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

Modulo 11

Struttura e organizzazione di Internet

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:

- Conoscere la storia, l'organizzazione e la struttura di Internet.
- Distinguere fra Internet e WWW e tra siti Intranet, Extranet, Internet.

Il modulo è strutturato nei seguenti argomenti:

- **Storia e organizzazione**
  - Descrivere le origini di Internet.
  - Descrivere a grandi linee la storia di internet.
  - Descrivere l'organizzazione di Internet come Internic, domini e *Request For Comment* (RFC).
  - Descrivere la struttura di Internet.
  - Distinguere fra Internet e WWW.
  - Distinguere tra siti Intranet, siti Extranet e siti Internet.

Introduzione

Storia e organizzazione di Internet

Simone Martini
11.1 (Internet)

Internet: un primo sguardo

Per il principiante, ***Internet***, il *World Wide* ***Web*** (**WWW**) e la rete del proprio ufficio sono più o meno la stessa cosa: una serie di calcolatori collegati tra loro che scambiano tra loro e con i loro utenti delle risorse, in particolare risorse di tipo informativo. È importante, invece, imparare a distinguere i vari livelli in cui la rete si struttura, per comprendere la logica interna dello sviluppo di questi livelli, e, dunque, le loro potenzialità, le loro criticità, i loro centri decisionali.

Scopo di questo intero modulo è descrivere la struttura e la storia della rete globale, rimandando invece ai moduli seguenti una discussione approfondita dei servizi forniti dalla rete stessa e delle tecniche per ricercare ed immettere in rete l'informazione. È un modulo più discorsivo e meno tecnico di altri. Alcuni approfondimenti discuteranno in modo tecnico alcune delicate questioni implementative.

Sappiamo già che una rete di calcolatori è una struttura di telecomunicazione in cui più calcolatori (in genere eterogenei, cioè diversi per *hardware* e sistema operativo) sono collegati allo scopo di condividere risorse e scambiarsi informazioni. Anche le reti sono tra loro eterogenee:

- per dimensioni: **LAN** (*local area network*), ***MAN*** (*metropolitan area network*), e **WAN** (*wide area network*);
- per supporto di telecomunicazione: cavo *ethernet*, cavo coassiale, doppino telefonico, fibra ottica, connessione senza fili (infrarosso, radio, UMTS, ecc.);
- per topologia della connessione: a stella, ad anello, a *bus* (tutti i nodi connessi su un'unica linea di comunicazione), punto a punto;
- per stabilità della connessione: connessioni dedicate (sempre attive), connessioni commutate (attivate su domanda di uno dei due nodi collegati), connessioni mobili (per esempio un cellulare UMTS).

L'eterogeneità delle reti, però, non impedisce la loro comunicazione. Sebbene alcune di queste reti usino al proprio interno protocolli di comunicazione specifici, mediante l'uso di un protocollo comune, le reti possono essere collegate tra loro a formare reti di reti (internets). Chiamiamo ***Internet*** (con la "I" maiuscola!) la rete planetaria di tutte le reti collegate tra loro e che comunicano con lo stesso protocollo. Nell'ultima sezione di questo modulo commenteremo la definizione ufficiale di ***Internet***: per comprenderne a pieno le implicazioni, tuttavia, è necessario analizzare nelle prossime sezioni la struttura di questa rete di reti, come e quando questa si è costituita, come si sviluppa, chi la gestisce, chi la paga.

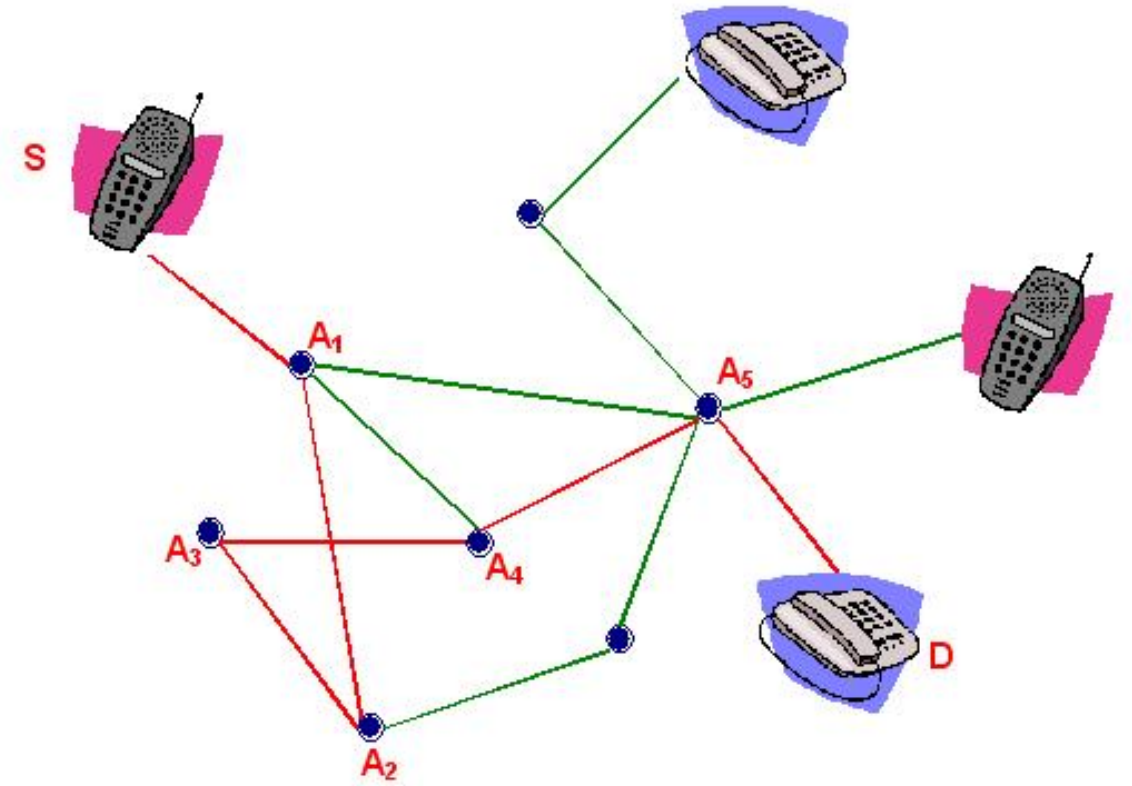
La struttura pervasiva di ***Internet*** rende la rete un veicolo formidabile per la trasmissione dell'informazione. Il *World Wide Web* è la principale architettura informativa basata su ***Internet***: si tratta di un insieme di ipertesti collegati tra loro e che

risiedono su nodi fisicamente diversi e molto distanti tra loro. Anche del **WWW** presenteremo la struttura e la storia del suo sviluppo.

Reti a commutazione di circuito

Internet si basa sulla interconnessione di diverse centinaia di migliaia di calcolatori e reti altrimenti indipendenti ed eterogenee. Come possono dialogare due nodi di questa rete? È evidente che non è ragionevole connettere direttamente ogni nodo ad ogni altro nodo. La connessione può però essere stabilita in modo indiretto:

per andare dal nodo S (sorgente) al nodo D (destinazione), passiamo prima per il nodo A1, poi per A2, ecc. fino a raggiungere il nodo An e da qui raggiungere D.



Schema di connessione tra due telefoni, passando per 5 nodi intermedi

Si tratta della stessa situazione della normale telefonia vocale: per effettuare una chiamata dal numero (nodo) S al numero (nodo) D, la chiamata passa per la centrale a cui è direttamente collegato S (questa sarà il nodo A1), viene instradata attraverso altre centrali (i nodi A2, A3, ecc.) fino a raggiungere la centrale An a cui D è direttamente collegato e raggiungere da qui, finalmente, D.

Nella telefonia vocale, le varie centrali (il cui ruolo era un tempo svolto da operatori manuali) funzionano come interruttori (deviatori): viene stabilito un collegamento "fisico" tra S e D, che, al momento in cui la connessione è stabilita, consiste in un (complesso) circuito elettrico che collega S a D. Per questo motivo questa tecnica si chiama connessione a commutazione di circuito. Le risorse (canali di comunicazione, interruttori, ripetitori, ecc.) che si trovano sul percorso S, A1, A2, .., An, D sono

assegnate alla connessione tra S e D e non sono disponibili per altri fino a quando S e D non le rilasciano, al termine della telefonata. A questo punto, tutte le risorse ritornano disponibili per altre telefonate.

Questa modalità di connessione è concettualmente semplice, ma mal si adatta alla comunicazione tra due calcolatori, perché:

- ci vuole troppo tempo per realizzare la connessione tra S e D.
  La commutazione del circuito è estremamente lenta rispetto alla velocità con la quale S produce i dati e a quella con cui D è capace di riceverli. In altre parole, il tempo speso per l'effettiva trasmissione dei dati tra S e D è quasi trascurabile rispetto alla somma del tempo necessario ad attivare il circuito e di quello che serve per rilasciarlo al termine della comunicazione.
- comporta un grande spreco di risorse
  La comunicazione tra calcolatori procede tipicamente per "picchi" di dati. Dopo che un picco è stato trasmesso, passa un certo tempo prima che il prossimo picco sia a sua volta inviato. Nel tempo che passa tra i due picchi il circuito rimane comunque impegnato e le risorse che lo compongono non possono essere rese disponibili per altre connessioni.
- è molto dipendente da eventuali guasti al circuito
  Se il circuito si guasta occorre "disfare" tutta la comunicazione e attivare la commutazione di un nuovo circuito. Inoltre non è semplice per D capire se la mancanza di dati che rileva sulla linea corrisponde al silenzio di S o ad un guasto intermedio.

Reti a commutazione di pacchetto

Per ovviare ai problemi delle commutazione di circuito, viene usata un'altra tecnica, cruciale per il successo di ***Internet***: la commutazione di pacchetto.

Siamo nella situazione seguente:

S vuole inviare a D il messaggio M costituto dai caratteri c1, c2, ..., ck.

Invece di richiedere alla rete la commutazione di un circuito che raggiunga D, S suddivide M in pacchetti, ciascuno composto di una quantità fissata (e piccola) di caratteri. Contraddistingue ogni pacchetto con la propria firma, con il numero d'ordine del pacchetto all'interno di M (così che D possa poi ricostruire il messaggio) e con un indirizzo che identifica in modo unico D sulla rete. A questo punto ogni pacchetto inizia una propria vita autonoma: S invia i pacchetti, uno alla volta, ad uno dei calcolatori che gli sono più vicini sulla rete, diciamo ad A1. Quando A1 riceve un pacchetto, si accorge di non essere il destinatario finale e così lo inoltra ad uno dei suoi vicini (diciamo ad A2), che a sua volta si comporta nello stesso modo, fino a quando il pacchetto riceve D. Via via che D riceve i pacchetti, li rimette in ordine, scarta eventuali duplicati e se necessario richiede a S la ritrasmissione di qualche pacchetto che non fosse giunto a destinazione.

Osserviamo che:

- ogni pacchetto utilizza tutta la banda di comunicazione disponibile in quel momento;

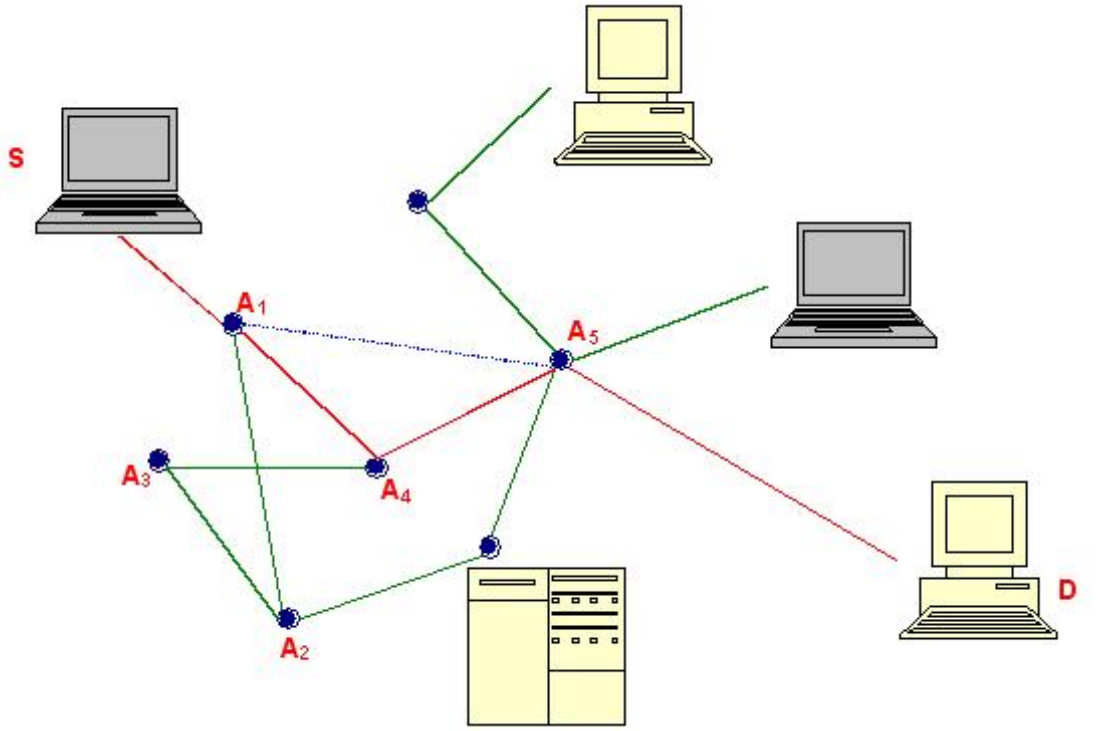
- ogni pacchetto che arriva su un nodo, viene memorizzato prima di essere ritrasmesso (meccanismo *store*-and-forward)
- la presenza di molti pacchetti che devono essere trasmessi su una specifica connessione può causare congestione: i pacchetti si accodano aspettando di usare la connessione;
- non è assolutamente garantito che tutti i pacchetti che compongono uno stesso messaggio compiano lo stesso percorso.

È evidente che questa descrizione lascia molte cose poco chiare:

- come decide un nodo a quale vicino trasmettere un pacchetto, in modo che, in qualche modo, il pacchetto si avvicini alla destinazione D?
- come viene indicato il codice che identifica D in modo univoco?
- come conosce S questo codice?
- come viene gestita l'eventuale presenza di errori durante la trasmissione?
- come viene gestita la congestione (per esempio con la scelta di cammini alternativi)?
- chi si occupa di tutti questi dettagli?
  Non certo l'utente finale, che non sa nulla di tutto questo. Forse le diverse applicazioni (browser, clienti di posta elettronica, ecc.)? O il sistema operativo?

Ad alcune di queste domande risponderemo nel seguito. Per comprendere la struttura di ***Internet***, tuttavia, la descrizione che abbiamo dato è sufficiente. Osserviamo che:

- non c'è spreco di tempo per connettere S a D, perché nessuna connessione viene stabilita. Viene solo impegnata una connessione da un nodo al proprio vicino.
- il circuito di comunicazione tra un nodo ed il vicino è usato per il messaggio M solo e quando si trasmette un pacchetto. Tra un pacchetto e l'altro possono essere trasmessi altri pacchetti di altri messaggi tra altri nodi.
- la comunicazione può avvenire anche se un circuito si guasta: basta aggirare il guasto mediante un altro percorso sulla rete.
  Nel disegno che segue i pacchetti tra S e D possono seguire un qualunque cammino che collega sorgente e destinazione. In particolare, la connessione guasta che collega A1 e A5 può essere facilmente aggirata.



Schema di connessione tra due calcolatori

La gerarchia dei protocolli

Rispondiamo in questa sezione alla domanda: chi si occupa dei dettagli della comunicazione?

La risposta è più complessa di quanto ci si possa aspettare, perché dipende dal livello al quale la domanda è posta. Nel caso di ***Internet***, possiamo distinguere cinque livelli. Ogni livello corrisponde alla gestione di una parte di dettagli (cioè ad un protocollo) e alla predisposizione di una serie di servizi per il livello superiore (il quale, dunque, non si preoccupa dei dettagli del livello inferiore).

Un protocollo definisce il formato e l'ordine delle comunicazioni spedite fra entità di rete, nonché le azioni da compiere al momento della trasmissione e del ricevimento della comunicazione.

- Livello fisico: i nodi sono fisicamente connessi da cavi e dispositivi di comunicazione (p.e modem); sono gestiti i parametri che dipendono dal mezzo di comunicazione (per esempio: qual è il voltaggio di 0 e di 1, quanti *bit* sono trasmessi al secondo, ecc.).
- Livello *data*-link: i nodi condividono un certo codice di trasmissione che consente loro di scambiarsi delle informazioni digitali e di correggere elementari errori di trasmissione che possono avvenire a livello fisico; fornisce al livello 3 la gestione dell'accesso al mezzo di comunicazione.
- Livello rete: i pacchetti sono inviati sulla rete e viaggiano attraverso più nodi; gestisce i servizi che aprono e chiudono le connessioni e, in

particolare, gestisce l'instradamento (*routing*) dei messaggi: a quale nodo devo inoltrare un pacchetto? In ***Internet***, questo è il livello del protocollo ***IP*** (***Internet*** *Protocol*).

- Livello trasporto: gestisce i dati che riceve dal livello 5, li organizza in pacchetti, li invia sulla rete sfruttando i servizi del livello 3; gestisce la ricostituzione del messaggio. In ***Internet***, questo è il livello del protocollo ***TCP*** (*Transmission Control Protocol*).
- Livello applicazione: le applicazioni dialogano tra loro per scambiarsi dati e documenti. Sfruttando i servizi del livello 4, richiedono, scambiano e ricevono dati. Questo è il livello dei protocolli **HTTP** (per lo scambio di servizi sul **WWW**), SMTP (per l'invio della posta elettronica), ecc.

Un approfondimento: i livelli ISO/OSI

L'Organismo Internazionale degli *Standard* ha definito una gerarchia *standard* di protocolli per reti, il cosiddetto modello ISO/OSI. Rispetto ai livelli che abbiamo elencato, ne aggiunge due, intermedi tra il livello trasporto e il livello applicazione. Sono il livello sessione (sopra a quello trasporto) e il livello presentazione (tra quello sessione e quello applicazione). ***Internet*** adotta il modello semplificato che abbiamo discusso; inoltre ***TCP*** e ***IP*** non rispettano del tutto lo *standard* ISO/OSI per i rispettivi livelli.

Il cuore di ***Internet*** (e la ragione del suo successo) è costituito dalla coppia di protocolli ***TCP***/**IP**.

Una rete di reti eterogenee

La strutturazione della comunicazione secondo una gerarchia di protocolli spiega come in ***Internet*** possano essere collegate tra loro reti indipendenti ed eterogenee.

Affinché la rete R1 possa comunicare con R2 è sufficiente che un nodo "di frontiera" di R1 (un *gateway*, o, come si dice più modernamente, un *router*) e un nodo di frontiera di R2 comunichino tra loro sfruttando la coppia di protocolli ***TCP***/**IP**. I *router*, ciascuno sfruttando i livelli inferiori della propria gerarchia, gestiscono le differenze tra le due reti (velocità di trasmissione, dimensione dei pacchetti, condizioni d'errore, ecc.). Nessuna delle due reti deve essere modificata per essere collegata: basta che i *router* siano in grado di interagire mediante ***TCP***/**IP**. Le due reti possono avere diversi comportamenti quanto a velocità e solidità. La commutazione di pacchetto permette alla rete rete più veloce di non sincronizzarsi con la più lenta, e mette in capo al destinatario (e non alla gestione della rete nel suo complesso) la responsabilità di rimettere insieme il messaggio e di scartare i pacchetti duplicati. Aggiungere una nuova rete a quelle già collegate è estremamente semplice: non occorrono modifiche alle reti esistenti e ai loro collegamenti; basta che la nuova rete abbia un *router* che supporta ***TCP***/***IP*** e che lo si possa collegare alla interconnessione di reti (inter-network, da cui ***Internet***) già esistente. L'unica cosa necessaria è la possibilità di indirizzare un pacchetto verso un nodo della nuova rete.

Indirizzare un nodo su Internet

Ogni nodo della rete deve poter essere individuato in modo univoco. La soluzione più semplice sembrerebbe quella di avere un'autorità centrale che gestisce tutta la rete e

assegna indirizzi univoci a chi li richiede, per esempio numeri assegnati in successione.

Questa soluzione centralizzata è semplice solo in apparenza. In particolare entra in conflitto con la scelta progettuale (evidente dalle sezioni precedenti) di mantenere ***Internet*** una rete aperta, cioè facilmente estensibile. La scelta di ***Internet*** è quella di avere un indirizzamento gerarchico:

- l'universo ***Internet*** è suddiviso in reti fisiche;
- ad ogni rete fisica è assegnato in modo centralizzato un certo numero (indirizzo);
- a ciascun nodo della rete fisica è assegnato un indirizzo composto dall'indirizzo della rete fisica concatenato con un altro numero, che individua in modo univoco il nodo all'interno della rete;
- se la rete fisica è suddivisa in sottoreti, l'assegnamento di indirizzi ai suoi nodi può avvenire, a sua volta, in modo gerarchico.

In questo modo esiste un'autorità centrale che si preoccupa di assegnare numeri alle reti; mentre i numeri ai nodi delle reti fisiche possono essere assegnati dai gestori delle reti stesse.

Indirizzi IP

Vediamo in dettaglio come è composto l'indirizzo di un nodo.

Gli indirizzi dei nodi sono definiti e gestiti a livello rete e dunque dal protocollo ***IP***, che specifica che ogni nodo sia univocamente identificato da un numero di 32 *bit*, il suo indirizzo ***IP***.

Un indirizzo ***IP*** viene in genere indicato come sequenza di 4 numeri decimali, ciascuno compreso tra 0 e 255, separati da un punto. Ad esempio, la macchina su cui sono state composte queste note ha indirizzo 130.136.2.37.

Un *byte* sono 8 *bit*. Un indirizzo ***IP*** è dunque composto da 4 *byte*. Con 8 *bit* si possono indicare i numeri decimali tra 0 e 28=255. Ciascuno dei quattro *byte* dell'indirizzo ***IP*** viene convenzionalmente letto come un numero decimale e separato con un punto dal successivo.

Quali dei quattro numeri costituiscono l'indirizzo della rete e quali l'indirizzo del nodo all'interno della rete?

Dipende dall'importanza e dalla dimensione della rete. Le reti sono classificate in tre categorie:

- Classe A: la rete è indicata dal solo primo *byte*; i restanti 3 *byte* indicano i nodi al suo interno. Sono reti di grandi dimensioni: con 3 *byte* a disposizione per i nodi, ciascuna rete di classe A può avere 224 nodi; per contro, possono esistere solo 256 reti di classe A.
- Classe B: la rete è indicata da due *byte*, i nodi sono indicati dai restanti 2 *byte*; ciascuna rete di classe B può avere 216 nodi.
- Classe C: la rete è indicata da tre *bit*, i nodi sono indicati dall'unico *byte* restante; ciascuna rete di classe C può avere solo 28=256 nodi.

Si comprenderà come l'assegnamento della classe ad una rete sia un'operazione di grande delicatezza. Una rete di classe C non ha grandi prospettive di crescita. Una rete di classe A è un risorsa estremamente rara, visto che ne possono esistere solo 256. La gestione dell'assegnamento di un numero (classe) - non ancora utilizzato - ad una rete è affidata ad un organismo centrale, la ***Internet*** *Assigned Number Authority*, IANA, www.iana.org. IANA delega poi analoghi organismi regionali all'assegnamento dei numeri ***IP*** all'interno delle relative *zone* geografiche. Per l'Europa, l'organismo di riferimento è la R**IP**E NCC (Réseaux ***IP*** Européens), www.ripe.net. I gestori delle singole reti fisiche sono responsabili dell'assegnamento dei numeri ai loro nodi.

Indirizzi simbolici di dominio

Gli indirizzi ***IP*** sono pensati per essere usati da... ***Internet*** Protocol!

Per l'uso umano essi sono difficili da ricordare e soggetti ad errori di trascrizione.

Sono stati pertanto introdotti indirizzi simbolici di dominio (o nomi logici), che indicano un nodo della rete con una sequenza di stringhe di caratteri (etichette, *label*) separate da punti. Per esempio, la macchina su cui sono state composte queste note ha nome logico papageno.cs.unibo.it. L'insieme e la struttura di questi nomi costituiscono il *Domain Name System*, o **DNS**, di ***Internet***.

Anche **DNS** è strutturato in modo gerarchico. La struttura a livelli si legge a partire da destra. Nel nostro esempio, la sequenza (etichetta) it contraddistingue il livello (il dominio) più alto nella gerarchia (è un TLD, Top *Level Domain*, o anche dominio di primo livello) e corrisponde al dominio italiano; unibo indica il livello successivo (il dominio di secondo livello), quello dell'università di Bologna; cs è il Dipartimento di Scienze dell'Informazione; papageno è la singola macchina. Non vi è alcun limite al numero di livelli di un nome logico.

Ad un nome logico corrisponde un unico indirizzo ***IP***, ma può accedere che ad uno stesso indirizzo ***IP*** siano assegnati più nomi logici. Per esempio i nomi pop.cs.unibo.it, www.cs.unibo.it, leporello.cs.unibo.it corrispondono tutti al medesimo nodo, con indirizzo ***IP*** 130.136.1.110.

La traduzione (in gergo: resolving) del nome logico di dominio nel corrispondente indirizzo ***IP*** è compito di un particolare servizio svolto da alcuni nodi della rete. Un ***Domain Name Resolver*** è un servizio fornito da un nodo che, mantenendo specifiche tabelle, è in grado di rispondere alla domanda: qual è l'indirizzo ***IP*** corrispondente ad un certo nome logico?

Un sinonimo di ***Domain Name Resolver*** è *Domain Name Server*, **DNS**. Si osservi che la sigla **DNS** non è usata in modo coerente. Talvolta indica l'insieme dei nomi simbolici (la S sta per *System*); altre volte indica un *resolver* (la S sta per *Server*). In particolare **DNS** come un particolare *resolver* è usata nelle finestre di configurazione di alcuni sistemi operativi (o meglio del loro modulo di rete che gestisce i protocolli ***TCP***/**IP**). In quel contesto nella finestrella indicata con **DNS** deve essere indicato l'indirizzo ***IP*** del *resolver* che verrà utilizzato da quella macchina.

La vastità della rete non permette ad un singolo *resolver* di mantenere tabelle complete. Ma se non conosce la risposta per uno specifico dominio, un *resolver* sa quale altro *resolver* potrebbe avere l'informazione, e gira la domanda a quest'ultimo. In

caso di bisogno può sempre risalire ad uno dei *resolver* radice (ce ne sono 13 sparsi per il mondo), ciascuno dei quali conosce l'indirizzo dei *resolver* di tutti i top *level domain*.

La presenza dei *resolver* permette di svincolare gli utenti dagli indirizzi ***IP***. Il gestore di una rete può cambiare la macchina fisica che corrisponde ad un nome logico senza che gli utenti dei servizi di quella macchina se ne accorgano. Basta comunicare alla gestione del **DNS** che un certo nome logico corrisponde ora ad un nuovo indirizzo ***IP***.

Non vi è legame tra la gerarchia dei nomi logici e la gerarchia degli indirizzi ***IP***; per esempio, i nodi del dominio it non fanno parte della medesima rete fisica.

Chi assegna i nomi logici di dominio

Come nel caso degli indirizzi ***IP***, la responsabilità dell'assegnamento dei nomi di dominio è suddivisa in modo gerarchico:

Il dominio .it

Il dominio it gestito da www.nic.it, dove hanno sede due organismi:

- Registration *Authority* Italiana: responsabile dell'assegnazione dei nomi e della gestione del **DNS** primario per il dominio .it;
- Naming *Authority* Italiana: organismo che stabilisce le procedure operative ed il regolamento in base al quale opera la *Registration Authority*.

Fisicamente, la *Registration Authority* italiana ha sede a Pisa, presso un instituto del Consiglio Nazionale delle Ricerche. La *Registration Authority* non interagisce direttamente con chi intende registrare un dominio. Il servizio di registrazione dei domin, per privati e aziende, svolto da intermediari, detti *maintainer*. Alcuni *maintainer* per il dominio it sono:

- www.register.it
- www.dominio.it

Chi coordina Internet

Si potrebbe pensare che la gestione di una rete così complessa e articolata richieda uno stetto controllo centralizzato. Nulla sarebbe più lontano dalla realtà.

Se si eccettuano le questioni tecniche che abbiamo discusso nelle prime sezioni, la potenza di impatto di ***Internet*** si deve in buona misura a due aspetti:

- la pubblicità delle sue scelte tecniche;
  I protocolli usati e tutte le scelte tecniche che permettono ai vari nodi e alle varie applicazioni di cooperare sono completamente pubblici e accessibili a chiunque abbia voglia di dar loro un'occhiata; lo erano sin dai primi momenti sfruttando il *File Transfer Protocol* (FTP); lo sono oggi in quanto pubblicati sul **WWW**. Nessuno deve pagare diritti per usarli. Torneremo su questo punto quando parleremo di **RFC**, che sono la modalità con la quale questa pubblicità si attua.

- la presenza di organismi di coordinamento che hanno gestito le sole scelte dalla quali dipendeva la interoperabilità della rete (per esempio l'assegnamento degli indirizzi e dei dominî).
  Internet, nella sua globalità, non è proprietà di nessuno. Tuttavia vi sono delle scelte tecniche che devono essere necessariamente condivise, pena la scomparsa stessa della rete. Queste scelte sono prese da alcuni organismi che, mediante procedure consultive delle parti coinvolte (tecnici, ricerca, aziende, commercio, enti pubblici) definiscono una sorta di *standard* per molte questioni.

La storia di Internet su un grafico

La figura che segue riassume in modo sintetico lo sviluppo di ***Internet*** dai suoi esordi alla fine degli anni 90. A partire dal basso troviamo:

- il numero di reti interconnesse;
- gli eventi più rilevanti;
- la linea del tempo;
- le organizzazioni che coordinano ***Internet*** da un punto di vista tecnico;
- le organizzazioni che sovrintendono a ***Internet*** da un punto di vista più vasto, incluso il suo finanziamento.

Fonte: *Internet Society*, www.isoc.org

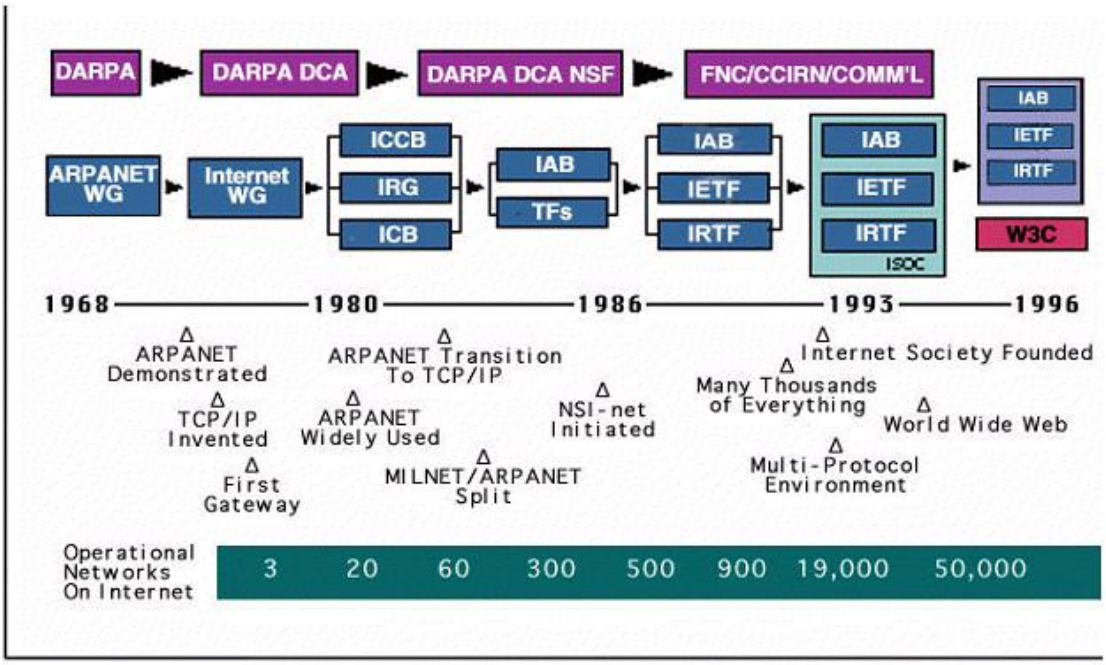


Grafico della storia di internet

Un approfondimento discute in modo più ampio, tra le altre cose, di alcuni aspetti storici.

Il procedimento di standardizzazione: le RFC

Internet non ha un centro direzionale capace di imporre delle scelte. Le organizzazioni di coordinamento suggeriscono e definiscono degli *standard* comuni, lasciando poi ai produttori, ai fornitori di connettività, agli utenti, il compito di uniformarsi allo *standard*. L'autorevolezza del coordinamento, e la necessità di non esser buttati fuori dal mercato, fa sì che le indicazioni degli *standard* abbiano nei fatti un amplissimo seguito.

Per divenire uno *standard* di ***Internet***, una proposta tecnica deve passare al vaglio di un lungo e approfondito procedimento di discussione, distribuito sulla rete e al quale tutti gli utenti, in linea di principio, possono partecipare.

Alla fine degli anni 60 la gestione di ***ARPANET*** decise che le proposte tecniche dovessero essere pubbliche. Tali rapporti tecnici furono denominati **RFC** (*Request for Comments*, richieste di commenti):

- sono disponibili sulla rete stessa (dapprima mediante ftp, poi mediante il www);
- sono numerati in ordine cronologico; **RFC** 1 è del 7 aprile del 1969; oggi (febbraio 2003) sono quasi 3500;
- tutti gli **RFC** sono disponibili a www.ietf.org/rfc.html;
- la consultazione e l'utilizzo del materiale degli **RFC** sono liberi e gratuiti.

Gli **RFC** sono redatti dai gruppi di lavoro tecnici dello IAB (***Internet*** Architecture Board) e, come dice il loro nome, il loro scopo iniziale è quello di descrivere una soluzione tecnica, al fine di suscitare commenti (supporto, critiche, modifiche, ecc.).

Se la proposta contenuta in un **RFC** genera abbastanza interesse e consenso nella comunità, il contenuto passa nello stato di *standard* proposto. Dopo che l'interesse preliminare si è concretizzato in due implementazioni funzionanti ed indipendenti, il documento passa nello stato di *standard* preliminare (draft *standard*) e sottoposto agli organi di governo di IAB. Se le ulteriori sperimentazioni sono *positive* e IAB è convinta che la proposta è sensata e corretta, e che le implementazioni sono funzionanti, lo **RFC** viene approvato e dichiarato *Standard* di ***Internet***.

