Configurazione di un router
Accesso e configurazione di base

L'accesso al *router* può avvenire:

- via rete (***telnet***);
- collegando un terminale (o un PC) alla *console* (fisicamente è una porta seriale asincrona classica) del *router*.

Nel primo caso è possibile la gestione da remoto; nel secondo caso è necessario essere in locale (oppure collegato al *router* tramite modem) e impostare i corretti parametri del programma di emulazione terminale (*HyperTerminal* nel mondo *Windows*). L'accesso in locale è obbligatorio nella fase di configurazione iniziale del *router*; successivamente è possibile utilizzare anche l'accesso *telnet* (se il *router* è raggiungibile).

Una terza possibilità è quella dell'utilizzo di strumenti di configurazione remota, quali gli strumenti di gestione basati su SNMP. Questa soluzione non consente tuttavia la completa configurazione della macchina; per alcuni aspetti è comunque necessario intervenire sulla configurazione del *router* manualmente.

Un'ultima possibilità, utilizzata soprattutto per controllare alcuni parametri di base del *router*, è quella di accedere alla macchina attraverso un *browser web* (il servizio può essere abilitato tramite il comando IP http *server* dalla modalità di configurazione). Questa modalità offre tuttavia funzionalità estremamente limitate (status di ogni interfaccia, ...) ed è utilizzato più come controllo che come strumento di configurazione.

Comandi fondamentali

Anziché presentare in questa sezione l'elenco dei principali comandi possibili, si presenta un esempio di configurazione reale di un *router* spoglio.

- *Enable*
  Entra in modalità amministrazione (richiede una *password*).
- *Erase startup_config*
  Cancella la configurazione della NVRAM ed azzera la configurazione del *router*. È importante notare che questo comando va impartito in modalità privilegiata e non in modalità di configurazione.
- *Configure terminal | memory | network*
  Entra in modalità di configurazione; i comandi di configurazione verranno impartiti dal *medium* specificato dalla seconda parola chiave. In altre parole, se viene specificata la *keyword terminal* i comandi verranno accettati dalla tastiera; se viene indicata la parola *memory* verrà copiata la configurazione di *startup* in quella volatile ed eseguita, mentre con la parola chiave *network* verrà ricercato un *server* TFTP sul quale è memorizzata la configurazione la quale verrà quindi caricata in memoria ed eseguita.
- *Hostname name*
  Assegnazione del nome al *router*.
- *Enable password* ena_pwd
  Abilitazione (e configurazione) della *password* del *router* locale (quella richiesta alla digitazione del comando *enable*).
- *Username name password* passwd
  Associa *password* a nomi. Può essere utilizzato sia per accedere ad un *router*, sia per configurare il *router* ad accedere in *dial/up* ad un altro apparato. Nel secondo caso, il *router* usa come *password* quella associata al proprio nome.
- *Line* vty 0 4

Configura i terminali virtuali: il primo numero dopo il VTY indica il numero del primo terminale virtuale; il secondo indica il numero dell'ultimo terminale virtuale (in questo caso è stata configurata la possibilità di 5 accessi contemporanei al *router*).

- *Login*
  Imposta l'obbligo di una fase di *login* nell'accesso via *telnet* (ma non impone una *password*).
- *Password telnet*_pwd
  Abilita (e configura) della *password* di accesso al *router* via *telnet*.
- *Exit*
  Esce dalla modalità di configurazione dei terminali virtuali.
- *Exit*
  Esce dalla modalità di configurazione.
- *Show running-config*
  Visualizza su *monitor* l'attuale configurazione (RAM) del *router*.
- *Copy running-config startup-config*
  Salva nella NVRAM la configurazione attiva.
- *Show startup-config*
  Visualizza su monitor la configurazione salvata su NVRAM.

NOTA: Nella visualizzazione di una configurazione (ad esempio *sh run*) vengono riportate solo le opzioni che non sono al valore standard.

Altri comandi fondamentali (attivabili solo in modalità *enable*)

- *copy running-config* tftp:nome*file*
  Salva su un *server* TFTP la configurazione attiva.
- *Reload*
  Effettua il *reboot* del *router*.

Modifiche tra IOS

Nelle più recenti versioni di IOS (fondamentalmente > 12.0) sono stati modificati pesantemente i comandi relativi alla visualizzazione e alla gestione della configurazione. In figura sono riportati i principali comandi e le principali variazioni del settore. I comandi tradizionali sono ancora supportati, ma è preferibile utilizzare i nuovi.

In breve i comandi sono stati unificati sotto le tre voci *copy*, per la gestione dei *file* di configurazione, *more*, per la loro visualizzazione, e *erase* per la loro cancellazione. L'argomento di questi comandi include quindi il *device* fisico dove il *file* è memorizzato (ad esempio tftp:, system:, nvram:, etc) e la sua locazione all'interno di questo *device*.

Vecchi comandi	Nuovi comandi
configure network	*copy ftp: system:running-config*
copy rcp running-config	*copy rcp: system:running-config*
copy tftp running-config	*copy tftp: system:running-config*
configure overwrite-network	*copy ftp: nvram:startup-config*
copy rcp startup-config	*copy rcp: nvram:startup-config*
copy tftp startup-config	*copy tftp: nvram:startup-config*
show configuration | *show startup-config*	*more nvram:startup-config*
write erase | *erase startup-config*	*erase nvram:*
write memory | *copy running-config startup-*	*copy system:running-config nvram:startup-*

config	*config*
write network	*copy system:running-config ftp:*
copy running-config rcp	*copy system:running-config rcp:*
copy running-config tftp	*copy system:running-config tftp:*
write terminal | *show running-config more*	*system:running-config*

Controllo e debug

I principali comandi di utilità, controllo e *debugging* sono solitamente disponibili solo in modalità privilegiata.

- *show* comando
  Visualizza i parametri relativi a comando;
- *show* ? (oppure *show* ip)
  Elenca ciò che è possibile visualizzare;
- *term mon* (*term no mon* per la sua disattivazione)
  Attiva il *debugging* sul monitor (necessario solo via *telnet*, per attivare l'*output* su terminale locale e non sulla *console* del *router*);
- *debug* comando
  Per attivare il *debug* su una funzione specifica;
- *debug* ?
  Mostra le attività su cui il *debugging* può essere attivato;
- *debug ip packet dump*
  Stampa su *monitoring* il *dump* esadecimale dei pacchetti che passano nel *router*; è un comando molto pericoloso per la sua capacità di saturare il *router*;
- *no debug all*
  Per disabilitare tutti i comandi *debug* attivati in precedenza.

Il *debug* deve essere lanciato con cura evitando di saturare la CPU e la capacità trasmissiva (nel caso di *debug* remoto) a disposizione del *router*. Non è infrequente che il *router* risulti saturato dalla gestione dei messaggi di *debug* e che non riesca più ad accettare altri comandi di nessun tipo. In queste condizioni il *debug* provoca la totale perdita di controllo sul *router* che può essere riattivato solamente attraverso l'utilizzo della *console* dello stesso.

