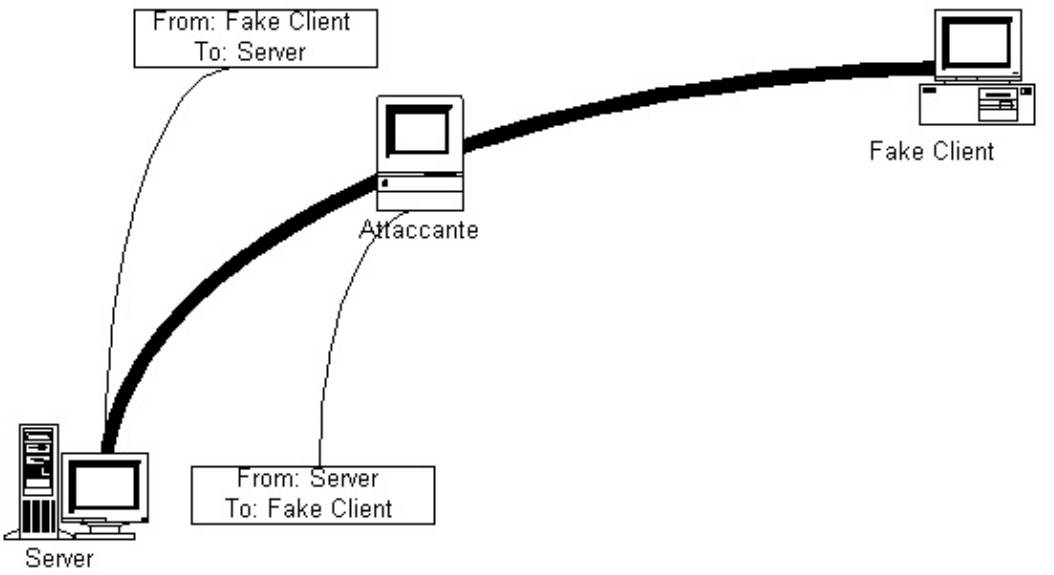
In un attacco di tipo *spoofing*, un attaccante spedisce pacchetti con un campo origine errato. Quando ciò accade, le risposte vengono inviate all'indirizzo di origine fittizio e non all'attaccante. Questo potrebbe sembrare un problema, ma in realtà ci sono tre casi in cui l'attaccante non si interessa di questo:

- l'attaccante può intercettare le risposte;
- l'attaccante non necessita di vedere le risposte;
- l'attaccante non è interessato alle risposte.

IP spoofing 2

L'attaccante può intercettare le risposte.

Se l'attaccante si trova in qualche punto nella rete tra la destinazione e l'origine, l'attaccante può vedere le risposte e continuare la conversazione fin quando lo ritenga necessario.

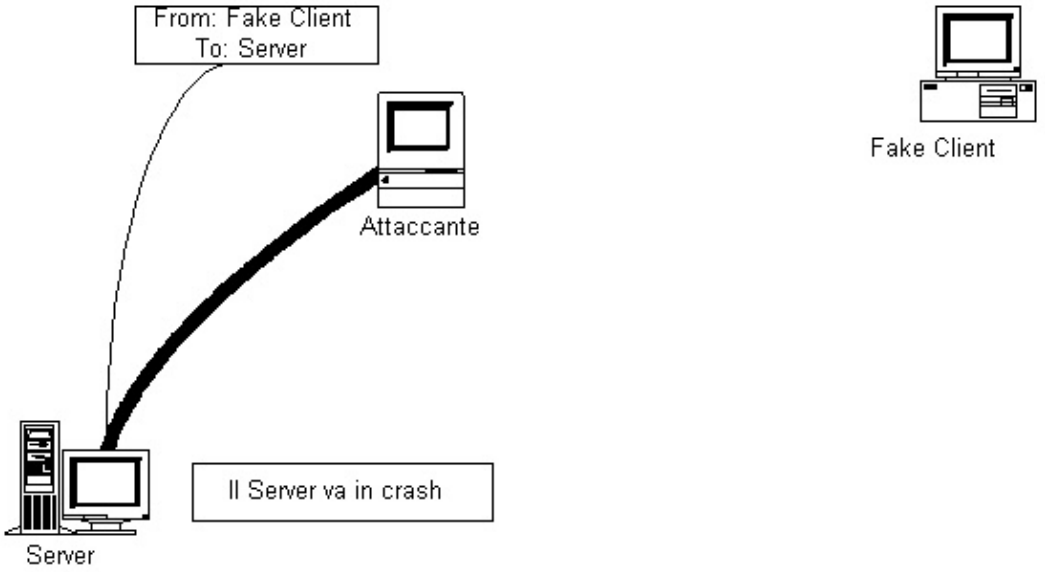


IP spoofing 3

L'attaccante non necessita di vedere le risposte.

Un attaccante potrebbe condurre un attacco di tipo DoS, la macchina attaccata non

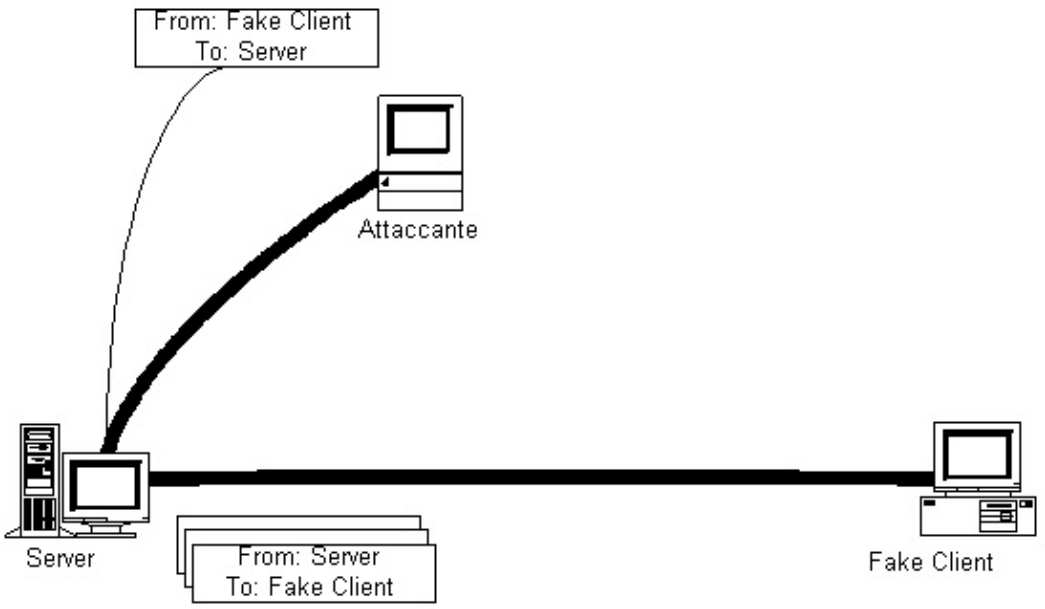
sarebbe in grado di rispondere comunque.