La procedura di standardizzazione è definita essa stessa nel **RFC** 2026.

Non tutti gli **RFC** corrispondono a *standard*. Alcuni sono di tipo informativo o storico, altri descrivono esperimenti. Per esempio **RFC** 2055 descrive la storia degli **RFC**.

Internet: la definizione ufficiale

La definizione ufficiale di cosa sia ***Internet*** è contenuta in una risoluzione approvata all'unanimità il 24 ottobre 1995 dalla FNC:

"**Internet**" *refers to the global information system that* -- (i) *is logically linked together by* a *globally unique address space based on the* ***Internet*** *Protocol* (***IP***) *or its subsequent extensions*/follow-ons; (ii) *is able to support communications using the Transmission Control Protocol/****Internet*** *Protocol* (**TCP**/**IP**) *suite or its subsequent extensions*/follow-ons, *and/or other* ***IP***-compatible *protocols*; *and* (iii) *provides, uses or makes accessible, either publicly or privately, high level services layered on the communications and related infrastructure described herein*.

Con "**Internet**" si indica il sistema informativo globale che: (i) è logicamente connesso mediante un unico spazio globale di indirizzi basato sul protocollo ***IP*** o sulle sue

estensioni; (ii) permette di supportare le comunicazione utilizzando la coppia di protocolli ***TCP/IP*** o le sue estensioni e/o altri protocolli compatibili con ***IP***; (iii) fornisce, utilizza o rende accessibili, in modo pubblico o privato, servizi ad alto livello sfruttando i livelli di comunicazione e le infrastrutture che sono stati descritti ai punti precedenti.

Comprendiamo adesso come (i) e (ii) si riferiscano alla complessa, ma elegante organizzazione tecnica responsabile della comunicazione e della connessione. Ma essenziale è anche il punto (iii): ***Internet***, al livello delle applicazioni, appare come un grande, universale sistema informativo. La descrizione approfondita dei servizi di ***Internet*** a questo livello è compito dei moduli successivi.

Tra tutti i servizi, tuttavia, il *World Wide* ***Web*** ha assunto negli ultimi 10 anni un ruolo centrale e cruciale. Discuteremo la sua storia e le sue caratteristiche principali nelle prossime sezioni.

Intranet

La tecnologia con la quale ***Internet*** è realizzata (cioè la coppia di protocolli ***TCP/IP*** ed le applicazioni che su di essi sono basate) può essere usata anche per realizzare reti di dimensioni più ridotte.

Una rete aziendale, per esempio, può essere realizzata collegando calcolatori e *server* e facendoli dialogare usando ***TCP/IP***. Anzi: questo è oggi il metodo più diffuso, e relativamente meno costoso, per realizzare una rete.

Una rete di questo tipo

- cioè una rete basata sulla tecnologia di interconnessione, sul *software* e le applicazioni che sono usati per ***Internet***;
- realizzata per le esigenze di un privato;

è chiamata intranet.

Una intranet può essere del tutto scollegata dalla rete pubblica, oppure vi può essere connessa per il tramite di appositi *server* (*firewall*) che limitano sia l'accesso dall'esterno che l'uscita da parte degli utenti interni.

Un'azienda o un'istituzione può così basare i propri strumenti interni di comunicazione e cooperazione sulle applicazioni *standard* (posta elettronica, *web* browsers, ecc.), mantenendo rigidamente riservato l'accesso alla propria intranet.

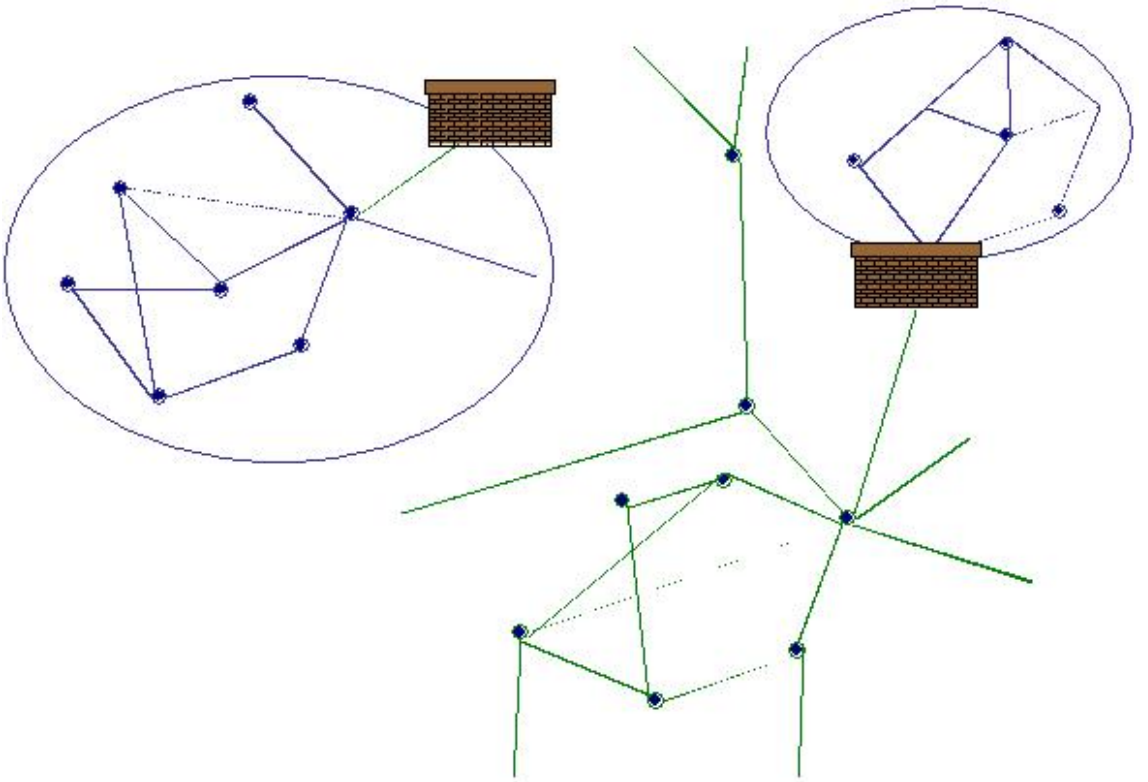
Talvolta due sottoreti della stessa intranet sono collegate tra loro per mezzo di connessioni facenti parte della rete pubblica di ***Internet***. Le questioni di sicurezza delle comunicazioni lungo i canali pubblici sono risolte mediante tecniche crittografiche.

I vantaggi di questa organizzazione:

- non è richiesto un investimento ulteriore (di *software, hardware* e di formazione del personale) per l'uso degli strumenti di cooperazione;
- lo sfruttamento delle connessioni pubbliche riduce i costi di connessione e di investimento nelle infrastrutture.

Nella figura che segue i due ovali azzurri costituiscono due sottoreti della stessa intranet. Le due sottoreti comunicano attraverso la parte pubblica di ***Internet***

(connessioni in verde). Sono indicati i *firewall* di separazione.



Una intranet

Extranet

Un'istituzione o un'azienda hanno spesso la necessità di condividere alcune informazioni (anche molto complesse e strutturate) con alcuni partner. Tali informazioni sono disponibili sulla intranet, ma non possono essere rese disponibili sulla parte pubblica di ***Internet***.

Chiamiamo extranet l'utilizzo della tecnologia ***Internet***/intranet per creare reti utilizzate da un insieme esteso di clienti o partner. Una extranet è sempre connessa alla parte pubblica di ***Internet*** e, come una intranet, si trova in genere al di là di uno o più *firewall*. È chiusa al pubblico generale, ma è aperta ad alcuni partner che non hanno accesso alla intranet.

I partner esterni alla intranet usano la parte pubblica pubblica di ***Internet*** per accedere al *firewall*; mediante una procedura di autenticazione (nel caso più semplice: nome e *password*) superano il *firewall* e accedono alla extranet. La loro autenticazione tuttavia non permette loro di accedere a tutte le informazioni riservate che fanno parte della intranet.

Come si capisce da ciò che abbiamo appena detto, il termine extranet non costituisce un concetto tecnico preciso. Spesso lo si utilizza per indicare la compresenza di reti pubbliche e *private*.

Si tratta in ogni caso di un modello organizzativo di largo uso.

Intranet e extranet: un esempio

Una università ha una propria intranet, costituita dalla rete su cui cooperano le varie sezioni amministrative (stipendi, personale, gestione delle carriere degli studenti, ecc.)

Le intranet corrispondenti alle sedi decentrate della stessa università sono collegate mediante connessioni pubbliche. I dati sensibili (per esempio tutti quelli riguardanti la carriera degli studenti) viaggiano in modo sicuro mediante tecniche crittografiche e sono protetti da ogni accesso dall'esterno della intranet.

Da ***Internet*** non è possibile accedere direttamente a tale intranet, per ovvi motivi di sicurezza e riservatezza.

Agli studenti, attraverso ***Internet***, è però permesso accedere alla propria carriera, per la modifica di alcuni dati anagrafici e la visione della carriera stessa. La rete dedicata a questi aspetti costituisce (parte del-) la extranet della rete che stiamo considerando.

I costi di Internet

I costi di ***Internet*** sono di due tipi:

- costo della struttura di interconnessione e dell'effettivo trasferimento dei dati;
- costo dei servizi, cioè delle informazioni messe a disposizione sulla rete stessa.

I costi della seconda categoria non ci interessano qui: ogni fornitore di servizi li vende (o li regala) ai propri clienti al costo e con le modalità di pagamento (a forfait, a blocchi, a tempo, a dimensione, ecc.) che ritiene più adeguati.

I costi della prima categoria sono, invece, i costi intrinseci di ***Internet***: senza queste spese ***Internet*** semplicemente non esisterebbe.

Una delle scelte progettuali originali di ***Internet*** è che i suoi costi intrinseci siano ripartiti tra tutti gli utenti, secondo il tipo di connessione di cui dispongono.

Ciascun utente paga una porzione di connessione: quella che lo collega al proprio (od ai propri) **ISP**. Un utente domestico pagherà la propria compagnia telefonica per la connessione commutata, a tempo o a forfait. Un utente più sofisticato pagherà il canone della fornitura ADSL. Un'azienda potrebbe pagare una connessione dedicata in fibra ottica.

A sua volta un **ISP** compra connettività:

- da altri **ISP** (comprando grandi quantità di banda otterrà prezzi più bassi del piccolo utente e questo gli permetterà di rivenderla con qualche margine);
- dai proprietari della connessione fisica.

Parte di quello che l'utente domestico paga alla propria compagnia telefonica per una piccola connessione commutata viene così girato al proprietario della connessione internazionale satellitare, il cui costo è effettivamente ripartito tra tutti gli utenti.

Non tutti gli utenti pagano però la stessa somma per lo stesso servizio.

La ripartizione terminale dei costi è fatta in ultima analisi dalle compagnie telefoniche, che spesso sono anche le proprietarie delle connessioni fisiche. In paesi dove la connessione telefonica è costosa, anche ***Internet*** viene pagato di più, a parità di servizio (cioè di banda disponibile e tempo di connesione).

La connessione telefonica è tanto più costosa quanta meno concorrenza vige in un certo mercato. OCSE stimava nel 1996 che il costo della connessione di base fosse tre volte più costoso nei paesi con servizio telefonico in regime di monopolio rispetto a quelli in regime di libera concorrenza. Sempre nel 1996, il costo di un'utenza ***Internet*** domestica italiana era di circa 10 volte il costo della stessa utenza in USA o in Canada.

I servizi basati su Internet

I servizi principali che ***Internet*** fornisce sono l'oggetto principale dei prossimi moduli. Qui ci basta ricordare che, fino ai primi anni 90, i servizi più usati erano:

- la posta elettronica;
- lo scambio di dati (sotto forma di *file*: *file transfer*);
- l'accesso remoto (*remote login*).

Quest'utilizzo era consistente con la diffusione di ***Internet*** principalmente tra centri di ricerca; la maggiore applicazione commerciale era la posta elettronica, un servizio che MCI vendeva ai propri clienti americani.

Nei primi anni 90 fa la sua comparsa un nuovo servizio, che sarà destinato a diventare in brevissimo tempo il più diffuso e pervasivo: il *World Wide* ***Web***, o **WWW**.

Si tratta di un sistema di (iper-)testi multimediali memorizzati in modo distribuito sulla rete, e che la tecnologia di ***Internet*** permette di visualizzare e visitare - seguendone i collegamenti - in modo semplice e intuitivo.

Nelle sezioni che seguono discuteremo brevemente di questi concetti. Alcuni moduli seguenti sono dedicati alle tecniche di progetto e di realizzazione di questi ipertesti.

Osserviamo intanto che il successo e la pervasività del **WWW** sono tali che, per molti, ***Internet*** è sinonimo di **WWW**. Si tratta di un errore tanto grave quanto diffuso.

Ipertesti

Un ipertesto è un documento che contiene al suo interno collegamenti ad altri documenti o a sezioni dello stesso documento. Le informazioni sono organizzate non in modo sequenziale, ma reticolare; l'utente può saltare da un punto all'altro del documento (o da un documento all'altro) seguendo i collegamenti.

Alcuni commenti critici di opere letterarie hanno la struttura di ipertesto:

- contengono collegamenti a passi paralleli della stessa opera;
- contengono riferimenti ad altre opere di critica, sotto forma di citazioni;
- contengono rimandi ad altre sezioni del commento.

La possibilità di creare ipertesti era ben nota all'ambiente letterario. Ma la struttura tradizionale del libro mal si presta a "realizzare" un ipertesto e a seguire i collegamenti interni ad esso.

Il **WWW** è un gigantesco ipertesto multimediale distribuito dotato di un'interfaccia di facile uso:

- ipertesto: la tecnologia informatica è usata per realizzare un ipertesto per il quale seguire un collegamento sia semplice quanto la lettura sequenziale di un testo;
- multimediale: ogni documento può essere composto non solo di testo, ma anche di informazioni espresse con *media* diversi: immagini, *video*, suono, ecc.
- distribuito: le diverse parti di questo gigantesco ipertesto risiedono su calcolatori diversi e distanti tra loro; l'utente non necessariamente è a conoscenza di dove e come le informazioni sono memorizzate;
- dotato di interfaccia di facile uso: la fruizione di questo ipertesto è mediata da un'interfaccia che si prende carico di visualizzare in modo uniforme i dati multimediali e di risolvere il problema del raggiungimento dei dati remoti. La disponibilità di questa interfaccia è la chiave di volta del successo del **WWW**. Chiamiamo *browser* le applicazioni che realizzano questa interfaccia per l'utente.

Il Web: un primo assaggio

Un ipertesto è realizzato sul ***Web*** come un documento in cui alcune parti (porzioni di testo o immagini) svolgono contemporaneamente due ruoli:

- descrivono l'informazione che denotano;
- rappresentano un collegamento.

Un'opportuno modo di evidenziare tali porzioni di testo permette all'utente di individuare i collegamenti e, se vuole, di seguirli.

I collegamenti sono attivabili: svolgono il ruolo di pulsanti, che, se premuti, permettono di seguire il collegamento stesso.

La struttura del documento - che comprende la posizione ed il valore dei collegamenti - è descritta in un apposito linguaggio, lo Hyper *Text Markup* Language (**HTML**).

HTML descrive la struttura del documento, non il modo in cui il documento deve essere stampato o visualizzato. **HTML** "marca" nel documento quello che costituisce un titolo, quello che costituisce un capitolo, un elenco, una citazione, ecc. Marca anche quello che è un collegamento (*link*), insieme al valore di tale collegamento, cioè del documento a cui trasferire l'utente se questi decide di seguire il collegamento.

Sono i *browser* a decidere come visualizzare la struttura descritta da **HTML**: *browser* diversi possono visualizzare in modo diverso lo stesso documento, a patto che rispettino la sua struttura (definita da un'opportuna "raccomandazione del **W3C**").

I collegamenti possono riferirsi a documenti remoti. Per realizzare ciò il ***Web*** sfrutta il sistema di indirizzamento di ***Internet***. Il valore di un collegamento è un Uniform Resource Identifier (**URI**), costituito in sostanza da tre parti:

- un protocollo da utilizzare per recuperare il documento (p.e. http);
- un nome simbolico di dominio (p.e. www.cs.unibo.it);

- un cammino d'accesso, relativo a quel dominio, e che specifica un documento (p.e. /~martini/didattica/index.html).

I documenti che costituiscono l'ipertesto sono residenti su calcolatori su cui girano opportune applicazioni dette *web server*.

Un *web server* risponde alla richiesta di un *browser*, formulata secondo l'opportuno protocollo, e fornisce al *browser* il documento richiesto. Il più comune dei protocolli è lo Hyper *Text Transfer Protocol*, **HTTP**, usato proprio per la richiesta e la trasmissione di ipertesti in modalità non sicura.

Il web è costruito su Internet

Il ***Web*** non esisterebbe senza la struttura di interconnessione e i servizi di livello inferiore che ***Internet*** mette a disposizione.

Sentiamo quello che ha da dire a questo proposito Tim Berners-Lee, l'inventore del ***Web*** (da: www.w3.org/People/Berners-Lee/FAQ.html):

Il ***Web*** è uno spazio astratto (immaginario) di informazioni. Sul *Net* [**Internet**, ndt] troviamo calcolatori - sul ***Web*** troviamo documenti, suoni, *video*, ... informazioni. Sul *Net*, le connessioni sono costituite da cavi tra calcolatori; sul ***Web*** le connessioni sono collegamenti (*link*) ipertestuali. Il ***Web*** esiste perché i programmi possono comunicare tra calcolatori sfruttando il *Net*. Il ***Web*** non potrebbe esistere senza *Net*. Il ***Web*** ha reso *Net* utile perché la gente è interessata all'informazione (senza contare la conoscenza e la saggezza!) e non vuole sapere granché di calcolatori e cavi.

Chi coordina il WWW

Il **WWW**, come ***Internet***, è nato in un ambiente scientifico, per promuovere la collaborazione tra ricercatori. Come ***Internet***, il ***Web*** nasce aperto, senza formati proprietari (cioè per l'uso dei quali si debbano pagare diritti) e senza una struttura di controllo centrale che potesse decidere cosa (e come) potesse stare sul ***Web*** e cosa no.

La rilevanza commerciale del ***Web*** era peraltro evidente sin dai suoi albori.

Era chiaro che qualche azienda avrebbe potuto introdurre dei formati proprietari. Sarebbe stato un grave handicap per la diffusione globale del **WWW**: formati e protocolli diversi sarebbero entrati in competizione.

Si pensi, per esempio, a quello che sta accadendo per i *video on-line*. Vi sono alcuni formati proprietari (per esempio Real Audio, © RealNetworks, o *Windows Media*, © Microsoft) che possono essere riprodotti solo mediante una specifica applicazione o *plug*-in. I vari formati, proprietari e non, competono per la propria fetta di mercato e di conseguenza l'utente deve disporre di più riproduttori, gratuiti o a pagamento a seconda delle scelte di *marketing* delle azionde proprietarie del marchio. Se la stessa situazione si fosse verificata col formato "standard" per i documenti testuali, la globalità e interoperabilità del ***Web*** sarebbero state compromesse: alcuni siti sarebbero stati visibili solo con determinati *browser*.

Come nel caso di ***Internet***, c'era bisogno di:

- un organismo di supervisione che garantisse l'esistenza unitaria del ***Web***;
- contemporaneamente tale organismo non doveva avere il potere di alterare in modo cruciale il carattere aperto e libero del ***Web*** stesso.

Sullo stile del governo di ***Internet***, nel 1994 fu creato il *World Wide* ***Web*** Consortium, o **W3C** (www.w3.org), un'organizzazione senza fine di lucro che sovrintendesse allo sviluppo del ***Web***, così come ISOC sovrintende a ***Internet***.

Il **W3C** ha sede presso il Massachusetts *Institute* of Technology; la sua sede europea è presso lo *European* Research Consortium in Informatics *and* Mathematics (www.ercim.org).

Lo scopo primario del **W3C** è quello di sviluppare protocolli, specifiche, *software* e strumenti che garantiscano l'interoperabilità del ***Web***.

La partecipazione al consorzio è aperta a organizzazioni di qualsiasi tipo, commerciali, educative, governative, ecc.

Le decisioni del **W3C** sono prese, in analogia con le **RFC** di ISOC, mediante un processo di proposte, commenti e raccolta di consenso. Quando una proposta ha raggiunto un sufficiente consenso, viene pubblicata come "raccomandazione" (recommendation). **W3C** non impone (né potrebbe in alcun modo imporre) le proprie raccomandazioni. Ma la sua autorevolezza le rende per molti aspetti simili a dei veri e propri *standard*.

I tentativi di essere più furbi di altri e di introdurre surrettiziamente caratteristiche proprietarie continuano anche in presenza delle raccomandazioni del **W3C**. Si sono visti (e purtroppo si vedono tuttora) "siti XYZ enhanced" dove XYZ è il nome commerciale di uno specifico *browser*. Si tratta di siti progettati in deroga ai criteri generali di interoperabilità definiti dal **W3C** nelle proprie raccomandazioni e che sfruttano invece caratteristiche "non standard" gestite solo dal *browser* XYZ (e non da ogni *browser* che rispetti le raccomandazioni **W3C**). Per fortuna questa linea di pensiero sembra essere stata sconfitta dallo scarso gradimento degli utenti...

Le raccomandazioni del **W3C** sono liberamente consultabili a www.w3.org/TR/

Conclusioni

Al termine di questa parte introduttiva, dovrebbero essere chiari:

- la struttura di ***Internet*** come rete aperta;
- le principali organizzazioni che sovrintendono al suo funzionamento e sviluppo, in assenza di un controllo centralizzato;
- il fatto che ***Internet*** è principalmente un'infrastruttura di comunicazione su cui sono fornite le effettive applicazioni utilizzate dagli utenti finali;
- la particolare rilevanza assunta negli ultimi anni da quell'applicazione che è il **WWW**.

Gli approfondimenti forniti sono di tue tipi.

- Un primo approfondimento, fruibile in tutti i percorsi, fornisce complementi sulla storia e l'organizzazione di ***Internet***. In particolare, è trattata più in dettaglio la storia di ***Internet***, mettendo in luce come le scelte progettuali

Approfondimento

Complementi sulla storia e l'organizzazione di Internet

Simone Martini

11.1 (Complementi sulla storia e l'organizzazione di Internet)

Introduzione

Questo breve approfondimento fornisce informazioni ulteriori circa la storia di *Internet* e del WWW, oltre ad alcuni parametri sulle loro dimensioni. Il suo scopo è principalmente quello di far vedere come le scelte progettuali chiave di *Internet*:

- una rete aperta;
- senza controllo centralizzato;

si radicano e si fondano nella storia del suo sviluppo. In particolare vedremo come la chiave del successo della rete è che questa, pur avendo sempre coinvolto insieme ricerca pubblica e iniziativa privata, è sempre stata una rete non proprietaria, le cui specifiche e caratteristiche erano e sono pienamente disponibili e utilizzabili da chiunque intenda investirvi.

Internet: i primi anni

Internet affonda le proprie radici in alcuni lavori pionieristici dei primi anni 60, in USA, che sia in ambito militare che nelle università studiano la connessione di calcolatori su area geografica e l'uso della commutazione di pacchetto.

J.C.R. Licklider (del *Massachusett Institute of Technology*, *MIT*, poi primo direttore del programma di ricerca in informatica del *DARPA*, *Defense Advanced Research Projects Agency*) è il primo a parlare di Reti galattiche, un insieme di calcolatori interconnessi su scala globale attraverso i quali ciascuno potesse avere un accesso veloce a programmi e dati residenti su un sito qualsiasi. Era il 1962, i calcolatori costavano centinaia di milioni (dell'epoca), stavano in stanze grandi come palestre e solo personale estremamente specializzato era in grado di usarli. Eppure l'idea di *Licklider* è molto simile a ciò che *Internet* è oggi...

DARPA sarà il centro trainante per la realizzazione della prima rete geografica. Sulla base delle idee pionieristiche di *Licklider*, i suoi successori alla direzione di *DARPA* (*Ivan Sutherland*, *Robert Taylor*, e *Lawrence G. Roberts*) comprendono lucidamente le potenzialità di una vasta rete interconnessa e riescono a incanalare in questo tipo di ricerca le grandi risorse necessarie. La prima connessione tra le due coste degli USA avverrà nel 1965, a commutazione di circuito, una soluzione che si rivelò immediatamente inadatta, confermando quanto già aveva sostenuto in via teorica *Leonard Kleinrock*, prima a *MIT* e poi all'Università di California a *Los Angeles* (*UCLA*).

Scelta la commutazione di pacchetto come modalità di comunicazione, si trattò di realizzare i primi semplici processori dedicati all'instradamento dei pacchetti sulle linee di comunicazione. L'appalto fu vinto dalla *Bolt, Beranek, and Newman* (BBN), che realizzò i primi dispositivi nel 1969: *DARPA* era pronta ad iniziare la realizzazione della

sua rete, che doveva collegare le principali installazioni militari e di ricerca americane.

Nell'ottobre del 1969 viene realizzata la prima comunicazione sulla nascente rete *ARPANET* tra *UCLA* (il gruppo di ricerca di *Kleinrock*) e lo *Stanford Research Institute* (*SRI*), a *Menlo Park*, non lontano da *San Francisco*. Alla fine del 1969 i nodi erano quattro e la ricerca relativa ad *ARPANET* riguardava, in parallelo, sia i protocolli di comunicazione che le applicazioni che potevano sfruttare la rete: una tradizione che ha caratterizzato tutto lo sviluppo di *Internet* e che continua ancora oggi immutata.

Nel 1972 *ARPANET* fu presentata per la prima volta ad un congresso pubblico. Nello stesso anno *Ray Tomlinson* (della BBN) scrive la prima applicazione per lo scambio di posta elettronica, affinché i ricercatori di *ARPANET* potessero scambarsi velocemente informazioni. Sarà l'applicazione più di successo della rete per oltre un decennio, che ebbe un impatto enorme sul modo di collaborare, prima per la costruzione dello stesso *Internet*, e poi per una gran parte della società.

Internet: le scelte cruciali

Una delle scelte cruciali per l'esistenza e il successo di *Internet* era già stata fatta, la commutazione di pacchetto. Ma *ARPANET*, essendo l'unica rete esistente, era una rete monolitica, chiusa.

Presto, tuttavia, le cose cambiarono. Quando si trattò di collegare a *ARPANET* due altre reti finanziate da *DARPA* - la rete satellitare SATNET e quella radio PRNET - fu chiaro che l'interconnessione di reti (lo *inter-networking*) doveva essere resa più semplice possibile. In particolare, il protocollo usato a quel tempo su *ARPANET* (NCP, *Network Control Protocol*) demandava alla rete tutta la responsabilità della consegna dei pacchetti: se un pacchetto non arrivava, la rete si bloccava. Questa situazione era inaccettabile su una rete radio, dove le mancate connessioni e le congestioni erano la regola e non l'eccezione.

Saranno *Robert Kahn* (un esperto di reti della BNN prima e a *DARPA* poi) e *Vint Cerf* (un esperto di sistemi operativi della *Stanford University*) a progettare e realizzare i protocolli che avrebbero reso *Internet* una rete aperta: *TCP* e *IP*.

Kahn fu guidato nel progetto di *TCP/IP* da quattro idee chiave:

- ogni rete interconessa doveva continuare ad esistere e funzionare per proprio conto: nessuna modifica interna doveva essere necessaria per collegarsi a *Internet*;
- la comunicazione doveva avvenire sulla base di un principio di miglior sforzo (*best effort*): se un pacchetto non ce la faceva a raggiungere la propria destinazione sarebbe stato ritrasmesso dal suo mittente originario (e non da qualche gestore intermedio);
- la connessione alla rete sarebbe stata assicurata da *router*, semplici processori che non mantenessero alcuna informazione sul flusso dei singoli pacchetti attraverso di essi: una scelta progettuale che rendeva i *router* leggeri ed economici e li sollevava dal compito di garantire la consegna dei pacchetti;
- non ci doveva essere alcun controllo globale della rete al livello della sua operatività.

TCP e *IP* impiegarono quasi dieci anni per raggiungere la completa maturità. *TCP/IP* sarà reso ufficiale sulla parte militare di *ARPANET* nel 1980 (formando così MILNET), mentre sulla pubblica *ARPANET* il cambio di protocollo avvenne il primo gennaio 1983. Da un punto di vista tecnico, era nata *Internet* come ancora oggi la conosciamo. Ma era ancora una rete di sola ricerca.

Internet: la storia recente

Gli anni 80 sono quelli in cui *Internet* si impone come *standard* de facto per la rete globale.

Non è l'unica rete esistente. Fin dagli anni 70 altre comunità di ricerca si sono dotate di reti indipendenti: HEPNet per la fisica delle alte energie, MFENet per la ricerca sulla fusione magnetica, SPAN per la ricerca spaziale della NASA, CSNET per la ricerca (accademica e industriale) in informatica. A queste si aggiungono reti proprietarie, come DECnet della *Digital Eq. Co.*, SNA della *IBM*, o XNS della *Xerox*. E poi vi sono, negli Stati Uniti, reti private commerciali per lo scambio di posta elettronica, tra cui quella gestita da MCI.

Le reti pubbliche, CSNET prima e NSFNET (la rete finanziata dalla *National Science Foundation* per supportare tutte le discipline accademiche) poi, effettuano alla metà degli anni 80 alcune scelte di grande importanza:

- scelgono *TCP/IP* come protocollo;
- forniscono delle dorsali di connessione (*backbones*) tra i nodi principali delle reti;
- mediante un accordo con gli enti finanziatori di *ARPANET*, viene permesso a CSNET e NSFNET di usare le infrastrutture (in particolare le dorsali che attraversano il continente americano) di *ARPANET*, con un costo a *forfait* e non a consumo.

Da parte sua NSF incentiva i nodi della propria rete a cercarsi utenti commerciali, non accademici, in modo che le risultanti economie di scala potessero essere sfruttate per abbassare i costi per tutti.

È interessante valutare quest'ultima scelta alla luce di un'altra decisione, apparentemente contrastante:

NSF vieta l'uso delle sue dorsali (e a maggior ragione quelle di *ARPANET* che NSF usa sotto convenzione) per tutti gli usi commerciali, cioè che non riguardassero ricerca e formazione.

Le due decisioni sono lungi dall'essere contraddittorie. Incoraggiando il traffico commerciale a livello locale e vietando invece il trasporto di quegli stessi pacchetti sulla lunga distanza, NSF stimolava la nascita e la crescita di aziende private che fornissero la connettività di lunga distanza per scopi commerciali. Nascono così le prime reti private interconnesse a *Internet* e i primi fornitori di connetività, gli *Internet service providers* (ISP). Come sottoprodotto si veniva anche a ridurre il carico sui *backbone* pubblici, perché in molti casi risultava conveniente usare quelli commerciali. Era iniziato il processo di privatizzazione della rete.

La combinazione di questi fattori:

- le oculate scelte di gestione che incentivano la nascita di reti private;
- la quantità di finanziamento veicolato in NSFNET (200 milioni di dollari dal 1986 al 1995);
- la qualità tecnica dei protocolli *TCP*/IP;

ha un impatto enorme. In particolare, vengono gradualmente marginalizzate le reti proprietarie, che fino ad allora avevano visto *TCP/IP* più o meno come una curiosità accademica, anche se lo fornivano insieme ai loro prodotti. Tutte, in un modo o nell'altro, si uniformano e si preparano a collegarsi spostandosi su, o comunque supportando, *TCP*/IP.

Rimane un ultimo vincolo: il divieto di usare i *backbone* pubblici per scopi commerciali. La restrizione, tuttavia, alla lunga appare sempre più una limitazione allo sviluppo di *Internet*. La prima connessione sperimentale avviene nel 1989, collegando a *Internet* la rete di posta elettronica MCI. Nel 1990 *ARPANET* viene dismessa come rete pubblica. Nel 1991 la restrizione di uso non commerciale cade definitivamente. Contemporaneamente, inizia il processo di de-finanziamento pubblico della rete, che cessa di essere finanziata del tutto nel 1995.

Internet: un bilancio storico

Guardando in retrospettiva la storia di *Internet*, vediamo in essa il gioco congiunto di quattro fattori distinti:

- un aspetto tecnologico, che sviluppa l'idea della commutazione di pacchetto mediante protocolli sempre più evoluti e in continuo divenire;
- un aspetto di gestione operativa e strategica di una infrastruttura globale complessa;
- un aspetto sociale, con la creazione di una vasta comunità di persone che, con l'ausilio della rete stessa, lavorano insieme per creare e sviluppare la tecnologia necessaria;
- un aspetto economico, caratterizzato da una transizione estremamente efficace da una struttura di ricerca ad un'infrastruttura informativa di vasta scala effettivamente disponibile.

La storia di Internet su un grafico

Riportiamo qui di nuovo la seguente figura, già presentata nel modulo introduttivo, che riassume in modo sintetico lo sviluppo di *Internet* dai suoi esordi alla fine degli anni 90. A partire dal basso troviamo:

- il numero di reti interconnesse;
- gli eventi più rilevanti;
- la linea del tempo;
- le organizzazioni che coordinano *Internet* da un punto di vista tecnico;
- le organizzazioni che sovrintendono a *Internet* da un punto di vista più vasto, incluso il suo finanziamento.

