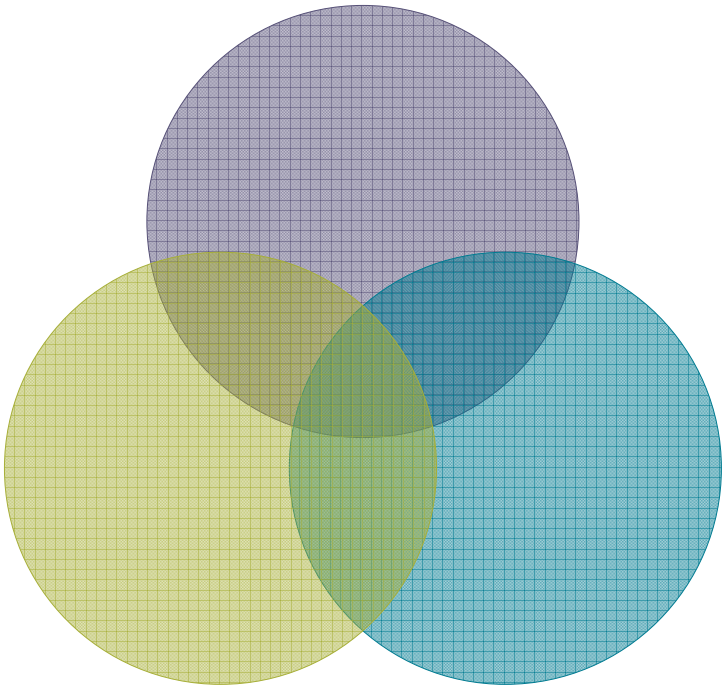


CONFIGURAZIONE DI UNA RETE CISCO: DAI PRIMI PASSI AL VIA

Edizione Febbraio 2005

Autore: Gianrico Fichera

gianrico.fichera@itesys.it

Sommario

gianrico.fichera@itesys.it

0. Approccio di base all'interfaccia

Descrizione dei comandi base per l'uso dell'interfaccia con il router

1. La tabella di routing in IOS

1.1 Cos'e' e come si interpreta

La tabella di routing e' una lista di destinazioni con indicato il percorso da prendere per il loro raggiungimento. Viene conservata da IOS per determinare come instradare i pacchetti IP. Tale tabella contiene tutti i percorsi conosciuti dal router ed e' dinamica in quanto si aggiorna in accordo con le informazioni pervenute dagli algoritmi di routing, dalla configurazione introdotta dall'operatore e dallo stato delle interfacce fisiche presenti nel router.

Per visionare la tabella di routing si utilizza il comando "show ip route". Si vedra' l'intera tabella di routing. Ecco un esempio di output:

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     116.0.0.0/24 is subnetted, 1 subnets
C       116.30.40.0 is directly connected, ATM0
C    192.168.30.0/24 is directly connected, Ethernet0
     151.117.0.0/24 is subnetted, 1 subnets
S       151.117.6.0 [1/0] via 192.168.30.2
```

Analizziamone il contenuto. La prima parte, denominata "Codes" e' una legenda che ci indica come interpretare il simbolismo della tabella di routing. La tabella stessa viene dopo, ad iniziare da 116.0.0.0/24 nell'esempio.

La prima colonna, se non vuota, indica la provenienza della route mediante una lettera da interpretare in base alla legenda. Eccone una interpretazione:

gianrico.fichera@itesys.it

C	La route e' generata dal router e dedotta dal file di configurazione delle interfacce fisiche, effettuata dall'operatore
S	La route e' stata definita esplicitamente dall'operatore nel file di configurazione
R	La route proviene da altri router tramite una interfaccia fisica con l'uso del protocollo di routing RIP
I	La route proviene da altri router tramite una interfaccia fisica con l'uso del protocollo di routing IGRP
D	La route proviene da altri router tramite una interfaccia fisica con l'uso del protocollo di routing EIGRP
O	La route proviene da altri router tramite una interfaccia fisica con l'uso del protocollo di routing OSPF
B	La route proviene da altri router tramite una interfaccia fisica con l'uso del protocollo di routing BGP

La seconda colonna indica la rete di destinazione con relativa netmask. Quando il router deve instradare un pacchetto IP, che al suo interno ha un indirizzo IP mittente e uno destinazione, ne preveva l'ip di destinazione e quindi va alla ricerca di un match nella tabella di routing. Quest'ultima operazione e' effettuata confrontando l'ip di destinazione con la seconda colonna in figura della tabella di routing. Nell'esempio vi sono informazioni per come raggiungere le reti 116.0.0.0, 192.178.30.0 e 151.117.0.0

La terza colonna, quando presente, nell'esempio e' nell'ultima riga e contiene "[1/0]" indica i due valodi di distanza amministrativa (AD) e la metrica. In questo caso si ha una AD pari ad 1 e una metrica pari a 0. Sono valori che verranno spiegati in seguito: in questo momento basti sapere che indicano provenienza e priorita' della riga di routing;

Nella parte restante della tabella viene indicato il NEXT-HOP ovvero a chi deve essere rigirato il pacchetto per raggiungere la destinazione indicata alla prima colonna. Il NEXT-HOP puo' essere il nome di una interfaccia o un indirizzo IP. Nel caso in cui sia un indirizzo IP si ha una chiamata ricorsiva nella tabella di routing, altrimenti si utilizza l'interfaccia indicata. Alla fine infatti deve risultare sempre una interfaccia fisica nel quale il router puo' inviare il pacchetto. Ad esempio se si vuole raggiungere l'indirizzo 151.117.6.2 il next-hop e' 192.168.30.2, ma il router non sa ancora in quale interfaccia fisica inviare il pacchetto. Con una chiamata ricorsiva si individua la rete 192.168.30.0 e quindi il next-hop finale cioe' l'ethernet0;

Nella tabella a seguire vediamo un esempio piu' complesso con alcune route provenienti da routing dinamico. Consiglio di ritornare su questo esempio dopo aver letto i capitoli sui protocolli di routing dinamico. In questo momento l'esempio ci serve per vedere un nuovo dato, l'eta' del route. E' una indicazione del tempo percorso da quando il router e' venuto a conoscenza dell'informazione. Si noti l'ultima riga, dove appare un valore di 25 secondi. Alcuni protocolli di routing come OSPF e EIGRP permettono a questa di crescere indefinitivamente, altri invece ne azzerano periodicamente il valore come RIP (ogni 30 secondi di default) o IGRP (ogni 90 secondi di default).

```
itesys1#sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 128.10.10.2 to network 0.0.0.0