IP spoofing 4

L'attaccante non è interessato alle risposte.

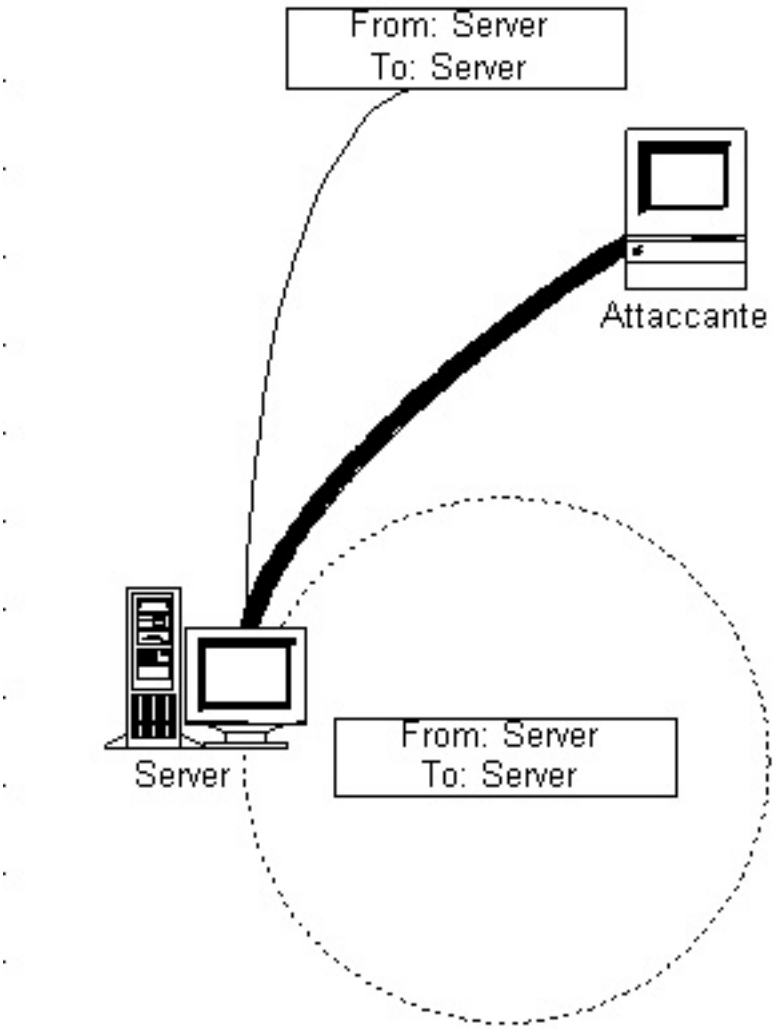
Alcuni attacchi sfruttano il fatto che le risposte raggiungono un altro *client* (ignaro).



L'attacco di tipo *smurf* utilizza indirizzi di origine fittizi per attaccare l'*host* che sembra l'origine; un attaccante invia ad un *server* (la vittima fittizia) un pacchetto con indirizzo di origine uguale a quello di un altro *host* (la vittima reale). A questo punto, gli amministratori della vittima reale e di quella fittizia iniziano a sospettare ciascuno

dell'altro.

Un altro tipo di attacco, chiamato attacco di tipo *land*, spedisce un pacchetto con indirizzo di origine uguale a quello di destinazione; tale pacchetto molte volte provoca un *loop*.



Port scanning

Il *port scanning* è il processo con cui si cercano i *PORT* su cui un *host* è in ascolto, al fine di individuare cosa si possa attaccare. Il *port scanning* incrementale (si inviano pacchetti TCP/IP con numeri di porta crescenti) è molto semplice da individuare, e quindi gli attaccanti cercano di camuffarlo con diverse tecniche. Ad esempio, molte macchine non effettuano il *logging* delle connessioni fino a quando non sono state interamente completate, quindi un attaccante potrebbe spedire un pacchetto iniziale con SYN uguale ad 1 e ACK uguale a 0, attendere la risposta (SYN e ACK uguali ad 1 (se il PORT è attivo;, RST uguale ad 1 se il PORT è chiuso) e bloccarsi. Questa tecnica è comunemente chiamata SYN *scan* oppure *Half open scan*.

Gli attaccanti possono anche trasmettere altri tipi di pacchetti, considerando la porta chiusa se ricevono un pacchetto RST, considerandola aperta se non ottengono risposta oppure se ottengono altri messaggi di errore. A volte possono essere utilizzati tutti i *flag*; in questo caso la tecnica in cui tutti i *flag* sono impostati viene chiamata *Christmas tree* mentre quella in cui tutti i flag resettati viene chiamata *null*.

Sniffing

La tecnica che consente di leggere i pacchetti quando attraversano la rete viene comunemente chiamata *packet sniffing*. Se si inviano informazioni non cifrate in rete, lo *sniffing* è uno strumento di utilizzo immediato per la loro cattura.

Il modo più semplice per applicare lo *sniffing* è controllare una macchina che si trova in una posizione privilegiata rispetto al traffico che ci interessa, ad esempio un *router* oppure un *server* (tipicamente tali macchine sono ben protette, anche dal punto di vista fisico, per cui ci si riduce ad attaccare macchine meno sicure).

Architetture per reti sicure

Franco Callegati
Paolo Zaffoni

8.4.1 (Spiegare i principali aspetti della sicurezza connessi alla trasmissione dei dati), 8.4.3 (Descrivere le funzioni e le caratteristiche di un firewall)

Terminologia

Non esiste una terminologia completa e consistente per le architetture e componenti di *firewall*. Per quanto riguarda i *firewall* sicuramente si può schematizzare quanto segue:

- *Firewall*: un componente o un insieme di componenti che limitano l'accesso tra una rete protetta ed Internet.
- *Host*: un *computer* connesso ad una rete.
- *Bastion host*: un *computer* che deve essere reso molto sicuro in quanto potrebbe essere oggetto di attacchi.
- *Dual-homed host*: un *computer* che ha almeno due interfacce di rete.
- *Network address translation*: una procedura mediante la quale un *router* modifica i pacchetti che lo attraverso cambiando gli indirizzi di rete in base ad una opportuna politica.
- Pacchetto: l'unità fondamentale di comunicazione in Internet.
- *Packet filtering*: l'azione intrapresa da un dispositivo per controllare in maniera selettiva il flusso dei dati proveniente e/o diretto verso la rete.
- Rete perimetrale: una rete aggiunta (interposta) tra una rete protetta ed una rete esterna (Internet) al fine di fornire un ulteriore livello di sicurezza. Una rete perimetrale viene qualche volta chiamata DMZ, *De-Militarized Zone* (Zona DeMilitarizzata, riferimento alla zona che separa le due Coree).
- *Proxy*: un'applicazione *software* che dialoga con *server* esterni per conto dei *client* interni.
- *Virtual Private Network* o VPN: una rete che trasporta pacchetti, appartenenti ad una rete privata implementata sull'infrastruttura pubblica, che non possono essere decifrati dagli attaccanti.