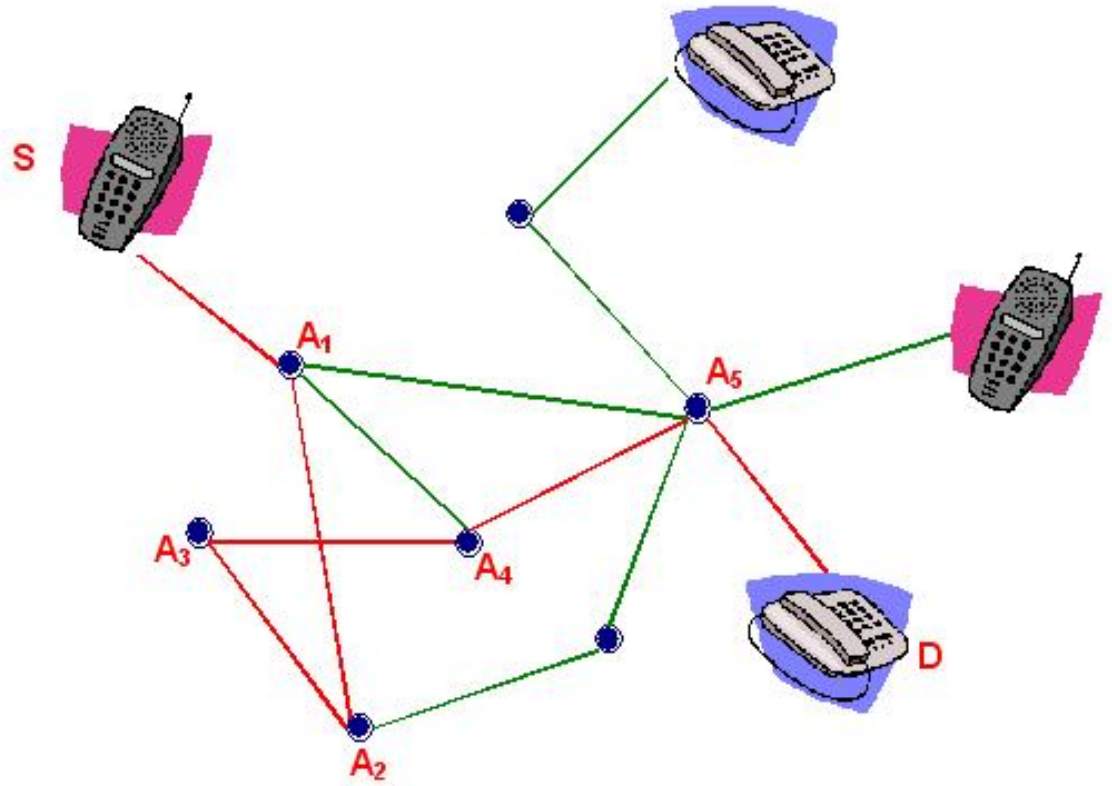


Grafico della storia di Internet

Fonte: *Internet Society*, www.isoc.org

Quanto è grande Internet: nodi

Ci sono aziende private e organismi pubblici che effettuano misurazioni periodiche di vari parametri della rete. Senza scendere in dettagli poco interessanti in questa sede, possiamo vedere alcune di queste misure.

Un primo dato è quanti nodi sono presenti sulla rete, nel senso di quanti indirizzi *IP* sono stati assegnati. Lo *Internet Software Consortium* (www.isoc.org) effettua questa misura ogni sei mesi. L'ultima misura disponibile è del gennaio 2003. Riportiamo i dati nella tabella che segue, assieme al grafico corrispondente.

Gen 2003	171,638,297
Lug 2002	162,128,493
Gen 2002	147,344,723
Lug 2001	125,888,197
Gen 2001	109,574,429
Lug 2000	93,047,785
Gen 2000	72,398,092
Lug 1999	56,218,000
Gen 1999	43,230,000
Lug 1998	36,739,000
Gen 1998	29,670,000
Lug 1997	19,540,000
Gen 1997	16,146,000
Lug 1996	12,881,000
Gen 1996	9,472,000
Lug 1995	6,642,000
Gen 1995	4,852,000
Lug 1994	3,212,000
Gen 1994	2,217,000
Lug 1993	1,776,000
Gen 1993	1,313,000

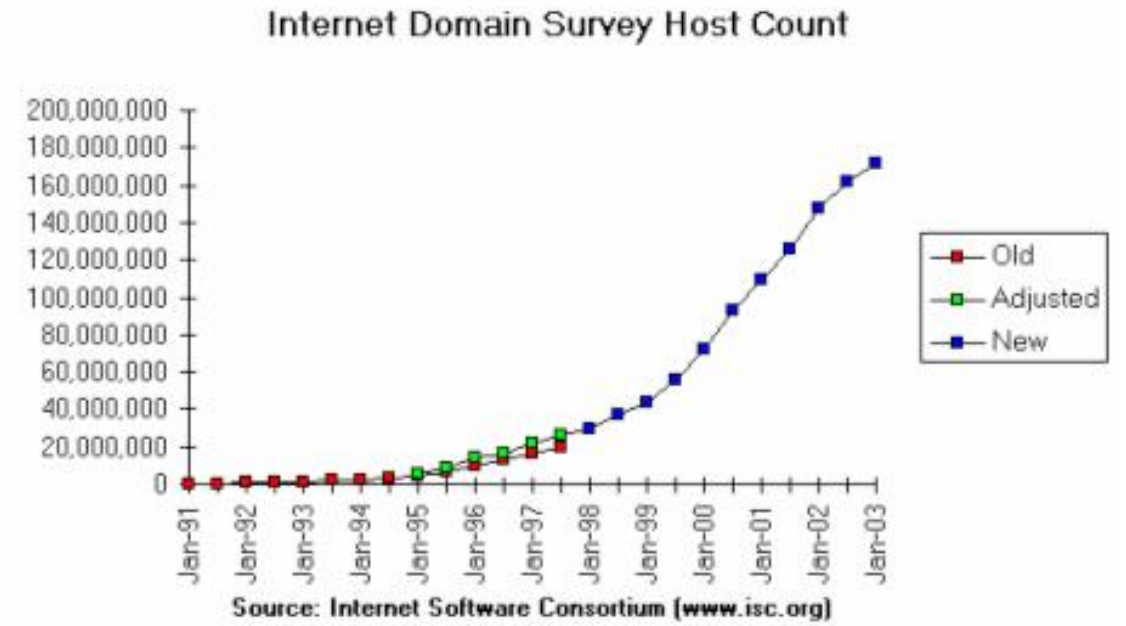


grafico tabella precedente

Si osservi la rapidissima diffusione negli anni 1999-2002, seguita da una relativa diminuzione dell'incremento nell'ultimo anno.

Quest'analisi può essere raffinata in base al dominio di primo livello. Ecco i primi 10 dominii, in ordine decrescente, al gennaio 2003 (fonte: ISC):

com	40555072	*Commercial*
jp	9260117	*Japan*
edu	7459219	*Educational*
arpa	6387463	*Mistakes*
it	3864315	*Italy*
ca	2993982	*Canada*
de	2891407	*Germany*
uk	2583753	*United Kingdom*
au	2564339	*Australia*
nl	2415286	*Netherlands*
br	2237527	*Brazil*
tw	2170233	*Taiwan, Province Of China*
fr	2157628	*France*
mil	1880903	*US Military*
us	1735734	*United States*
es	1694601	*Spain*
se	1209266	*Sweden*
dk	1154053	*Denmark*
fi	1140838	*Finland*

Per quanto riguarda l'Italia, nel 1994 erano registrati 154 dominii di secondo livello con radice it. Lo sviluppo esponenziale della rete in Italia è ben rappresentato dal seguente grafico (elaborazione di dati ISC):

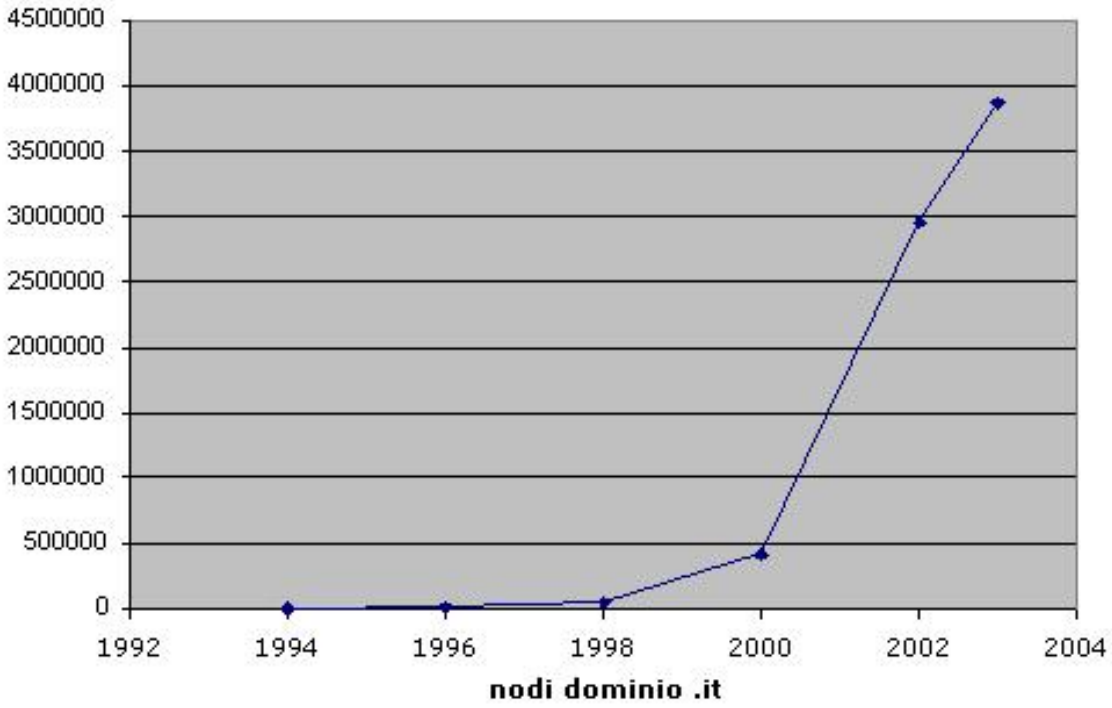
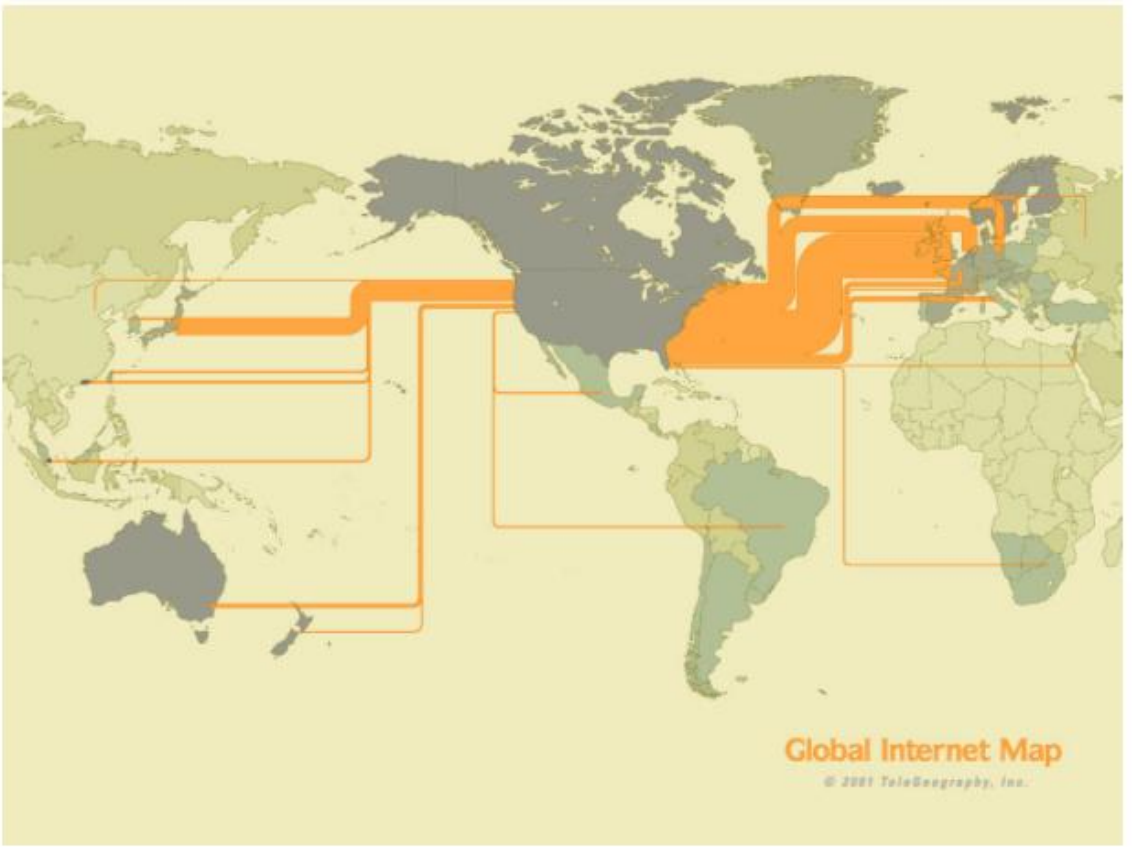


Grafico dei nodi italiani

Quanto è grande Internet: disponibilità di banda

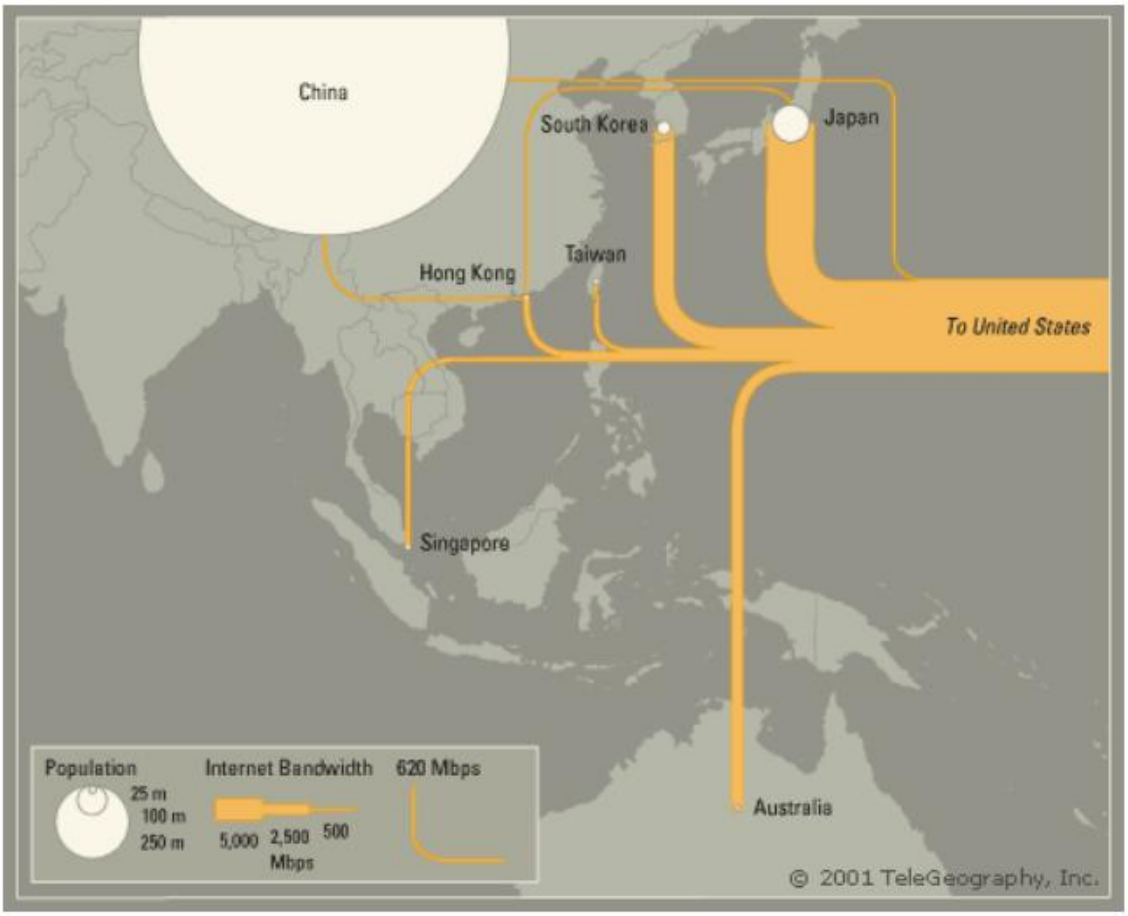
Il numero di nodi è una misura significativa, ma per certi versi non cruciale. Più interessante, per comprendere dove *Internet* sia davvero diffuso e utilizzabile con profitto, è la disponibilità di banda, cioè di quanti *bit* al secondo possono viaggiare sulle connessioni principali tra grandi reti fisiche.

La seguente carta mostra la disponibilità di banda sulle principali dorsali internazionali. Linee arancioni più larghe corrispondono a banda più larga.



Connettività sulle principali dorsali internazionali

Possiamo confrontare la disponibilità di banda con la popolazione residente. Il seguente grafico riporta la connettività transpacifica rapportata alla popolazione: cerchi bianchi più grandi corrispondono a maggior popolazione.



Banda in rapporto alla popolazione in estremo oriente

© *TeleGeography, Inc*. 2001, www.telegeography.com

Quanto è grande Internet: quante persone usano la rete

Un'altra misura, in qualche modo indipendente da quelle appena presentate, è il numero di persone che usano *Internet*, definito come il numero di adulti e bambini che hanno avuto accesso a *Internet* almeno una volta negli ultimi 3 mesi (si contano le persone e non gli *account*: una persona può possedere più *account*, o un *account* può essere condiviso da più persone). La seguente tabella stima questo numero al settembre 2002 (fonte: *NUA Internet Surveys*, www.nua.ie/surveys).

Tutto il mondo	605.60 milioni
Africa	6.31 milioni
Asia/Pacifico	187.24 milioni
Europa	190.91 milioni
Medio Oriente	5.12 milioni
Canada e USA	182.67 milioni
America Latina	33.35 milioni

Il gap tecnologico tra i paesi occidentali e gli altri è così evidente da non aver bisogno di commenti (si osservi che Asia comprende sia Giappone, che Australia, che Cina).

Una misura dello stesso tipo riguarda quanto *Internet* è usata dagli utenti per uso personale. La tabella seguente è relativa a tutta *Internet* e mostra i dati dei mesi di settembre e agosto 2002 (fonte: *Net Ratings*, www.netratings.com); riguarda la stima (ovviamente ottenuta con metodi statistici) de:

- il numero di sessioni per mese;
- il numero di domini diversi visitati per mese;
- il numero di pagine richieste per sessione;
- il tempo speso nel mese per le connessioni;
- il tempo per sessione;
- quanto tempo passa l'utente in *media* su una pagina;
- dimensione della parte attiva di *Internet* (attivo: che risponde ai pacchetti a lui inviati);
- dimensione di *Internet* (quantità di indirizzi *IP* assegnati).

Si osservi la (sostanziale) diversità di questi due ultimi dati rispetto a quelli prima riportati determinati da ISOC.

SEPTEMBER 2002 GLOBAL INTERNET INDEX AVERAGE USAGE*

	September	August	% Change
Number of Sessions per Month	19	19	1.99
Number of Unique Domains Visited	49	48	0.77
Page Views per Month	778	785	-0.97
Page Views per Surfing Session	40	41	-2.90
Time Spent per Month	10:17:45	10:17:44	-0.00
Time Spent During Surfing Session	0:31:44	0:32:22	-1.95
Duration of a Page Viewed	0:00:48	0:00:47	0.98
Active Internet Universe	220,444,008	218,038,452	1.10
Current Internet Universe Estimate	385,564,028	385,998,080	-0.11

*Home Internet Access

Uso di Internet

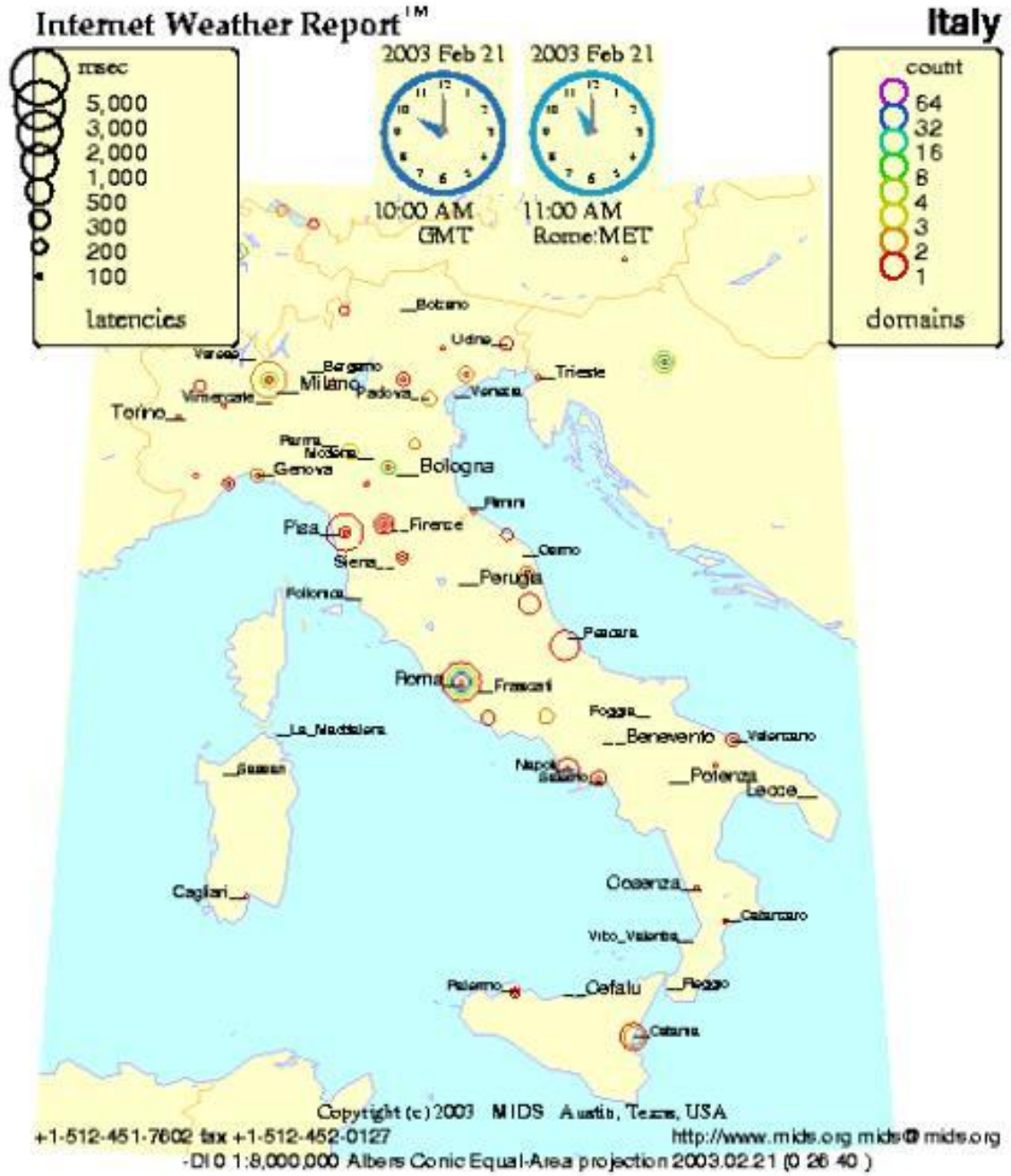
Internet: alcuni parametri di funzionamento

Lo stato di salute di *Internet* è costantemente monitorato da istituzioni pubbliche e private. Lo scopo di queste misure è sia quello di rilevare anomalie (che possono corrispondere a malfunzionamenti o attacchi alla sicurezza) sia quello di individuare le strategie più efficaci per garantire una certa qualità di servizio.

Alcune aziende fanno della misura di alcuni parametri di funzionamento della rete il

proprio *core business*.

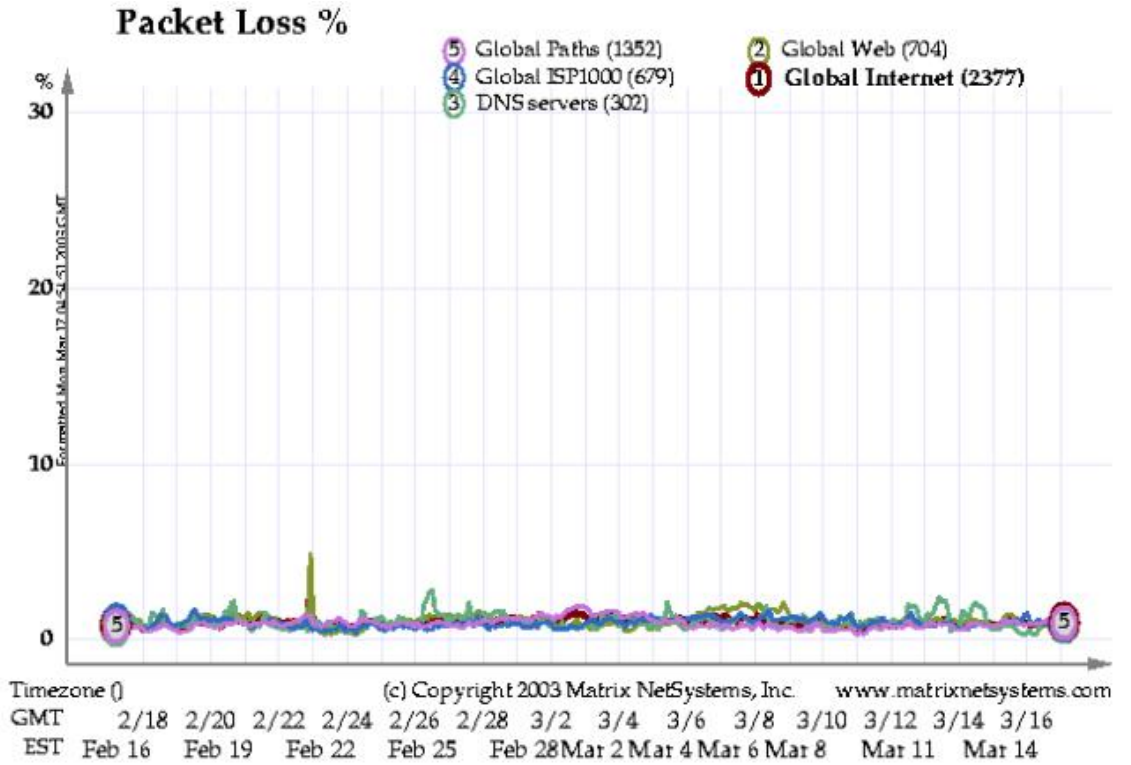
A titolo di esempio riportiamo i seguenti dati, che si riferiscono al tempo di latenza, cioè al tempo che un pacchetto impiega ad andare e tornare tra due siti noti. La *MIDS* (situata ad *Austin*, *Texas*) effettua queste misure molte volte al giorno verso alcuni siti sparsi in tutto il mondo e chiama il risultato *Internet weather report*. La figura che segue riporta graficamente il tempo di latenza verso l'Italia misurato il 21 febbraio 2003 alle ore 11. La dimensione dei cerchi è proporzionale alla latenza. Il colore dei cerchi indica rispetto a quanti calcolatori è ottenuta la misura.



Latenza

MIDS, www.matrix.net

La stessa azienda misura anche la perdita di pacchetti, in percentuale sul numero totale di pacchetti inviati. Il grafico seguente mostra tale valore per i 28 giorni compresi tra il 18 febbraio e il 16 marzo 2003. Si osservi un valore particolarmente alto (quasi il 3%) il 23 febbraio.



Perdita di pacchetti

MIDS, www.matrix.net

I valori riassuntivi per lo stesso periodo (riferiti all'intera *Internet*) sono (in percentuale sui pacchetti scambiati):

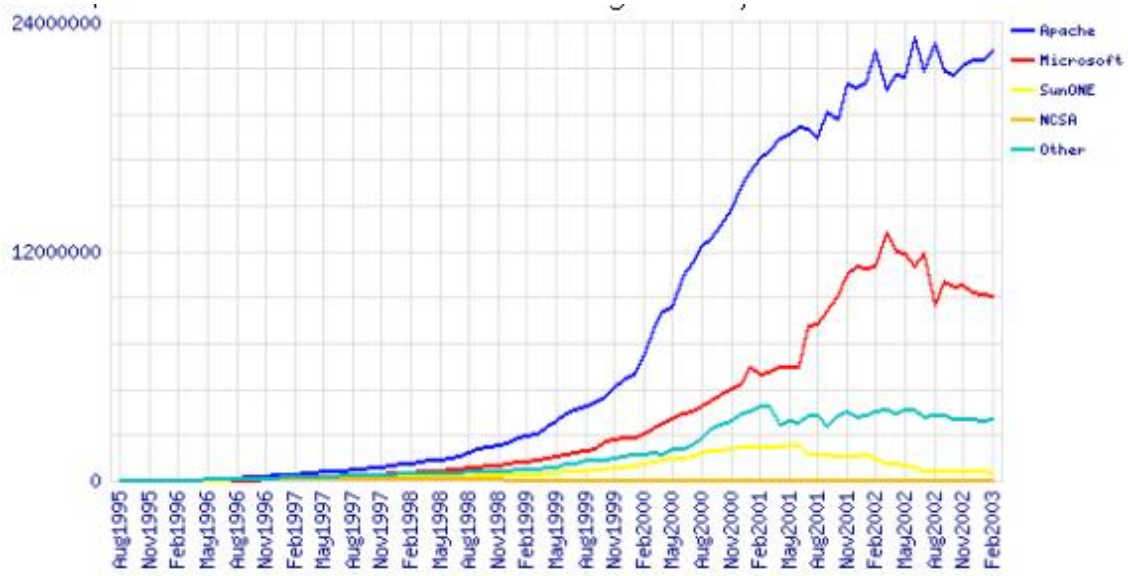
min	mediana	max	media
0.46	1.01	*+90	1.01

Quanto è grande il Web

Analogamente a quanto abbiamo per fatto per *Internet*, è interessante cercare di capire alcune dimensioni del *Web*.

Un primo dato riguarda il numero di siti, determinato come numero di *server* che rispondono ad una richiesta HTTP. Il grafico seguente riporta questo numero ripartito secondo il produttore del *server*. Il numero totale di *server* si ottiene sommando le ordinate delle cinque curve. Nel febbraio 2003 *Netcraft* ha rilevato 35,863,952 *server*, stimando che circa la metà di essi siano attivi (gli altri corrispondono a siti in cui non è stata posta alcuna informazione: esistono perché qualcuno ha registrato quel dominio e

vi ha piazzato un *Web server*, ma senza che vi siano documenti significativi).



Numero di Web server

©*Netcraft* 2003, www.netcraft.com

Più difficile è stimare la dimensione del *Web* in termini di documenti disponibili. Una prima stima ci viene *data* dai motori di ricerca (che saranno descritti in dettaglio nel prossimo modulo: i servizi di *Internet*). Al marzo 2003 il motore di ricerca *Google* indicizza 3,083,324,652 pagine.

Il Web o i Web ?

Il numero di pagine indicizzate da un motore di ricerca è una (notevole) sottostima del numero di documenti che costituiscono il *Web*.

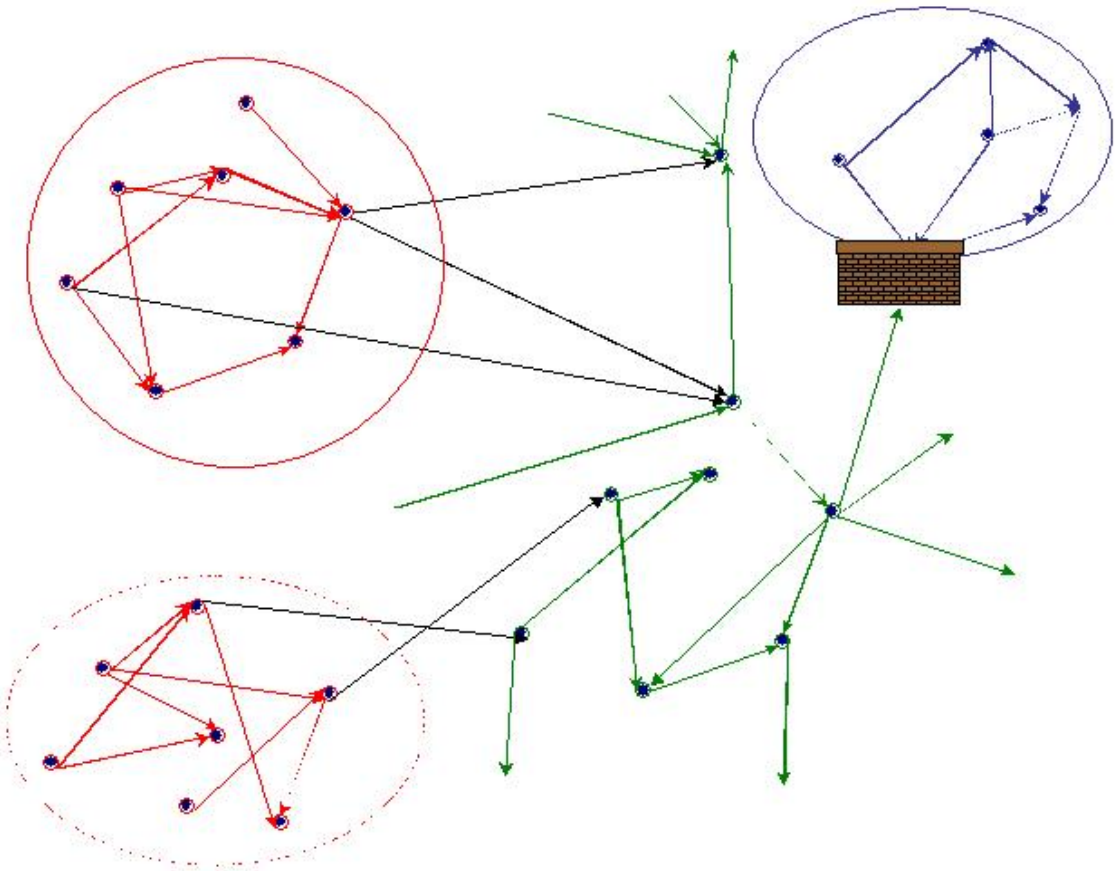
Vi sono documenti che, per varie ragioni, non sono indicizzati:

- documenti che mediante opportune indicazioni (riportate dal *server* che li ospita) sono esplicitamente esclusi dall'indicizzazione.
  Un opportuno *file* (robots.txt) sul *server* può indicare ai motori di ricerca quali informazioni escludere: i motori accreditati seguono tali indicazioni. Motori malandrini che scandagliano la rete alla ricerca di informazioni particolari (per esempio indirizzi di posta elettronica) ovviamente si fanno beffe di tali indicazioni.
- Documenti costituiti da materiale multimediale non indicizzabile.
  Per esempio suoni, o video. Fino a poco tempo fa anche le immagini non venivano indicizzate, cosa che invece avviene adesso, sulla base del contesto in cui si trovano e del nome assegnato al *file* che le contiene.
- Documenti che non hanno alcun collegamento verso di loro.
  Un motore di ricerca spazza il *Web* seguendo in modo sistematico tutti i collegamenti che trova. Se non c'è alcun collegamento verso un

documento, questo non sarà mai scoperto da un motore di ricerca. D'altra parte un utente che conosca lo URI di uno di questi documenti può accedervi senza problemi.