Altri strumenti di controllo

Altri comandi utili per il controllo dell'operabilità del *router* sono quelli classici dell'ambiente TCP/IP, e cioè:

- *ping* [indirizzo]
  Controlla la raggiungibilità di indirizzo;
- *trace* [indirizzo]
  Visualizza il percorso verso la destinazione; nel caso di più percorsi, li visualizza tutti;
- *telnet* [indirizzo]
  Apre un terminale virtuale con la destinazione.

Può essere importante ricordare che questi strumenti di diagnostica sono molto approssimativi. Ad esempio una mancanza di risposta al comando *ping* non implica automaticamente la mancanza di una *route* per raggiungere la destinazione, ma può anche essere l'eventuale mancanza di una *route* per il ritorno. È quindi importante accertarsi in prima battuta che i vicini al *router* in esame siano raggiungibili, per poi proseguire il *debug* secondo cerchi concentrici a raggio sempre maggiore.

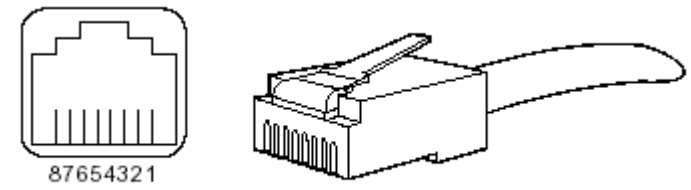
Interconnessione con la rete fisica

Cavi RJ-45

I prodotti *Cisco* utilizzano i seguenti tre tipi di cavi RJ-45:

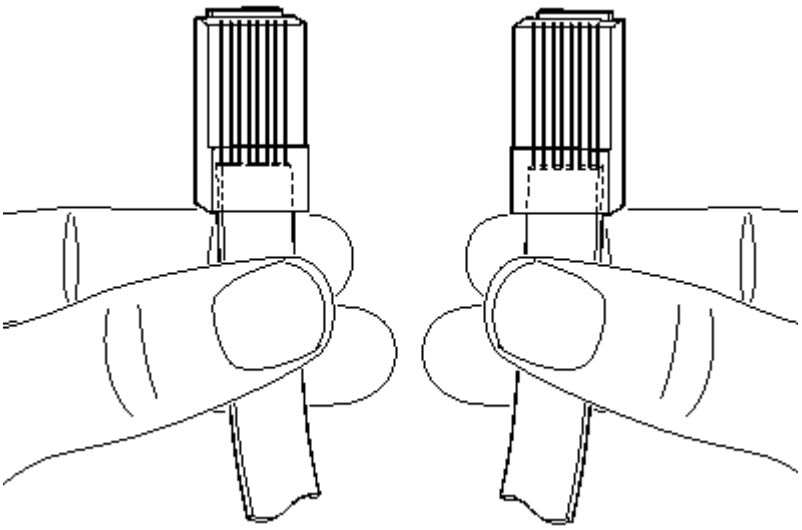
- *Straight-through* (dritto).
- *Crossover* (incrociato).
- *Rolled*.

Cisco tipicamente non li fornisce e sono in genere largamente disponibili da terze parti.



RJ-45 connector

Per identificare il tipo di cavo RJ-45, si deve tenere in mano i due estremi del cavo in modo da poter vedere la colorazione dei cavi all'interno del *plug*, come mostrato in figura.



Si esamini la sequenza di colorazione dei cavi internamente al *plug* RJ-45 (trasparente):

- *Straight-through* [dritto]
  la colorazione dei cavi nel *plug* procede identicamente in ambedue gli estremi.
- *Crossover* [incrociato]
  il primo (più a sinistra) colore di cavo su di una lato del cavo è il terzo sull'altro lato.
- *Rolled*
  i colori dei cavi nel *plug* di un estremo del cavo sono in ordine esattamente opposto a quelli dell'altro estremo.

Layout cavi

Cavo straight-trought

Signal Pin Pin Signal

Tx+	1	1	Tx+
Tx-	2	2	Tx-
Rx+	3	3	Rx+
-	4	4	-
-	5	5	-
Rx-	6	6	Rx-
-	7	7	-
-	8	8	-

Cavo cross-over

Signal	***Pin***	***Pin***	***Signal***
Tx+	1	3	Rx+
Tx-	2	6	Rx-
Rx+	3	1	Tx+
-	4	4	-
-	5	5	-
Rx-	6	2	Tx-
-	7	7	-
-	8	8	-

Cavo straight-trought

Signal	***Pin***	***Pin***	***Signal***
-	1	8	-
-	2	7	-
-	3	6	-
-	4	5	-
-	5	4	-
-	6	3	-
-	7	2	-
-	8	1	-

Configurazione delle interfacce

L'IOS assegna ad ogni interfaccia fisica di rete un identificativo univoco all'interno del sistema. Questo identificativo è formato dal loro nome tecnologico più un identificativo numerico (quindi si troveranno *Ethernet0, Ethernet1, Serial0, Serial1*) in ordine crescente. Nel caso di apparati composti da *chassis*, il numero dell'interfaccia comprende anche il numero dello *chassis* (ad esempio *Ethernet0/1* indica la seconda *Ethernet* del primo *chassis*).

Da questo punto in poi si seguiranno le seguenti regole:

- i comandi, eccetto quando chiaramente specificato, saranno comandi disponibili solamente in

modalità configurazione (o da un suo sottomenu)
- a questa regola fanno eccezione i comandi inizianti per *show*, i quali sono disponibili esclusivamente in modalità *enable*.

Comandi generali

Sono quei comandi che, impostati in modalità *enable*, permettono successivamente la configurazione opportuna delle interfacce vere e proprie.

Sono normalmente dei comandi di tipo generale che hanno validità per tutto il *router*.

ip *subnet-zero*

Abilita l'uso della *subnet* zero sulle interfacce e sulle *routing updates*. In mancanza di questo comando le reti terminanti con "0" non sono ammesse se non con *netmask* naturali (/24, /16 e /8); ad esempio non è ammessa la rete 130.192.1.0/30, mentre lo è la 130.192.1.4/30

Comandi di interfaccia

Sono comandi che vanno dati all'interno della configurazione delle interfacce.

- *interface name*
  Entra nel sottomenu di configurazione dell'interfaccia *name*. Questo comando permette l'entrata nel sottomenu di configurazione delle interfacce abilitando quindi la digitazione dei comandi successivi
- ip *address* indirizzo maschera [*secondary*]
  Assegna all'interfaccia l'indirizzo indirizzo. L'opzione *secondary* indica che l'indirizzo è secondario e permette la configurazione di più indirizzi IP sulla stessa interfaccia fisica
- *description* descrizione_interfaccia
  Assegna una stringa letterale per la descrizione dell'interfaccia
- *shutdown*
  Disabilita il funzionamento di quell'interfaccia (può essere utilizzato ad esempio dalle interfacce ISDN per forzare la terminazione della chiamata corrente); per riattivare l'interfaccia è necessario digitare *no shutdown*
- mtu valore
  Definisce una MTU diversa rispetto a quella standard
- ip *proxy* arp
  Abilita il *proxy* arp su quell'interfaccia

Indirizzi delle interfacce

Per configurare ed attivare le interfacce di rete è necessario seguire alcuni semplici ma indispensabili passi. Le interfacce dei *router Cisco* vanno configurate ponendo attenzione alla tipologia delle stesse e alla porta fisica.

```
Router(config)# interface tipo porta
```

Ad esempio per configurare la porta *Ethernet0* i comandi da eseguire saranno i seguenti:

```
Router<enable
Router#config
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface ethernet0
Router(config-if)#
```

```
Router(config-if)# media-type 10baset
```

Lo stato del *prompt* cambia in *Router(config-if)#* per permettere ulteriori configurazioni dell'interfaccia stessa. La specifica *media-type* è d'obbligo per tutti i *router* che hanno disponibilità di interfacce *Ethernet* con connettore sia RJ45 che AUI. Per *default* il media-type è di tipo AUI.