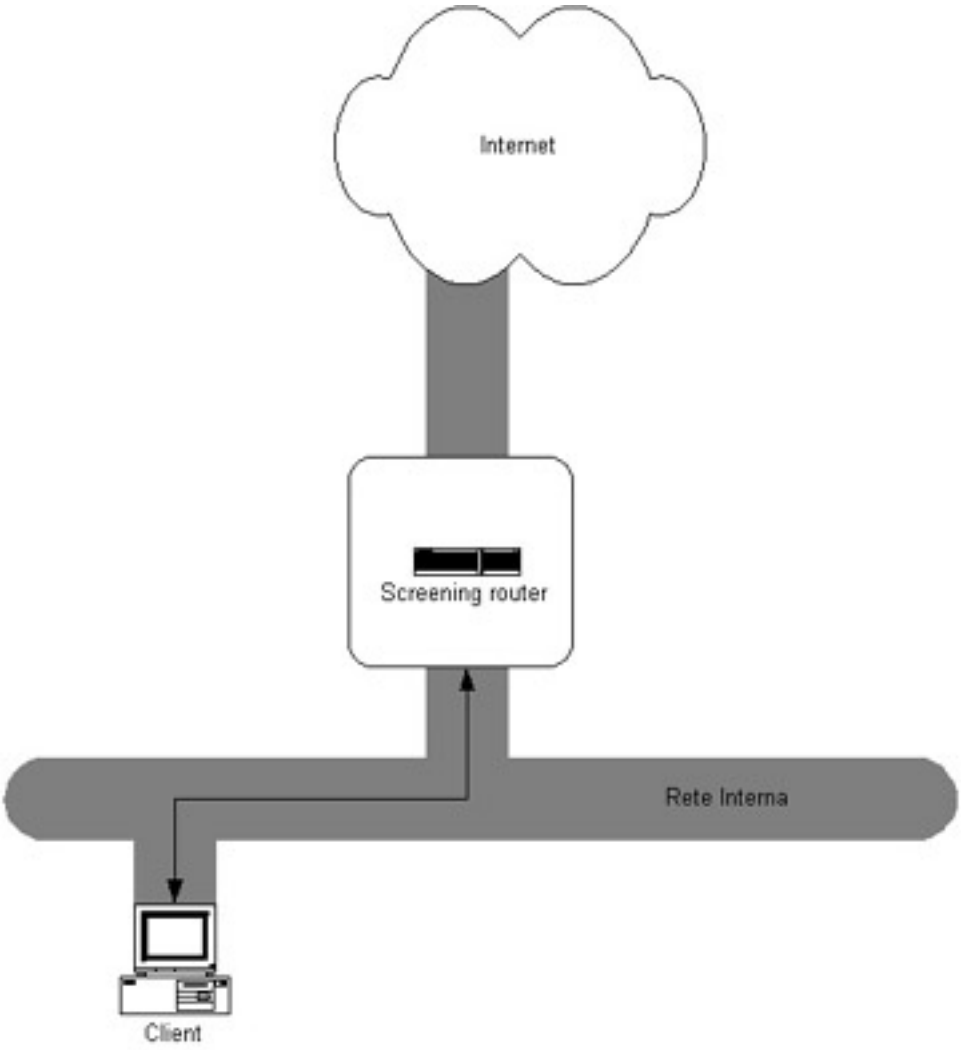
Packet filtering firewall 1

I sistemi per il *packet filtering* instradano in maniera selettiva i pacchetti tra *host* interni ed *host* esterni, vietando il passaggio a determinati tipi di pacchetti in conformità con la politica di sicurezza dell'organizzazione di appartenenza. Il *router* utilizzato come *packet filtering firewall* viene comunemente chiamato *screening router*.

Le informazioni di un pacchetto, elaborate dal *packet filter* sono:

- l'indirizzo IP di origine;
- l'indirizzo IP di destinazione;
- il protocollo a cui si riferisce il contenuto (TCP, UDP, ICMP, eccetera);
- la porta di origine TCP o UDP;
- la porta di destinazione TCP o UDP;
- il tipo di messaggio ICMP;
- la dimensione del pacchetto.

Il *router* è anche in grado di analizzare la parte dati di un pacchetto. Il *router* può anche assicurarsi che il pacchetto sia valido, impedendo la possibilità di attacchi basati su pacchetti incorretti.



Il *router* conosce altre informazione relative al pacchetto che si riferiscono a strati più bassi dell'architettura di comunicazione:

- l'interfaccia da cui proviene il pacchetto;
- l'interfaccia a cui è destinato il pacchetto.

Infine, un *router* può tenere traccia di una connessione, memorizzando, ad esempio:

- i pacchetti che costituiscono la risposta ad altri;
- il numero di pacchetti trasmessi o ricevuti da un *host*;
- eventi che indicano l'uguaglianza di pacchetti ricevuti in momenti diversi altro pacchetto;
- eventi che indicano la ricezione di pacchetti frammentati.

Packet filtering firewall 2

Per comprendere il funzionamento del *packet filtering* analizziamo le differenze tra un *router* ordinario e uno *screening router*. Un *router* ordinario controlla semplicemente l'indirizzo IP di destinazione di ogni pacchetto e seleziona il *path* ottimale per raggiungere la rete di destinazione, in base a politiche di *routing* preimpostate, in base all'indirizzo di destinazione, in base al contenuto delle tabelle di *routing*.

Uno *screening router* analizza i pacchetti molto più attentamente, decidendo se instradarlo o meno in base alle politiche di sicurezza dell'organizzazione a cui appartiene.

Le tecniche di *packet filtering* possono anche essere implementate da dispositivi che svolgono funzioni di *routing*; tali apparati prendono il nome di *packet filtering bridge*.