- Documenti che si trovano in una intranet, dietro un *firewall*.
- documenti generati dinamicamente sulla base di richieste di utenti.
  Non esistono statistiche affidabili di quanta informazione sia disponibile in questo modo. Si tratta in ogni modo di enormi quantità di dati. Un esempio tipico è costituito dagli elenchi del telefono: i dati veri e propri non sono accessibili dal *Web*. Mediante un'opportuna richiesta, un *server* genera dinamicamente una pagina che contiene i dati richiesti, che non vengono memorizzati sul *server*, ma solo visualizzati dal *browser* che ha effettuato la richiesta.

Ci dobbiamo dunque rappresentare il WWW come composto da più componenti, non tutte connesse tra loro. Nella seguente figura vediamo una rappresentazione schematica di questa situazione. I due ovali rossi contengono documenti tra loro collegati, ma per i quali non esiste alcun collegamento verso di loro. L'ovale azzurro è il *Web* di una intranet. I collegamenti verdi sono la parte connessa del *Web*.



Componenti inaccessibili del WWW

Ipertesti: la storia

Nel contesto tecnologico, l'idea di ipertesto e dell'organizzazione della conoscenza su base reticolare e associativa - invece che sequenziale - fa la sua comparsa nel 1945. *Vannevar Bush*, ex presidente del *Massachusetts Institute of Technology* e Direttore dello *Office of Scientific Research and Development* del governo USA, in un lavoro giustamente famoso immagina un'organizzazione ipertestuale ante-litteram, che chiama memex.

Vannevar Bush, *As We May Think*, specialmente le sezioni 6 e seguenti; *The Atlantic Monthly*, *July* 1945; www.theatlantic.com/unbound/flashbks/computer/bushf.htm

La parola ipertesto sarà usata per la prima volta nel 1965, da *Ted N. Nelson*. Ispirandosi al memex di *Bush*, *Nelson* introduce un ipertesto come:

un corpus di materiali scritti o grafici interconnessi in un modo così complesso da non poter essere ragionevolmente presentato o rappresentato su carta. Può contenere sommari, o schemi dei suoi stessi contenuti e delle loro relazioni reciproche; può contenere annotazioni, aggiunte e note [...] Un tale sistema potrebbe crescere senza limiti, inglobando gradualmente una parte sempre più ampia della conoscenza scritta esistente al mondo.

Il prefisso iper- è scelto per indicare che un ipertesto:

non può essere essere veicolato in modo sensato utilizzando *media* sequenziali.

Ted N. Nelson, *A file structure for the complex, the changing, and the indeterminate*; *20th National ACM Conference, New York*

Disponibile *on-line* a: http://elib.cs.berkeley.edu/cgi-bin/pl_dochome?collection=Digital+Documents&id=4

Nelson non era il solo a sperimentare con organizzazioni reticolari della conoscenza. Fin dal 1962 *Doug Engelbart*, allora allo *Stanford Research Institute*, lavorava su un sistema basato su calcolatore per la costruzione e la realizzazione di (quelli che *Nelson* avrebbe chiamato) ipertesti. Nel 1968 ne effettua la prima dimostrazione pubblica, presentando NLS (*an oNLine System*). Per seguire i collegamenti interni di un documento le schede perforate, le telescriventi e i (costosi) terminali video a linea di testo non sono sufficienti. *Engelbart* risolve il problema con il tocco del genio, inventando un piccolo strumento per puntare sullo schermo. È nel corso di quella (famosa) presentazione del 1968 che fa la sua comparsa per la prima volta il *mouse*.

Il filmato dell'intera presentazione (90 minuti) è disponibile *on-line* per concessione della *Stanford University*:

http://sloan.stanford.edu/mousesite/1968Demo.html

Da allora sono stati progettati molti sistemi per realizzare ipertesti. Ma nessuno aveva realizzato la possibilità di sfruttare l'intera *Internet* come supporto.

Dagli ipertesti al Web

È alla fine degli anni '80 che al CERN di Ginevra (un centro di ricerche in fisica delle alte energie) si inizia a progettare un sistema di documenti ipertestuali distribuiti, affinché i membri dei vari gruppi di ricerca - sparsi su tutto il mondo - potessero

collaborare.

È l'opera di *Tim Berners-Lee*, che specifica il proprio sistema ipermediale nel 1989. L'idea nuova che Berners-Lee introduce è che un documento possa essere indicato in modo univoco attraverso un *Universal Document Identifier*, quello che oggi chiamiamo URI. Per specificare un documento progetta HTML, e affinché i documenti potessero essere recuperati viene progettato il protocollo HTTP.

Il primo prototipo di editore di ipertesti di questo tipo sarà prodotto nel 1990 (col nome *WorlDwidEWeb*), insieme al primo *Web server*, ed utilizzato dalla comunità dei fisici delle alte energie mediante un *browser* in sola modalità testuale.

Il primo *browser* multimediale verrà rilasciato nel 1993: è *Mosaic*, scritto da uno studente, M. *Andreesen*.

Andreseen si metterà in proprio nel 1994, fondando *Netscape*, Inc.

Un'idea dell'esplosione del *Web* tra il 1992 e il 1994 la si può avere dal grafico che segue. Mostra il carico sostenuto dal primo *Web server* (info.cern.ch) dal 1992 al 1994. Si osservi che la scala delle ordinate è logaritmica: il carico è aumentato di 1000 volte...

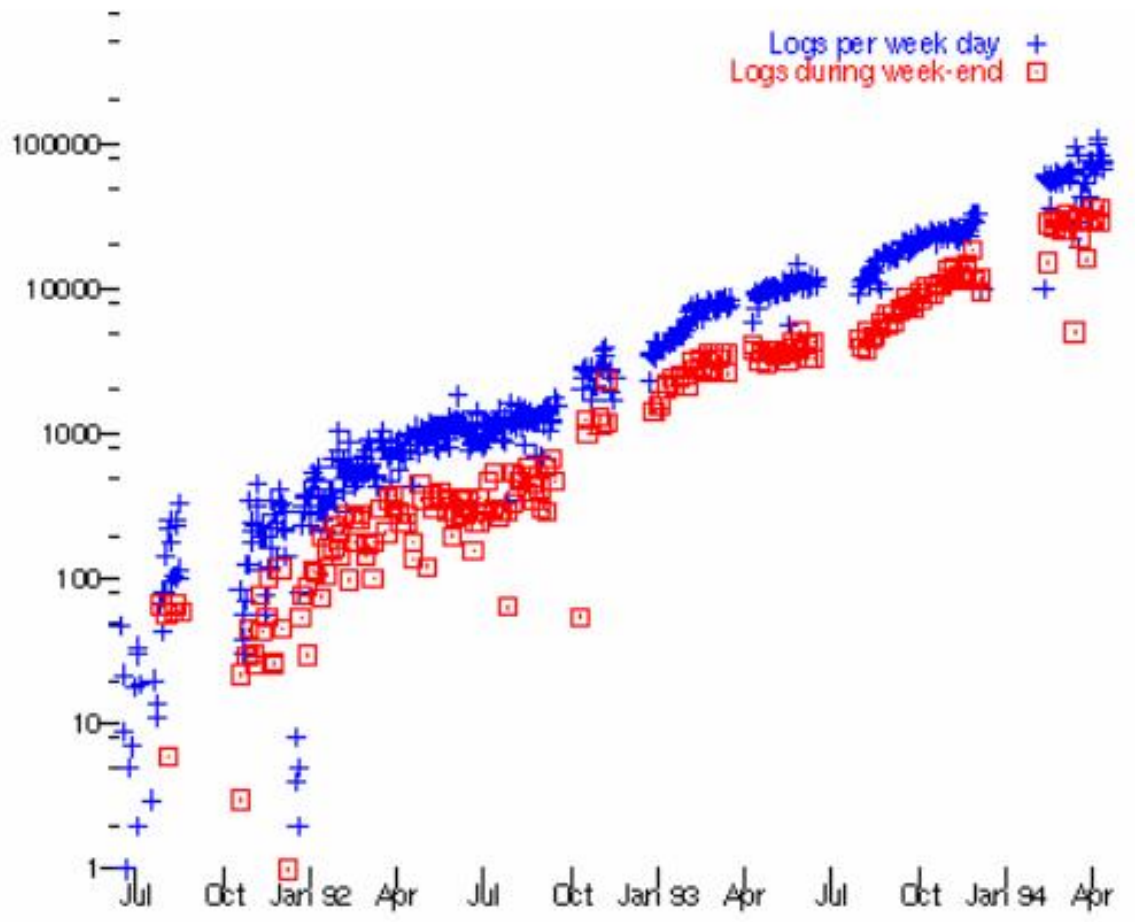


Grafico carico sostenuto da info.cern.ch

© W3C, www.w3.org/History.html

Conclusioni

Abbiamo presentato la storia di *Internet* e del WWW. Si sarà osservato che:

- non si tratta di una storia di sola tecnologia;
- anzi, fondamentale per il successo, anche tecnico, della rete è l'aspetto di gestione dell'innovazione e di formazione di una comunità di internauti omogenea, ma, allo stesso tempo, aperta;
- il ruolo di diffusione libera dell'informazione ha svolto un ruolo fondamentale nello sviluppo di *Internet*;
- le idee fondamentali e più innovative non si sono prodotte all'interno di aziende informatiche, ma come risultato di una stretta interazione tra ricerca pubblica, ricerca accademica, industria.

Struttura di Internet ed il livello rete

Maurizio Gabbrielli

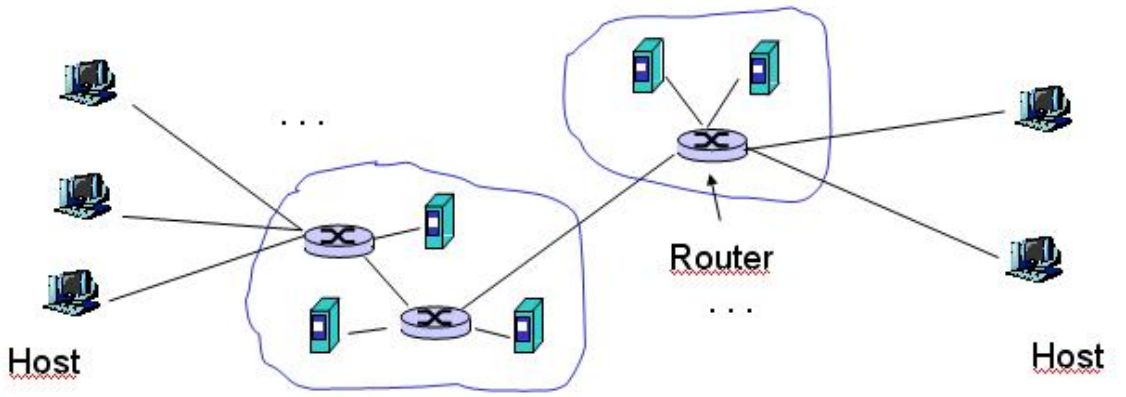
11.2 (Struttura di Internet ed il livello rete)

Indice

La struttura della rete

Internet, così come ogni altra rete di calcolatori possiamo vederla suddivisa nei seguenti componenti:

- Estremità della rete: contengono gli *host* e le applicazioni
- Nucleo della rete: contiene
  - *router*;
  - rete di reti.
- Rete di accesso e mezzi fisici: questa parte è quella costituita da collegamenti (*link*) fisici



La struttura della rete

Le estremità della rete:

Sistemi finali (*host*, terminali):

- sono calcolatori su cui girano i programmi applicativi (che usano i protocolli di livello applicazione e di livello trasporto) quali ad esempio:
  - *browser* WWW;
  - programmi per *email*;
  - ...

A livello di comunicazione fra terminali della rete si usano due modelli di interazione:

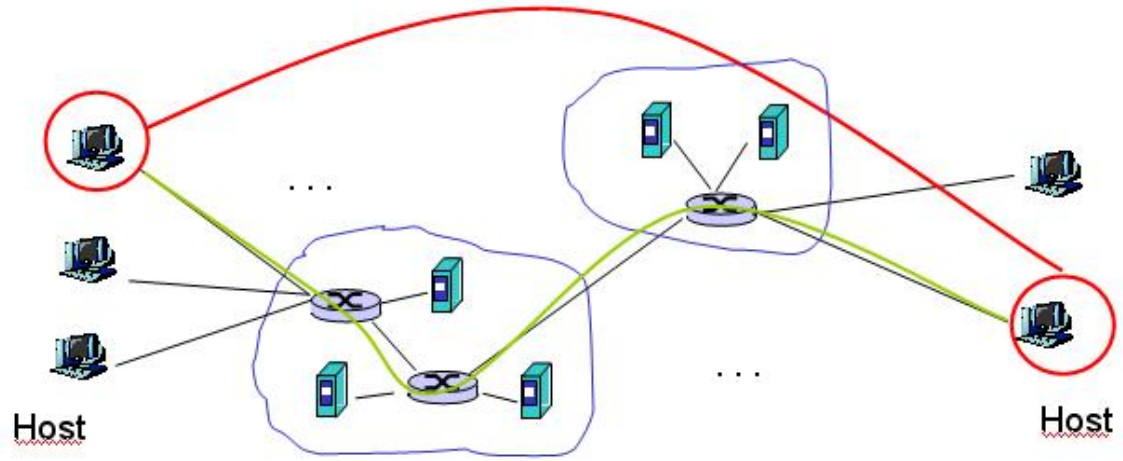
- *client/server*:

    - il *client* richiede e riceve servizi dal *server*;
    - ad esempio, nella navigazione nel *Web* abbiamo: *client* (*browser*)/ *server* (*Web server*);
- *peer-to-peer*:
    - interazioni simmetrica fra *host* pari;
    - esempio: teleconferenza.

Comunicazione fra estremità della rete

Dal punto di vista logico la comunicazione avviene fra terminali.

Dal punto di vista fisico invece avviene passando attraverso i vari nodi della rete.



Comunicazione tra estremita' della rete

Il nucleo della rete

È costituito da una maglia (*mesh*) di *router* interconnessi che permettono la comunicazione di dati.

- i *router* sono calcolatori specializzati ai quali sono collegati vari *link* di ingresso e di uscita. La funzione del *router* è quella di instradare (o commutare)i dati che arrivano sui *link* di ingresso verso gli opportuni *link* di uscita, realizzando così fisicamente l'indirizzamento dei dati. Sui *router* girano i protocolli del livello rete e del livello fisico, ma non quelli dei livelli trasporto e applicazione.

Due modalità principali per il trasferimento dei dati attraverso la rete:

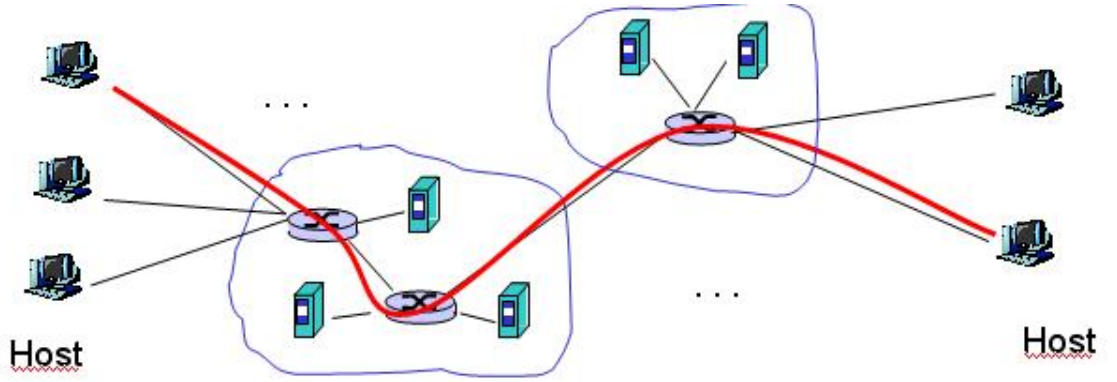
- Commutazione di circuito:
    - c'è un circuito dedicato per ogni chiamata (Esempio: rete telefonica)
- Commutazione di pacchetto:
    - i dati sono spediti attraverso la rete in pacchetti discreti. Ci sono due tipi di reti a commutazione di pacchetto:
    - reti circuiti virtuali (ad esempio, reti ATM):
        - viene stabilito un percorso (circuito virtuale) prima di iniziare a

inviare i dati; i pacchetti che costituiscono uno stesso messaggio vengono spediti tutti su un solo circuito virtuale (identificato da un numero);
    - reti *datagram* (questo è il caso di *Internet*):
        - non c'è alcun percorso prestabilito: pacchetti diversi dello stesso messaggio possono seguire percorsi diversi

Commutazione di circuito

Risorse riservate per ogni chiamata:

- la banda del *link* è divisa in parti, ogni parte è assegnata ad una chiamata. Ci sono due modalità di divisione della banda:
    - FDM divisione di frequenza (Freq. Div. *Multiplexing*);
    - TDM divisione di tempo (*Time* Div. *Multiplexing*).
- La risorsa è mantenuta per tutto il collegamento (nessuna condivisione): se la chiamata non la utilizza la risorsa questa è inattiva.
- *Performance* garantita in modo simile ad un circuito fisico.
- È richiesta una fase di inizializzazione della chiamata (*call-setup*).



Commutazione di circuito

Commutazione di pacchetto

- Il flusso di dati fra terminali è suddiviso in pacchetti:
    - i pacchetti di utenti diversi condividono le risorse;
    - ogni pacchetto usa tutta la banda del link;
    - le risorse sono usate in base al bisogno.
- Competizione per le risorse:
    - la domanda totale può eccedere la disponibilità;
    - si può avere congestione: pacchetti si accodano (in opportuni *buffer* sui *router*) aspettando di usare il *link*; nel caso in cui il *router* non ce la faccia a smistare abbastanza velocemente i pacchetti le code si allungano e si possono perdere pacchetti quando i *buffer* sono saturi;
    - meccanismo *store and forward*: i pacchetti si muovono un passo alla volta ripetendo i seguenti due passi:
        - trasmetti sul link;
        - aspetta il turno al prossimo link.

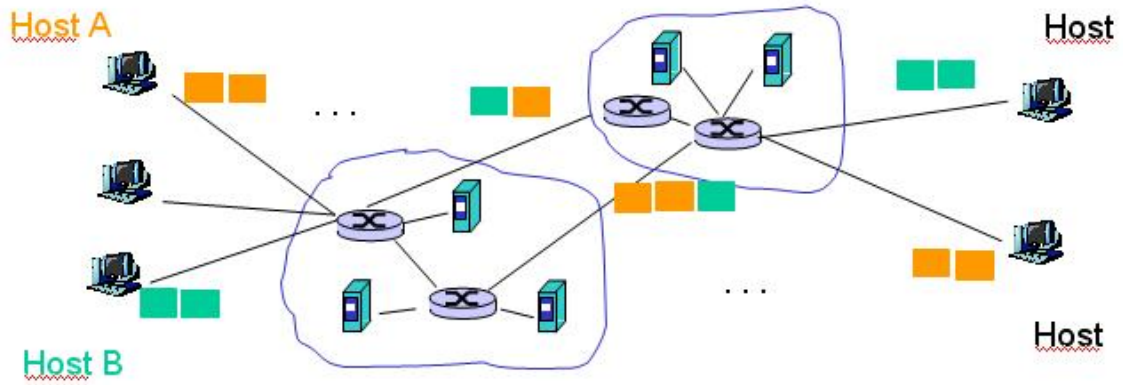
- Non c'è:
  - divisione della banda in parti;
  - allocazione dedicata;
  - prenotazione risorse.

Commutazione di pacchetto (rete datagram)

Pacchetti dello stesso messaggio possono seguire percorsi diversi.

Pacchetti di messaggi diversi possono condividere gli stessi link.

I messaggi sono suddivisi in pacchetti di dimensioni più piccole per ottimizzare le prestazioni: pacchetti diversi (di uno stesso messaggio) possono transitare in parallelo su *link* diversi verso la destinazione finale.



Commutazione di pacchetto (rete datagram)

Commutazione di pacchetto o commutazione di circuito?

Commutazione pacchetto:

- ottima per dati a ondate perchè permette la condivisione di risorse e non c'è alcuna inizializzazione;
- possibile congestione: l'accumulo di pacchetti che il *router* non riesce a smistare può provocare ritardo e perdita di pacchetti;
- necessari opportuni protocolli per trasferimento affidabile (vedi *TCP*);
- necessario controllo della congestione.

Commutazione di circuito:

- va bene per applicazioni che inviino un flusso di dati costante e abbiano bisogno della garanzia di una banda minima.

Esempio: se *link* di 1 Mbit *link* e ogni utente ha bisogno di 200 Kbps ed è attivo il 5% del tempo, con la commutazione di circuito si possono avere al massimo 5 utenti, mentre con la commutazione di pacchetto si possono avere 20 utenti in contemporanea (con una probabilità molto bassa che ci siano più di 5 utenti attivi nel stesso momento).

Il livello rete

Facendo riferimento alla gerarchia ISO/OSI, il livello rete fornisce un servizio di trasferimento dati fra terminali e *router*.

A differenza dei livelli trasporto e applicazione, il livello rete è implementato anche nei *router* della rete.

il livello rete sostanzialmente realizza le seguenti tre funzioni:

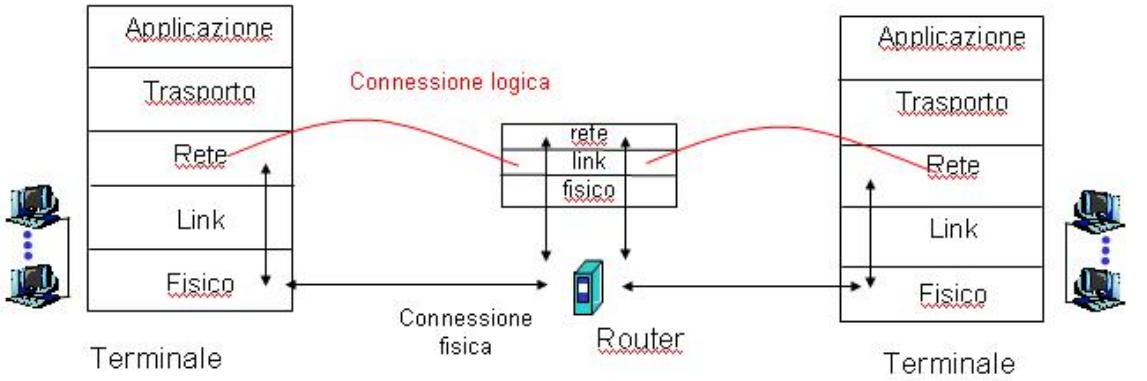
- determinazione del cammino: al livello rete viene deciso il cammino che i pacchetti devono seguire per raggiungere la destinazione. vengono usati allo scopo opportuni algoritmi di *routing* che sono implementati nei *router*.
- *Switching*: i pacchetti devono essere fatti transitare dal collegamento di *input* di un *router* all'opportuno collegamento di *output*.
- *Call setup*: alcune architetture di rete richiedono una fase di inizializzazione della chiamata (call *setup*) prima che possa iniziare la spedizione dei pacchetti (esempio ATM e circuiti virtuali).

Il livello rete in Internet: protocollo IP

Il protocollo del livello rete di *Internet* si chiama protocollo IP

Il protocollo *IP* prevede una rete a commutazione di pacchetto di tipo *Datagram* con le seguenti caratteristiche:

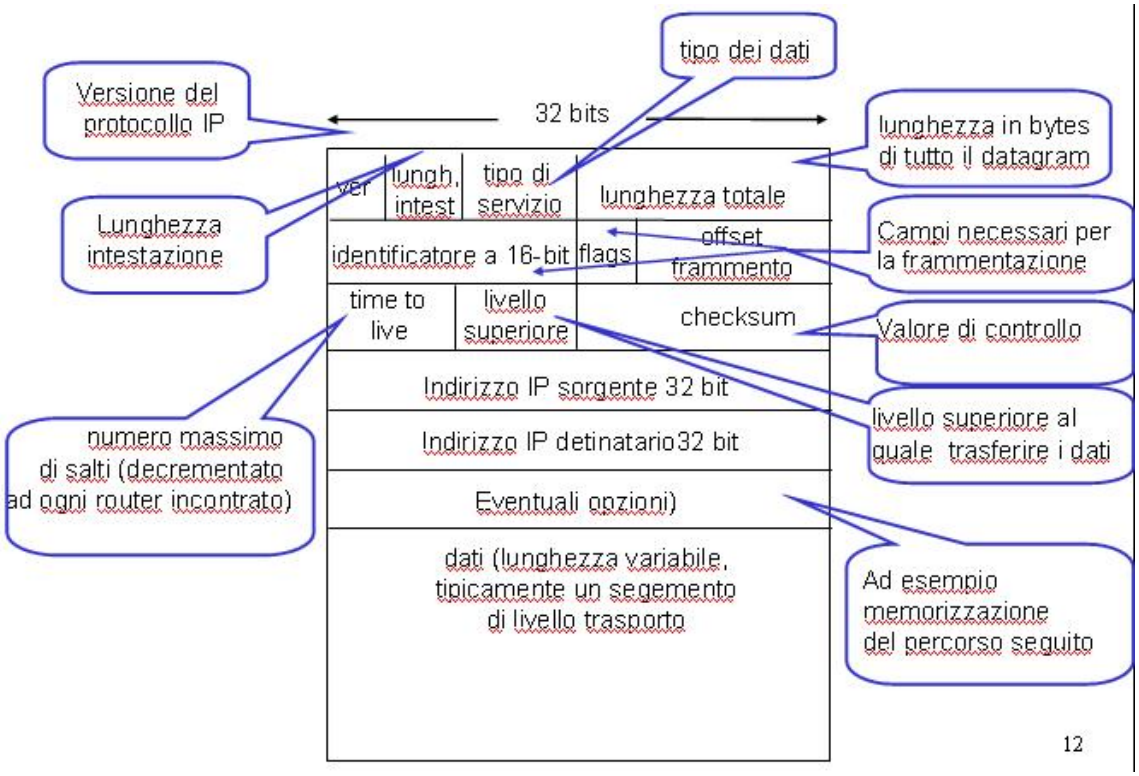
- permette la spedizione di pacchetti dal mittente al destinatario;
- non c'è alcuna garanzia di affidabilità: i pacchetti si possono perdere;
- non c'è nessuna inizializzazione della chiamata;
- I *router* non mantengono lo stato delle comunicazioni;
- nessun concetto di connessione;
- ogni pacchetto contiene l'indirizzo *IP* del destinatario (per poter permettere ai *router* la determinazione del cammino).



Il livello rete in Internet: protocollo IP

Formato del pacchetto (datagram) IP

Secondo il protocollo *IP* i dati sono organizzati in pacchetti fatti come mostrato qui sotto:

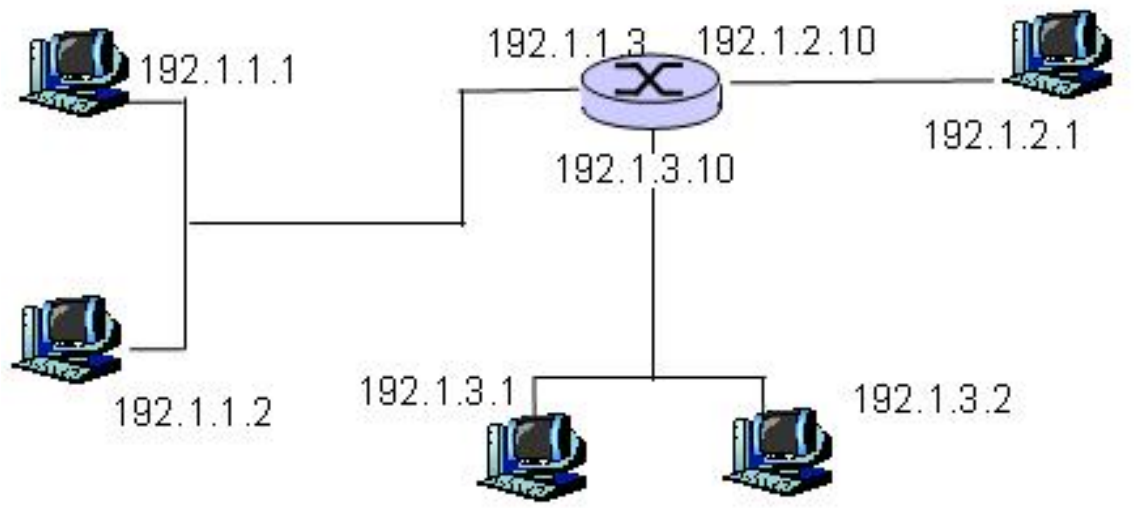


Formato del pacchetto (datagram) IP

Cosa è un indirizzo IP

- Una interfaccia è una connessione fra *host, router* e *link* fisico: i *router* tipicamente hanno molte interfacce, gli *host* possono avere o no molte interfacce.
- Un indirizzo *IP* è un identificatore a 32-*bit* (divisi in 4 gruppi di 4 *bit*) per interfaccia di un *host* oppure di un *router*. SI noti che anche se comunemente si parla di indirizzp *IP* di un *host*, in realtà l'indirizzo è associato all'interfaccia e non all'*host* o il *router*. Il seguente è un esempio di indirizzo IP 192.1.1.1 che corrisponde (in binario) a 11000000 00000001 00000001 00000001.

Sotto abbiamo una esempio di vari indirizzi *IP* associati a diverse interfacce:



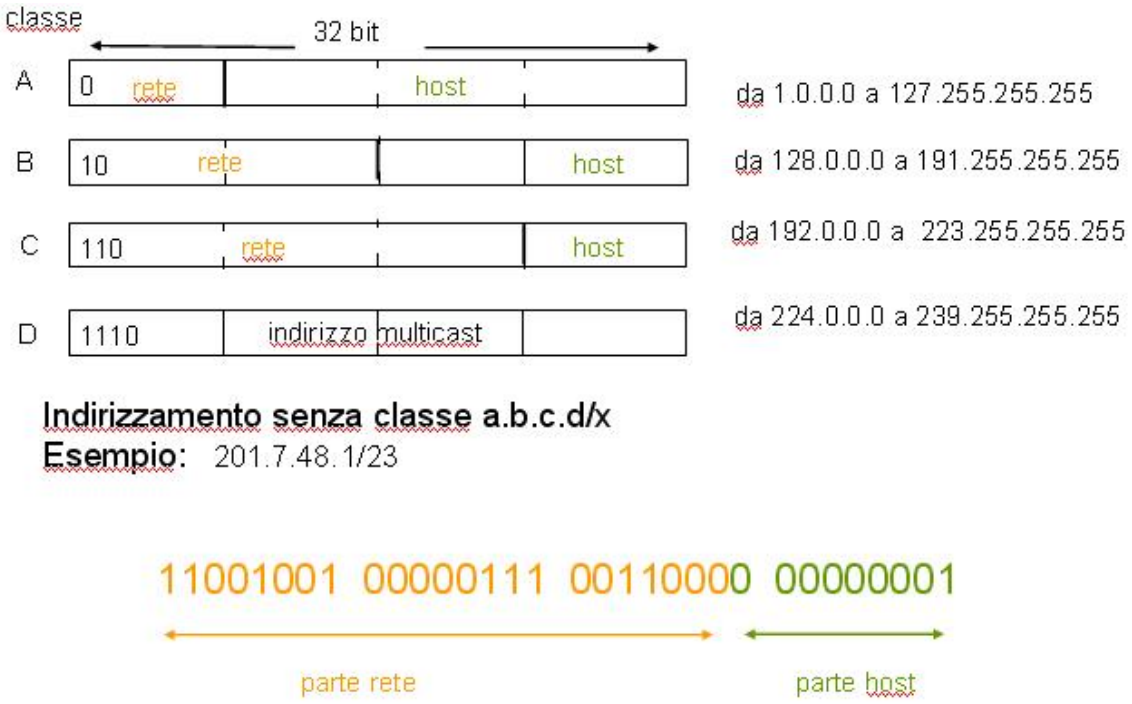
Indirizzi IP

Tipi di indirizzo IP

- Un indirizzo *IP* è costituito da due parti: la parte rete (a sinistra) e la parte *host* (a destra) . La parte rete identifica i numeri *IP* che appartengono ad una stessa rete *IP*, ovvero quelle interfacce tali che si può andare da una all'altra seza passare da un *router* o da un terminale. All'interno di una rete *IP* poi il numero di *host* identifica una particolare interfaccia. Ad esempio possiamo avere gli indirizzi 127.0.0.1 e 127.0.0.02 che hanno la stessa parte rete (127) e due parti *host* diverse (0.0.1 e 0.0.2).
- La lunghezza delle due parti varia a seconda del metodo di indirizzamento usato. Si distinguono infatti le seguenti modalità:
  - Indirizzamento con classe: in questo caso la parte rete ha una lunghezza fissa che può essere di 8 *bit* e inizia con il *bit* più a sinistra 0 (classe A);16 *bit* e inizia con i due *bit* più a sinistra 10 (classe B); 24 *bit* (classe C) e inizia con i tre *bit* più a sinistra 110. Esiste anche la una classe D che si usa per l'indirizzamento *multicast*.
  - Indirizzamento senza classe: in questo caso la parte rete ha una lunghezza variabile che è indicata esplicitamente: gli indirizzi di questo tipo hanno la forma a.b.c.d/*x* dove *x* indica il numero di *bit* più a sinistra che sono usati per la parte rete.
- PerchÈ due modalità: inizialmente l'indirizzamento era con classe, poi si è passati a quello senza classe per migliorare l'uso dello spazio degli indirizzi: ad esempio, un'organizzazione che avesse bisogno di indirizzi per 2000 *host* (nella stessa rete *IP*) doveva usare indirizzi di classe B (la classe A non basta). Però per un dato indirizzzo di rete di casse B ci sono circa 65000 indirizzi disponibili per gli *host*.