Proseguendo nella configurazione è necessario assegnare l'indirizzo IP (diamo per scontato che si utilizzi il protocollo IP) e la *subnetmask*.
La sintassi impone questa metodologia di imputazione dei comandi:

```
Router(config-if)# ip address netmask address
```

Per configurare la nostra interfaccia *Ethernet* con l'IP privato 192.168.150.1 dovremo eseguire i seguenti comandi

```
Router(config-if)#ip address 192.168.150.1 255.255.255.0
Router(config-if)#no shutdown
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet 0, changed state to up
Router(config-if)#
```

Nella configurazione di un *router Cisco* assume importanza l'identificazione e l'impostazione dei protocolli di *routing*.

Nomi delle interfacce

In un *router* ogni interfaccia ha un indirizzo IP ben preciso e differente dalle altre (tralasciando i casi di interfacce *unnumbered* o *negotiated*). È buona pratica assegnare un nome simbolico ad ogni interfaccia, inserendolo sia nel *router* tramite il comando IP *host* che in un DNS. Ad esempio:

```
ip host s0_bbone 10.25.23.3
ip host e0_bbone 10.25.24.1
ip host roma_bbone 10.25.24.1
ip domain-name rete.it
```

Scegliamo nel caso sopra come interfaccia principale del *router* l'interfaccia *Ethernet0* (e0). Quindi roma_bbone.rete.it corrisponderà allo stesso indirizzo di e0_bbone.rete.it. Occorre utilizzare un dominio fittizio (rete.it) oppure un dominio reale assegnato dal NIC (questo a seconda che sia una rete privata o una rete pubblica).

Nella configurazione delle interfacce è inoltre buona regola:

- Utilizzare sempre il comando ***shutdown*** se l'interfaccia non è attiva.
- Ricordare di rimuovere indirizzi, descrizione, ecc. se l'interfaccia è stata disattivata. Questo soprattutto se la gestione della rete è affidata a più persone.
- Utilizzare sempre il comando ***description***, che permette di assegnare ad una interfaccia una descrizione. È buona regola in questo caso inserire nella descrizione informazioni aggiuntive, come la tecnologia del collegamento (ISDN, *Frame-Relay*, CDN, ecc.) il numero della linea (DLCI, N. CDN, N. ISDN), punti collegati o cliente collegato all'interfaccia, eventualmente la velocità. Ad esempio: `interface Serial0`
  `description Collegamento Terni-L'Aquila - CDN N. xxxxxx/yy - Vel. 128KBit.`
  Inoltre se si ha un buon sistema di gestione o ci si vuole cimentare nello *scripting*, si può prelevare queste informazioni tramite SNMP e costruire automaticamente una tabella *excel* o un *database* dei collegamenti attivi. Molto utile anche per generare inventari sempre aggiornati, che alternativamente andranno prodotti manualmente.

- Specificare sempre tramite il comando *bandwidth* la velocità del collegamento. In questo modo le statistiche di occupazione che si possono visualizzare tramite il comando *show interface* indicheranno il carico corretto della linea (xxx/255). Ad esempio: `interface Serial0`
  `bandwidth 128`.

Notare che lo specificare la velocità del collegamento sull'interfaccia non limita in alcun modo le *performance* del collegamento ma serve solo a fini statistici per aggiornare in modo corretto le stime di occupazione. Quindi se ad esempio si specifica *bandwidth* 1024 su un collegamento a 2MBit/Sec (2048KBit/Sec) non si limita l'utilizzo del collegamento ad un solo MBit/secondo, ma semplicemente le statistiche di occupazione saranno calcolate dal *router* come se il collegamento avesse una velocità di un solo MBit/secondo.

Infine se si dispone di reti libere a disposizione (ad esempio si dispone di una rete privata e con molti o tutti gli indirizzi liberi) utilizzare sempre una interfaccia di *loopback* configurata su ogni *router*.

In questo modo si avrà a disposizione una interfaccia sempre *up* e raggiungibile, indipendentemente dal fatto che le altre siano *up* o *down*:

```
interface loopback0
ip address 10.149.250.5 255.255.255.252
```

I vantaggi dell'utilizzare una interfaccia di *loopback* su ogni *router* sono diversi, questi alcuni:

- Si dispone di una interfaccia sempre *up* a cui far fare riferimento dai sistemi di gestione per fare i *poll* SNMP, prelevare le configurazioni, andare in *telnet*, ecc.
- Si può utilizzare l'indirizzo assegnato alla *loopback* per rendere *unnumbered* delle interfacce ad esempio seriali o ISDN (non consigliabile se si dispone di indirizzi a disposizione poiché rende più complessa la gestione) ma se si è a corto è una soluzione.
- Si può usare la *loopback* come peer del processo BGP per renderlo più stabile (l'interfaccia è sempre UP).
- *Tunneling*: usare l'interfaccia di *loopback* come estremo di un tunnel rende il tunnel stesso più stabile (se si usa ad esempio STUN o GRE o IPSEC).

Nomi degli apparati

Un'altra cosa da prendere in considerazione sono i nomi che vengono dati agli apparati. Si vedono spesso reti nelle quali gli apparati hanno nomi di fantasia (ad esempio Zeus, Athena, ecc.). Questo tipo di pratica è senz'altro divertente ma molto meno efficace dal punto di vista della gestibilità della rete. Molto meglio dare agli apparati nomi che denotino, ad esempio, la loro locazione fisica. Sarà poi più semplice per il gestore di rete capire su quale apparato e in quale punto della rete è collegato in un certo momento.

Per esempio, un utilizzo sensato del comando *hostname* dell'IOS è il seguente.

```
hostname roma_bbone

hostname terni_access

hostname aquila_ISDN
```

denotando ad esempio la locazione fisica e l'utilizzo che viene fatto del *router* (*router* di *backbone*, *router* di accesso ISDN, ecc.).

Inoltre l'*host name* può essere reperito tramite SNMP da un eventuale sistema di gestione: questo permetterà di rappresentare la mappa fisica della rete su, ad esempio, HP *Openview* NNM in modo molto più semplice. È buona norma documentare in una tabella *excel* i nomi degli apparati, la locazione fisica, gli indirizzi, ecc; quindi occorre configurare un DNS dove inserirete i nomi degli apparati con i rispettivi indirizzi IP: sarà in questo modo molto più semplice raggiungerli utilizzando nomi, invece di indirizzi. Se si utilizza un DNS, bisogna configurare tutti i *router* in modo che facciano riferimento ad esso per risolvere i nomi:

```
ip domain-server 10.5.4.1
```

dove 10.5.4.1 è l'indirizzo IP del vostro DNS. Se invece non si dispone di un DNS *server* si inserisca il comando:

```
no ip domain-lookup
```

per evitare che il *router* cerchi di risolvere ogni nome a lui sconosciuto che si inserisce da linea di comando.