Una volta esaminate le informazioni di interesse, uno *screening router* compie una delle seguenti azioni:

- spedisce (*Permit*) il pacchetto verso la destinazione;
- scarta (*Drop*) il pacchetto, senza notificare l'evento al mittente;
- rifiuta (*Reject*) il pacchetto, ed invia un messaggio di errore al mittente;
- effettua il *logging* del pacchetto (registra l'evento);
- attiva un allarme per notificare ad un sistema di supervisione (*console* dello *screening router*) la presenza del pacchetto.

Router più sofisticati sono anche in grado di:

- modificare il pacchetto (ad esempio, per effettuare il *network address translation*);
- spedire il pacchetto ad una destinazione diversa da quella prevista (ad esempio, per forzare le transazioni attraverso un *proxy* oppure per effettuare un *load balancing*);
- modificare le regole di *filtering* (ad esempio, per bloccare tutto il traffico proveniente da un sito che ha spedito pacchetti ostili).

Packet filtering firewall 3

Una importante regola di filtraggio prevede l'utilizzo dei numeri di *port* associati ai servizi Internet e i *Flag* contenuti nei pacchetti TCP.

I dispositivi per il *packet filtering* che tengono traccia dei pacchetti analizzati sono di solito chiamati *statefull packet filter*, o *packet filter* dinamici, in quanto memorizzano lo stato delle connessioni, ovvero modificano il proprio comportamento in base alla storia del traffico. I filtri che operano azioni anche sul campo dati del pacchetto sono frequentemente chiamati *packet filter* intelligenti.

Un sistema per il *packet filtering* può costituire il punto in cui vengono forniti i servizi per realizzare reti private virtuali (VPN) e per il *network address translation*. Poiché il *packet filter* già analizza i pacchetti, può facilmente identificare i pacchetti che sono destinati ad un particolare *host* che si trovi nella VPN, cifrare tali pacchetti e spedirli verso la destinazione.

Packet filtering firewall 4

Vantaggi del packet filtering

- Un solo *screening router* può aiutare a proteggere una rete intera, purché ben collocato nella topologia della rete stessa.
- Semplici tecniche di *packet filtering* risultano molto efficienti. Poiché il *packet filtering* richiede l'elaborazione di un numero limitato di campi dei pacchetti, può essere utilizzato con un minimo *overhead*. L'uso di un *proxy* implica invece operazioni più complesse.
- Il *packet filtering* è disponibile in molti prodotti sia commerciali, sia *freeware*.

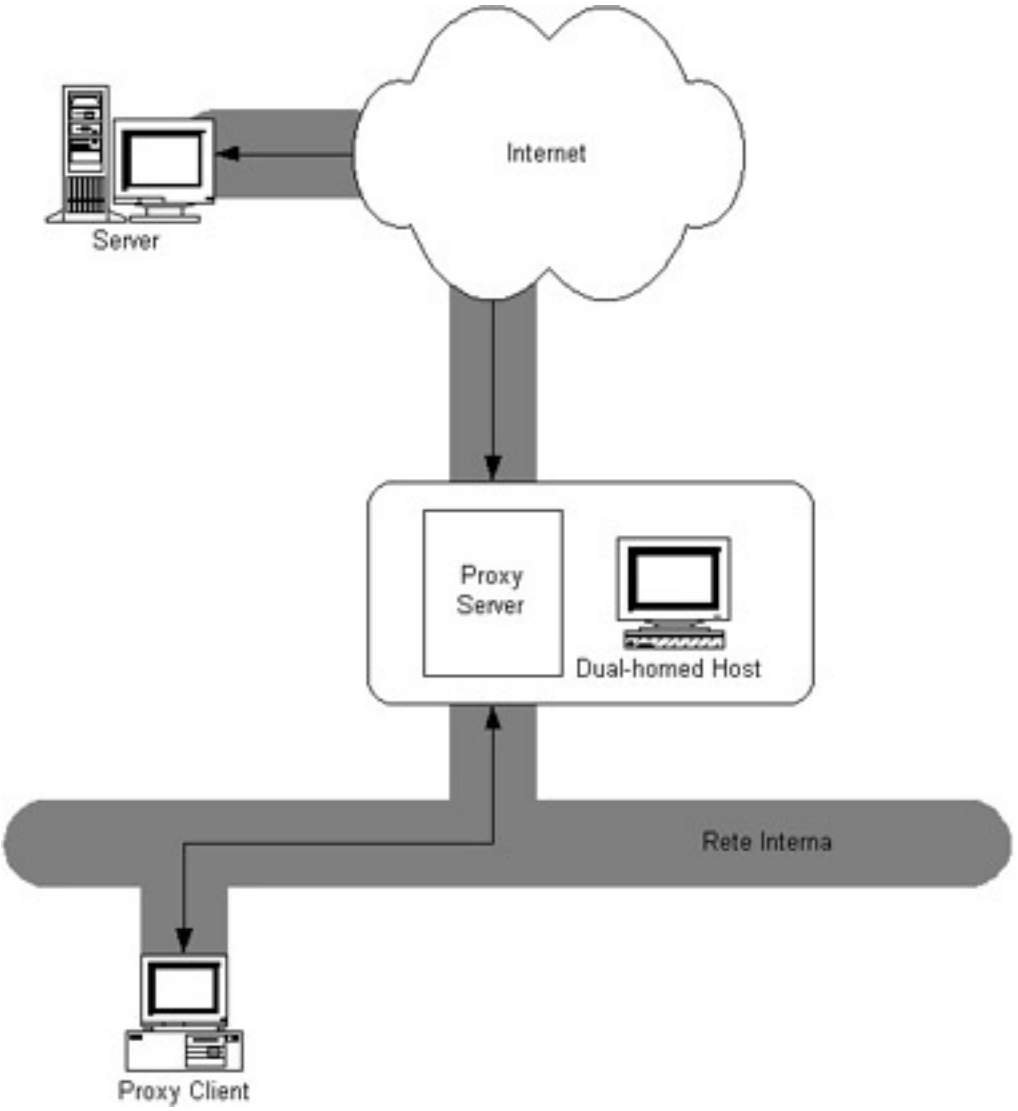
Packet filtering firewall 5

Svantaggi del packet filtering

- Gli strumenti di *packet filtering* non sono semplici da configurare, per via della complessità delle regole e delle difficoltà operative che occorre affrontare in fase di *testing*.
- Il *packet filtering*, se attivato su macchine per il *routing*, riduce le prestazioni, poiché determina un carico elaborativo aggiuntivo sul *router*.
- Alcune politiche non possono essere rafforzate da *screening router*. Ad esempio, i pacchetti possono essere associati agli *host* che li spediscono non agli utenti che ne hanno richiesto la trasmissione.