Indirizzi IP

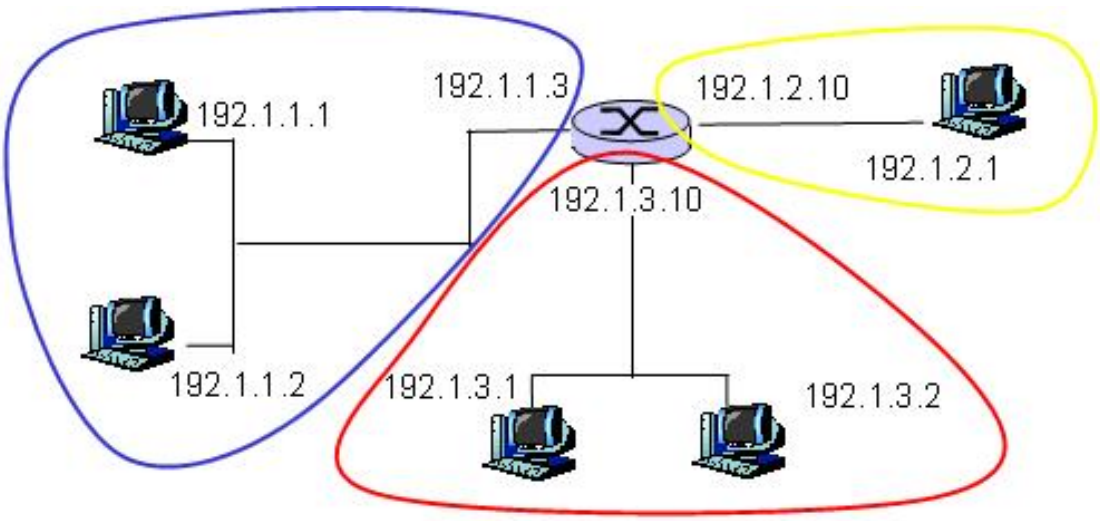
Indirizzamento con classe:



Indirizzi IP

Rete IP

Una rete *IP* (dal punto di vista dell'indirizzamento *IP*) è costituita da un insieme di dispositivi di interfaccia con la stessa parte rete di indirizzo *IP*. In altri termini, si può andare da un punto all'altro della stessa rete *IP* senza passare da un *router* o *host*. Sotto abbiamo un esempio dove sono evidenziate tre reti *IP* (in blu, rosso e giallo):



Rete IP

Per determinare quali sono le reti *IP* dunque basta staccare ogni interfaccia da *host* e *router* e creare isole di reti isolate.

Come ottenere un indirizzo IP?

Parte Rete: per ottenere un blocco di di indirizzi *IP* (corrispondenti ad un indirizzo specifico per a parte rete) ci si rivolge a ICANN (*Internet Corporation for Assigned Names and Numbers*) che provvede a:

- decidere gli indirizzi;
- gestire il DNS;
- assegnare i nomi di dominio e risolvere eventuali dispute e controversie.

Parte *Host*: l'indirizzo della parte *host* invece può essere assegnato staticamenente dall'amministratore di sistema (è scritto in un opportuno *file*) oppure può essere assegnato dinamicamente. Questo avviene quando ci colleghiamo a *Internet* usando un modem ed i servizi di un ISP (*Internet Service Provider*). In questo caso si usa il *DHCP* (*Dynamic Host* Configuration *Protocol*), un protocollo che prevede le seguenti fasi:

- *host* manda un messaggio cerca DHCP;
- DHCP *server* risponde con un messaggio offerta DHCP;
- *host* richiede un indirizzo *IP* con un messaggio: richiesta DHCP;
- DHCP *server* manda l'indirizzo assegnato con un messaggio : *ack* DHCP.

Il livello trasporto ed il protocollo TCP

Maurizio Gabbrielli

11.3 (Il livello trasporto ed il protocollo TCP)

Indice

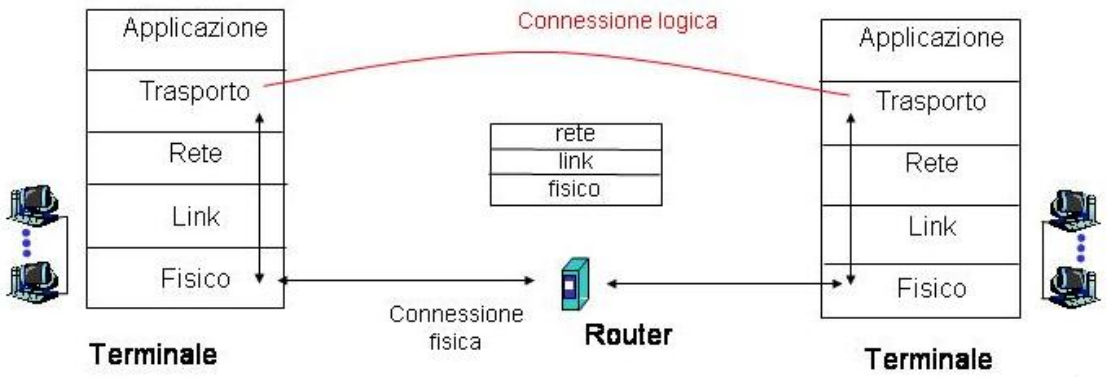
Cosa è il livello trasporto

Il *livello trasporto* fornisce una *comunicazione logica fra processi applicativi* che girano su terminali diversi. I processi del livello applicazione per spedire dei dati li passano al livello trasporto.

I protocolli di trasporto girano solo nei terminali della rete (*end systems*) e non nei *router*.

Confronto servizi trasporto/rete:

- livello rete: realizza il trasferimento dati fra terminali usando le funzionalità del livello link
- livello trasporto: realizza il trasferimento dati fra processi: si basa sui (ed estende i) servizi di livello rete.



La comunicazione logica a livello trasporto

Servizi e protocolli del livello trasporto

Servizi di trasporto in *Internet*: il livello trasporto realizza due diversi tipi di servizio e quindi prevede due protocolli principali:

- Servizio orientato alla connessione:
  - realizza il trasferimento di dati fra *end-systems* (terminali).
  - prevede un fase iniziale di *handshaking* nella quale viene preparato lo stato dei due terminali che devono comunicare
  - TCP (*Transmission Control Protocol*) è il protocollo che realizza il servizio orientato alla connessione di Internet
- Servizio senza connessione:
  - realizza il trasferimento di dati fra *end-systems* (terminali).
  - trasferimento dati inaffidabile
  - no controllo flusso
  - no controllo congestione
  - **UDP** (*User Datagram Protocol*) realizza il servizio senza connessione di Internet
- Non sono disponibili al livello trasporto i seguenti servizi:
  - *real-time*
  - banda minima garantita
  - multicast affidabile

Segmento del livello trasporto

Segmento: l'unità di dati trasferita fra entità del livello trasporto è detta segmento (o anche TPDU *transport protocol data unit*)

In prima approssimazione il segmento del livello trasporto (sia *TCP* che **UDP**) ha il formato illustrato qui sotto

I numeri di porta (sorgente e destinazione) servono per il *multiplexing* ed il *demultiplexing*

Struttura del segmento a livello trasporto

Multiplexing/demultiplexing 1

Il *Multiplexing-demultiplexing* permette una comunicazione fra processi di tipo molti a molti, ovvero processi diversi che girano su uno stesso *host* (con un unico numero di *IP*) possono inviare messaggi a processi diversi che girano su un altro *host* (con un altro numero di *IP*). La corrispondenza fra processo mittente e processo destinatario è possibile grazie ai numeri di porta che identificano i vari processi.

Multiplexing: raccolta di dati da più programmi applicativi e aggiunta ad essi di opportune intestazioni (che serviranno dopo per identificare il destinatario nella fase di *demultiplexing*). Basato su:

- indirizzo *IP* (presente a livello di pacchetti *IP*);
- numeri porta sorgente e porta destinazione presenti in ogni segmento.

Demultiplexing: consegna dei segmenti ricevuti ai corretti processi destinatari del livello applicazione.

Nota: alcuni numeri di porta sono riservati per applicazioni specifiche quali, ad esempio, posta elettronica (protocollo SMTP), trasferimento *file* (FTP), *Web* (HTTP).

Multiplexing/demultiplexing 2

La spedizione di una messaggio comporta le seguenti fasi (vedi figura nella **pagina successiva**):

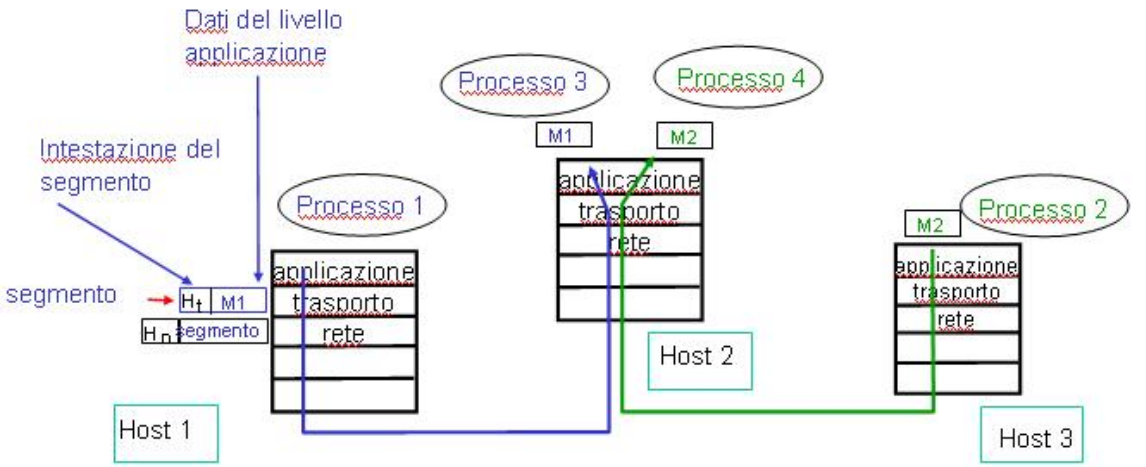
- Nell'host mittente il messaggio M viene spedito dal livello applicazione al

livello rete:il messaggio:

- viene passato al livello trasporto e qui diviso in tanti segmenti;
- ad ogni segmento sono aggiunte le intestazioni (numeri di porta) necessari per il *multiplexing*;
- il segmento viene passato al livello rete;
- il livello rete del mittente scompone ogni segmento in pacchetti.

- La rete quindi fà transitare i pacchetti dall'host sorgente all'host destinazione (usando l'indirizzo *IP* del destinatario).
- Nell'host destinatario il messaggio M viene spedito dal livello rete al livello applicazione:
  - il livello rete del destinatario ricompone i pacchetti per ricostruire i segmenti;
  - da ogni segmento viene estratta la parte dati e inviata al processo opportuno del livello applicazione;
  - il processo è identificato dal numero di porta destinazione presente nell'intestazione del segmento.
  - Al livello applicazione il messaggio è ricostruito mediante i segmenti che lo compongono.

Multiplexing/demultiplexing 3



Esempio di multiplexing descritto nella pagina precedente

UDP: User Datagram Protocol [RFC 768]

UDP è un protocollo del livello trasporto *Internet* molto semplice.

Offre un servizio *best effort* che in pratica non garantisce nulla: i segmenti **UDP** possono essere:

- persi;
- consegnati in ordine errato alle applicazioni.

Offre un servizio senza connessione:

- non c'è *handshacking* iniziale fra il mittente (*sender*) **UDP** *sender* ed il ricevente (*receiver*), ovvero non esiste una fase iniziale nella quale il mittente ed il ricevente stabiliscono una connessione.
- Ogni segmento **UDP** è gestito indipendentemente dagli altri.

Non c'è controllo della congestione e non c'è controllo del flusso: i dati possono essere spediti dal *sender* alla velocità che vuole, senza tenere conto dello stato della rete e senza che il *receiver* possa rallentare la velocità di spedizione.

Trasmissione sia *unicast* che *multicast*.

Usare UDP o TCP?

Perchè usare **UDP**?

- Non viene stabilita una connessione (che può aggiungere fattori di ritardo);
- è semplice: non c'è stato della connessione;
- piccola intestazione di ogni segmento;
- nessun controllo della congestione: **UDP** può mandare i segmenti alla velocità che vuole;
- quindi maggiore efficienza (minore ritardo e maggiore banda) al prezzo di mancanza di affidabilità.

UDP è quindi usato da quelle applicazioni che possono tollerare perdita di dati ma che hanno bisogno della massima banda possibile e di ritardi più bassi possibile:

- tipiche applicazioni che hanno queste caratteristiche e usano **UDP** sono le seguenti: applicazioni multimediali, teleconferenze, giochi interattivi telefonia via *Internet*. Da notare che il DNS (servizio del livello applicazione per la traduzione dei nomi in indirizzi *IP*) usa **UDP**.

Viceversa, *TCP* è usato da quelle applicazioni per le quali la banda ed i ritardi non sono essenziali, mentre è importante l'affidabilità della trasmissione:

- Applicazioni che hanno queste caratteristiche e usano *TCP*: HTTP (WWW), FTP (*file transfer*), Telnet (*remote login*), SMTP (*email*).

Requisiti dei servizi di trasporto per alcune applicazioni comuni

Applicazione	Perdita di dati	Banda elastica	Rilevanza ritardo
Trasferimento *file*	no	si	no
Posta elettronica	no	si	no
Documenti *Web*	si	si	no
audio/video *real time*	si	no (audio: 5kb-1Mb, *video* 10Kb-5Mb)	si 100msec.
audio/*video* no *real time*	si	no (audio: 5kb-1Mb, *video* 10Kb-5Mb)	si pochi sec.
giochi interattivi	si	> alcuni Kbps	si 100msec.
applicazioni finanziarie	no	elastica	si e no

Perdita dati: no -> non tollerata; si -> tollerabile.

Banda elastica: si -> l'applicazione funziona con qualsiasi banda; no -> è indicata la banda minima necessaria per il funzionamento.

Rilevanza ritardo: no -> l'applicazione funziona con qualsiasi ritardo; si -> è indicato il ritardo massimo che permette il funzionamento corretto dell'applicazione.

Applicazioni di Internet ed i relativi protocolli

Applicazione	Protocollo del livello applicazione	Protocollo del livello trasporto usato
Posta Elettronica	smtp [RFC 821]	TCP
Accesso terminale remoto	telnet [RFC 854]	TCP
Web	HTTP [RFC 2068]	TCP
Trasferimento *file*	FTP [RFC 959]	TCP
Streaming multimedia	proprietario	tipicamente **UDP**
Telefono via Internet	proprietario	tipicamente **UDP**

Formato del segmento UDP

UDP è usato tipicamente per applicazioni multimediali che possono tollerare la perdita di dati ma che invece sono sensibili ai ritardi ed alla banda di trasmissione. Il segmento **UDP**, dato che non deve memorizzare informazioni necessarie per realizzare meccanismi di trasporto affidabile, risulta particolarmente snello: infatti contiene soltanto i seguenti campi

- dati: questo campo contiene i dati inviati dall'applicazione;
- porta mittente e porta destinatario: questi due campi, come visto, contengono i numeri di porta (mittente e destinazione) e servono per il *multiplexing*;
- lunghezza: indica la lunghezza in *bytes* del segmento inclusa l'intestazione;
- *checksum*: permette un controllo (parziale) sulla correttezza dei dati trasmessi.

Trasferimento affidabile su **UDP**: è possibile soltanto gestendo al livello applicazione i meccanismi che controllano l'affidabilità della trasmissione: deve quindi essere

realizzato un meccanismo di gestione dell'errore specifico per ogni applicazione.

Struttura segmento UDP

UDP checksum

A che serve: il campo *checksum* contiene un valore di controllo che serve per il rilevamento di eventuali errori nel segmento. Tale controllo avviene secondo le seguenti modalità che coinvolgono sia il mittente che il destinatario del segmento.

- *Mittente*: il mittente considera il contenuto del segmento come una sequenza di interi a 16-*bit* esegue le seguenti operazioni:
  - esegue la somma del contenuto del segmento (visto come sequenza di interi);
  - esegue il complemento a 1 della somma calcolata (il complemento a 1 si ottiene cambiando gli 0 in 1 e gli 1 in 0) il valore così ottenuto è messo nel campo *checksum*.
- *Destinatario*: il destinatario controlla la correttezza del segmento ricevuto mediante le seguenti operazioni:
  - calcola la somma del contenuto del segmento ricevuto (sempre visto come sequenza di interi);
  - somma la somma calcolata al valore contenuta nel campo *checksum*:
    - risultato diverso da sequenza di 1 ---> rilevamento errore (infatti questo significa che la somma fatta dal mittente è diversa dallasomma fatta dal destinatario);
    - risultato costituito da sequenza di 1 ---> nessun errore rilevato.

N.B. Il fatto che non si siano rilevati errori non garantisce l'assenza di errori: la

checksum permette sono un controllo di correttezza parziale.

Servizio orientato alla connessione: TCP

TCP offre le seguenti funzionalità [RFC 793]:

- trasmissione *unicast* (ovvero punto a punto);
- trasferimento di *stream* di *byte* affidabile e ordinato: è garantito che i dati spediti arrivino tutti a destinazione e nell'ordine in cui sono spediti. Vengono usati meccanismi di riscontro (*ack*) e di *time-out* per controllare la perdita di dati;
- controllo della congestione: sono implementati opportuni meccanismi che evitano il sovraccarico (congestione) della rete.: se la rete è congestionata il *sender* rallenta la velocità di spedizione dei segmenti;
- controllo del flusso: il destinatario può obbligare il mittente a non superare una *data* velocità (variabile dinamicamente) di trasmissione per evitare la saturazione del *buffer* di ricezione;
- *setup* della connessione: prima della trasmissione dei dati c'è una fase di inizializzazione della connessione.

TCP in prima approssimazione fà le seguenti cose:

- prende dati dal livello applicazione;
- aggiunge informazioni per indirizzamento e affidabilità;
- forma dei segmenti;
- spedisce segmenti ad un pari su di un altro terminale;
- aspetta la ricevuta (*ack*).

Principi del trasferimento dati affidabile

Il livello rete offre una trasferimento dati che non garantisce in alcun modo l'affidabilità. Il livello trasporto deve quindi implementare dei meccanismi che controllino la ricezione corretta dei segmenti spediti.

Le caratteristiche del canale non affidabile determineranno la complessità del protocollo per il trasferimento affidabile (rdt).

Assumendo che il canale possa:
- perdere dati;
- introdurre errori;
- alterare l'ordine di arrivo dei dati rispetto a quello di spedizione.

L'idea di base del trasferimento affidabile è comunque la seguente per tutti i protocolli: il mittente per ogni segmento spedito si attende che il destinatario gli invii un messaggio di riscontro (detto *ack*) che attesti la corretta ricezione dei dati (l'analogo della ricevuta di ritorno delle raccomandate). Se tale riscontro non arriva entro un dato tempo limite, il mittente assume che i dati siano andati perduti e quindi li rispedisce. Inoltre il mittente rispedisce i dati anche nel caso in cui il destinatario gli abbia mandato un riscontro negativo nel quale lo avverte che i dati sono arrivati con degli errori (questo è controllabile usando la *checksum*, come visto). L'identificazione dei singoli segmenti avviene usando opportuni numeri di sequenza associati ai segmenti. Si deve

cercare di evitare che due pacchetti siano identificati dallo stesso numero di sequenza.

Tipi di trasferimento affidabile

Si possono individuare due categorie di protocolli per il trasferimento affidabile:

- protocolli *stop and wait*: per ogni segmento spedito si aspetta l'*ack* prima di inviare il segmento successivo. L'efficienza dei protocolli *stop and wait* è molto scarsa perchè il canale non viene utilizzato al meglio: dopo aver spedito un segmento per tutto il tempo necessario a questo per raggiungere la destinazione e all'*ack* per tornare indietro nessun altro dato è spedito sul canale che quindi è inutilizzato per la maggior parte del tempo.
- Protocolli *pipeline*: vengono inviati più segmenti senza aspettare l'*ack* dei precedenti. Tali segmenti poi possono poi essere riscontrati individualmente o in modo cumulativo. I protocolli reali del livello trasporto sono di tipo pipeline. Si distinguono due modelli di protocolli pipeline:
    - *GO-BACK* n: i segmenti sono riscontrati in modo cumulativo.
    - Ripetizione Selettiva: i segmenti sono riscontrati singolarmente.

TCP usa un protocollo di tipo *pipeline* con riscontri gestiti mediante una combinazione delle tecniche go-*back* n e ripetizione selettiva

TCP: Introduzione

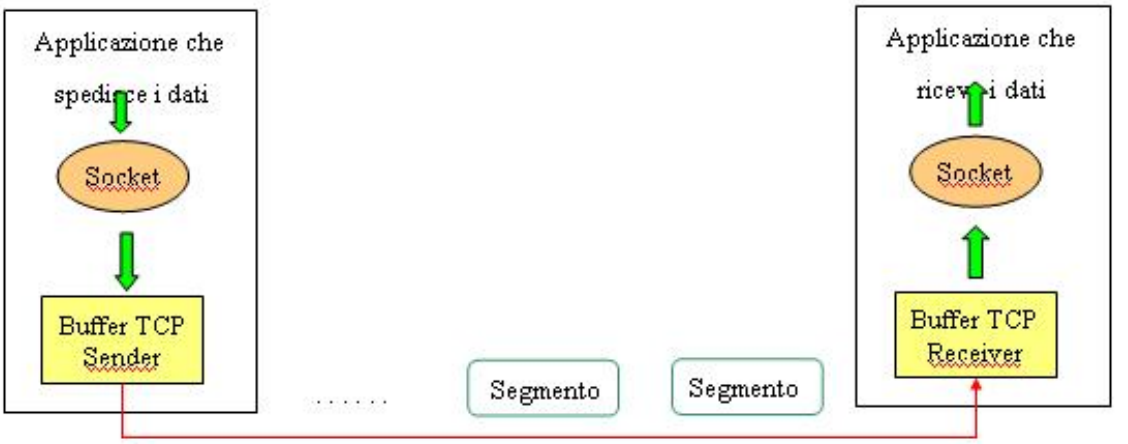
TCP è descritto in dettaglio nelle RFC 793, 1122, 1323, 2018, 2581. Come accennato in precedenza le principali caratteristiche di *TCP* sono le seguenti:

- protocollo punto a punto ovvero si ha un solo mittente ed un solo destinatario *receiver*;
- trasferimento affidabile, ordinato di *byte steam*: non vi sono confini dei messaggi: il livello applicazione dopo aver aperto una connessione *TCP* vede l'invio e l'arrivo di un flusso (*stream*) di *byte*;
- *pipeline*: *TCP* è di tipo *pipeline*. Il numero di segmenti che possono essere spediti senza che vi sia riscontro è modificabile dinamicamente dai meccanismi di controllo del flusso e della congestione;
- *send & receive buffers*: sono presenti dei *buffer* sia nel mittente che nel destinatario;
- comunicazione *full-duplex*: nella stessa connessione si ha un flusso di dati in entrambe le direzioni;
- orientato alla connessione: all'inizio della connessione vengono scambiati dei messaggi di controllo fra mittente e destinatario per inizialiizare lo stato della connessione (*handshaking*);
- controllo del flusso e della congestione.

TCP: spedizione dei dati

I dati sono passati dal livello applicazione al livello trasporto attraverso i *socket*.

Il *socket* (introdotto in *Unix*, 1981) è un'interfaccia locale fra livello applicazione e livello trasporto creata, usata e rilasciata esplicitamente dalle applicazioni e controllata dal sistema operativo. Il *socket* è associato ad un indirizzo *IP* ed ad un numero di porta che permettono di individuare il processo applicativo al quale devono arrrivarare i dati. Processi applicativi possono spedire messaggi a un processo (remoto o locale) scrivendo nel *socket* ricevere messaggi da un processo leggendo dal *socket*.



Comunicazione TCP

Struttura del segmento TCP

Struttura del segmento TCP descritta nella pagina successiva

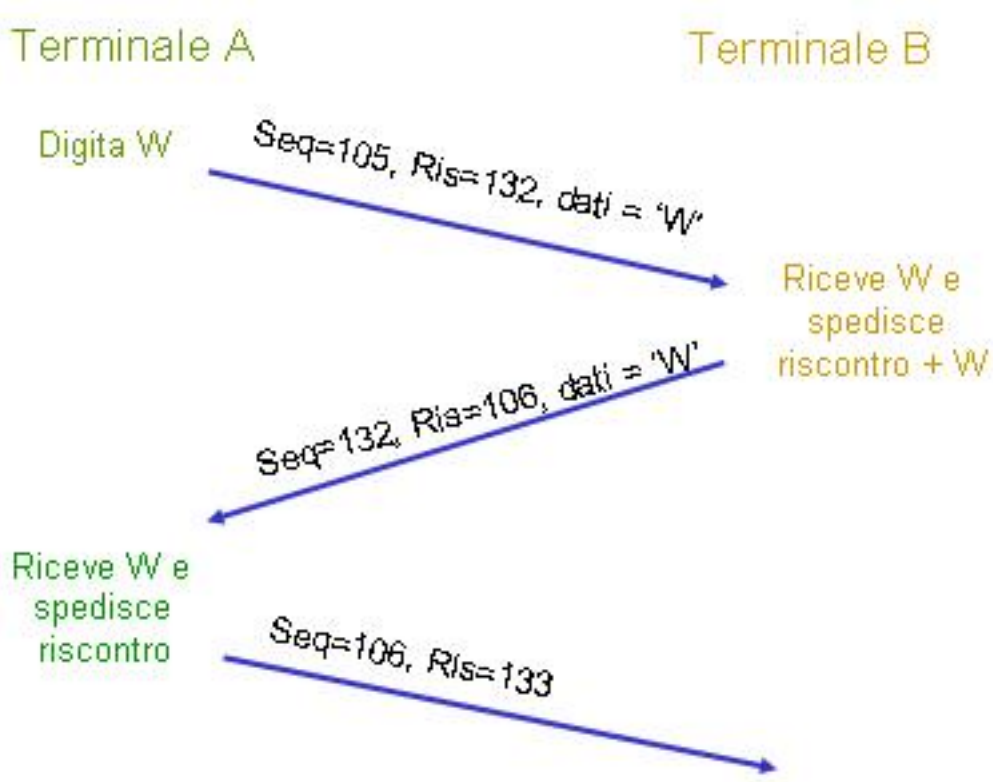
Componenti del segmento TCP

- Porta sorgente e destinazione: numeri di porta sorgente e destinazione per *multiplexing*;
- numero sequenza: numero che identifica ogni segmento e che serve per realizzare il trasferimento affidabile. Il numero indica il numero del primo *byte* nella parte dati del segmento, dove i *byte* sono numerati a partire dall'inizio del flusso di *byte* delle connessione *TCP*. Ad esempio, se nel primo segemento sono spediti 500 *bytes* di dati, numerati da 0 a 499, il numero di sequenza del secondo segmento sarà 500;
- numero riscontro: questo èil numero che indica il segmento per il quale il destinatario ha inviato un riscontro (*ack*): il numero indica il numero del prossimo segmento che il destinatario si aspetta. Ad esempio, se un segmento con numero di sequenza 1000 contenente 450 *bytes* di dati è ricevuto, il destinatario invierà come riscontro un segmento contenente nel

campo riscontro il valore 1451 (il numero del prossimo *byte* che ci aspetta);
- lungh. intestazione: lunghezza del campo intestazione;
- U (URG): *bit* che indica dati urgenti (usato raramente);
- A (ACK): *bit* che indica se si tratta di un segmento che contiene un riscontro (ACK);
- R (RST), S (SYN), F (FIN): *bit* usati nella fase di inizializzazione della connessione;
- dimensione finestra destinatario: campo (aggiornabile dinamicamente dal destinatario) che contiene la dimensione della parte libera del *buffer* destinatario e che è usata per il controllo del flusso. Il mittente controllerà che il numero di *byte* spediti e non ancora riscontrati sia inferiore al valore contenutoin questo campo;
- *checksum*: campo che contiene il valore della *checksum* per il controllo di errore;
- punt. dati urgenti: campo che contiene l'indirizzo degli eventuali dati urgenti (raramente usato).

Un esempio di uso dei numeri di sequenza: una sessione telnet

- Seq= **Numero sequenza** (numero del primo *byte* di dati nel segmento)
- Ris = Numero riscontro: (umero sequenza del prossimo *byte* che ci si aspetta dall'altro lato)
- Nell'esempio in figura, l'utente sul terminale A apre una sessione telnet. Ricordiamo che telnet è un protocollo di livello applicazione per il collegamento remoto che usa *TCP* come protocollo di livello trasporto. Dopo aver aperto la sessione telnet e quindi aver iniziato la connessione *TCP* avviene quanto segue:
  - L'utente sul terminale A digita la lettera W
  - B riceve W, spedisce il riscontro per (Ris = 106) e rispedisce anche W (per l'eco sul terminale)
  - A riceve W e rispedisce il riscontro



Esempio di sessione telnet

Trasferimento affidabile TCP: lato mittente 1

Il seguente pseudo-codice descrive il funzionamento di un *sender* che realizza il trasferimento affidabile di *TCP* (:= indica il comando di assegnamento).

base_send := numero di sequenza iniziale:
prossimo_num_seq := numero di sequenza iniziale
ripeti {
se vengono ricevuti dati dall'applicazione:

- crea segmento con numero di sequenza prossimo_num_seq;
- fai partire il *timer* per il segmento con numero di sequenza prossimo_num_seq;
- passa il segmento al livello rete;
- prossimo_num_seq : = prossimo_num_seq + lunghezza dati spediti;

se si verifica un *timeout* per il segemento con numero di sequenza x:

- ristrametti il segmento con numero di sequenza x;
- calcola il nuovo valore da assegnare al *timer* del segmento x;
- fai ripartire il *timer* per il segemento con numero di sequenza x;

se si riceve un riscontro con valore y nel campo numeri riscontro:
se X> base_send;
cancella tutti i *timer* per i segmenti con numeri di sequenza < Y;
/* riscontro cumulativo*/
base_send : = y;

se *X* <= base_send
incrementa il numero dei ricontri duplicati ricevuti per il segmento Y;
se numero dei ricontri duplicati = 3;
ristrametti il segmento con numero di sequenza x;
fai ripartire il *timer* per il segmento con numero di sequenza x;
}

Trasferimento affidabile TCP: lato mittente 2

Come si può notare nello pseudo codice visto in precedenza in *TCP* si ha:

- **Riscontro** Cumulativo: il riscontro per un certo numero di sequenza vale anche per tutti i numeri di sequenza minori.
- *Timer*: al momento della spedizione di un segmento viene fatto partire un orologio (*timer*); quando tale orologio raggiunge una certa soglia, ovvero quando si verifica un *time*-out, se non si è ricevuto un riscontro per un segmento si assume che il segmento sia andato perduto. Il valore della soglia è modificabile dinamicamente in base a criteri statistici che tengono conto del tempo necessario ad un segmento per raggiungere il destinatario e quindi tornare indietro (detto RTT ovvero *Round Trip Time*).
- Ritrasmissione selettiva: solo un segmento viene rispedito al verificarsi di un *time out* e non tutti i segmenti che ancora non sono stati riscontrati.
- Ritrasmissione veloce: dato che *TCP* non usa riscontri negativi, l'unico modo che ha il destinatario per avvertire il mittente che un segemento non è stato ricevuto è quello di ritrasmettere un riscontro per il precedente segmento ricevuto correttamente. Quando il mittente riceve tre riscontri duplicati per un segmento rispedisce il segmento successivo (anche se il suo *timer* non è ancora arrivato alla soglia). Il mittente si accorge che un segmento non è arrivato ricevendo un segmento con numero di sequenza maggiore di quello che si sarebbe aspettato.

TCP lato destinatario

Il seguente pseudo-codice descrive il funzionamento di un destinatario che realizza il trasferimento affidabile di TCP.

ripeti {
Se arriva un segmento in ordine, non ci sono buchi e tutti gli altri segmenti arrivati sono già stati riscontrati
aspetta fino a 500 millisecondi. Se non arriva nessun altro segmento spedisci un riscontro per il segmento arrivato
Se arriva un segmento in ordine, non ci sono buchi e e c'e un riscontro sospeso
spedisci immediatamente un *ack* cumulativo
Se arriva un segmento non in ordine, con numero di sequenza maggiore di quello atteso (ovvero è rilevato un buco)
spedisci un riscontro duplicato che indichi il prossimo numero di sequenze atteso
Se arriva un segmento che riempie parzialmente o totalmente un buco
spedisci immediatamente un *Ack*

}

Controllo del flusso in TCP

Controllo del flusso: fa sì che il *buffer* del destinatario non venga saturato con i dati dal mittente. Da non confondersi con il controllo di congestione. Viene realizzato dal mittente facendo si che il numero di *byte* trasmessi e non ancora riscontrati sia minore del valore contenuto nel campo dimensione finestra destinatario dell'ultimo semento ricevuto dal mittente (valore inserito dal destinatario). Tale valore indica lo spazio libero nel *buffer* del destinatario ed è inserito nel segmento dal destinatario.

Buffer del destinatario

Gestione della connessione TCP

La connessione *TCP* passa attraverso tre fasi:

- inizializzazione;
- trasferimento dati;
- chiusura.

Nella fase di inizializzazione il mittente ed il destinatario *TCP* si scambiano alcuni segmenti che servono solo per stabilire alcuni valori di controllo quali:

- numero di sequenza;
- dimensioni dei *buffer*;
- dimensioni del campo dimensione finestra destinatario per il controllo del flusso.

Nella fase di trasferimento dati vengono inviati i dati ricevuti dal livello applicazione.

Nella fase di chiusura vengono scambiati dei segmenti per avvertire delle chiusura della connessione il partner.

Fase di apertura della connessione TCP

(handshaking)

La connessione è iniziata da un processo applicativo che chiede al livello trasporto i tabire un collegamento *TCP* con un altro processo applicativo. Il processo che richiede la connessione è detto *client* mentre l'altro è detto *server*. La connessione è dunque iniziata dal *client* ad esempio con la seguente istruzione Java

Socket *client* = *new* ***Socket***(indirizzo IP,numero porta);
Il *server* dovrà essere pronto ad accettare la richiesta di connessione, ad esempio usando la seguente istruzione Java

Socket connessione = Iniziale.accept()
dove Iniziale è un oggetto della classe Server***Socket***

La connessione viene stabilita seguendo i seguenti tre passi:

- Il *client* invia una richiesta di connessione: il *client*:
  - sceglie un numero di sequenza;
  - invia al *server* un segmento (con il numero di sequenza scelto) che non contiene dati e contiene il valore 1 per il *bit* SYN (per questo tale segmento è detto segmento SYN).
- Quando il *server* riceve la richiesta esso:
  - alloca i *buffer* necessari;
  - sceglie un proprio numero di sequenza per il primo segmento che dovrà spedire;
  - spedisce al *client* un segmento (con il numero di sequenza scelto) che contiene un riscontro per il segmento SYN ricevuto.
- Quando il *client* riceve il segmento contenente il riscontro per il segmento SYN:
  - alloca i *buffer* necessari;
  - invia al *server* un segmento con il *bit* SYN a 0 che contiene un riscontro per il segmento ricevuto dal *server*;
  - inserisce nella parte dati i dati del livello applicazione che vuole spedire.

La connessione a questo punto è stabilita.

Chiusura della connessione TCP

La connessione può essere chiusa sia dal *client* che dal *server*. Ad esempio, il *client* la può chiudere facendo

client.close();
dove *client* è il *socket* visto in precedenza. Al verificarsi di una tale richiesta di chiusura della connessione:

- Il *client* manda un segmento che contiene il *bit* FIN posto a 1 (segmento FIN);
- Il *server* riceve il segmento FIN e risponde con segmento che contiene il riscontro per tale segmento FIN;

- Il *server* invia al *client* un altro segmento che contiene il *bit* FIN a 1;
- Il *client* riceve questo segmento FIN dal *server* e invia un segmento di 1 di riscontro per esso;
- Il *client* entra quindi in una fase di attesa (della durata tipica di 30 secondi) durante la quale se necessario può rispedire l'ultimo segmento di riscontro.