Dial on Demand

Le funzionalità di *Dial on Demand* sfruttano la possibilità di avere connessioni alla rete telefonica pubblica (ad esempio utilizzando un modem, un *Terminal Adapter* per la rete **ISDN**, una scheda di rete ISDN, la rete **X.25**) che vengono utilizzate per attivare dei circuiti di comunicazione di tipo temporaneo tra due *end points*. In generale, questo collegamento è attivato sulla base di un certo evento predefinito, ad esempio la caduta di un circuito fisico primario, la necessità di trasportare un determinato pacchetto dati, e altro. L'attivazione del collegamento comporta l'invio di opportuni comandi di segnalazione da parte dell'interfaccia fisica verso la rete pubblica, in modo da permettere lo scambio dati.

Le motivazioni che portano all'impiego del *Dial on Demand routing* sono principalmente:

- Connessioni *Dial-up*: vengono utilizzate per attivare un canale di comunicazione primario ad-hoc nel caso in cui questo canale non debba essere permanentemente attivo; tipico esempio di impiego è una sede periferica di un'azienda la quale attiva un collegamento per lo scambio dati con la sede centrale solo nel momento in cui ve ne è la necessità. Questa scelta presuppone un traffico ridotto in volume e concentrato in alcuni momenti particolari; ad esempio non è la soluzione ideale per collegare un *server web* ad Internet.
- *Backup*: è utilizzato in caso di soluzione temporanea in caso di caduta di un altro link primario. Il collegamento di *backup* consente di ovviare al disservizio utilizzando un canale temporaneo per lo scambio di dati. Normalmente la connessione di *backup* viene abbattuta nel momento in cui il collegamento primario ritorna operativo.
- Trabocco: è utilizzato quando la capacità del link primario non è più sufficiente a smaltire il traffico. Viene quindi attivato un nuovo collegamento in parallelo al primo in grado di espandere le capacità del canale (inteso come aggregato di due canali). Questa soluzione può essere valida nel caso in cui il traffico in eccesso è concentrato in particolari momenti, e non conviene quindi adottare un canale dedicato ad-hoc, allocato permanentemente.

È da rimarcare come *Backup* e *Dial on Demand routing* siano due cose separate: è possibile abilitare le funzionalità di *backup* su qualunque link, anche se normalmente i link di tipo *dial-up* sono quelli più utilizzati. In linea di principio è quindi possibile utilizzare due collegamenti seriali in parallelo, permanentemente allocati, di cui uno *backup* dell'altro. Il DDR è conveniente nel momento in cui l'ammontare di traffico è limitato ed è concentrato in brevi periodi di tempo. Anche nel caso di funzionalità di *backup*, è necessario che la linea primaria non abbia disservizi troppo spesso, altrimenti può essere più conveniente una seconda linea dedicata di *backup* anziché una

linea ISDN.

Una criticità per il DDR è, ad esempio, l'attivazione di protocolli di *routing* sul collegamento *dial-up* in quanto lo scambio di messaggi di servizio tra *router* adiacenti può mantenere attivo il collegamento 24 ore su 24.

Attivazione e disattivazione di una chiamata

Il DDR si basa su un concetto di evento: il collegamento *dial-up* viene instaurato automaticamente nel momento in cui l'evento viene verificato. Esempi di eventi possono essere la perdita del segnale di portante su un link primario, il superamento di una certa soglia di traffico su un link, oppure più banalmente la richiesta di attivazione manuale da parte dell'utente.

Un evento abbondantemente utilizzato in pratica è la definizione di alcune categorie di traffico (ad esempio alcuni pacchetti dati) che possono scatenare la chiamata. Ad esempio, si potrebbe volere che solamente i pacchetti destinati ad un indirizzo IP (quello del *server* aziendale) abbiano il diritto di far partire la chiamata verso la sede centrale, mentre tutte le altre destinazioni devono essere ignorate.

La fase di riconoscimento dell'evento viene seguita dalla fase di attivazione del canale; i dati verranno inoltrati solo a canale instaurato. È evidente come l'utilizzo in produzione di funzionalità DDR richiede che la fase di connessione sia veloce onde non indisporre gli utenti. Una connessione DDR attraverso un modem, tecnicamente possibile, ha dei tempi di collegamento decisamente elevati e quindi sono preferite connessioni di tipo **ISDN**.

La disattivazione del canale ha un andamento speculare alla sua attivazione: viene verificato un determinato evento (ad esempio la riduzione del traffico sotto una certa soglia, la riattivazione del canale primario, la mancanza di traffico per un certo tempo) al seguito del quale il canale viene abbattuto.

L'evento di chiamata può essere verificato più volte (ad esempio nel caso di soglia di traffico, oppure mediante l'indirizzo IP destinazione); nel caso in cui la chiamata sia già attiva, il nuovo evento viene ovviamente ignorato e il traffico relativo ad esso viene veicolato sulla connessione precedentemente instaurata.

Opzioni di chiamata

Una connessione *dial-up* (di tipo **ISDN**) necessita di alcuni parametri di configurazione:

- Numero di telefono del chiamato: è necessario specificare il numero di telefono da comporre per l'attivazione della chiamata. Il numero di telefono può essere unico, oppure diverso a seconda di alcune scelte. Ad esempio, una sede centrale che ha la necessità di collegarsi con più sedi periferiche può utilizzare un solo accesso ISDN. La chiamata alla sede periferica giusta verrà effettuata in base all'indirizzo IP destinazione contenuto nel pacchetto che scatena la chiamata. In questo modo è possibile risparmiare sull'*hardware* installato, ma è necessario che il *router* abbia più numeri di telefono chiamabili e che sia in grado di discriminare tra essi in base ad un opportuno parametro (ad esempio l'indirizzo IP di destinazione del pacchetto in transito).
- Protocolli di autenticazione: spesso è necessario che la chiamata venga autenticata; è possibile ad esempio specificare un controllo sul numero di telefono del chiamante, oppure richiedere l'utilizzo di *username* - *password* (ad esempio mediante **PAP** o **CHAP**), o altro.

Backup

I *router Cisco* hanno due possibili soluzioni per la realizzazione del *backup* di un link:

- *backup* fisico: l'interfaccia di *backup* (ISDN) è logicamente collegata alle sorti di un'altra interfaccia fisica (ad esempio una seriale) ed entra in azione immediatamente non appena il *router* rileva una mancanza di connettività sul link primario (ad esempio la caduta della portante)
- *backup* logico: l'interfaccia di *backup* è indicata come strada possibile verso la destinazione remota al pari del collegamento primario. A differenza di questo, però, il costo di raggiungimento della destinazione è maggiore, quindi i pacchetti dati scelgono normalmente la prima strada per il raggiungimento della destinazione.

Il *backup* fisico ha il vantaggio di intervenire non appena si rileva un problema sul collegamento primario. Questo, però, può essere anche un fatto negativo, in quanto l'utente potrebbe non essere interessato a pagare l'instaurazione di una connessione anche nel momento in cui non ci sia traffico. D'altro canto questa è l'unica strada possibile nel caso di scambio dati con protocolli che non hanno un proprio livello 3 (ad esempio netbeui).