Proxy 1

Un *Proxy* è un oggetto che esegue delle azioni al posto di un altro oggetto.



Si tratta di applicazioni specializzate che ricevono richieste di servizi Internet da parte degli utenti e le inviano ai *server* reali. I sistemi *Proxy* possono essere utilizzati sia per ragioni di sicurezza, sia per ragioni di *performance*, sia per motivi di necessità.

I sistemi *Proxy* si collocano, più o meno trasparentemente, tra una rete di *client* interni ed i *server* esterni (es. Internet). La trasparenza è il maggior beneficio di un sistema *Proxy*; l'utente durante la navigazione sui *server Web*, ad esempio non percepisce la presenza del *proxy*.

Il *Proxy* opera funzioni di filtraggio; non inoltra cioé sempre le richieste all'esterno, soprattutto se la politica di sicurezza dell'organizzazione a cui appartiene prevede l'inibizione di alcuni siti *Web* o altro.

Proxy 2

Vantaggi dei sistemi Proxy

- Un sistema *Proxy* è adatto per il *logging*. Poiché un *Proxy* può comprendere il protocollo applicativo, tale sistema può effettuare un *logging* più efficiciente e completo.
- Un sistema *Proxy* consente operazioni di *caching*. Poiché tutte le richieste passano attraverso il *Proxy*, tale sistema può mantenere una copia locale dei dati richiesti. Se il numero delle richieste che si ripetono è significativo, allora il *caching* può migliorare le prestazioni.
- Un sistema *Proxy* consente una autenticazione a livello di utente.
- Un sistema *Proxy* è in grado di effettuare tecniche di *content filtering*.

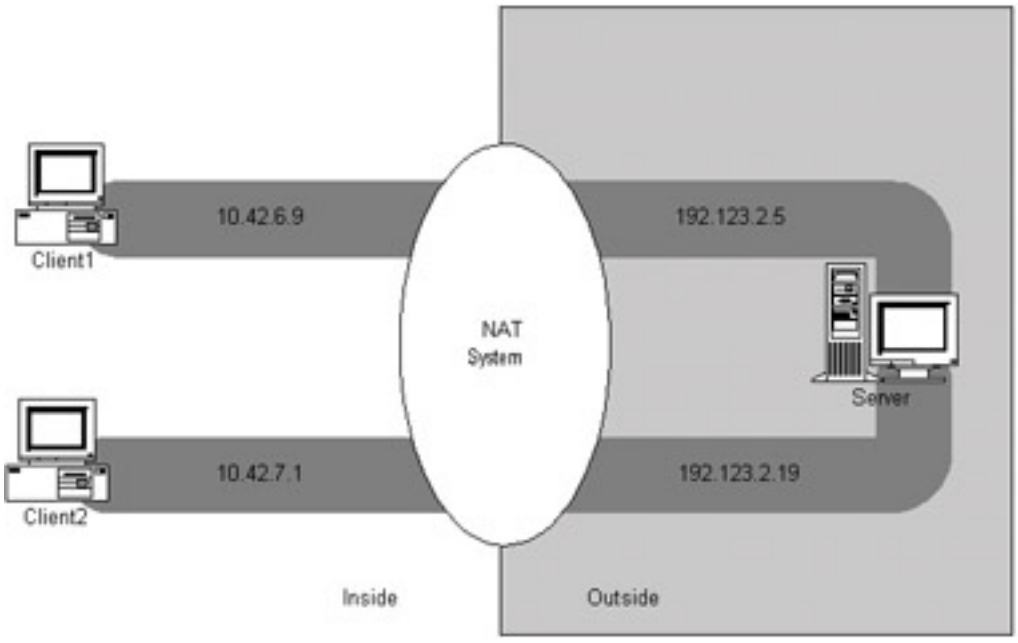
Proxy 3

Svantaggi dei sistemi Proxy

- Non tutti i servizi sono stati progettati per essere utilizzati facilmente attraverso un *Proxy*.
- I sistemi *Proxy* richiedono un differente applicativo *Proxy server* per ogni servizio.
- I sistemi *Proxy* solitamente richiedono di apportare modifiche ai *client*, alle applicazioni o alle procedure.

Network address translation 1

La prestazione *network address translation* consente ad una rete di usare internamente un insieme di indirizzi differente da quello utilizzato verso l'esterno. Un sistema per il NAT non fornisce sicurezza, ma aiuta a nascondere la struttura della rete interna e forza le connessioni a passare attraverso un *choke point*.

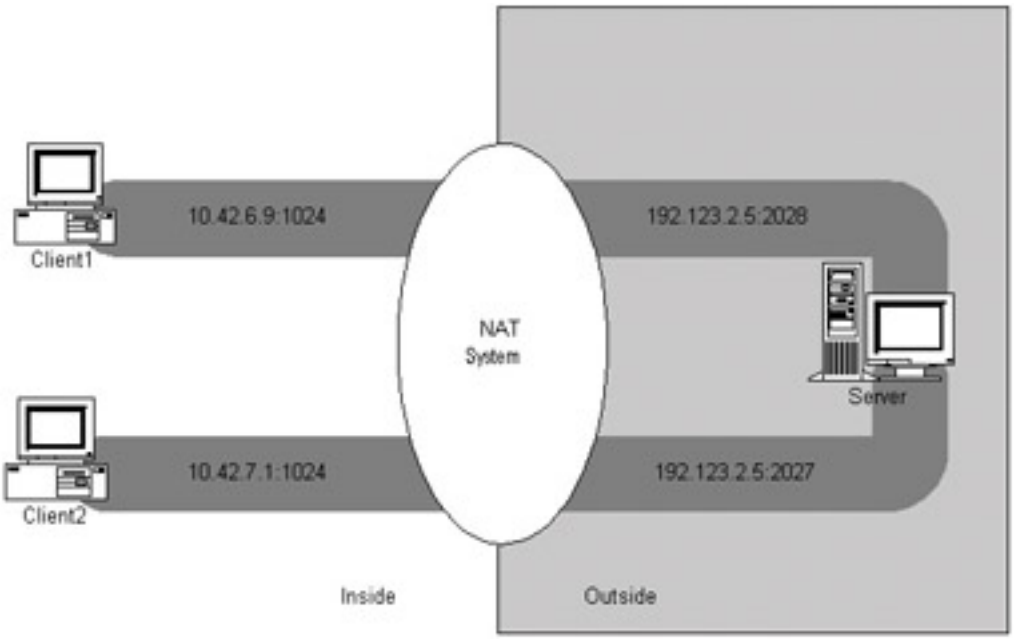


Come nel caso del *packet filtering*, anche il NAT richiede ad un *router* (in generale ad un *server*), di eseguire elaborazioni sui pacchetti. Nel caso di un *router* con funzionalità

di NAT *server*, i pacchetti vengono analizzati, modificati ed istradati

Network address translation 2

Il NAT *server* è in grado di modificare anche i numeri di porta di origine e di destinazione; in questo caso viene chiamato PAT (*Port and Address Translation*).