Terminata la fase di attesa la connessione è chiusa e tutte le risorse necessarie per la comunicazione sono de-allocate.

Principi di controllo della congestione

Congestione: informalmente, si ha congestione in una rete quando ci sono troppi mittenti che spediscono più dati di quanti la rete riesca a gestirne. Gli effetti della congestione vanno dai lunghi ritardi di arrivo dei dati alla perdita di pacchetti. Difatti i *router* e i nodi intermedi della rete contengono dei *buffer* sia di ingresso che di uscita nei quali sono accodati i pacchetti in attesa di essere smistati sui collegamenti opportuni. Quando il numero di tali pacchetti aumenta oltre le capacità di gestione del *router* le *code* nei *buffer* si allungano con conseguente aumento dei ritardi e, quando si oltrepassa la capacità dei *buffer*, perdita di pacchetti.

Controllo della congestione: è uno dei problemi più rilevanti nello studio delle reti attuale. Vi sono due tecniche principali:

- Controllo assistito dalla rete: i *router* ed i nodi intermedi forniscono informazioni ai terminali sullo stato di congestione della rete e possono anche imporre ai terminali di limitare la velocità di spedizione. Questo tipo di controllo è usato nell reti ATM.
- Controllo senza assistenza della rete: in questo caso lo stato di congestione della rete è osservato dai terminali in base ai ritardi e all'eventuale perdita dei pacchetti. Non c'è alcuna informazione da parte dei *router*. Questo è l'approccio usato da *TCP*.

Controllo della congestione in TCP

In *TCP* in controllo della congestione avviene come segue: il mittente mantiene in una opportuna variabile (detta finestra di congestione) il numero di segmenti massimo che possono essere spediti senza riscontro. Il valore della finestra di congestione è aumentato dinamicamente dal mittente fino a che non si verifica una perdita di un segmento; quando si verifica una perdita si modifica il valore della finestra riportandolo ad un valore prefissato (più basso) e quindi si inizia di nuovo a spedire segmenti aumentando il valore della finestra.

Idealmente quindi *TCP* cerca di sfruttare al massimo la banda disponibile.

Diverse implementazioni di *TCP* possono usare varianti dell'algoritmo di controllo della congestione qui accennato e descritto più in dettaglio nella pagina seguente (tale algoritmo è di solito detto *Tahoe*).

Controllo della congestione in TCP: algoritmo Tahoe

Questo algoritmo usa due variabili:

- FinCong = finestra di congestione, contiene il numero massimo di segmenti che possono essere spediti senza riscontro;
- Soglia = contiene un valore che varia dinamicamente che modifica il modo in cui cresce FinCong.

e consiste delle seguenti fasi:

- Partenza Lenta: FinCong viene inzializzata a 1. Fino a quando FinCong è minore di Soglia vengono ripetute le seguenti operazioni:
  - spedisci un numero di segmenti pari a FinCong; aspetta il riscontro (cumulativo) per tali segmenti; raddoppia il valore di FinCong.
- Annullamento della congestione: quando FinCong diventa maggiore di Soglia viene ripetuto il seguente ciclo fino al momento in cui si verifica una perdita di un segmento:
  - spedisci un numero di segmenti pari a FinCong; aspetta il riscontro (cumulativo) per tali segmenti; aumenta di 1 il valore di FinCong.
- Perdita: al verificarsi di una perdita (segnalata da un *timeout* per un segmento) vengono eseguite le seguenti operazioni:
  - Il valore di Soglia diventa quello quello di FinCong/2; si ritorna alla fase di partenza lenta.

L'idea dell'algoritmo è quindi quella di aumentare esponenzialmente il numero di segmenti che vengono spediti (partendo da 1) senza aspettare il riscontro fino ad arrivare al valore della soglia; da tale valore in poi tale numero viene aumentato di 1 e infine, quando si verifica una perdita, si ritorna al valore 1, ponendo allo stesso tempo come nuovo valore di soglia quello dell'ultimo numero di segmenti spediti (ovvero quello contenuto in FinCong) diviso 2.

Fairness di TCP

Il meccanismo di controllo della congestione visto con l'algoritmo *Tahoe* è detto AIMD (aumento addittivo, decremento moltiplicativo) dato che, a parte la fase dalla partenza lenta, il numero di segmenti che possono essere spediti aumenta di 1 ad ogni passo (ovvero ad ogni RTT) mentre si dimezza ad ogni passo quando la rete è congestionata.

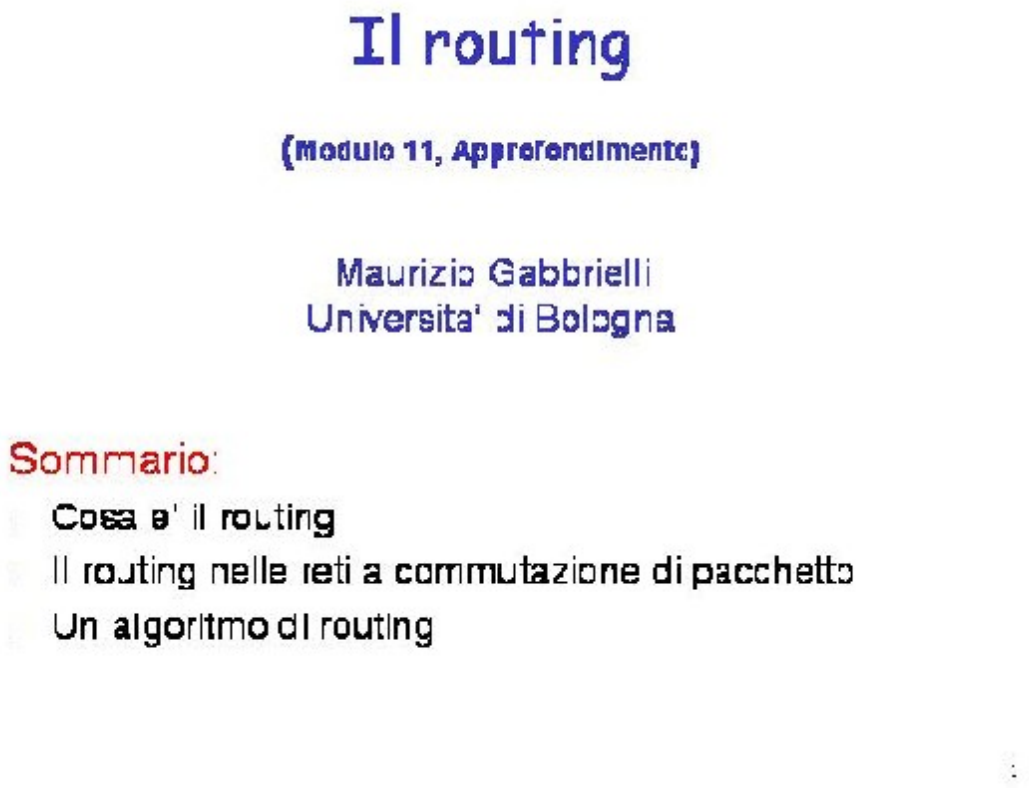
Questo meccanismo AIMD fa si che *TCP* sia *fair* (equo), nel senso che se ci sono più connessioni attive contemporaneamente su uno stesso canale, dopo un fase iniziale a regime ogni connessione userà la stessa percentuale della banda disponibile. Intuitivamente questo avviene perchè la connessione che usa più banda al momento del decremento (moltiplicativo) viene penalizzata più di quella che usa meno banda, mentre l'aumento (addittivo) è lo stesso per entrambe.

Routing

Simone Martini

11.4 (Routing)

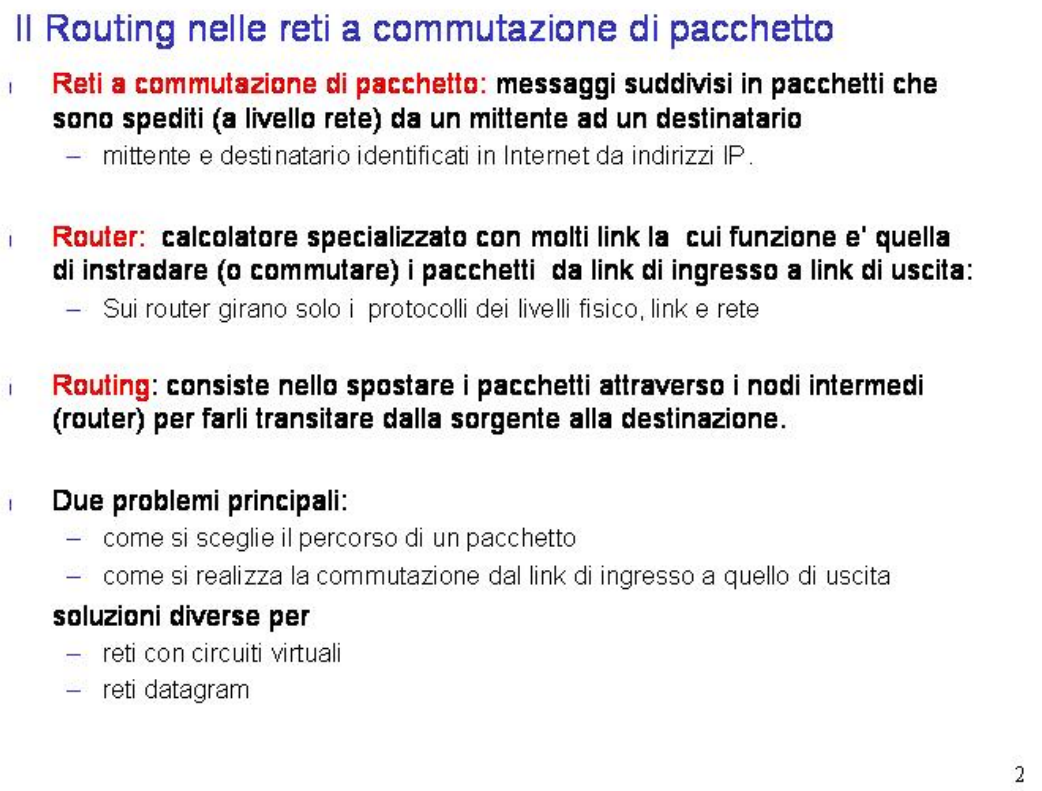
Sommario



Sommario

Bungiorno, il mio nome è Maurizio Gabbrielli, sono dell'Università di Bologna ed in questa lezione parleremo del *routing*, quindi vedremo un approfondimento del modulo 11, nel contesto del quale sono state già viste altre caratteristiche delle reti di calcolatori e di *Internet* in particolare. I punti che affronteremo, come è esposto qui nel sommario sono: cosa si intende per *routing*, come avviene il *routing* nelle reti a commutazione di pacchetto, infine vedremo un algoritmo di *routing* specifico, che è quello di *Dijsktra*, usato in alcuni contesti.

Il routing nelle reti a commutazione di pacchetto 1



Il routing nelle reti a commutazione di pacchetto

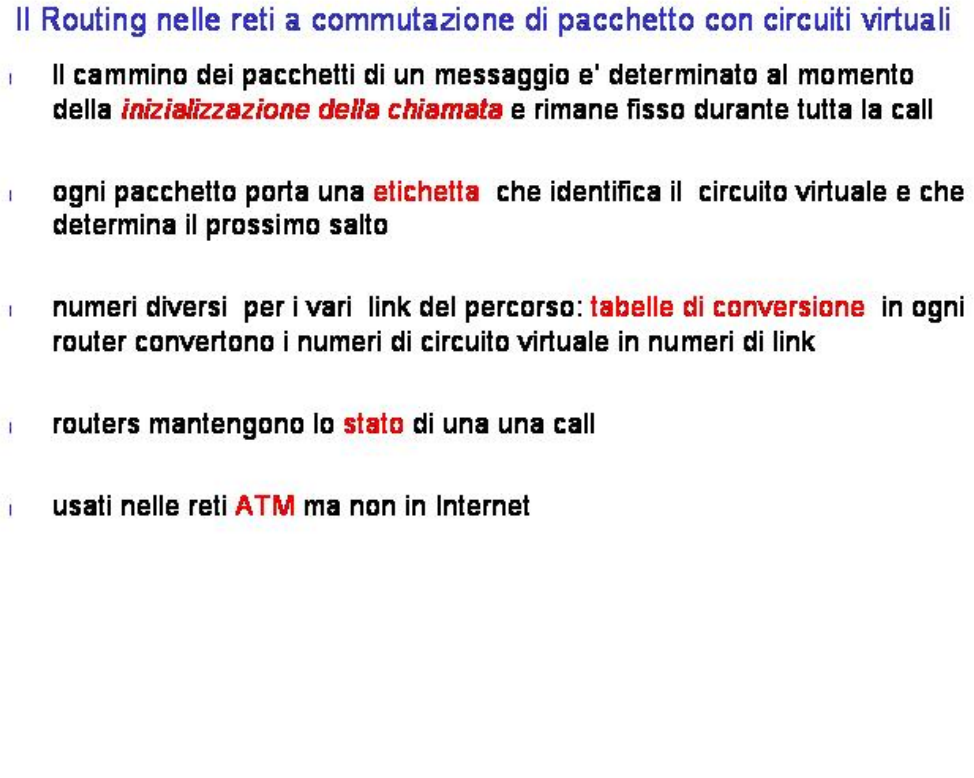
Iniziamo ricordando cosa si intende per reti a commutazione di pacchetto. In tale tipologia di rete i messaggi sono appunto suddivisi in pacchetti, tutti di una certa dimensione prefissata, che vengono spediti a livello di rete dal mittente al destinatario, identificati entrambi da un *IP-address* (indirizzo IP), cioè un indirizzo che consiste di 32 bit divisi in quattro gruppi da 8 bit ciascuno. Quindi, rispetto alle reti a commutazione di circuito, la differenza sostanziale è la mancanza di un uso esclusivo delle risorse offerte dalle reti, nel senso che le risorse di rete sono condivise fra più comunicazioni diverse, che possono quindi spedire i pacchetti usando gli stessi canali, gli stessi *router*, con un meccanismo di condivisione delle risorse non presente nelle reti a commutazione di circuito, dove invece una volta che sia stata stabilita la comunicazione, la risorsa per quel canale di quel circuito è assegnata in uso esclusivo ai due *partner* della comunicazione e viene mantenuta fino a che la comunicazione stessa non cessi. Nelle reti a commutazione di pacchetto tutto questo non avviene, se si spediscono vari pacchetti che devono transitare sulla rete, esistono i cosiddetti *router*, cioè dei calcolatori specializzati la cui funzione è quella di instradare, commutare i pacchetti.

Il routing nelle reti a commutazione di pacchetto 2

Il routing nelle reti a commutazione di pacchetto

Questi *router* avranno appunto un certo numero di link in entrata e un certo numero di link in uscita e lo scopo del loro funzionamento è quello di prelevare un pacchetto da un link di ingresso e di instradarlo su un opportuno link di uscita. Va ricordato il fatto che sui *router* sono implementati soltanto i primi tre livelli a partire dal basso della gerarchia ISO-OSI per quanto riguarda le reti, cioè i protocolli del livello fisico, del livello link e del livello di rete, mentre non sono implementati i protocolli del livello di trasporto e di applicazione. Il *routing* consiste nell'attività specifica di un *router*, cioè nell'instradare pacchetti, in modo tale da farli arrivare dalla sorgente, cioè dal mittente, alla destinazione, facendoli transitare attraverso i nodi intermedi che costituiscono la rete. Nella realizzazione di questo meccanismo di instradamento ci sono due problemi principali che devono essere affrontati: il primo è la scelta del percorso del pacchetto, cioè come decidere dinamicamente il cammino che il pacchetto deve seguire per arrivare dalla sorgente alla destinazione; il secondo problema è come realizzare la commutazione dal link di ingresso a quello di uscita, cioè come far transitare un pacchetto in uno specifico *router* da un link di ingresso ad un link di uscita. Vedremo che ci sono soluzioni diverse a seconda del tipo di reti a commutazione di pacchetto considerato, cioè reti con circuiti virtuali oppure reti di tipo *datagram*.

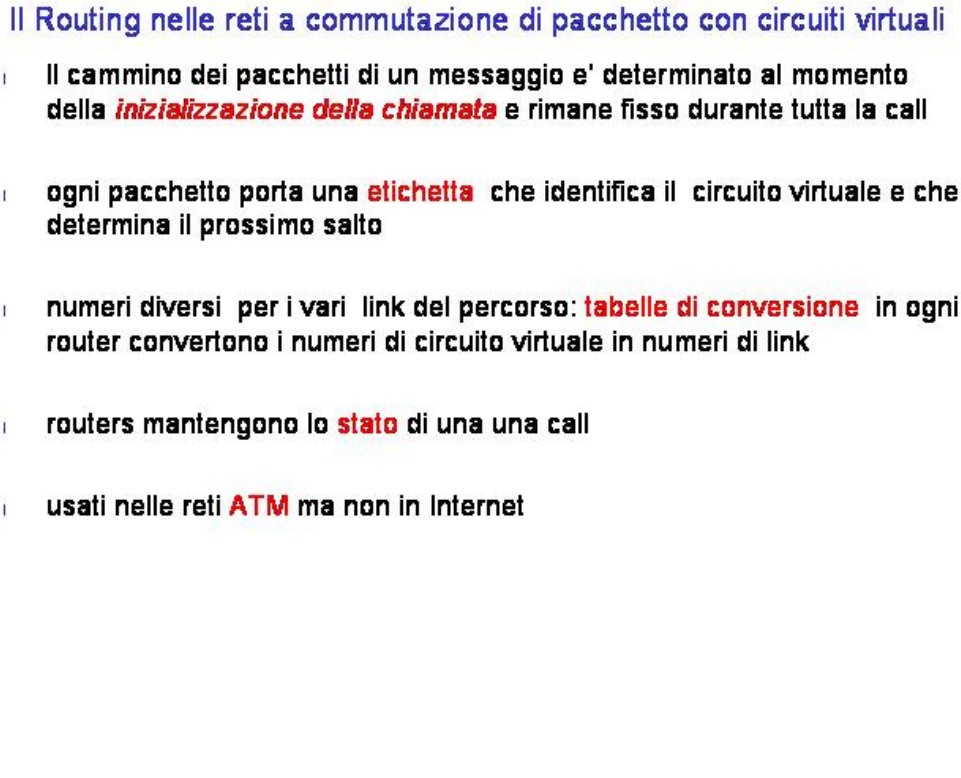
Il routing nelle reti a commutazione di pacchetto con circuiti virtuali 1



Il routing nelle reti a commutazione di pacchetto con circuiti virtuali

Nelle reti a commutazione di pacchetto con circuiti virtuali la situazione è la seguente: prima che venga iniziato il trasferimento effettivo dei dati, cioè la spedizione dei messaggi tramite pacchetti, c'è una fase di inizializzazione della chiamata, nella quale appunto i vari *router* che si trovano nel cammino tra la sorgente e il destinatario, si "mettono d'accordo" e stabiliscono una sorta di circuito virtuale, che sarà poi usato dal mittente e dal destinatario per spedire i pacchetti. Quindi in qualche modo queste reti ricordano un po' le reti a commutazione di circuito, la differenza sta nel fatto che, come è stato detto in precedenza, l'uso delle risorse non è esclusivo, cioè anche se viene stabilito un circuito virtuale tra mittente e destinatario, questo non vuol dire che essi abbiano in uso esclusivo le risorse che compongono questo circuito virtuale.

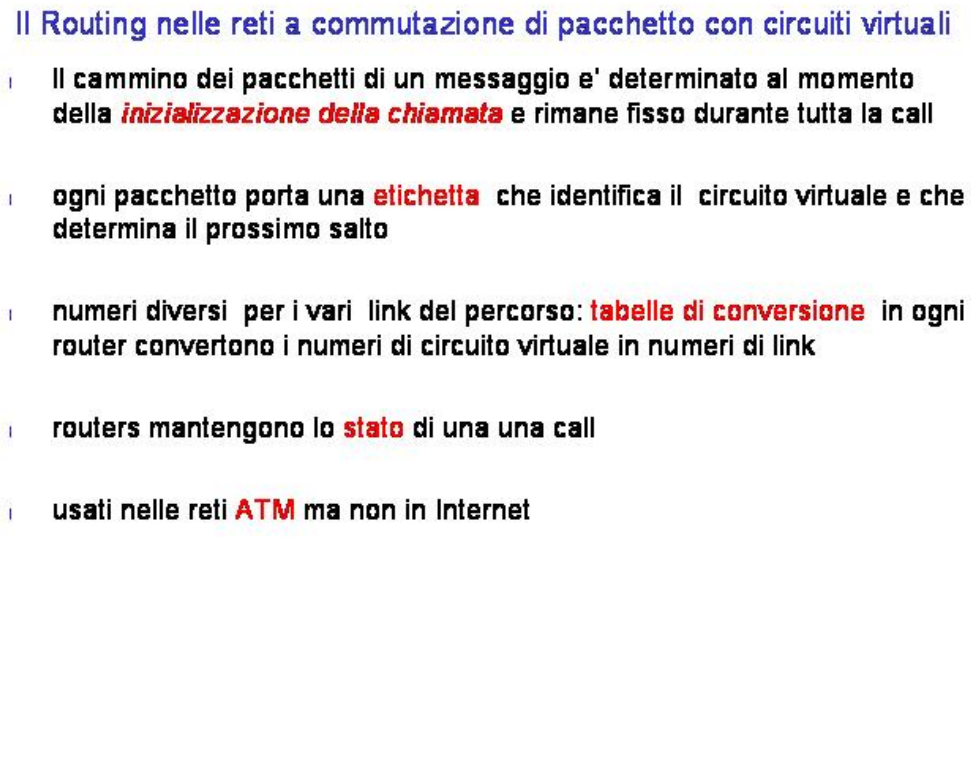
Il routing nelle reti a commutazione di pacchetto con circuiti virtuali 2



Il routing nelle reti a commutazione di pacchetto con circuiti virtuali

Tornando alla rete a commutazione di pacchetto con circuiti virtuali dopo questa prima fase di inizializzazione della chiamata, viene quindi stabilito il circuito virtuale, cioè il percorso che devono seguire i pacchetti, il quale è identificato a livello di pacchetti tramite un'etichetta contenuta in ogni pacchetto. Nella realtà la situazione è un po' più complicata, nel senso che questi codici che identificano un circuito virtuale non sono gli stessi su tutti i pezzi del cammino che costituiscono l'intero circuito virtuale, cioè in punti diversi del circuito, esso viene identificato da numeri diversi e per tale motivo servono le tabelle di compressione di *router*, che associano al numero di circuito virtuale in arrivo il numero di circuito virtuale in uscita. A livello concettuale possiamo rimanere con l'idea che ogni circuito virtuale venga identificato da una certa etichetta inserita all'interno del pacchetto per indicare quale circuito virtuale deve essere seguito.

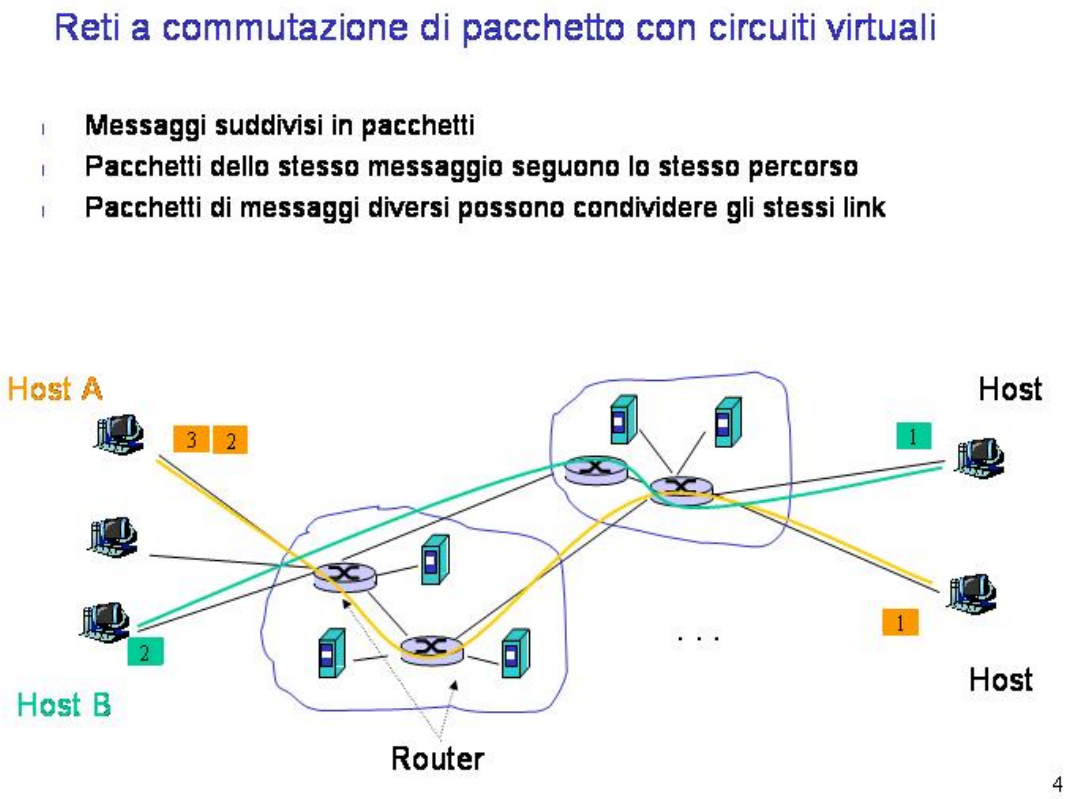
Il routing nelle reti a commutazione di pacchetto con circuiti virtuali 3



Il routing nelle reti a commutazione di pacchetto con circuiti virtuali

Da quanto detto appare abbastanza chiaro il fatto che i *router* mantengano lo stato di una chiamata, cioè sanno quanti e quali sono i circuiti virtuali che li attraversano, quindi ad ogni istante hanno un controllo sulla situazione della trasmissione dei dati per quanto riguarda i circuiti virtuali che attraversano quello specifico *router*. Le reti a commutazione di pacchetto con circuiti virtuali sono usate nelle reti ATM, che sono reti di origine telefonica, non molto usate attualmente per la trasmissione di dati; tali circuiti virtuali non sono invece usati in *Internet*, dove invece sono preferite le reti di tipo *datagram* come vedremo in seguito.

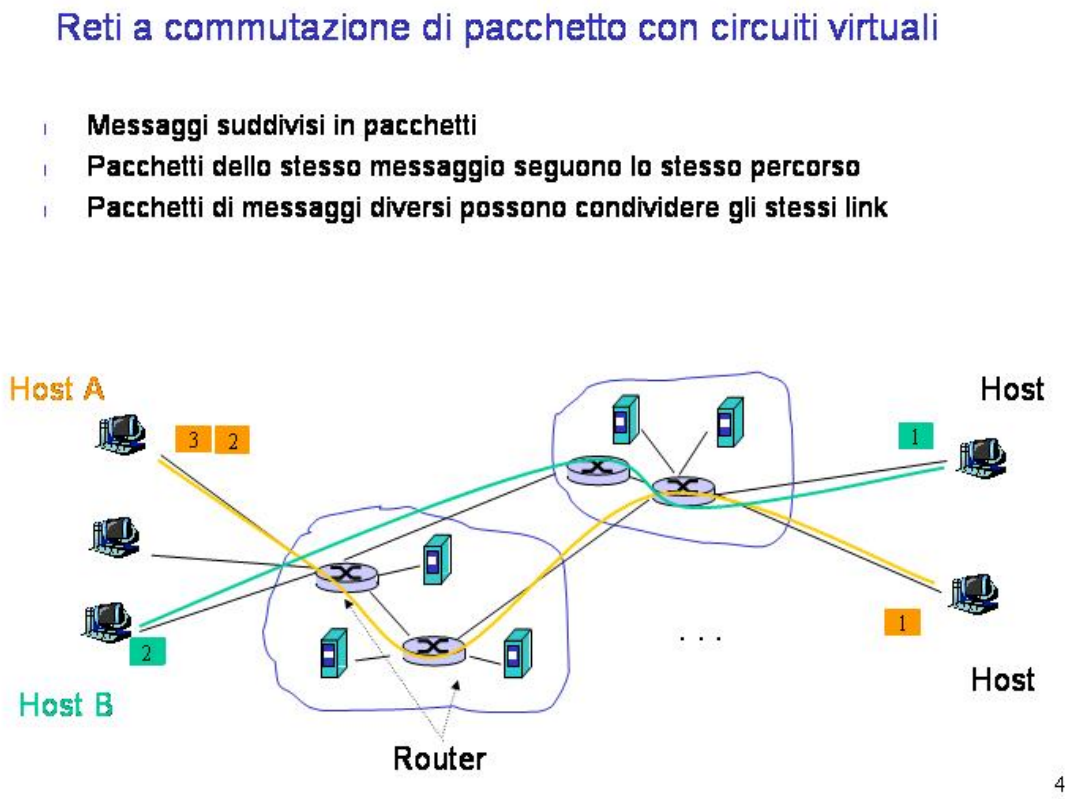
Reti a commutazione di pacchetto con circuiti virtuali 1



Reti a commutazione di pacchetto con circuiti virtuali

Per ricapitolare quanto detto sui circuiti virtuali, possiamo vedere questa visualizzazione di tipo grafico: abbiamo questa prima situazione con due *host* A e B che vogliono inviare dei pacchetti a degli altri *host* che si troveranno dall'altra parte della rete, anche a distanze geografiche notevoli. Tra i mittenti e i destinatari esiste tutta la struttura della rete costituita dai vari *router*. In questa rete a commutazione di pacchetto con circuiti virtuali se l'*host* A vuole inviare dei dati a questo altro *host* che si trova nella parte destra in basso, prima di tutto stabilisce un circuito virtuale, rappresentato in figura da una linea arancione, dopodiché i pacchetti vengono spediti lungo questo circuito virtuale, quindi i pacchetti 1,2 e 3 dell'*host* A usano esclusivamente il cammino che è stato definito al momento dell'inizializzazione del circuito virtuale.

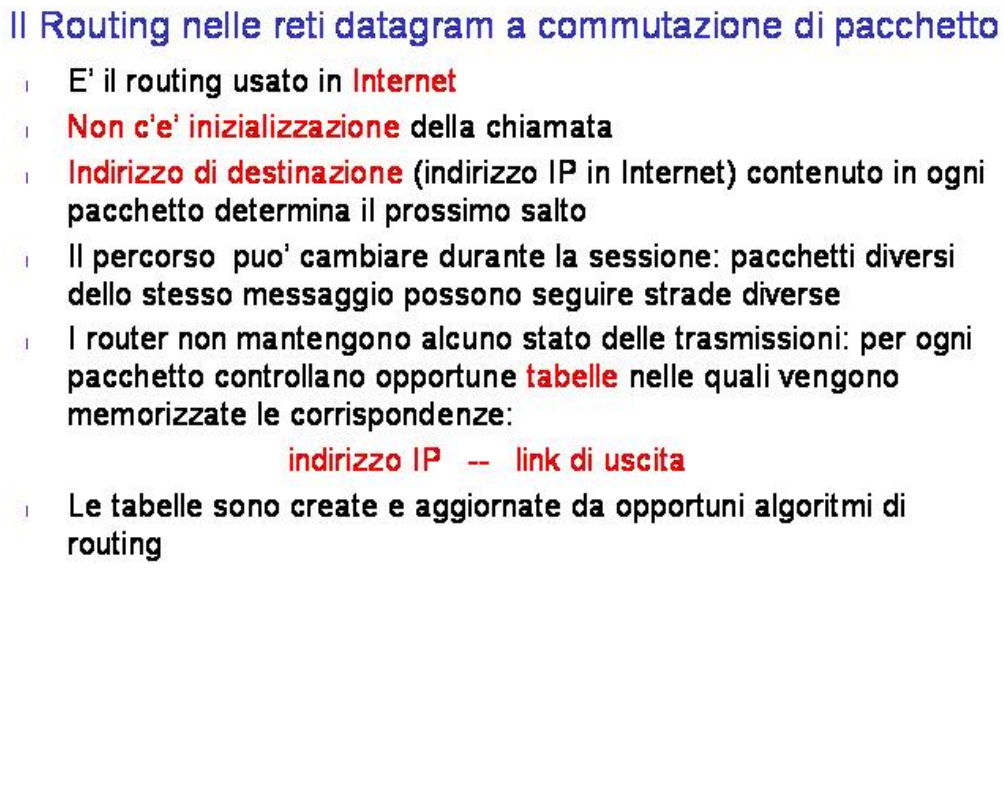
Reti a commutazione di pacchetto con circuiti virtuali 2



Reti a commutazione di pacchetto con circuiti virtuali

Analogamente l'*host* B che vorrà inviare dei pacchetti ad un altro *host*, nella figura lo troviamo a destra in alto, prima di inviare i pacchetti stabilirà un circuito virtuale, dopodiché i pacchetti transitano dal mittente al destinatario, verrà inviato prima il pacchetto 1 dell'*host* A, poi il pacchetto 1 dell'*host* B e così via, con un ordine casuale nell'alternarsi tra pacchetti dell'*host* A e dell'*host* B. È quindi importante ricordare che i messaggi sono suddivisi in pacchetti, se questi ultimi appartengono allo stesso messaggio, in queste reti a circuiti virtuali seguiranno lo stesso percorso e infine pacchetti di messaggi diversi possono condividere gli stessi collegamenti, dato che le risorse sono condivise tra le varie comunicazioni.