Dal punto di vista della configurazione degli apparati, il *backup* fisico comprende la configurazione dell'interfaccia primaria con la parola chiave *backup <interface>*, il che indica che, a fronte di un guasto su quell'interfaccia, quella di *backup* deve essere attivata immediatamente. Il *backup* logico invece funziona principalmente nell'ambito di rete IP e consiste nella definizione di una *route* statica verso la destinazione remota ad un costo superiore alla *route* standard. A differenza della *route* standard, che instrada sull'interfaccia primaria, la *route* statica secondaria instrada attraverso l'interfaccia ISDN. Il vantaggio è che la connessione viene stabilita solo quando c'è un'effettiva necessità di scambio di dati, ma, d'altra parte, funziona con *route* IP e quindi non è in grado di attivare la connessione a fronte di traffico di altro tipo. In ogni caso è necessario che l'interfaccia primaria sia in grado di rilevare un malfunzionamento del link fisico. Questo è facilmente ottenibile nel caso in cui il link primario sia gestito attraverso un'interfaccia seriale; viceversa può essere più critico accorgersi di un link interrotto nel caso di tecnologia *Ethernet*.

Sia nel caso di trabocco che nel caso di *backup* è possibile definire il ritardo tra la caduta della linea primaria e l'attivazione della linea secondaria, e il ritardo tra la riattivazione della linea primaria e l'abbattimento della connessione di *backup*.

Trabocco

La gestione del trabocco è molto simile concettualmente a quella del *backup*. L'attivazione del trabocco richiede la definizione di due soglie di traffico, la prima che indica quando deve essere attivato il collegamento supplementare, il secondo che indica a che livello di traffico questo collegamento dovrà essere abbattuto. Il *router* misura la quantità di traffico attraverso un'opportuna media in modo da nascondere variazioni istantanee del carico.

Dialer Profiles

Il sistema operativo IOS mette a disposizione una funzionalità interessante che può essere vista come una virtualizzazione di interfacce di tipo *dial-up*. I *Dialer Profile*s, infatti, sono una nuova interfaccia virtuale che può essere utilizzata per mascherare la mancanza di un numero adeguato di interfacce fisiche. Un esempio chiarisce meglio la situazione. Si supponga un *router* con 10 collegamenti geografici. Ipotizzando di abilitare un *backup* fisico su ogni collegamento, sono necessarie 10 interfacce ISDN: il *backup* fisico richiede che sia rigidamente definita un'interfaccia come *backup* di un'altra. Tuttavia, è altamente improbabile che si guastino 10 collegamenti geografici tutti insieme: è molto probabile che un paio di interfacce ISDN siano più che sufficienti a mettere il *router* al riparo da sorprese.

L'idea di *Dialer Profile*s permette quindi di definire delle nuove interfacce *dial-up* virtuali in modo che sia possibile assegnare ad ogni interfaccia fisica un'interfaccia di *backup*. Queste, poi, condividono uno stesso insieme di interfacce fisiche da cui attingono in caso di bisogno. Le interfacce fisiche, a questo punto, non dispongono di parametri di rete propri, ma acquisiscono quelli delle interfacce virtuali nel momento in cui queste vengono mappate su quelle fisiche.

Nel momento in cui viene rilevata la necessità di attivazione di un'interfaccia *dial-up*, il *dialer* viene attivato e i suoi parametri (indirizzo IP, numero di telefono da chiamare, ...) vengono trasferiti alla prima interfaccia fisica disponibile, attivando così la chiamata. Se esiste già un collegamento remoto verso quella destinazione la chiamata non viene effettuata, con la stessa modalità già vista in precedenza per le interfacce fisiche.

Configurazione ISDN

I passi fondamentali per la configurazione di ISDN sono:

- configurazione delle interfacce;
- configurazione dei gruppi;
- configurazione delle eventuali *access-list*.

I principali comandi di interfaccia sono:

```
isdn switch-type basic-net3
```

Imposta il tipo di *switch* con cui operare (euro-ISDN). Nelle ultime versioni di IOS questo comando è diventato comando di interfaccia (prima era globale), per cui diventa possibile avere interfacce ISDN attaccate a diversi tipi di centralini con segnalazione diversa.

```
dialer string num
```

Utilizza sempre il numero num per aprire una chiamata. Questo comando (oppure in alternativa quello successivo) è sempre obbligatorio.

```
dialer map prot indir [name name] num
```

Permette di definire più numeri a cui instradare la chiamata a seconda del pacchetto che si presenta all'interfaccia, con il significato per instradare un pacchetto del protocollo prot verso il *next-hop* indir apri una connessione con il numero ISDN num. Il pacchetto deve essere di interesse per l'interfaccia; inoltre la corrispondenza tra indir e num è utilizzata anche in fase di accettazione delle chiamate (Se devo raggiungere l'indirizzo indir devo comporre il numero num ma anche: Riconosco come pacchetti validi provenienti da indir solo quelli che provengono da una connessione col numero num).

```
dialer-group num
```

Indica il tipo di filtro da applicare ai pacchetti che attraversano l'interfaccia (specifica il gruppo di accesso cui appartiene l'interfaccia); è un comando obbligatorio.

```
dialer idle-timeout sec
```

Tempo dopo il quale, se non viene rilevato traffico su quell'interfaccia, il collegamento viene disattivato. Se non specificato viene adottato quello standard (120 sec).

isdn answer1 num, isdn answer2 num

Indica che può accettare chiamate provenienti dal numero ISDN num; è usato per il controllo accessi. È possibile indicare fino a due numeri chiamanti (oppure num:subaddr). È possibile impostare *wilcards* (isdn *answer1* 345:2x).

isdn caller num

Come il precedente, ma lo *screening* viene fatto sul valore dell'ISDN caller ID. Se ne possono impostare fino a 64 per ogni interfaccia, ed accetta *wilcard* (isdn *caller* 345xx).

Esempio di configurazione

- *configure terminal*: Entra in modalità configurazione
- *interface* BRI0: Entra in configurazione interfaccia
- *isdn switch-type basic-net3*: Definisce il tipo di ISDN utilizzato
- ip *address* 10.0.0.33 255.255.255.240: Definisce l'indirizzo IP
- *encapsulation* ppp: Definisce l'*encapsulation* di livello 2
- *dialer map* ip 10.0.0.34 5178046: Abbina il numero telefonico da chiamare con l'indirizzo IP remoto
- *dialer-group* 1: Assegna l'interfaccia al gruppo numero 1
- *exit*: Esce dalla configurazione dell'interfaccia
- ip *route* 10.0.0.0 255.255.255.0 10.0.0.34: Configura le *route* statiche necessarie per raggiungere il resto della rete attraverso ISDN

route statiche necessarie per raggiungere il resto della rete attraverso ISDN.