I sistemi per il NAT possono usare differenti schemi per tradurre tra indirizzi interni ed indirizzi esterni:

- Assegnare staticamente un indirizzo esterno per ogni *host* interno, senza modificare i numeri di *port*. Questo approccio non fornisce alcun risparmio per quel che riguarda lo spazio di indirizzamento ed inoltre rallenta le prestazioni.
- Assegnare dinamicamente un indirizzo esterno ogni volta che un *host* interno inizia una connessione, senza modificare i numeri di porta.
- Creare un *mapping* fisso tra indirizzi interni ed indirizzi esterni, ma usare il *port mapping* per consentire a più macchine interne di usare uno stesso indirizzo esterno.
- Allocare dinamicamente una nuova coppia di indirizzo esterno-porta ogni qual volta che un *host* interno inizia una connessione.

Network address translation 3

Vantaggi del network address translation

Il *network address translation* consente di economizzare sul numero degli indirizzi esterni (pubblici) e determina vantaggi per la sicurezza:

- Migliora il controllo sulle connessioni in uscita, poiché tutti gli *host*

possiedono un indirizzo che non funziona sulla rete esterna (tipicamente indirizzi interni sono quelli di classe C, B, A, di tipo privato).
- Limita il traffico in ingresso.
- Maschera la configurazione della rete interna. Tanto meno un attaccante conosce di una rete, tanto più la rete è sicura. Un sistema per il NAT rende molto difficile ad un attaccante la possibilità di determinare quanti *computer* comprende la rete, che tipo di macchine siano e le modalità di interconnessione.

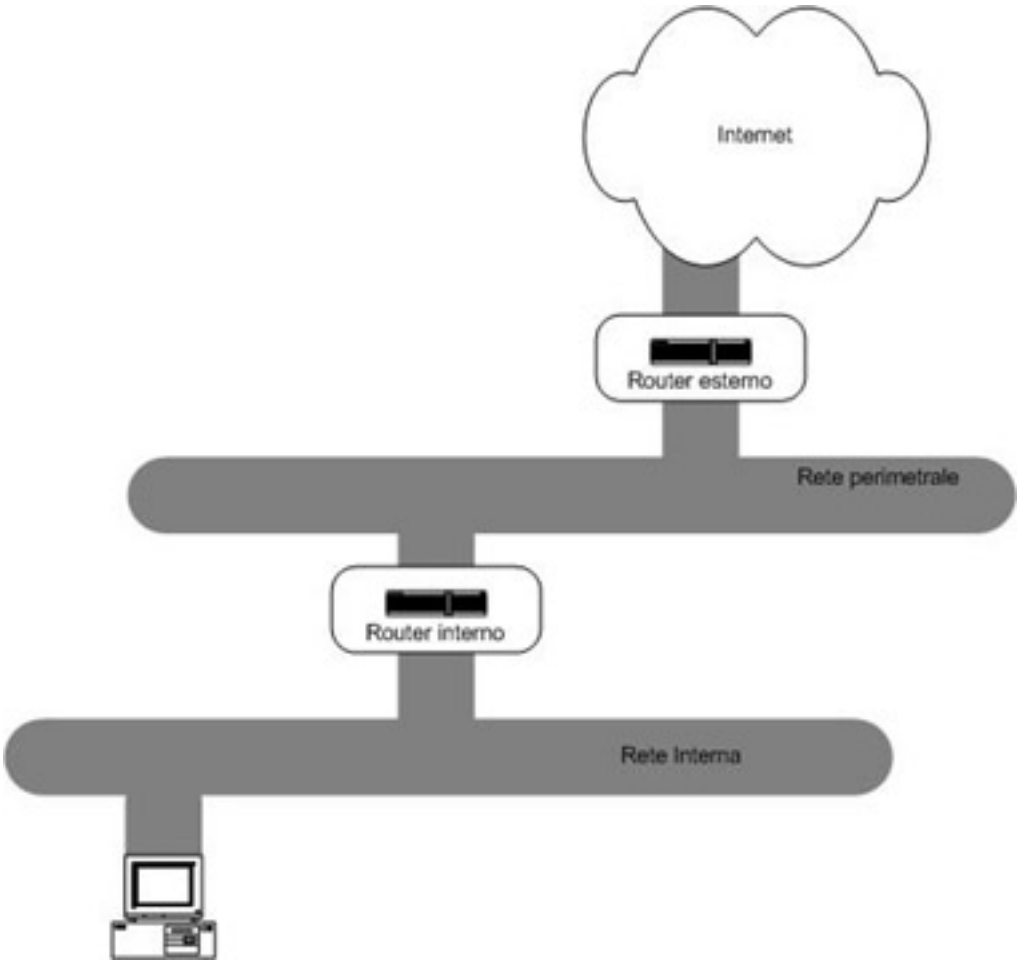
Network address translation 4

Svantaggi del network address translation

- L'allocazione dinamica richiede informazioni sullo stato che non sono sempre disponibili. Risulta molto semplice ad un sistema per il NAT sapere quando un *host* ha terminato di usare una connessione TCP, ma non esiste alcun modo per stabilire se un pacchetto UDP faccia parte di una nuova transazione o meno.
- Il NAT interferisce con alcuni sistemi di cifratura ed autenticazione. I sistemi che cifrano i dati spesso tentano di garantire anche la loro integrità in modo tale da assicurarsi che nessuno modifichi i pacchetti mentre sono in transito verso la destinazione.
- Non tutte le applicazioni funzionano correttamente se mediate dal NAT *server*.
- L'allocazione dinamica degli indirizzi interferisce con il *logging*. I *log* mostrano infatti gli indirizzi tradotti e quindi è necessario correlare le informazioni dei *log* con quelle gestite dal sistema per il NAT per poter interpretare correttamente gli eventi registrati.
- L'allocazione dinamica delle porte interferisce con il *packet filtering*. I sistemi per il *packet filtering* utilizzano le porte di origine e di destinazione per capire quale protocollo è coinvolto nell'interazione. Modificare la porta di orgine può influire sull'accettabilità dei pacchetti.