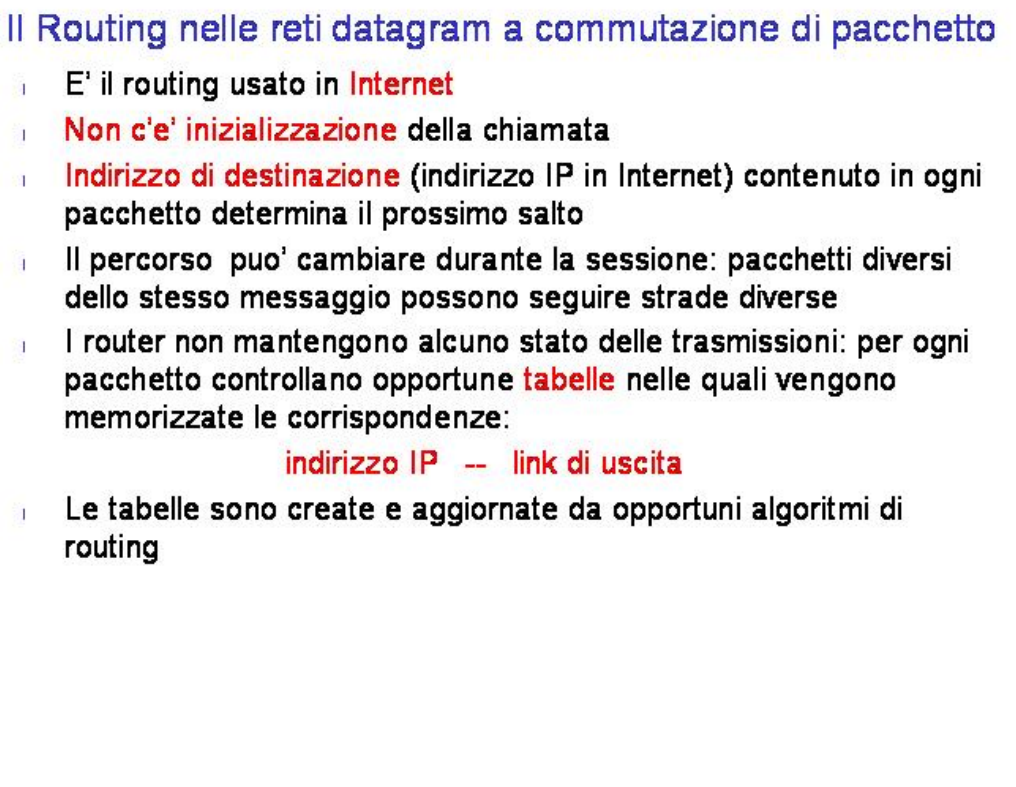
Il routing nelle reti datagram a commutazione di pacchetto 1



Il routing nelle reti datagram a commutazione di pacchetto

Per quanto riguarda invece il *routing* nelle reti a commutazione di pacchetto di tipo *datagram*, che è poi quello usato in *Internet*, diciamo innanzitutto che non esiste alcuna inizializzazione della chiamata, cioè non viene stabilito un circuito virtuale come abbiamo visto in precedenza. In questo caso infatti ogni pacchetto che viene spedito contiene l'indirizzo di destinazione, sottoforma di indirizzo IP. Quindi ogni destinatario spedisce un pacchetto al prossimo *router* che si trova sulla rete e il percorso che il pacchetto è fatto in passi successivi, nel senso che il pacchetto viene spedito dal mittente al *router* più vicino, questo ultimo analizza l'indirizzo di destinazione del pacchetto e in base a questo indirizzo decide a quale altro *router* successivo inviare il pacchetto. Nel *router* successivo si procede allo stesso modo, cioè viene analizzato nuovamente l'indirizzo IP di destinazione e deciso il nodo seguente e così via fino a che il pacchetto non arriva a destinazione. Parlando informalmente tale processo è analogo al tragitto compiuto da un'automobile per andare da una città ad un'altra e il conducente si ferma durante il viaggio a chiedere informazioni sulla strada da seguire.

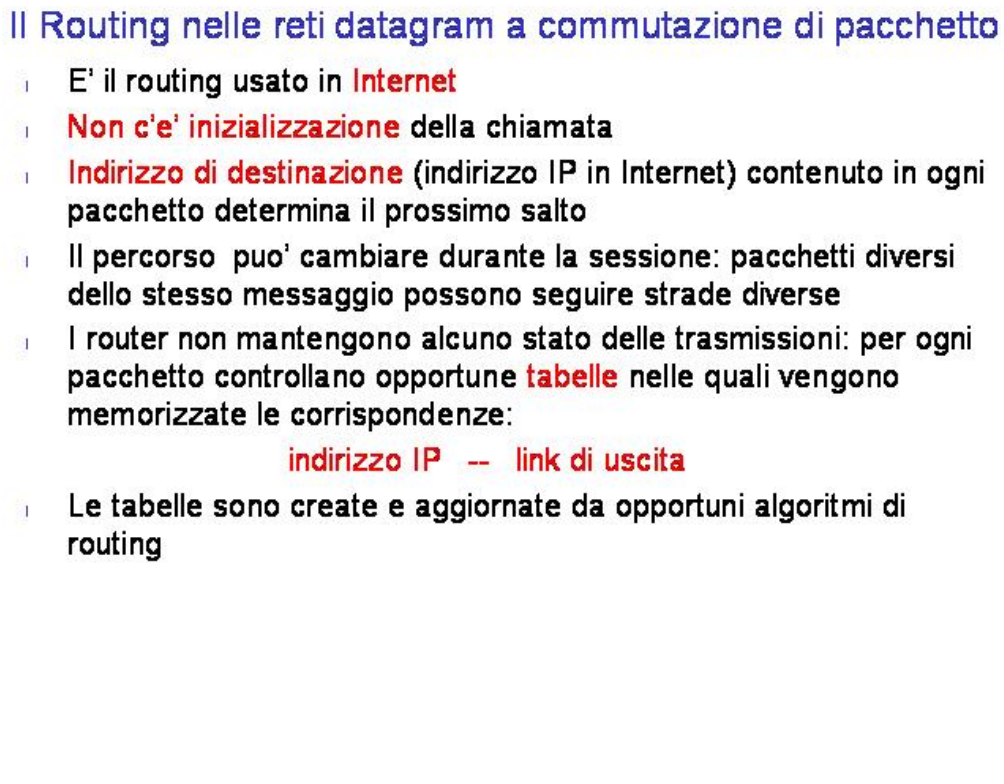
Il routing nelle reti datagram a commutazione di pacchetto 2



Il routing nelle reti datagram a commutazione di pacchetto

Quindi da quanto detto appare chiaro che i pacchetti che sono spediti ad un destinatario e che costituiscono lo stesso messaggio in generale possono non seguire tutti la stessa strada, lo stesso cammino per arrivare a destinazione, cioè possono seguire cammini diversi a seconda dello stato della rete, quindi in generale il percorso può cambiare durante la sessione di trasmissione dei dati. Inoltre, sempre per quanto abbiamo detto sul meccanismo di trasmissione di pacchetti con *datagram*, appare evidente che i *router* non mantengono sulle trasmissioni in corso, nel senso che uno specifico *router* non sa quante e quali sono le trasmissioni che in quel momento lo stanno usando. Il *router* vede soltanto un insieme di pacchetti che gli arriva nei link di ingresso e che lui deve in qualche modo smistare, instradare nei link di uscita, senza sapere se questi pacchetti appartengono o meno alla stessa trasmissione. Il *router*, per poter instradare i pacchetti mantiene delle opportune tabelle nelle quali vengono memorizzate delle corrispondenze tra indirizzo IP, inteso come indirizzo di destinazione del pacchetto e link di uscita; quindi il lavoro del *router* consiste per ogni pacchetto che gli arriva in ingresso, controllare l'indirizzo IP di tale pacchetto, andare nella tabella che mantiene queste corrispondenze, vedere quale link di uscita corrisponde per quello specifico indirizzo IP e instradare il pacchetto su quello specifico link di uscita.

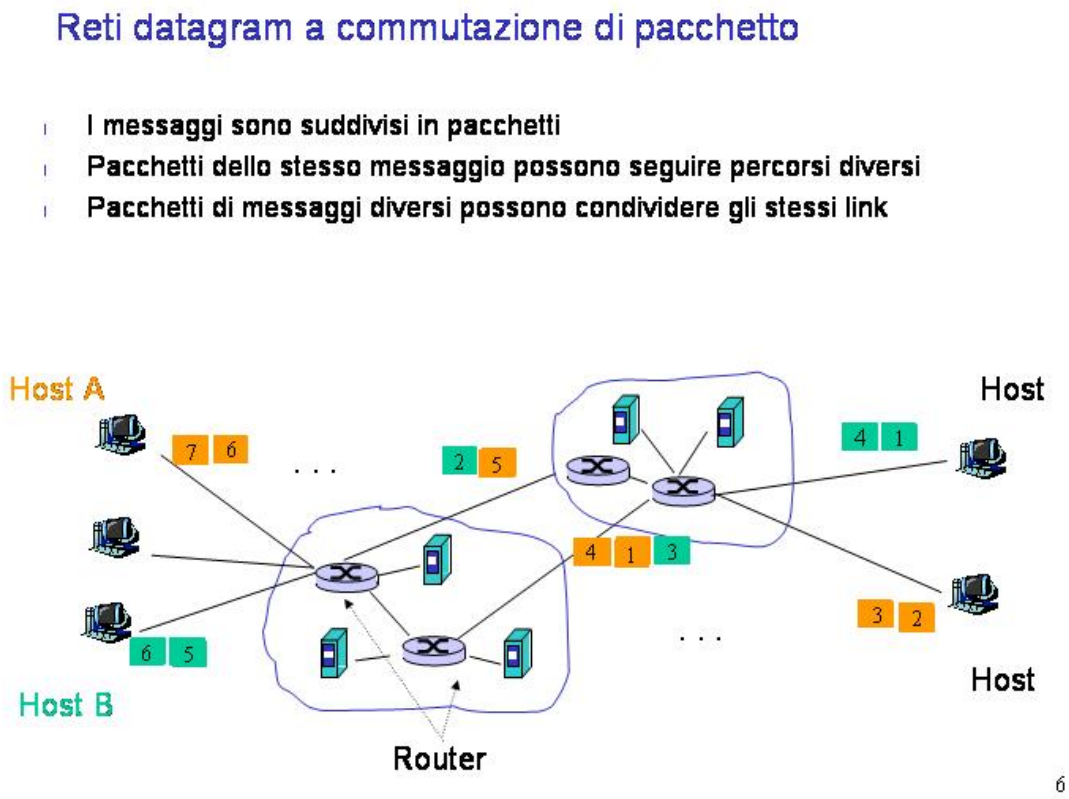
Il routing nelle reti datagram a commutazione di pacchetto 3



Il routing nelle reti datagram a commutazione di pacchetto

Quindi il problema essenziale che si pone alla realizzazione di *router* nelle reti a commutazione di pacchetto di tipo *datagram* è la gestione di queste tabelle, sia nel senso di come realizzare l'accesso alle tabelle in modo efficiente, sia soprattutto decidere come creare e aggiornare queste tabelle, cioè decidere come definire i cammini tra la sorgente e la destinazione e quindi instradare i pacchetti nella rete. Questo avviene usando degli opportuni algoritmi di *routing* dei quali ne vedremo in seguito uno in particolare che è quello di *Dijsktra*.

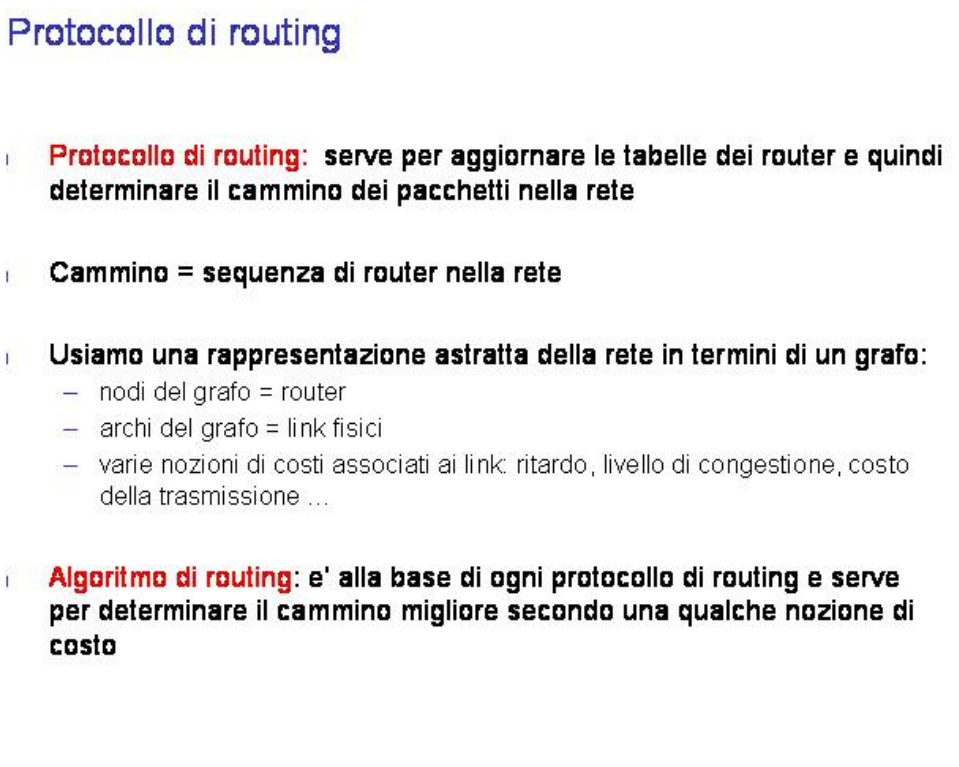
Reti datagram a commutazione di pacchetto



Reti datagram a commutazione di pacchetto

Anche qui abbiamo una figura che riassume quello che abbiamo detto finora relativamente alle reti a commutazione di pacchetto di tipo *datagram*. Ciò che vedete è una situazione analoga a prima nella quale abbiamo due *host* A e B che vogliono spedire dei dati a due *host* che si trovano nella parte destra della figura. Come nel caso precedente i dati sono suddivisi in pacchetti, però ora i pacchetti che costituiscono un singolo messaggio possono seguire percorsi diversi. Quindi vedete ad esempio che i pacchetti 2 e 3 spediti dall'*host* A sono già arrivati, i pacchetti 1 e 4 seguono invece quest'altra strada e passano attraverso questo collegamento da questo *router* all'altro in alto, mentre il pacchetto 5 sempre dell'*host* A segue un percorso diverso. Vedete, come è esemplificato nella figura, i pacchetti 2 e 3 arrivano prima del pacchetto 1, questo vuol dire che in generale non c'è alcuna garanzia del fatto che venga mantenuto correttamente l'ordine di spedizione dei pacchetti, vale a dire che io spedisco i pacchetti nell'ordine 1, 2, 3, 4 quello che accade è che arrivano prima i pacchetti 2, 3 poi l'1 e infine il 4. Più in particolare, non c'è alcuna garanzia che i pacchetti arrivino effettivamente a destinazione, nel senso che i pacchetti si possono perdere nella rete. Se vogliamo quindi essere sicuri che i pacchetti arrivino effettivamente a destinazione nell'ordine corretto, dobbiamo implementare degli opportuni meccanismi e questo viene fatto in genere a livello superiore, a livello di trasporto, ad esempio dal protocollo TCP che garantisce la trasmissione affidabile dei dati. L'ultima cosa da notare in questa figura è il fatto che pacchetti originati da messaggi diversi, di colore arancione e verde in figura, possono condividere gli stessi collegamenti.

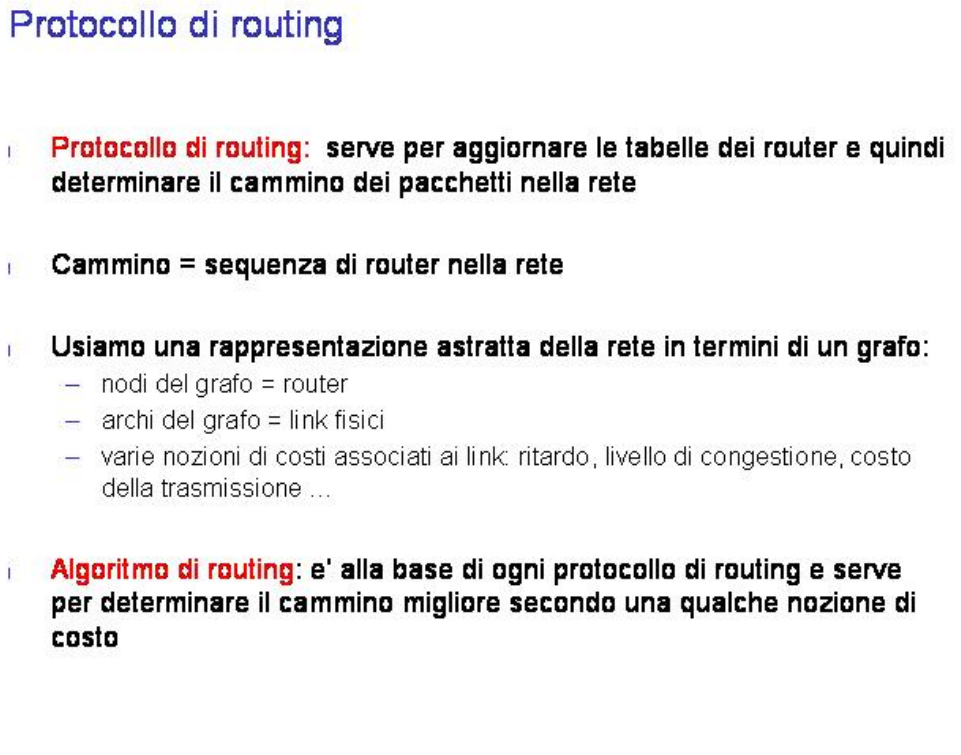
Protocollo di routing 1



Protocollo di routing

Parlando in generale del *routing*, abbiamo capito a questo punto che servono opportuni protocolli di *routing*, che i *router* usano per scambiarsi informazioni in modo tale da poter aggiornare le tabelle delle quali abbiamo parlato prima, in modo tale da poter determinare il cammino che i pacchetti devono seguire nella rete, dove per cammino intendiamo la sequenza di *router* che devono essere incontrati nel percorso dalla sorgente alla destinazione. Possiamo dal punto di vista concettuale identificare una rete di calcolatori con un grafo, dove un grafo è appunto un insieme di nodi alcuni dei quali sono collegati con degli archi, nel quale i nodi sono appunto i terminali ed i *router* mentre gli archi del grafo rappresentano i link, i collegamenti fisici, tra i nodi stessi.

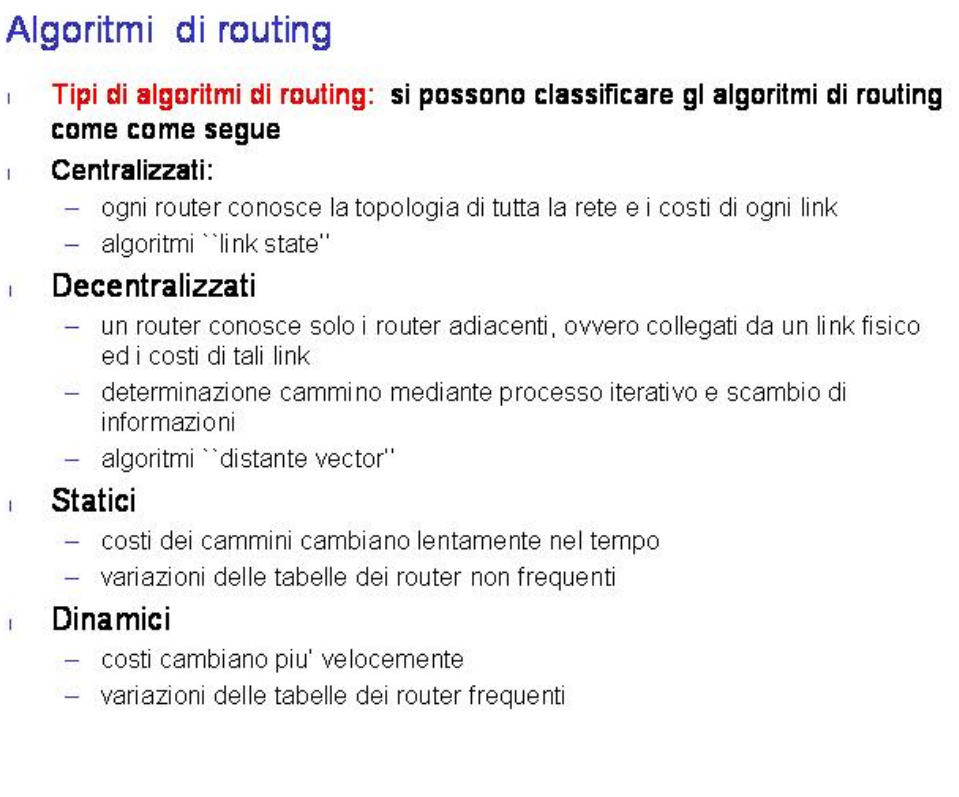
Protocollo di routing 2



Protocollo di routing

Associati a questi archi possiamo avere varie nozioni di costo, nel senso che possiamo essere interessati a sapere qual è il ritardo associato ad un certo link, qual è il costo in termini economici della trasmissione su quello specifico link, possiamo pensare che il costo ci indichi anche qual è il livello di congestione di quel link e altre informazioni di questo genere. Quindi possiamo avere varie nozioni di costo ed avremo più in generale un algoritmo che decide sostanzialmente il cammino su questo grafo, sceglie un cammino all'interno del grafo e solitamente la scelta avviene tenendo conto dei costi associati a tale percorso. Ad esempio un criterio, che è poi quello seguito nella pratica da moltissimi algoritmi di *routing*, è quello di scegliere cammini di costo minimo in base alla funzione di costo scelta: se ad esempio siamo interessati a trasmettere pacchetti nel più breve tempo possibile, evidentemente sceglieremo la funzione di costo che ci indica il ritardo associato ai vari link e poi useremo un algoritmo opportuno per scegliere il cammino che ha il costo minimo.

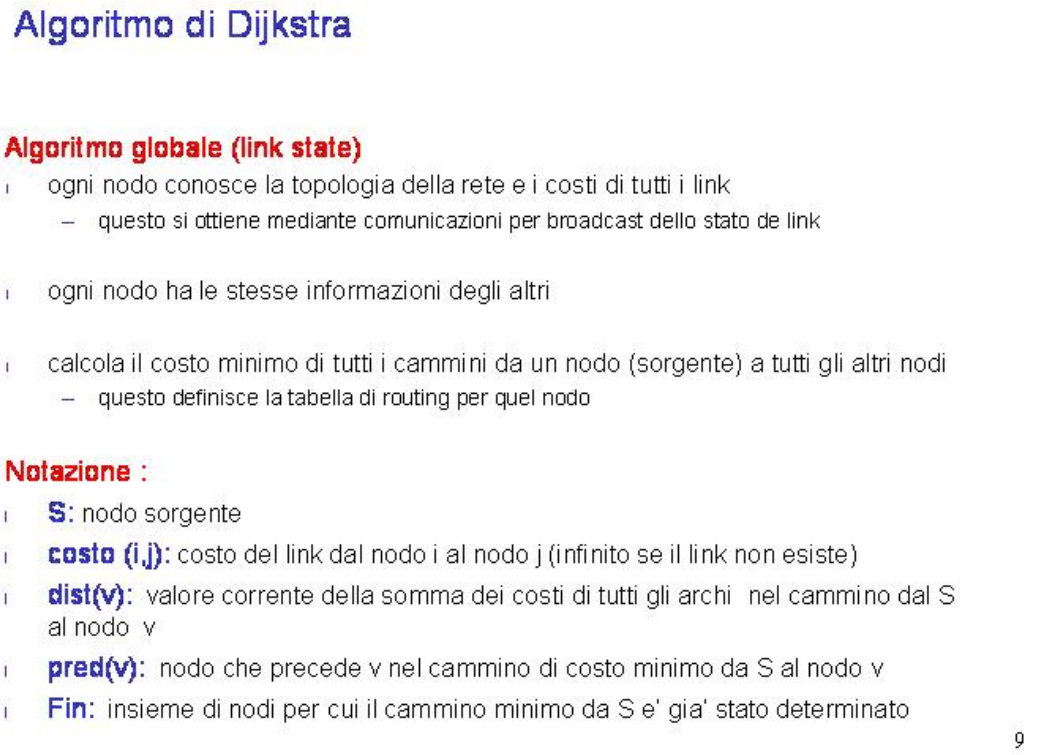
Algoritmi di routing



Algoritmi di routing

Esistono quindi vari algoritmi di *routing* che si usano per selezionare i cammini nelle reti a commutazione di pacchetto e quindi per definire le tabelle di instradamento e tra i vari algoritmi di *routing* possiamo distinguere sostanzialmente due tipi: gli algoritmi centralizzati e quelli decentralizzati. Gli algoritmi centralizzati sono quelli nei quali ogni nodo della rete, ogni *router* conosce la topologia di tutta la rete e i costi di tutti i link, quindi ogni nodo ha una visione completa di tutta la rete; questi algoritmi sono detti di solito *link state*. Negli algoritmi decentralizzati invece ogni *router* ha soltanto una visione locale della rete, ad esempio conosce soltanto i *router* adiacenti e i costi dei link che lo collegano ad essi. In tal caso quindi il singolo *router* non ha la visione di tutta la rete, per poter determinare quale sia il cammino per arrivare ad una certa estremità della rete, in una zona che non conosce direttamente, ma avrà bisogno di informazioni che gli arrivano dai *router* vicini. Quindi in questo tipo di algoritmi, un esempio è il famoso algoritmo chiamato *distante vector*, il calcolo del cammino migliore avviene attraverso un processo iterativo di scambio di informazioni tra diversi *router*.

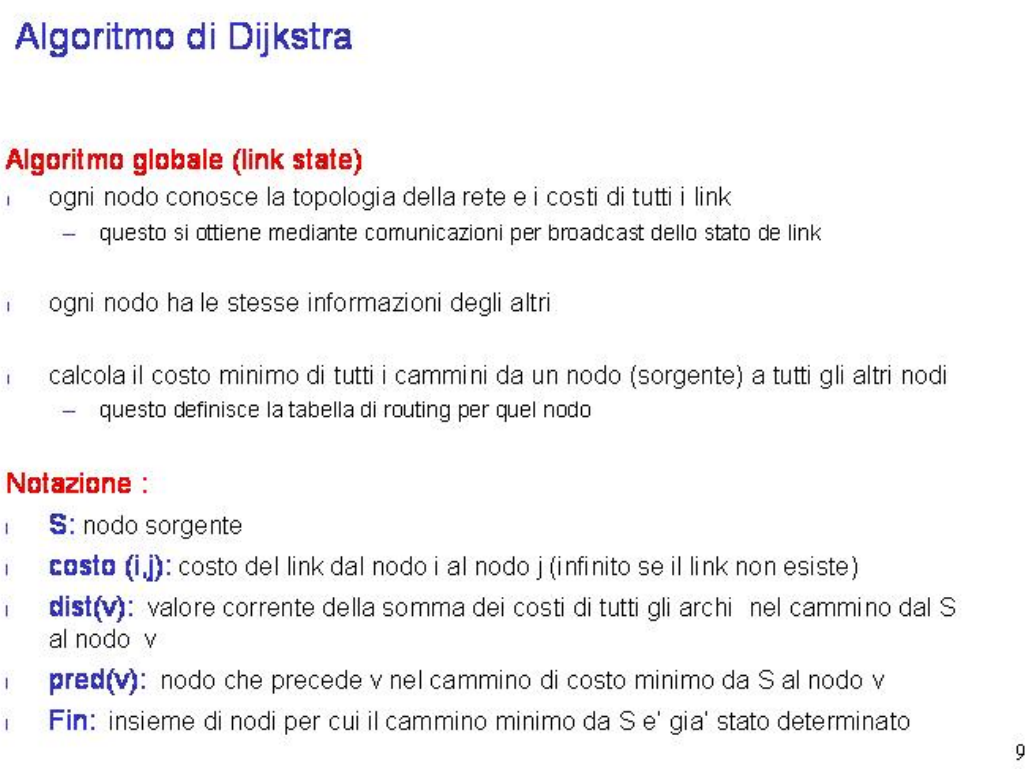
Algoritmo di Dijkstra 1



Algoritmo di Dijkstra

Altra distinzione che è possibile fare tra gli algoritmi di *routing* è quella tra algoritmi statici e dinamici, dove per statici si intendono quegli algoritmi che si applicano quando i costi dei cammini cambiano abbastanza lentamente nel tempo, cioè le variazioni all'interno alle tabelle di *routing* non sono molto frequenti, mentre gli algoritmi dinamici sono quelli che si usano quando queste informazioni cambiano abbastanza velocemente e quindi le tabelle devono essere aggiornate più frequentemente. Vediamo ora un esempio di un algoritmo centralizzato globale, cioè un algoritmo di tipo *link state*, nel quale quindi si assume che ogni nodo conosca esattamente la tipologia di tutta la rete; evidentemente questo tipo di algoritmo non può essere usato a livello geografico, nel senso che è molto difficile che un nodo possa conoscere la topologia di tutta la rete in ambito geografico, ma questo tipo di algoritmi si usa normalmente a livello di reti locali, dove l'estensione della rete, della quale siamo interessati a conoscerne i cammini minimi, non sia particolarmente estesa, cioè il numero di nodi non sia particolarmente elevato. Questo algoritmo è dovuto, come dice il nome stesso, a *Dijkstra*, un informatico olandese che ha dato notevoli contributi in vari settori dell'informatica, e sostanzialmente questo algoritmo calcola tutti i cammini minimi all'interno di un grafo a partire da un certo nodo che identifichiamo con nodo sorgente per arrivare poi a tutti i nodi che compaiono in questo grafo.

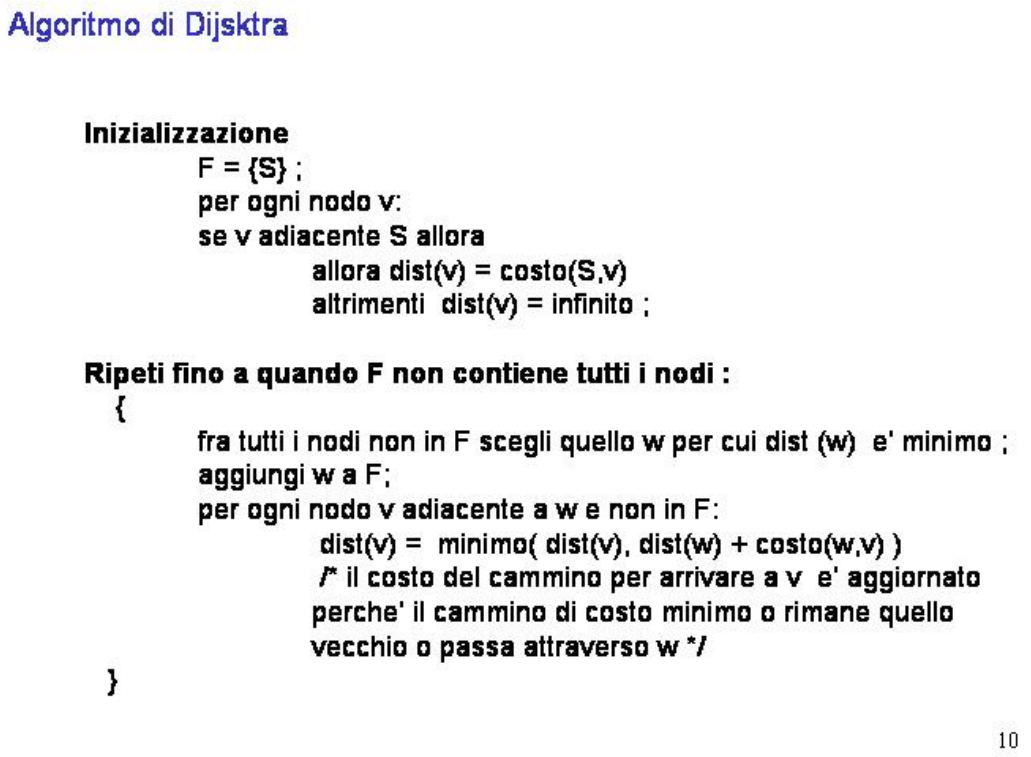
Algoritmo di Dijkstra 2



Algoritmo di Dijkstra

Per poter illustrare questo algoritmo introduciamo un po' di notazione. Come già detto identifichiamo il nodo sorgente, cioè il nodo dal quale vogliamo partire, con S; poi con costo(i,j) denotiamo il costo di un link dal nodo i al nodo j, dove, come abbiamo già detto, il costo può rappresentare informazioni di tipo diverso ad esempio il livello di congestione, il costo economico, eccetera, ma sarà in ogni caso un certo valore numerico e in particolare avrà valore infinito nel caso in cui il link non esista. Indichiamo con dist(v) il valore corrente della somma dei costi di tutti gli archi del cammino da S al nodo v, pred(v) è il nodo che precede v nel cammino di costo minimo dal nodo sorgente S al nodo v. Infine Fin è l'insieme dei nodi del grafo per i quali il cammino di costo minimo dal nodo sorgente è già stato determinato. Quindi lo scopo dell'algoritmo è partire dal nodo S e arrivare ad includere nell'insieme Fin tutti i nodi del grafo.

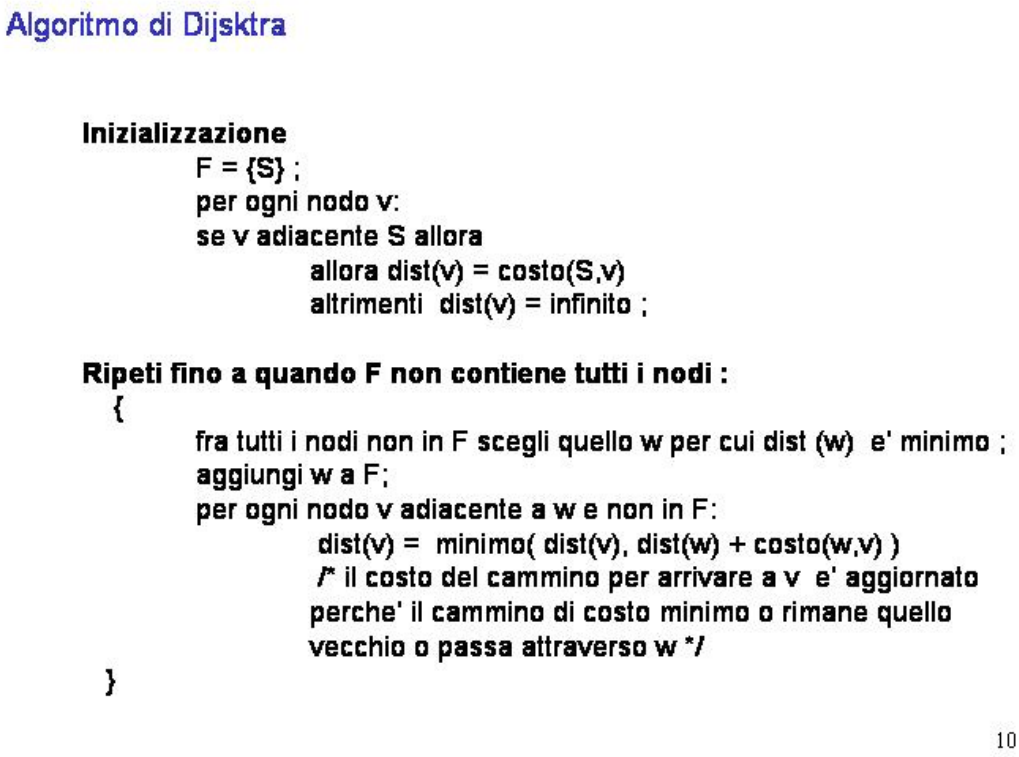
Algoritmo di Dijkstra 3



Algoritmo di Dijkstra

L'algoritmo di *Dijkstra* si compone di queste due parti che vedete qui in figura: abbiamo come al solito una fase di inizializzazione, dove viene inizializzata la variabile F come un insieme costituito dal solo nodo S, cioè il nodo sorgente, quindi per ogni nodo v che si trova nel grafo fornito in *input* dall'algoritmo eseguiamo la seguente istruzione: se il nodo v è adiacente ad S, cioè esiste un arco, un link fisico, tra v ed S, allora assegniamo a dist(v), cioè la distanza di v, il costo dell'arco da v ad S, altrimenti, cioè se questo arco non esiste, assegniamo a dist(v) il valore infinito. Notate che in questo esempio stiamo usando uno pseudo-linguaggio di programmazione, cioè dei costrutti che ricordano in qualche modo un linguaggio di programmazione di tipo imperativo, senza per questo usare uno specifico linguaggio.