- *dialer-list* 1 *protocol* ip *list* 101
  Dialer-list: definisce come interessanti tutti i pacchetti IP, quindi associa il *dialer-group* 1 alla *access-list* 101 per un miglior raffinamento della politica di accesso
- *access-list* 101 *permit* ip *any any*
  Filtro (numero 101) da applicare ai pacchetti. In questo caso, una *access-list* così semplice è superflua in quanto sarebbe bastato il semplice comando *dialer-list*. Una *access list* migliore potrebbe essere: *access-list* 101 *permit* ip 126.0.0.0 0.255.255.255 128.16.64.0 0.0.0.255, in cui si accettano in ingresso i pacchetti della rete 126.x diretti verso la 128.16.64.x
- *end*
  Termina la configurazione corrente

Accesso ATM tramite interfaccia Seriale 1

Questo paragrafo descrive come configurare *router* che utilizzano una interfaccia seriale per accesso **ATM** attraverso una ***A**TM **D**ata **S**ervice **U**nit* (**ADSU**). Si procederà con:

- abilitare l'incapsulamento *Asynchronous Transfer Mode-Data Exchange Interface* (ATM-DXI),
- selezionare un metodo di incapsulamento multiprotocollo utilizzando ATM-DXI,
- impostare un PVC per l'incapsulamento selezionato.

Accesso ATM tramite interfaccia Seriale

Nei *router* con interfaccia seriale, un ADSU è necessario per fornire l'interfaccia ATM alla rete, convertire i pacchetti uscenti in celle ATM, e riassemblare le celle ATM entranti in pacchetti.

Ogni interfaccia seriale può essere configurata per l'incapsulazione multiprotocollo su ATM-DXI, come specificato nella RFC 1483. Al ADSU, l'intestazione DXI viene eliminata, e i dati del protocollo vengono segmentati in celle per il trasporto sulla rete ATM.

La RFC 1483 descrive due metodi di trasporto traffico di interconnessione di reti di tipo *connectionless* su una rete ATM. Un metodo consente il *multiplexing* di più protocolli su di un singolo PVC. L'altro metodo utilizza differenti circuiti virtuali per trasportare protocolli differenti. L'implementazione della RFC 1483 da supporto a tutti e due i metodi consentendo il trasporto di *Apollo Domain*, *AppleTalk*, *Banyan VINES*, DECnet, IP, *Novell* IPX, ISO CLNS, e XNS.

Task List per la configurazione di accesso seriale ATM

La configurazione di accesso ATM su interfaccia seriale, si completa nei seguenti passaggi.

- Abilitare la *Serial Interface*
- Abilitare l'incapsulazione ATM-DXI
- Impostare il PVC ATM-DXI
- Mappare gli indirizzi di protocollo presso il PVC ATM-DXI
- Controllo (opzionale)

Abilitare la Serial Interface

Per iniziare a configurare l'interfaccia seriale per l'accesso ATM, occorre abilitare l'interfaccia seriale, eseguendo i seguenti passi a partire dal modalità *global* di configurazione.

Il primo comando è il seguente.

```
interface serial number
```

Per ogni protocollo trasportato, si assegni un indirizzo di protocollo per interfaccia.

```
appletalk address network.node
ip address address mask
ipx network number
```

I protocolli supportati sono *Apollo Domain*, *AppleTalk*, *Banyan VINES*, DECnet, IP, *Novell* IPX, ISO CLNS, e XNS.

Accesso ATM tramite interfaccia Seriale 2

Abilitare l'incapsulazione ATM-DXI

Per abilitare l'incapsulamento ATM-DXI su una seriale ***H**igh-**S**peed **S**erial **I**nterface* (**HSSI**), eseguire il seguente comando in *interface configuration mode*:

Comando

```
encapsulation atm-dxi
```

Impostare il PVC ATM-DXI

Un ATM-DXI PVC può essere definito quale elemento in grado di supportare uno o più protocolli, come descritto in RFC 1483 (singolo protocollo) e in RFC 1490 (multiprotocollo)

Per impostare ATM-DXI PVC e selezionare un metodo incapsulato, occorre eseguire il seguente comando impostabile nel *configuration mode*:

```
dxi pvc vpi vci [snap | nlpid | mux]
```

L'opzione MUX (*multiplex*) definisce il PVC per supportare un solo protocollo; ogni protocollo deve essere trasportato su un differente PVC. L'opzione **SNAP** (*SubNetwork Access Protocol*) riguarda l'incapsulazione multiprotocollo LLC/SNAP, compatibile con RFC 1483; SNAP è l'opzione corrente di *default*. L'opzione *Network Layer Protocol IDentification* (**NLPID**) riguarda l'incapsulazione multiprotocollo, compatibile con RFC 1490; questa opzione è fornita per la compatibilità verso il basso con le impostazioni di *default* della precedente versione nel *software Cisco* IOS.

Mappare gli indirizzi di protocollo presso il PVC ATM-DXI

Questa sezione descrive come mappare gli indirizzi di protocollo all'identificatore di canale virtuale (*virtual channel identifier*, VCI) e all'identificatore di cammino virtuale (*virtual path identifier*, VPI) di un PVC che supporta traffico multiprotocollo. Il protocollo indirizza tutti i sistemi *host* fino all'altro capo del collegamento. Per mappare un indirizzo di protocollo a un ATM-DXI PVC, occorre eseguire il seguente comando nella interfaccia *configuration mode*.

```
dxi map protocol protocol-address vpi vci [broadcast]
```

Occorre ripetere questo comando per ogni protocollo che dovrà essere supportato in PVCI protocolli supportati sono *Apollo Domain*, *AppleTalk*, *Banyan VINES*, DECnet, IP, *Novell* IPX, ISO CLNS, e XNS.

Controllo

Dopo aver configurato l'interfaccia seriale per ATM, si può visualizzare lo stato dell'interfaccia, ATM-DXI PVC, o la mappa ATM-DXI. Per mostrare l'interfaccia, PVC, o le informazioni sulla mappa, occorre eseguire i seguenti comandi in EXEC *mode*.

```
show interfaces atm [slot/port]
show dxi pvc
show dxi map
```

Esempio

Il seguente esempio mostra come configurare un'interfaccia seriale per l'accesso ATM.

Nell'esempio, l'interfaccia seriale 0 è configurata per ATM-DXI con incapsulazione MUX. Poiché viene utilizzato l'incapsulamento MUX, è supportato solo un protocollo in PVC. Questo protocollo è identificato in modo esplicito da un comando di mappatura DXI, che identifica altresì l'indirizzo di protocollo del nodo remoto. Questo PVC può supportare il traffico IP di *broadcast*.

```
interface serial 0
ip address 172.21.178.48
encapsulation atm-dxi
dxi pvc 10 10 mux
dxi map ip 172.21.178.4 10 10 broadcast
```

Incapsulamento Frame relay

Per incapsulare in *Frame relay* a livello di interfaccia, si devono eseguire i seguenti comando in *global configuration mode*:

Comando (*configuration mode*):

```
interface type number
encapsulation frame-relay [ietf]
```

Frame relay supporta Incapsulamento di tutti i protocolli in conformità con la RFC 1490, consentendo interoperabilità tra produttori diversi.

Occorre usare la forma Internet *Engineering Task Force* (IETF) dell'incapsulazione *Frame relay* se il *router* o l'accesso del *server* è connesso a un diverso sistema tramite una rete *Frame Relay*.