C 192.168.94.0/24 is directly connected, FastEthernet0/0
128.10.0.0/30 is subnetted, 1 subnets
C 128.10.10.0 is directly connected, Serial0/0
I 192.168.0.0/24 [100/8486] via 128.10.10.2, 00:00:00, Serial0/0
R* 0.0.0.0/0 [120/1] via 128.10.10.2, 00:00:25, Serial0/0
```

1.2 Distanza amministrativa (AD)

La distanza amministrativa e' un numero compreso tra 0 e 255. Questo indica il peso, il valore di importanza associato ad ogni informazione di routing. L'informazione di routing, proveniente dagli algoritmi di routing, dal file di configurazione definito dall'operatore e dalle interfacce presenti nel router, ha necessita' di una priorita' in quanto, di fatto, non tutte le informazioni hanno il medesimo valore e, in molti casi, vi sono piu' indicazioni differenti per raggiungere la medesima destinazione, provenienti da fonti diverse.

Poiche' solo le informazioni 'migliori' e non duplicate vanno inserite nella tabella di routing (tranne quando lo si forza con statiche o l'algoritmo di routing lo consente) e' necessario un processo di selezione. La AD consente di effettuare questo passo basandosi su considerazioni logiche piuttosto che sui limiti fisici della rete stessa (come la larghezza di banda o il tempo di latenza). Non si pensi che le scelte di priorita' siano basate sull'ordine di arrivo o dettati dal caso.

Ha priorita' maggiore la route che possiede un valore piu' vicino allo zero mentre ha priorita' minore chi piu' si avvicina a 255. In particolare una AD pari a 255 indica una riga di routing che non verra' utilizzata in nessun caso. D'altra parte una AD pari a 0 dara' precedenza alla route su tutte le altre con la medesima destinazione. I valori di AD sono assegnati per default ma in generale sono modificabili dall'utente. Eccoli riepilogati:

Connessa	**C**	AD pari a 0
Statica	**S**	AD pari a 1
RIP	**R**	AD pari a 120
IGRP	**I**	AD pari a 100
EIGRP	**D**	AD pari a 90 (interno)
EIGRP	**EX**	AD pari a 170 (esterno)
EIGRP	**D**	AD pari a 5 (summarization)
OSPF	**O**	AD pari a 110
EBGP	**B**	AD pari a 20
IBGP	**B**	AD pari a 200

Cerchiamo di comprendere il criterio con il quale sono stati assegnati questi valori di default. E' evidente che un'interfaccia direttamente connessa ad una rete (ad esempio una Ethernet di un router sulla LAN) determina il percorso piu' breve ed efficiente per raggiungere la stessa da cui una AD pari a 0. Una statica, in quanto introdotta per forzare un instradamento (e' stata imposta dall'operatore nel file di configurazione), verra' subito dopo da cui AD 1.

Tra gli algoritmi di routing dinamico i piu' preziosi sono coloro che calcolano con piu' accuratezza il percorso migliore per raggiungere una destinazione e che hanno tempo di convergenza minore (ovvero che calcolano piu' rapidamente il percorso migliore) pertanto EIGRP (proprietario Cisco) precede IGRP (da considerare obsoleto ma ratificato da uno standard) che a sua volta precede RIP (poco scalabile e poco efficiente nel determinare i percorsi migliori) che e' il meno efficiente in assoluto a meno di non usarne la versione aggiornata, RIPv2. Nella tabella comunque figurano valori piu' alti del RIP come per OSPF, EIGRP EX o IBGP. *Ma questi non operano peggio di RIP*. OSPF semplicemente non e' confrontabile con RIP o IGRP o EIGRP perche' opera in contesti differenti. In genere in una internetwork RIP o IRGP o EIGRP operano all'interno di WAN medio/piccole mentre OSPF/BGP vanno sul medio/grande. Solo reti almeno medio/grandi utilizzano OSPF (sull'ordine del centinaio di router) pertanto informazioni provenienti da OSPF non sono confrontabili con quelle degli algoritmi minori.

Infine EBGP (exterior BGP). Il BGP viene utilizzato per collegare tra loro macroreti su scala ancora superiore rispetto OSPF. Si utilizza per lo scambio di informazioni di routing tra operatori di telecomunicazioni e ISP. BGP scambia informazioni tra ISP o piu' in generale tra diverse organizzazioni tecniche e viene utilizzato nella rete mondiale INTERNET per scambiare informazioni di routing di diversi operatori. Se un router ha una sessione EBGP aperta riceve informazioni per raggiungere reti distanti non presenti nell'unita' amministrativa dove si trova (che puo' contenere pochi o centinaia di routers). Siamo certi che queste informazioni hanno priorita' su qualsiasi