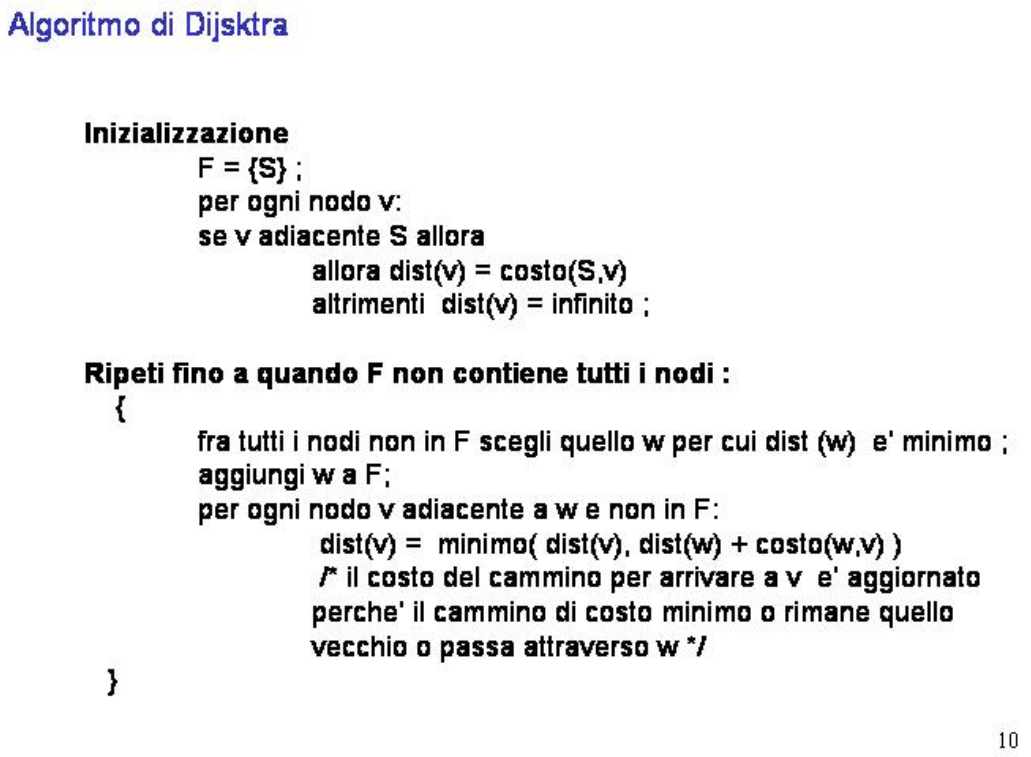
Algoritmo di Dijkstra 4



Algoritmo di Dijkstra

Terminata la fase di inizializzazione abbiamo un ciclo nel quale vengono ripetute un certo insieme di operazioni fino a quando l'insieme F non contenga tutti i nodi. Innanzitutto consideriamo tutti i nodi del grafo che ancora non sono in F (dove F è l'insieme di tutti i nodi del grafo per i quali il cammino di costo minimo è già stato determinato), fra questi nodi rimasti scegliamo il nodo per cui il valore di dist(w) è minimo; in altri termini scegliamo il nodo per cui il costo del cammino dal nodo in questione al nodo sorgente S è minimo. Questo nodo lo aggiungiamo all'insieme F, avendone determinato il cammino minimo a partire da S. Dobbiamo ora aggiornare la situazione dei nodi adiacenti al nodo w che abbiamo appena aggiunto, cioè per tutti i nodi che sono adiacenti al nodo w che non siano già anch'essi appartenenti all'insieme F facciamo la seguente operazione.

Algoritmo di Dijkstra 5



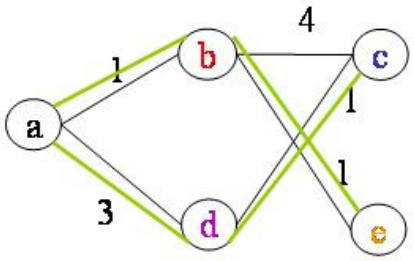
Algoritmo di Dijkstra

Confrontiamo la distanza del nodo dal nodo sorgente, indicata con dist(v), con la somma dist(w) + costo(w,v) ,cioè la distanza del nodo w a S sommata al costo dal nodo w al nodo v. Se il vecchio valore della distanza del nodo v da S è minore della somma calcolata non facciamo nulla, lasciamo cioè questo valore, altrimenti se è maggiore cambiamo il valore di dist(v) assegnandogli il valore restituito dalla somma calcolata. In altri termini, noi sappiamo che per un certo nodo v adiacente ad un nodo w, che abbiamo appena aggiunto ad F, esiste un cammino che parte da S e arriva a questo nodo v facendo una certa strada con un relativo costo; con questa operazione controlliamo se esista un cammino che arrivi a v passando da w e se questo cammino abbia costo minore di quello che già conoscevamo. Se ciò è vero, essendo interessati a trovare cammini con costi minimi, evidentemente sceglieremo il cammino di costo minimo, cioè sceglieremo il cammino che passa attraverso w. Queste operazioni vengono appunto ripetute fino a quando l'insieme F non contenga tutti i nodi e, quando terminiamo il ciclo, abbiamo determinato tutti i cammini minimi dal nodo sorgente a tutti i nodi del grafo e, evidentemente, per ricordarsi quali sono questi cammini minimi, basta che ogni volta che noi aggiungiamo un nodo f ci ricordiamo da dove siamo arrivati, cioè ci ricordiamo qual è il predecessore del nodo appena aggiunto nel cammino che dalla sorgente S arriva a quel nodo.

Esempio di applicazione dell'algoritmo 1

Esempio di applicazione dell'algoritmo

Passo	F	dist(b), pred(b)	dist(c), pred(c)	dist(d), pred(d)	dist(e), pred(e)
0	{a}	1,a	∞	3,a	∞
1	{a,b}		5,b	3,a	2,b
2	{a,b,e}		5,b	3,a	
3	{a,b,e,d}		4,d		
4	{a,b,e,d,c}				



11

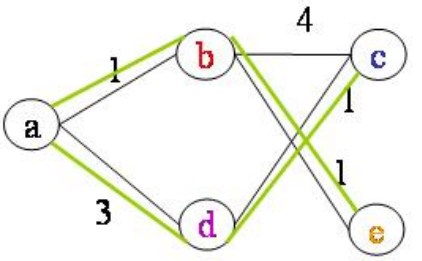
Esempio di applicazione dell'algoritmo

Per cercare di capire un po' meglio questo algoritmo vediamo un esempio di applicazione dell'algoritmo su un caso semplicissimo, quasi banale, che contiene 5 nodi, 5 archi, con i costi che vedete indicati nella figura. Siamo interessati a trovare i cammini minimi dal nodo A, che è quindi il nodo sorgente, a tutti i nodi del grafo. Vediamo che cosa avviene con l'applicazione dell'algoritmo: nel primo passo, indicato con passo 0, l'insieme F sarà costituito dal solo nodo A, dopodiché abbiamo che: la distanza di B dal sorgente A è 1, il predecessore di B è A (questo ci servirà per ricordarsi qual è la sequenza dei nodi del cammino minimo), la distanza di C dal nodo A è infinito, dato che il nodo C non è un nodo adiacente ad A, non esiste un arco che collega direttamente A con C, quindi non ci sarà nemmeno un predecessore di C; esisterà evidentemente un cammino, ma noi al momento non lo conosciamo, dato che nel primo passo controlliamo solamente i collegamenti diretti tra A e gli altri nodi.

Esempio di applicazione dell'algoritmo 2

Esempio di applicazione dell'algoritmo

Passo	F	dist(b), pred(b)	dist(c), pred(c)	dist(d), pred(d)	dist(e), pred(e)
0	{a}	1,a	∞	3,a	∞
1	{a,b}		5,b	3,a	2,b
2	{a,b,e}		5,b	3,a	
3	{a,b,e,d}		4,d		
4	{a,b,e,d,c}				



11

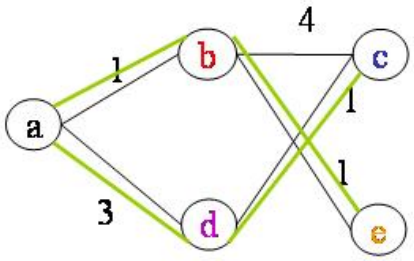
Esempio di applicazione dell'algoritmo

Analogamente possiamo vedere che per il nodo D abbiamo un cammino di costo 3, quindi la distanza da D ad A è 3, il predecessore di D è A. Infine per il nodo E la distanza è infinito, perché non esiste un arco che collega direttamente E ad A e quindi E non ha alcun predecessore. Veniamo ora al passo 1. Tra tutti i valori delle distanze, come abbiamo visto prima nella specifica dell'algoritmo, scegliamo il nodo che ha distanza minore; quindi i valori delle distanze sono rispettivamente 1, infinito,3, infinito, la distanza minore è quella di B, scegliamo quindi il nodo B e inseriamolo nell'insieme F, questo vuol dire che abbiamo determinato un cammino minimo che va da A a B e questo è rappresentato dall'arco che abbiamo aggiunto indicato in figura dalla linea verde. A questo punto ripetiamo il ciclo, abbiamo 3 distanze: 5, 3, 2 e di nuovo tra queste distanze dobbiamo scegliere quella minore, cioè 2 relativa al nodo E (vi ricordo che le distanze sono calcolate dal nodo in questione al nodo sorgente A); quindi E viene aggiunto all'insieme F dei nodi per i quali i cammino sono stati determinati, il predecessore di E è B, ciò significa che il cammino di costo 2 da E ad A è quello che passa attraverso il nodo B; come già detto il predecessore ci serve appunto per ricordare il cammino fatto per andare da A ad E con tale costo. Dal punto di vista grafico ciò è indicato in figura ancora dall'arco verde che viene aggiunto al grafo.

Esempio di applicazione dell'algoritmo 3

Esempio di applicazione dell'algoritmo

Passo	F	dist(b), pred(b)	dist(c), pred(c)	dist(d), pred(d)	dist(e), pred(e)
0	{a}	1,a	∞	3,a	∞
1	{a,b}		5,b	3,a	2,b
2	{a,b,e}		5,b	3,a	
3	{a,b,e,d}		4,d		
4	{a,b,e,d,c}				



11

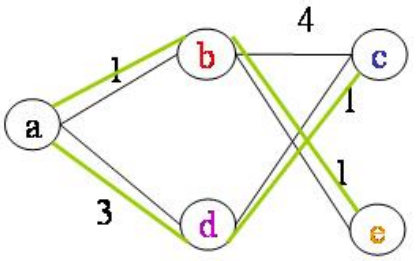
Esempio di applicazione dell'algoritmo

Quindi a questo punto abbiamo 3 nodi, si ripete questo ciclo perché non abbiamo esaurito tutti i nodi possibili, quindi abbiamo 2 nodi da scegliere: il nodo C e il nodo D, al solito scegliamo il nodo di costo minimo; scegliamo il costo minimo 3 relativo al nodo D che viene aggiunto all'insieme F. Aggiungendo tale nodo abbiamo un cammino che arriva a C passando da D, cammino che prima non conoscevamo, quindi mentre prima il collegamento, il cammino da A a C conosciuto era solo quello che passava da B e aveva costo 5, come vedete scritto lì nella terza riga, adesso aggiungendo il nodo D ci accorgiamo che esiste un altro cammino che permette di arrivare a C, passando da A a B e da B a C, che ha costo 4. Per cui scegliamo questo cammino e aggiorniamo la tabella, con il comando visto prima nell'algoritmo, dove sceglievamo il minimo tra i cammini possibili per raggiungere un certo nodo. Ricapitolando, abbiamo trovato un cammino che ha costo minore per arrivare al nodo C di quello visto in precedenza, aggiorniamo allora la tabella di C, dove il valore viene posto a 4 invece di 5 e il predecessore non è più B ma D. Questo lo facciamo soltanto adesso perché nei casi precedenti l'aggiunta di un altro nodo non modificava la situazione degli altri nodi; in generale però ad ogni passo dell'algoritmo, ogni volta che si aggiunge un nuovo nodo all'insieme F, viene controllato se l'aggiunta di questo nodo modifica i costi dei cammini che arrivano ai nodi che ancora non sono nell'insieme F.

Esempio di applicazione dell'algoritmo 4

Esempio di applicazione dell'algoritmo

Passo	F	dist(b), pred(b)	dist(c), pred(c)	dist(d), pred(d)	dist(e), pred(e)
0	{a}	1,a	∞	3,a	∞
1	{a,b}		5,b	3,a	2,b
2	{a,b,e}		5,b	3,a	
3	{a,b,e,d}		4,d		
4	{a,b,e,d,c}				



11

Esempio di applicazione dell'algoritmo

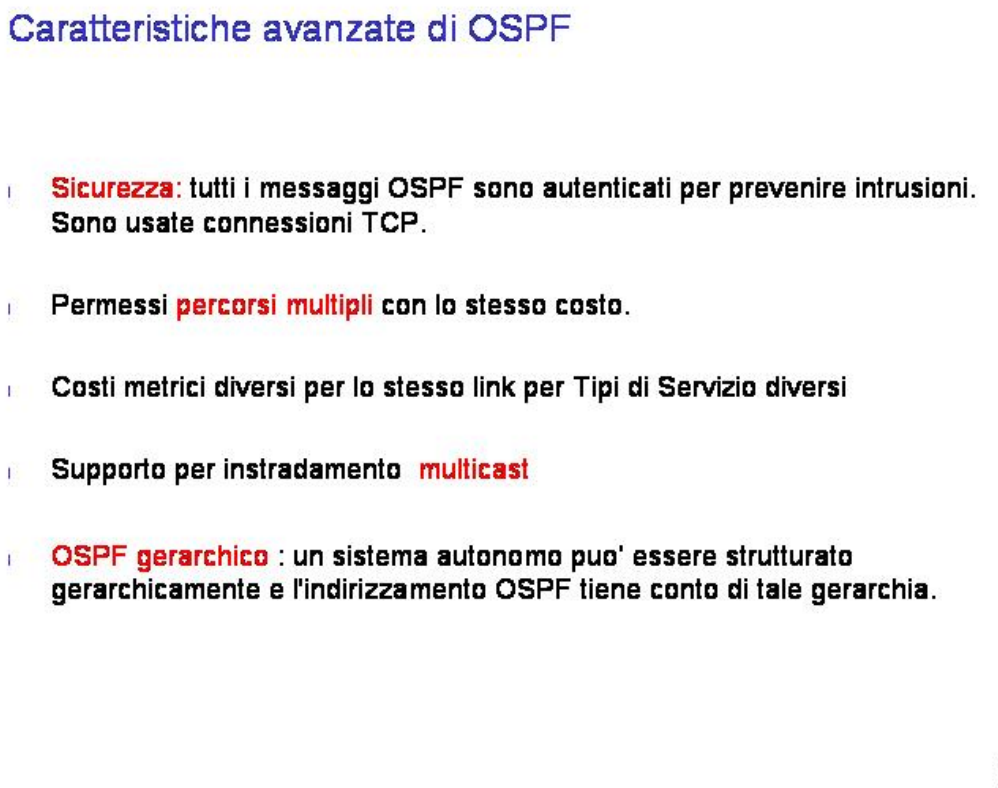
Per completare l'algoritmo, a questo punto ovviamente basterà aggiungere il nodo C. Ora abbiamo quindi tutti i nodi A, B, C, D, E del grafo nell'insieme F, l'algoritmo termina e sappiamo tutti i cammini di costo minimo, rappresentati in figura dagli archi verdi, per arrivare a tutti i nodi del grafo con i relativi costi. Questo appena descritto è appunto l'algoritmo di *Dijkstra*, si può dimostrare che tale algoritmo ha complessità quadratica, è abbastanza buono e, dal punto di vista pratico non può sicuramente essere usato per determinare il cammino minimo di *routing* all'interno di reti geografiche particolarmente vaste, ma può essere usato all'interno di sistemi autonomi, cioè sistemi abbastanza limitati sia dal punto di vista geografico che dal punto di vista del numero di nodi che lo compongono.

OSPF (Open Shortest Path First) RFC 2178

OSPF (Open Shortest Path First) RFC 2178

Un protocollo di *routing* reale che usa l'algoritmo di *Dijkstra* è l'OSPF (*Open Shortest Path First*), che è un protocollo di tipo *Open*, cioè non proprietà di una particolare azienda, è un protocollo dove ogni nodo ha una mappa della topologia della rete, essendo infatti applicabile l'algoritmo di *Dijkstra* ogni nodo ha bisogno di conoscere la situazione di tutta la rete, cioè i costi di tutti gli archi; per avere queste informazioni ogni *router* teoricamente invia delle informazioni di instradamento a tutti gli altri *router* della rete. Come già detto questo protocollo usa l'algoritmo di *Dijkstra*.

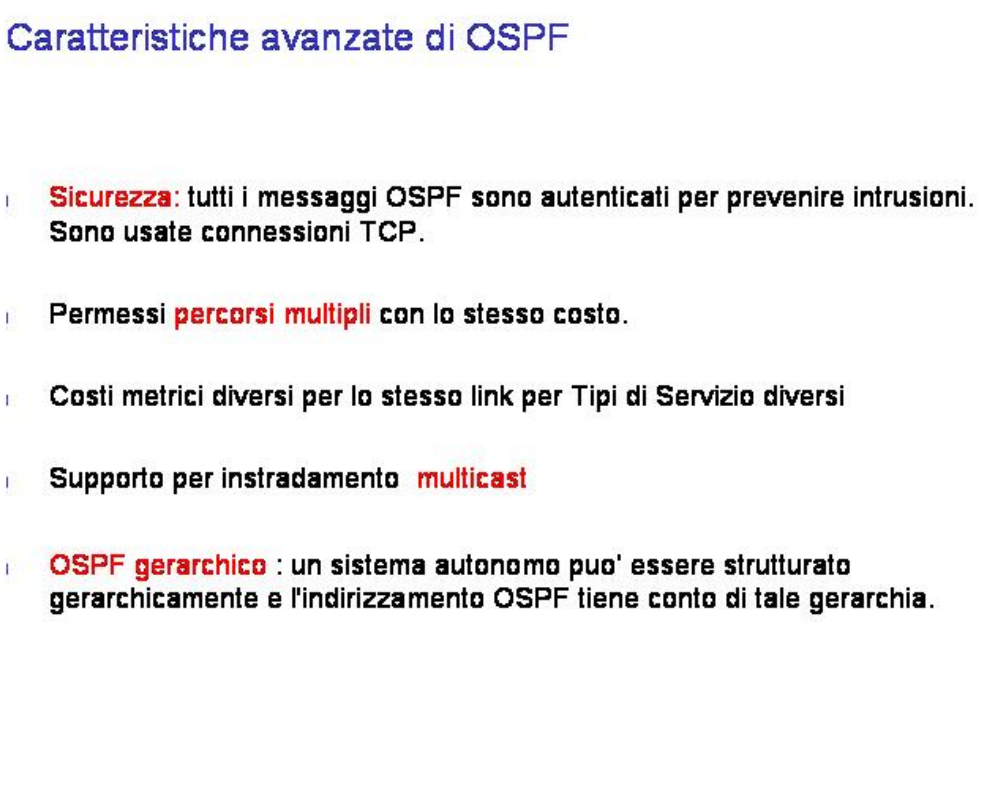
Caratteristiche avanzate di OSPF 1



Caratteristiche avanzate di OSPF

Come già detto questo protocollo usa l'algoritmo di *Dijkstra* con alcune caratteristiche aggiuntive, tra queste ce ne sono alcune che permettono un maggior controllo sulla sicurezza del protocollo stesso e altre che invece permettono di migliorare l'instradamento, ad esempio tenendo conto della struttura gerarchica che può esistere in un certo sistema, dove per struttura gerarchica si intende il fatto che un sistema può essere organizzato a livelli, non visto solo come un grafo ma anche come un albero e di questi livelli dell'albero si deve tener conto per migliorare l'instradamento. Nell'OSPF si tiene appunto conto di queste caratteristiche e anche del fatto che si possono usare costi metrici diversi per lo stesso link a seconda del tipo di servizio del quale si ha bisogno; sono permessi anche percorsi multipli con lo stesso costo e esiste anche il supporto per l'instradamento di tipo *multicast*.

Caratteristiche avanzate di OSPF 2



Caratteristiche avanzate di OSPF

Per quanto riguarda la sicurezza, come vi accennavo prima, per trasmettere informazioni vengono usate semplicemente connessioni TCP-IP in modo tale da essere sicuri che i dati inviati arrivino effettivamente a destinazione e tutti i messaggi OSPF sono autenticati per prevenire intrusioni, perché chiaramente quello che vogliamo evitare è che informazioni necessarie al calcolo del cammino minimo, e quindi al calcolo dell'instradamento dei pacchetti, cioè le informazioni necessarie per valutare il traffico nella rete, possano essere influenzate da intrusioni esterne, potrebbero esserci degli intrusi, interessati per vari motivi a modificare lo stato di instradamento di una rete e ad arrivare a provocare la congestione della rete stessa. Per evitare questo tipo di situazioni vengono usati i meccanismi di autenticazione. Questo termina ciò che avevo da dirvi sull'aspetto del *routing* sulle reti, ovviamente abbiamo visto soltanto una panoramica generale riguardo a cosa sia il *routing* in reti a commutazione di pacchetto, un esempio di algoritmo di *routing*, ne esistono altri altrettanto importanti, in particolare l'algoritmo di *distante vector*, che permette il calcolo di cammini minimi in modo distribuito e che è abbastanza usato dal punto di vista pratico. Con questo avrei concluso.

Bibliografia

Siti

I seguenti articoli - scritti da chi Internet l'ha fatto - costituiscono una fonte di prima mano sulla storia della rete. Il primo è, trai due, quello più agile; descrive anche l'organizzazione a commutazione di pacchetto:

A Brief History of the Internet and Related Networks;
V. Cerf
http://www.isoc.org/internet/history/cerf.shtml

A Brief History of the Internet;
B. Leiner, V. Cerf, D. Clark, R. Kahn, L. Kleinrock, D. Lynch, J. Postel, L. Roberts, e S. Wolff
http://www.isoc.org/internet/history/brief.shtml

Altri riferimenti sulla storia di Internet si trovano a partire da

http://www.isoc.org/internet/history/; http://www.isoc.org/internet/history/

Sulla struttura di Internet, in risposta alla domanda 'Quali sono le componenti principali di Internet e quali sono i più importanti protagonisti di ogni componente', si può vedere

The big picture;
Russ Haynal
http://www.navigators.com/internet_architecture.html

Alcune considerazioni relative al WWW, da parte del suo creatore, Tim Berners-Lee:

http://www.w3.org/People/Berners-Lee/FAQ.html;
http://www.w3.org/People/Berners-Lee/FAQ.html

Una cronologia dei primi anni del WWW (non aggiornata dopo il 1995):

http://www.w3.org/History.html; http://www.w3.org/History.html

Alcune informazioni biografiche sulle figure principali

che hanno creato Internet e il Web sono disponibili a

www.ibiblio.org/pioneers/index.html; http://www.ibiblio.org/pioneers/index.html

Glossario

ARPANET : Rete a commutazione di pacchetto per applicazioni scientifiche e militari creata da **DARPA**; è il nucleo dal quale evolverà **Internet**.

Browser : Applicazione "client" che interagisce con web server attraverso il protocollo **HTTP** per la fornitura di ipertesti multimediali; il browser è responsabile della visualizzazione del documento, sulla base delle informazioni riportate nel documento stesso in linguaggio **HTML**.

Checksum : valore di controllo che serve per il rilevamento di eventuali errori nel segmento.

Commutazione di circuito : Modalità di comunicazione tra due nodi di una rete, nella quale viene stabilito un circuito fisico di comunicazione tra il nodo sorgente e il destinatario; le risorse non vengono rilasciate che al termine della comunicazione.

Commutazione di pacchetto : Modalità di comunicazione tra due nodi di una rete, nella quale il messaggio che il nodo sorgente intende mandare al destinatario è suddiviso in più pacchetti; ogni pacchetto è inviato indipendentemente dagli altri e raggiunge la propria destinazione dopo un certo numero di "salti" tra nodi contigui. È compito del destinatario ricomporre il messaggio, sulla base delle informazioni presenti nei pacchetti. È la modalità usata in **Internet** a livello rete.

Congestione : situazione nella quale ci sono sulla rete più dati di quanti il nucleo della rete riesca a gestirne. Gli effetti della congestione vanno dai lunghi ritardi di arrivo dei dati alla perdita di pacchetti.

DARPA (Defence Advanced Research Project Agency) : Agenzia del Dipartimento della Difesa statunitense che nel corso degli anni 60 e 70 crea **ARPANET**, la prima rete a commutazione di pacchetto.

DNS (Domain Name System) : L'insieme e la struttura dei nomi simbolici di dominio. Talvolta **DNS** sta per Domain Name Server ed è in questo caso sinonimo di **Domain Name Resolver**.

Domain Name Resolver : Applicazione responsabile della traduzione di un nome simbolico di dominio nel corrispondente indirizzo **IP**.

Dominio : insieme di nodi (logici) caratterizzati dalla condivizione delle ultime etichette del proprio nome simbolico. Un dominio di primo livello è l'insieme di tutti i nodi che condividono l'ultima etichetta (per esempio tutti i nodi il cui nome simbolico termina per .it).

Extranet : Quella parte di una rete privata - realizzata con le tecnologie **Internet** - che è accessibile in modo controllato da parte di utenti esterni all'organizzazione proprietaria della rete.

Firewall : Sistema connesso alla rete con lo scopo di filtrare i pacchetti in transito; viene utilizzato con lo scopo di creare una barriera difensiva che aumenti il grado di sicurezza perimetrale di una rete.

Gateway : Vedi router.

HTML (HyperText Markup Language) : Linguaggio di marcatura (markup) mediante il quale viene descritta la struttura di un ipertesto multimediale. Le informazioni **HTML** sono sfruttare da un browser per la visualizzazione del documento.

HTTP (HyperText Transfer Protocol) : **Protocollo** mediante il quale avviene la comunicazione tra un browser e un web server per la fornitura di un documento multimediale.

Indirizzo IP : Numero di 32 bit che identifica in modo univoco ogni nodo di **Internet**; un indirizzo viene in genere indicato come sequenza di 4 numeri decimali, ciascuno compreso tra 0 e 255, separati da un punto.

Instradamento (routing) : Nelle reti a commutazione di pacchetto indica l'attività con la quale un nodo che deve inviare un pacchetto (perché l'ha prodotto o perché l'ha a sua volta ricevuto) passa tale pacchetto ad un altro nodo; componente essenziale di tale attività è la decisione di quale sia il nodo al quale passare il pacchetto.

Internet : Rete planetaria di reti interconesse che condividono un unico spazio di indirizzi e che comunicano sulla base dei protocolli **TCP**/**IP**.

Intranet : Rete privata realizzata con le tecnologie di interconnessione, col software e le applicazioni di **Internet**; in genere è protetta dagli accessi che provengono dalla parte pubblica di **Internet** mediante un firewall.

IP (Internet Protocol) : Il protocollo di livello rete su cui è basato **Internet**; è responsabile dell'instradamento dei pacchetti. Attualmente è in uso la versione 4; presto sarà usata la versione 6.

Ipertesto : corpus di materiali che contiene al proprio interno collegamenti ad altri documenti o a sezioni; le informazioni sono organizzate non in modo sequenziale, ma reticolare; la complessità di tale organizzazione non permette la veicolazione sensata di essa utilizzando media sequenziali.

ISP (Internet Service Provider) : Azienda o ente che fornisce connettività ai propri clienti.

LAN (Local Area Network) : Reti di modeste dimensioni (la sede di un'azienda), realizzate con supporti di telecomunicazione a banda molto larga, i cui nodi principali sono costituiti dai calcolatori degli utenti e dai server.

Livello trasporto : uno dei livelli in cui sono strutturati i meccanismi di comunicazione di rete secondo lo standard ISO/OSI. Il livello rete realizza una comunicazione logica fra processi applicativi che girano su terminali diversi.

MAN (Metropolitan Area Network) : Reti che coprono un'area geografica limitata, realizzate in modo da avere una banda abbastanza larga, i cui nodi sono in genere router che collegano tra loro delle **LAN**.

Multimediale : composto da informazioni veicolate mediante più media (cioè testuali, grafiche, sonore, ecc.)

Nome simbolico di dominio : Stringa che indica un nodo della rete mediante un sequenza di etichette separate da punti.

Numero di porta : numero che permette di identificare un processo del livello applicazione al quale i dati devono essere destinati. Permette il multixplexing. Alcuni numeri di porta sono riservati per applicazioni specifiche (posta elettronica, ftp ecc.).

Numero sequenza : numero che identifica univocamente ogni segmento e che serve per realizzare il trasferimento affidabile.

Pacchetto : Una sequenza di dati, di lunghezza fissata, contraddistinta da informazioni che permettono di ricostruire il mittente, il destinatario, la posizione della sequenza in un messaggio più grosso.

Protocollo : Modalità di comunicazione tra due agenti software o hardware.

Protocollo pipeline : protocollo che permette l'invio di più segmenti prima di attendere i messaggi di riscontro.

Raccomandazione : Documento del **W3C** che fissa alcuni aspetti tecnici del **WWW** che consentono la sua interoperabilità; è l'analogo, all'interno del **WWW**, di quello che una **RFC** è per **Internet**.

Resolving : Vedi **Domain Name Resolver**.

RFC (Request for comments) : Documenti della **Internet** Architecture Board (IAB) coi quali vengono definiti gli standard tecnici di **Internet**.

Riscontro : messaggio con il quale il destinatario di un segmento comunica al mittente la ricezione del segmento.

Router : Processore dedicato, incaricato dell'instradamento dei pacchetti.

Routing : Vedi instradamento.

Segmento : l'unità di dati trasferita fra entità del livello trasporto.

Sito web : Termine non tecnico che indica l'insieme dei documenti ipermediali accessibili attraverso un unico **URI** (o anche attraverso un unico web server).

Socket : interfaccia locale fra livello applicazione e livello trasporto creata, usata e rilasciata esplicitamente dalle applicazioni e controllata dal sistema operativo (introdotto in Unix nel 1981).

TCP (Transmission Control Protocol) : Il protocollo di livello trasporto su cui è basato **Internet**. Realizza un servizio orientato alla connessione e permette il trasferimento affidabile dei dati

UDP (User Datagram Protocol) : protocollo di livello trasporto che realizza il servizio senza connessione di Internet e che permette il trasferimento di dati senza garanzia di affidabilità.

URI (Uniform Resource Identifier) : Stringa che identifica in modo univoco su **Internet** una determinata risorsa; è composto da un protocollo, un nome di dominio, e un cammino d'accesso relativo al dominio.

URL (Uniform Resource Locator) : Termine obsoleto. Usa invece

W3C (World Wide Web Consortium) : Organizzazione senza fine di lucro che sovrintende allo sviluppo del **WWW**.

WAN (Wide Area Network) : Reti di grande dimensione (nazionale o più), realizzate mediante connessioni dedicate punto a punto gestite da aziende di telecomunicazione, i cui nodi sono costituiti da router che collegano tra loro **LAN** o **MAN**.

Web server : Applicazione "server" che interagisce con un browser (il client) attraverso il protocollo **HTTP** per la fornitura di ipertesti multimediali; un web server, insieme ai documenti ipermediali di cui assicura la fornitura, costituisce un sito web.

Web : Vedi **WWW**.

WWW (World Wide Web) : **Ipertesto** multimediale distribuito realizzato su **Internet** e basato sul protocollo **HTTP**.

Autori

Hanno realizzato il materiale di questo modulo:

Prof. Maurizio Gabbrielli

Maurizio Gabbrielli è professore straordinario di informatica presso il Dipartimento di

Scienze di Informazione dell'Università di Bologna. Ha conseguito il titolo di dottore di ricerca in informatica nel 1992 presso l'Università di Pisa ed è stato quindi ricercatore presso il CWI (centro di ricerca per la matematica e l'informatica) di Amsterdam, ricercatore presso l'Università di Pisa e professore associato presso l'Università di Udine. I suoi interessi di ricerca includono i linguaggi di programmazione, i metodi formali per la verifica e l'analisi, la programmazione con vincoli e la teoria della concorrenza. È autore di più di 50 articoli pubblicati su riviste e atti di congressi internazionali, ha partecipato a numerosi progetti italiani ed internazionali ed è stato membro e presidente del comitato di programma di varie conferenze internazionali. È attualmente presidente del comitato esecutivo di PPDP, presidente della associazione italiana di programmazione logica, membro del comitato esecutivo di ALP (*Association for Logic Programming*) e membro del comitato esecutivo di EAPLS (*European Association for Programming Languages and Systems*).

Prof. Simone Martini

Simone Martini è professore ordinario di Informatica nell'Università Bologna. Laureato in Scienze dell'Informazione e Dottore di Ricerca in Informatica (Università di Pisa), ha insegnato nelle Università di Pisa e Udine. È visitatore presso il *Systems Research Center* della *Digital Equipment Corporation* a Palo Alto, la *Stanford University*, l'*École normale supérieure* di Parigi. I suoi interessi di ricerca riguardano i fondamenti logici dei linguaggi di programmazione.