DMZ 1

L'architettura DMZ (*De-Militarized Zone*) aggiunge un ulteriore livello di sicurezza (rete perimetrale) all'architettura in cui si usa un *packet filtering*. La rete perimetrale infatti isola la rete interna dalla rete Internet. Il modo più semplice per realizzare una DMZ è quello di utilizzare due *screening router*. Uno risiede tra la rete perimetrale e la rete interna (*router* interno o *choke router*) ed un altro risiede tra la rete perimetrale e la rete esterna (*router* esterno o *access router*). Per penetrare all'interno della rete privata è necessario violare due *router*.



La rete perimetrale comporta quindi un livello di sicurezza maggiore. Se qualcuno penetra all'interno di un *host* che si trova nella rete perimetrale può catturare solamente il traffico presente nella rete perimetrale, senza poter analizzare il traffico riservato della rete interna.

DMZ 2

router interno

Il *router* interno (o *choke router*) protegge la rete interna sia dagli attacchi provenienti da Internet che da quelli provenienti dalla rete perimetrale. Tale *router* effettua la maggior parte del *packet filtering*. Consente l'uso dei servizi esterni da parte dei siti interni (traffico di tipo *outbound*). Tali servizi possono essere tranquillamente utilizzati dall'organizzazione e facilmente gestiti da un *packet filter* piuttosto che da un *proxy server*. Il *router* interno permette anche l'accesso ai servizi attivi nella rete perimetrale, che potrebbero essere anche distinti da quelli a cui si accede nella rete esterna.

router esterno

Il *router* esterno (o *access router*) protegge sia la rete perimetrale che quella interna dagli attacchi provenienti da Internet. Permette il passaggio della maggior del traffico

outbound ed effettua controlli solo sul traffico in ingresso (traffico *inbound*). Il tipo di attacco che solitamente viene controllato da tale *router* è quello relativo all'IP *spoofing*.

Bibliografia

Libri

Infrastrutture per reti di calcolatori

A. S. tanembaum*Reti di computer - terza edizione;* 1998UTET/Prentice Hall

W. Stallings*Local and Metropolitan Area Networks - sixth edition;* 2000Prentice Hall

F. Wilder*A Guide to the TCP/IP Protocol Suite - second edition;* 1998Artech House

S. Gai*Reti locali: dal cablaggio all'internetworking;* 1995Scuola superiore "G. Reiss Romoli"

IEEE*Std 802: Overview and Architecture;* IEEE

ISO/IEC*8802.3;* ISO

IBM*Token-Ring Network: Architecture Reference;* IBM

ISO/IEC*IEEE Std 802.5 Overview and Architecture;* ISO

R. Perlman*Interconnections: Bridges, Routers, Switches, and Internetworking Protocols - second edition;* 1999Addison Wesley

Stalling*ISDN and Broadband ISDN with Frame Relay and ATM;* 2002Prentice Hall

R. P. Davidson*Internetworking LANs: Operation, Design and Management;* 1992Artech House

A cura di S. Giorcelli (TILAB)*Collana ATM;* 1996UTET

S. Gianotti*ATM;* 1998HOEPLI

O. Kyas*ATM networks;* 2002Prentice Hall

Reti di computer in tecnica TCP/IP

U. Black*Internet Architecture: An Introduction to IP Protocols;* 2000Prentice Hall

A. G. Blank*TCP/IP Jumpstart: Internet Protocol Basics - second edition;* 2002Sybex

D. Comer*Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture - fourth edition;* 2000Prentice Hall

B. A. Forouzan*TCP/IP Protocol Suite - third edition;* 1999McGraw Hill

W. R. Stevens*The Protocols (TCP/IP Illustrated, Volume 1);* 1994Addison Wesley

Reti di computer in tecnica Windows 2000

Microsoft*MCSE Training Kit - Microsoft Windows 2000 Professional;* 2000Microsoft Press

Microsoft*MCSE Training Kit - Microsoft Windows 2000 Server;* 2000Microsoft Press

Microsoft*Microsoft Windows 2000 Server Resource Kit;* 2000Microsoft Press

Microsoft*Microsoft Windows 2000 Server Administrator's Companion;* 2000Microsoft Press

Gestione di reti

Dr. Feit*SNMP, a guide to Network Management;* 1993McGraw Hill

W. Stallings*SNMP, SNMPv2, SNMPv3, and RMON 1 and 2 - third edition;* 1999Addison-Wesley

Divakara K. Udupa*TMN Telecommunications Management Network;* 1999McGraw Hill

Sicurezza e gestione della sicurezza

B. Schneier*Secrets and Lies: Digital Security in a Networked World;* 2000John Wiley < Sons

Giustozzi*Segreti spie codici cifrati;* 1999Apogeo

Stuart McClure*Hacking Exposed: Network Security Secrets < Solutions - third edition;* 2001Osborne McGraw Hill

B. Schneier*Applied Cryptography: Protocols, Algorithms, and Source Code in C - second edition;* 1995Wiley Computer Publishing

Zwicky*Building Internet Firewalls - second edition;* 2000O'Reilly < Associates

W. Stallings*Network and Internetwork Security: Principles and Practice - second edition;* 1999Prentice Hall

Siti

Infrastrutture per reti di calcolatori

ATM forum; http://www.atmforum.com

RFC 1490: multiprotocol over frame relay; http://www.ietf.org

Frame relay: normative, documenti di survey, glossario; http://www.frforum.com

Frame relay: approfondimenti sulla tecnica e sul protocollo;

http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/frame.htm

Approfondimenti su reti frame relay in tecnica Ericsson; http://www.ericsson.se

Cisco: documentazione su architettura TCP/IP;
Cisco
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ip.htm

Helmig (tutorial TCP/IP); http://www.wown.com/j_helmig/tcpip.htm

Specifiche di servizi Internet;
IETF
http://www.ietf.org/rfc.html

IEEE - Institute of Electrical and Electronics Engineers;
IEEE
http://www.ieee.org/portal/index.jsp

Forum DSL; http://www.dslforum.org/

Tutorial IBM su TCP/IP;
IBM
http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/gg243376.html?Open

Reti di computer in tecnica TCP/IP

RFC 791 (IP, Internet Protocol);
IETF
http://www.ietf.org/rfc/rfc0791.txt

RFC 793 (TCP, Transmission Control Protocol);
IETF
http://www.ietf.org/rfc/rfc0793.txt

RFC 950 (Procedura per il supporto del subnetting);
IETF
http://www.ietf.org/rfc/rfc0950.txt

RFC 1009 (Requirements for Internet Gateways);
IETF
http://www.ietf.org/rfc/rfc1009.txt

RFC 1517 (Classless Interdomain routing);
IETF
http://www.ietf.org/rfc/rfc1517.txt

RFC 1518 (Classless Interdomain routing);
IETF
http://www.ietf.org/rfc/rfc1518.txt