Si raccomanda di 'spegnere' l'interfaccia precedente prima di cambiare il tipo di incapsulamento. Sebbene ciò non sia richiesto, porre in *shut down* l'interfaccia, assicura che la stessa sia resettata per il nuovo incapsulamento.

Esempio

Di seguito viene impostato l'incapsulamento IETF a livello di interfaccia. Il secondo esempio imposta l'incapsulamento IETF su basi per-DLCI.

```
encapsulation frame-relay IETF
frame-relay map ip 131.108.123.2 48 broadcast
frame-relay map ip 131.108.123.3 49 broadcast

encapsulation frame-relay
frame-relay map ip 131.108.123.2 48 broadcast ietf
frame-relay map ip 131.108.123.3 49 broadcast ietf
```

Impostazione del routing statico: comandi generali

```
ip routing
```

Abilita il *router* ad instradare pacchetti IP (processo di *forwarding*). Questo comando: è utile anche in forma negata (no ip *routing*) per cancellare completamente la precedente configurazione di *routing* e lasciare il *router* spoglio. A questo punto è possibile riabilitare il *routing* e procedere alla nuova configurazione.

```
ip classless
```

Nel momento in cui il *router* riceve un pacchetto per cui non ha una *route* specifica (e nemmeno la *default route*), usa la migliore supernet *route* possibile.

route statiche e di defaul

È il modo più semplice per abilitare il *routing*; non è tuttavia molto robusto in quanto tutto deve essere fatto manualmente e quindi sono estremamente frequenti gli errori (oltre alla mancanza di aggiornamento automatico da parte della rete).

```
ip route indirizzo maschera router [distanza]
```

I pacchetti destinati alle reti comprese nel *range* (indirizzo, maschera) devono essere instradati

verso *router*, che deve essere (1) in una sottorete direttamente collegata a una delle interfacce, oppure (2) una porta del *router* corrente nel caso in cui l'interfaccia sia *unnumbered*. La *route* può essere sostituita da una appresa dinamicamente e avente distanza inferiore.

`ip default-network indirizzo`

Configura una *route* di *default*. È immessa da uno *smart router* il quale normalmente conosce le *route* per qualsiasi destinazione e diffonde la *route* di *default* tramite i protocolli di *routing*. La modalità con cui questa è propagata dipende dal protocollo di *routing*: RIP annuncia 0.0.0.0 0.0.0.0, IGRP annuncia indirizzo indicandola come *route* esterna e candidata per la *route* di *default*.

Configurazione dei protocolli di routing dinamico: Comandi comuni ai protocolli

Il *routing* dinamico necessita di comandi specifici per i diversi protocolli di instradamento, ma alcuni comandi sono comuni ai vari protocolli:

`router proto [ID]`

Abilita il protocollo di *routing* specificato; entra in modalità di configurazione di tale protocollo; ha modalità leggermente diverse per ogni protocollo

`network indirizzo_di_rete`

Specifica contemporaneamente due informazioni:

- le reti (direttamente connesse al *router*) che sono nel dominio di *routing* in esame (e che verranno annunciate dal protocollo);
- le interfacce che dovranno partecipare a quel dominio di *routing* (il *router* automaticamente capisce quali sono le sue interfacce interessate dal dominio, ed abilita l'invio e le ricezione di messaggi di *updates* attraverso quelle interfacce)

La sintassi del comando è leggermente diversa in OSPF e in BGP.

`passive-interface interfaccia`

Inibisce l'invio di messaggi di *update* sull'interfaccia (che, ad esempio, è al bordo del dominio di *routing*). Può essere una ragione amministrativa (evitare di propagare messaggi in una specifica direzione) oppure economica (impedire la generazione di messaggi di *routing* su linee commutate quali ISDN). Questa interfaccia è comunque in grado di accettare e processare *routing update* che arrivano ad essa (inibisce l'invio ma non la ricezione)

`neighbor indirizzo`

Indica al *router* di inviare i messaggi all'indirizzo indirizzo specificato; è usato su reti senza capacità *broadcast* oppure per prevenire l'invio dei messaggi di aggiornamento a specifici *router* (ad esempio su LAN in congiunzione al comando *passive-interface*, per abilitare solo specifici *neighbors*, per ragioni di *policy*)

Configurazione dei protocolli di routing dinamico: Comandi specifici per i protocolli 1

Comandi specifici per RIP

`router rip`

Abilita il protocollo di *routing* RIP. Dal momento che questo comando non ha il parametro ID, non possono coesistere più istanze di RIP sulla stessa macchina

`version 1 | 2`

Abilita l'invio di messaggi secondo la versione 1 o 2 (*default* 1); nella ricezione capisce ambedue le versioni

Comandi specifici per IGRP-EIGR

`router igrp process_id - router eigrp process_id`

Attiva il processo di *routing*. Il *process_id* identifica il particolare processo di *routing* in esecuzione, che deve essere uguale in tutti i *router* del dominio IGRP/EIGRP in quanto l'informazione viene inclusa negli annunci. Se si è in un AS registrato è buona norma porre questo identificativo pari al numero dell'AS; nel caso si voglia impiegare contemporaneamente IGRP e EIGRP (per esempio per necessità di transizione), IGRP e EIGRP possono scambiarsi informazioni solo se *process_id* = AS

`metric weights tos k1 k2 k3 k4 k5`

Cambia il valore dei parametri utilizzati per il calcolo del costo per uno specifico codice *Tipe Of Service* (anche se è fortemente sconsigliato cambiarli); il significato dei termini è analogo al comando *default-metric*. I valori di *default* sono tos: 0, k1=k3= 1, k2=k4=k5= 0

`no metric holddown (solo IGRP)`

Disabilita l'algoritmo di *hold down* di IGRP, migliorando il tempo di convergenza a scapito di possibilità di *loop*.

Configurazione dei protocolli di routing dinamico: Comandi specifici per i protocolli 2

Comandi specifici per OSPF

`router ospf process_id`

Abilita un processo di *routing* OSPF. Il *process_id* identifica il processo di *routing* OSPF all'interno del *router* ed ha significato locale (contrariamente a IGRP/EIGRP non viene trasmesso all'esterno del *router*)

`network indirizzo wildcard area id_area`

Il protocollo OSPF prevede che le reti da annunciare vengano indicate esplicitamente con la coppia <indirizzo,*wildcard*>. Queste informazioni individuano una o più interfacce che si trovano nell'area id_area sulle quali vengono inviati e ricevuti i messaggi OSPF. La maschera è di tipo *wildcard* (come le *access list*), mentre id_area è codificato su 4 byte, ed è possibile utilizzare sia la notazione decimale che quella decimale puntata

`area id_area stub`

Dichiara l'area id_area una stub area

`area id_area range indirizzo maschera`

Specifica un *address range* da annunciare all'esterno dell'area id_area, consentendo l'aggregazione di informazioni per la propagazione all'esterno dell'area id_area (se all'interno dell'area c'è almeno un'interfaccia con l'indirizzo che cade all'interno dell'*address range*, all'esterno è annunciato l'*address range* invece dei singoli indirizzi)

```
area id_area virtual-link ID_router
```

Crea un link virtuale con il *router* che ha ID_*router*, dove questo valore è individuabile visualizzando i *database* di OSPF; l'area id_area è comune ai due *router*

```
default-information originate [aways]
```