RFC 1519 (Classless Interdomain routing);
IETF
http://www.ietf.org/rfc/rfc1519.txt

RFC 1520 (Classless Interdomain routing);
IETF
http://www.ietf.org/rfc/rfc1520.txt

RFC 1631 (The IP Network Address Translator);
IETF
http://www.ietf.org/rfc/rfc1631.txt

RFC 768: User Datagram Protocol;
IETF
http://www.ietf.org/rfc/rfc768.txt

RFC 791: Internet Protocol;
IETF
http://www.ietf.org/rfc/rfc791.txt

RFC 792: Internet Control Message Protocol;
IETF
http://www.ietf.org/rfc/rfc792.txt

RFC 793: Transmission Control Protocol;
IETF
http://www.ietf.org/rfc/rfc793.txt

Reti di computer in tecnica Windows 2000

MCP Magazine Online;
Microsoft
http://www.mcpmag.com

Microsoft Web Site;
Microsoft
http://www.microsoft.com/italy

Microsoft Windows 2000 Web Site;
Microsoft
http://www.microsoft.com/italy/windows2000

Windows and .NET Magazine;
Microsoft
http://www.win2000mag.com

Microsoft Technet Technical Plus;
Microsoft
http://www.microsoft.com/italy/technet

Reti di computer in tecnica Unix/Linux

Documenti su Linux; http://www.linuxdoc.org/

Libri elettronici su Linux; http://www.oreilly.com/catalog

Aspetti di sicurezza per Linux; http://www.seifried.org/lasg/

Servizi di rete integrati in ambienti Microsoft (SAMBA); http://www.samba.org/

Documentazione su Linux; http://www.pluto.linux.it/ildp

Rivista su Linux; http://www.linuxjournal.com

FAQ sul tema Linux; http://www.tldp.org/FAQ/Linux-FAQ

Gestione di reti

Request For Comments (RFC);
IETF
http://www.ietf.org/rfc.html

International Telecommunication Union;
ITU
http://www.itu.int/home/index.html

SNMP Version 3 (snmpv3);
IETF
http://www.ietf.org/html.charters/snmpv3-charter.html

HP Openview (Piattaforma HP: overview);
Hewlett-Packard
http://www.openview.hp.com/

HP online manuals;
Hewlett-Packard
http://docs.hp.com/

Tivoli (Piattaforma IBM);
IBM
http://www-3.ibm.com/software/tivoli/

System management Server (Piattaforma Microsoft);
Microsoft
http://www.microsoft.com/italy/smserver/

Sicurezza e gestione della sicurezza

Sicurezza di Windows 2000;

Microsoft
http://www.microsoft.com/windows2000/security/

Smart Card;
Microsoft
http://www.microsoft.com/whdc/hwdev/tech/input/smartcard/default.mspx

Windows catalog, SmartCard Reader;
Microsoft
http://www.microsoft.com/windows/info/smart404.asp?404;http://www.microsoft.com/windows/cata

Cisco Router Access Control List;
Cisco
http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt3/scac

Sicurezza di Linux; http://www.seifried.org/lasg/

TCSEC - Trusted Computer System Evatuation Criteria;
http://www.radium.ncsc.mil/tpep/library/tcsec/index.html

CERT Coordination Center; http://www.cert.org

The International PGP Home Page; http://www.pgpi.org

PGP documentation; http://www.pgpi.org/doc/

Glossario

Brouter : Dispositivo di rete che combina gli attributi di un *bridge* e di un *router*. Un *brouter* può instradare uno o pi#249; protocolli specifici, come il TCP/IP, e *bridge* tutti gli altri.

Crittologia : Disciplina che tratta delle scritture segrete, dei documenti in cifra.

Crittografia : Insieme delle tecniche che consentono di realizzare la cifratura di un testo e la decifratura di un crittogramma.

Crittoanalisi : Disciplina che studia come forzare i cifrari.

IDEA - *International Data Encryption Algorithm* : Algoritmo di crittografia sviluppato da Xuejia Lai e *James Massey* dell'Istituto Federale Svizzero di Tecnologia. È nato per cercare di migliorare e progressivamente sostituire l'algoritmo DES. Grazie alla sua efficienza ed affidabilità è stato incluso nel progetto PGP.

Message Digest algorithm : Algoritmo di crittografia sviluppato da Ron *Rivest* al MIT. La logica MD5 prevede di utilizzare come input un messaggio di lunghezza

arbitraria e produrre come output un messaggio di 128-bit. L'input è processato in blocchi di 512 bit.

Radix 64 *CONVERSION* : Tecnica di codifica che trasforma o mappa elementi di ingresso in forma binaria in sequenze di uscita di caratteri "stampabili" in codice ASCII.

Autori

Hanno realizzato il materiale di questo modulo:

Prof. Franco Callegati

Franco Callegati è professore associato di Reti di Telecomunicazioni presso il Dipartimento di Elettronica, Informatica e Sistemistica (D.E.I.S.) dell'Università di Bologna. Presso la Facoltà di Ingegneria di Bologna prima ed ora presso la Facoltà di Ingegneria di Cesena ha tenuto e tiene corsi di base di Reti di Telecomunicazioni e corsi avanzati su teoria del traffico e progettazione di reti. Si interessa di problematiche di dimensionamento e progettazioni di reti di telecomunicazione a larga banda e la sua attività di ricerca più recente ha come oggetto le reti ottiche ad altissima velocità, argomento sul quale ha pubblicato numerosi lavori, partecipando a progetti di ricerca nazionali ed internazionali con ruoli di coordinamento.

Dott. Ing. Paolo Zaffoni

Paolo Zaffoni si è laureato in Ingegneria delle Telecomunicazioni presso l'Università di Bologna nel giugno del 2001. È iscritto al secondo anno del Corso di Dottorato di Ricerca in Ingegneria Elettronica, Informatica e delle Telecomunicazioni presso l'Università degli Studi di Bologna. Svolge attività di ricerca nel campo dell'analisi del traffico, del progetto e della gestione di reti ad alte prestazioni. Ha svolto ed è attualmente impegnato in attività di supporto alla didattica per gli insegnamenti di Reti di Telecomunicazioni relativi al Corso di Laurea in Ingegneria dell'Informazione presso l'Università degli Studi di Bologna e fornisce servizi di consulenza ad imprese attive nel settore delle tecnologie dell'informazione.

Modulo realizzato sulla base di materiali prodotti nell'ambito di un piano di formazione di 12.000 tecnici delle pubbliche amministrazioni e messi a disposizione del MIUR dall'Autorità per l'Informatica nella Pubblica Amministrazione (AIPA).