Abilita il *router* di annunciare una *route* di *default* all'interno del suo dominio OSPF, comportandosi da AS *Boundary Router* (lo stesso scopo può essere ottenuto con il *redistribute*). Un AS *boundary router* non annuncia necessariamente la *route* di *default*, in quanto potrebbe annunciare più *route* esterne imparate da altri protocolli. Se il *router* non ha alcuna *route* di *default*, questo comando è ininfluente, a meno che si specifichi la *keyword always*, nel qual caso viene comunque sempre immessa una *route* di *default* in quel dominio anche se il *router* non ne ha una propria

Comandi specifici per BGP

La configurazione di BGP è più complessa degli altri protocolli di *routing* e comprende i seguenti passi fondamentali:

```
abilitazione del routing BGP
```

configurazione dei BGP *peers*

```
router bgp AS
```

Attiva il processo di *routing* BGP nell'*Autonomous System* AS

```
network indirizzo [mask netmask]
```

Identifica questa rete come appartenente al dominio BGP locale e la inserisce nella propria *routing table*. Il significato è diverso dai protocolli IGP in quanto il comando *network* non definisce le interfacce sulle quali bisogna inviare gli annunci. Contrariamente ad OSPF, la *netmask* è nella forma classica

```
neighbor indirizzo remote-as AS
```

Dichiara come *peer (neighbor)* il *router* indirizzo dell'*Autonomous System* AS. I *neighbor* possono essere *Internal* o *External*. I *peer* di tipo *external* sono contraddistinti da un link fisico in comune, mentre quelli di tipo *internal* sono posizionati in un qualunque locazione dell'AS

```
aggregate-address indirizzo maschera
```

Se esiste almeno una *route* per una rete che rientra nel range di indirizzi (indirizzo, maschera) BGP annuncia questo range

```
default-information originate
```

Abilita la propagazione della *route* di *default* (0.0.0.0) all'interno del dominio BGP. La generazione

di questa *route* non è fatta in automatico dal BGP, ma deve essere appresa da altre parti (ad esempio mediante redistribuzione).

Configurazione dei protocolli di routing dinamico: Comandi specifici per i protocolli 3

Redistribuzione

È quel processo che permette di collegare due domini di *routing* diversi scambiandosi vicendevolmente le *route* apprese in ognuno di essi. A differenza di avere un dominio unico, il processo di redistribuzione provoca un compattamento delle informazioni di *routing* in modo da rendere il processo più scalabile. In altre parole un dominio di *routing* non conoscerà completamente la topologia dell'altro dominio, ma solo delle informazioni tendenti a dire quali reti sono presenti, ma non qual è il percorso esatto che i pacchetti faranno per raggiungerle. La redistribuzione può essere uni o bi-direzionale.

Uno dei problemi di questo meccanismo è che ogni protocollo di *routing* ha un meccanismo di computo della metrica diverso dagli altri. È allora necessario dire esplicitamente al *router* la metrica con la quale gli annunci dell'altro dominio dovranno essere propagati, con il comando *default-metric*.
I principali comandi connessi alla redistribuzione sono:

```
redistribute protocollo [id]
```

Distribuisce nel dominio del *router* in questione le informazioni raccolte tramite il protocollo protocollo; è un sottocomando della modalità *router*. Il valore ID è necessario per discriminare tra più processi dello stesso protocollo (es. EIGRP).

```
default-metric metrics
```

Comando abbinato al *redistribute*, indicando che tutte le *route* apprese dall'esterno sono da ridistribuire con metrica *metrics*; ha modalità leggermente diverse per ogni protocollo.

```
default-metric k1 k2 k3 k4 k5 (IGRP - EIGRP)
```

Comando abbinato al *redistribute*, indicando che tutte le *route* apprese dall'esterno sono da redistribuire con metrica indicata. Ad esempio il comando *default*-metric 10000 100 255 1 1500 corrisponde ai termini: banda, ritardo, affidabilità, carico, MTU.

```
redistribute static
```

Redistribuisce all'interno del protocollo di *routing* in esame tutte le sue *route* statiche.

```
redistribute connected
```

Ridistribuisce le *route* che vengono create automaticamente per il fatto di avere una interfaccia in esse. Le *route* interessate da questo comando sono quelle non specificate da un esplicito comando *network*; per OSPF e IS-IS queste *route* sono ridistribuite come appartenenti all'esterno dell'AS.

Comandi di controllo

Questi comandi, contrariamente a quelli precedenti, sono attivabili dalla modalità privilegiata.

`show ip route`

Mostra la *routing table* del protocollo IP

`clear ip route {network [mask] | *}`

Va eseguito in modalità privilegiata, permette la cancellazione di una o più *route* che si suppongono non più valide. Questo comando non permette la cancellazione delle *route* statiche

`show ip proto`

Visualizza lo stato di ogni protocollo di *routing* attivo (tempistiche, parametri (es per EIGRP), redistribuzioni, ...)

`show ip eigrp interfaces | neighbors | topology | traffic`

Comandi di controllo del funzionamento del processo EIGRP

`show ip ospf | bgp`

Visualizza le informazioni generali sul processo in esame

`show ip ospf database`

Mostra il *database* dei link *state advertisement* ricevuti

`sh ip ospf neighbor`

Visualizza tutti i *router* OSPF adiacenti, indipendentemente dall'area a cui appartengono; il campo *neighbor*_ID mostra l'identificativo (*router*_ID) del *router* remoto. Nel caso in cui il *router* sia su una *Ethernet*, mostra anche chi è il *Designated Router* e il *Backup* DR, il loro indirizzo su quella rete e l'indirizzo attraverso il quale sono raggiungibili

`sh ip ospf border-routers | interface | virtual-links`

Altri comandi per la visualizzazione di aspetti specifici di OSPF

Lettura del database OSPF

OSPF offre il grande vantaggio di visualizzare un'ottima descrizione della rete. Il comando sh ip ospf *database* può però essere ostico nella sua interpretazione. Per la lettura dei risultati è allora necessario ricordare che:

- *Router Link State*: rappresenta l'elenco dei *router* presenti sull'area in esame.
- *Network Link State*: rappresenta l'elenco delle reti *broadcast* contenute nell'area in esame.
- *Summary Net Link State*: rappresenta l'elenco delle reti presenti nelle altre aree; sparisce qui la distinzione tra reti di transito e reti tradizionali.
  Ogni *entry* (del tipo specificato sopra) è composta da due informazioni, il LinkID e l'*Advertising Router*. Il significato di questo campi è variabile a seconda del tipo di *entry* ed è

schematizzato in figura. Nel caso di un *Router* Link, il LinkID e l'*ADVRouter* coincidono (perché è il *router* che annuncia sé stesso).

Il RouterID è solitamente dato dal più alto indirizzo configurato sulle sue interfacce, tranne nel caso in cui sia stato configurato l'indirizzo di *Loopback* che diventa automaticamente il nuovo *RouterID*. Il *database* deve essere ovviamente uguale all'interno dei vari *router* appartenenti alla stessa area. Ogni *router* può comunque avere uno o più *database*, a seconda che sia un *internal router* oppure un *router* di bordo tra più aree.