algoritmo di routing locale (da RIP a OSPF) che non dovrebbe essere in grado di fornirne. Non pretendo di essere stato chiaro per il lettore che non conosce il routing dinamico. Abbia la pazienza di leggere i capitoli successivi e quindi di ritornare su questo paragrafo.

1.3 Metrica

Abbiamo ormai compreso che l'AD e' misura della priorita' di una riga di routing stabilita secondo parametri di efficienza non basati sulle caratteristiche fisiche di una rete ma piuttosto sull'efficienza e sulla tipologia dei vari protocolli di routing. La metrica invece e' una misura del valore di una route determinata a partire dalle caratteristiche fisiche specifiche della rete su cui si opera. Queste ultime sono l'hop-count per il RIP, il delay e bandwidth (di default) per IGRP e EIGRP o altre metriche composte per altri algoritmi di routing.

Questo significa che l'algoritmo RIP considera piu' efficiente il percorso con il numero minore di router (un hop e' un passaggio cioe' normalmente un router), una filosofia abbastanza comprensibile. Infatti se per andare da A a B vi sono due percorsi, uno con 5 router in mezzo e uno con 10 la scelta migliore sembra proprio la prima. IGRP e EIGRP invece non considerano il numero di hop per andare ad esempio da A a B ma piuttosto i ritardi di propagazione dei vari router sul percorso (delay) e la capacita' di banda dei vari link (bandwidth). A questo punto abbiamo informazioni sufficienti per capire come un router seleziona tutte le informazioni di routing e decide cosa inserire in tabella di routing.

Abbiamo detto che devono essere scartate le informazioni duplicate ma il problema era come fare selezione. Cio' che ha metrica minore va scelto. Pero' attenzione, i confronti si possono fare solo tra route che hanno la stessa distanza amministrativa perche' queste hanno metriche calcolate con lo stesso criterio. Non ha senso confrontare metriche provenienti da algoritmi di router distinti in quanto sono valori numerici calcolati con formule diverse e quindi non confrontabili. A parita' di AD invece si opera nell'ambito dello stesso algoritmo pertanto i dati sono confrontabili.

E' evidente anche perche' prima si considerino le AD piuttosto che le metriche nel processo di selezione: una riga di IGRP ha priorita' su RIP indipendentemente dai valori di metrica in quanto le informazioni che fornisce hanno una precisione maggiore rispetto al RIP. In un contesto in cui vi sono due percorsi, per raggiungere, ad esempio, un router B da un router A, di 2 e 3 hop rispettivamente, RIP darebbe priorita' al primo percorso anche se di link con poca banda e con alti delay mentre scarterebbe il secondo anche se di link con grande ampiezza di banda e bassi delay.

1.4 Scelte all'interno della tabella di routing

Abbiamo compreso 'cosa' arriva nella tabella di routing dopo il processo di selezione. Adesso si tratta di scoprire come IOS opera delle scelte all'interno della tabella di routing. Gli algoritmi RIP e IGRP non conservano (tranne dal 12.0 per RIP) nessuna delle informazioni scartate perche' non efficienti in termini di AD o metrica.

Pertanto tutto cio' che proviene dai router vicini e non entra nella tabella di routing viene perduto. OSPF e EIGRP conservano invece anche parte delle informazioni non inserite in tabella di routing e si vedra' che questo e' il loro segreto per la veloce convergenza.

Per riepilogare tutto cio' che ha duplicati con stessa AD ma metrica inferiore viene scartato. Cio' che ha differente AD a parita' di destinazione viene scartato se ha AD maggiore. Ma cosa accade quando vi sono *due righe di routing dove la prima sia un sottoinsieme dell'altra*? Cosa scegliere tra 172.43.0.0 e 172.43.20.0 quando la destinazione e' 172.43.20.23? Questa domanda e' di fondamentale importanza ed e' importante averne la risposta che e' data dalla regola del "*longest match rule*". **Questa regola dice che viene scelta la rete di destinazione che ha piu' bit in comune con l'indirizzo di destinazione**. In questo caso 172.43.20.0. Ed e' la scelta piu' saggia. Ovvero il routing piu' specifico vince sul piu' generico. Si consideri la seguente tabella di routing:

```
Router#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     116.0.0.0/24 is subnetted, 1 subnets
C       116.30.40.0 is directly connected, ATM0
C    192.168.30.0/24 is directly connected, Ethernet0
     151.117.0.0/16 is variably subnetted, 2 subnets, 2 masks
S       151.117.0.0/16 [1/0] via 192.168.30.3
S       151.117.6.0/24 [1/0] via 192.168.30.2
```

Qual'e la scelta per 151.117.6.5? La risposta e', per la regola del *longest-match,* 'via 192.168.30.2'. Nel caso di 151.117.8.0? Si andra' via 192.168.30.3. Per la destinazione 151.116.0.5? In quest'ultimo caso il router SCARTA il pacchetto! Per capire la scelta che fara' un router per raggiungere una destinazione ci viene di nuovo in aiuto il comando "show ip route". Ecco nel dettaglio cosa succede:

```
Router#show ip route 151.117.6.5
Routing entry for 151.117.6.0/24
Known via "static", distance 1, metric 0
Routing Descriptor Blocks:
* 192.168.30.2
Route metric is 0, traffic share count is 1

Router#show ip route 151.117.8.0
Routing entry for 151.117.0.0/16
Known via "static", distance 1, metric 0
Routing Descriptor Blocks:
* 192.168.30.3
```

```
Route metric is 0, traffic share count is 1

Router#show ip route 151.116.5.0
% Network not in table
```

Il router non sa come raggiungere la rete 151.116.5.0. investigando ulteriormente sulle scelte del router sempre per la classe 151.116.5.0:

```
Router#ping 151.116.5.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 151.116.5.1, timeout is 2 seconds:

01:33:26: IP: s=192.168.30.1 (local), d=151.116.5.1, len 100, unroutable
01:33:26: ICMP type=8, code=0.
01:33:28: IP: s=192.168.30.1 (local), d=151.116.5.1, len 100, unroutable
01:33:28: ICMP type=8, code=0.
01:33:30: IP: s=192.168.30.1 (local), d=151.116.5.1, len 100, unroutable
01:33:30: ICMP type=8, code=0.
01:33:32: IP: s=192.168.30.1 (local), d=151.116.5.1, len 100, unroutable
01:33:32: ICMP type=8, code=0.
01:33:34: IP: s=192.168.30.1 (local), d=151.116.5.1, len 100, unroutable
01:33:34: ICMP type=8, code=0.
Success rate is 0 percent (0/5)
```

Aggiungiamo una riga di routing statico. Si utilizza di comando "ip route" indicando la rete con netmask e quindi il NEXT-HOP. Un caso particolare e il routing di default che indica il percorso per raggiungere una qualsiasi rete:

ip route 0.0.0.0 0.0.0.0 192.168.30.2

ora riproviamo:

```
Router#ping 151.116.5.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 151.116.5.1, timeout is 2 seconds:

01:36:35: IP: s=192.168.30.1 (local), d=151.116.5.1 (Ethernet0), len 100, sending
01:36:35: ICMP type=8, code=0.
01:36:37: IP: s=192.168.30.1 (local), d=151.116.5.1 (Ethernet0), len 100, sending
01:36:37: ICMP type=8, code=0.
01:36:39: IP: s=192.168.30.1 (local), d=151.116.5.1 (Ethernet0), len 100, sending
01:36:39: ICMP type=8, code=0.
01:36:41: IP: s=192.168.30.1 (local), d=151.116.5.1 (Ethernet0), len 100, sending
01:36:41: ICMP type=8, code=0.
01:36:43: IP: s=192.168.30.1 (local), d=151.116.5.1 (Ethernet0), len 100, sending
01:36:43: ICMP type=8, code=0.
```

Questa volta il pacchetto e' partito, grazie alla regola del longest-match. *Infatti nella peggiore delle ipotesi un pacchetto fara' matching con 0.0.0.0. Cosi' abbiamo definito un gateway di default con la riga "ip route 0.0.0.0 0.0.0.0 192.168.30.2".* Si presume che 192.168.30.2 sia un router in possesso delle informazioni necessarie per raggiungere le destinazioni cercate.

La regola del *longest-match* non funziona in un caso: **"quando una parte di una major network e' conosciuta, i pacchetti per la parte restante sono scartati"**. Per capire il significato di questa seconda regola cancelliamo dalla nostra tabella di routing il riferimento a 151.116.0.0:

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 192.168.30.2 to network 0.0.0.0

116.0.0.0/24 is subnetted, 1 subnets
C 116.30.40.0 is directly connected, ATM0
C 192.168.30.0/24 is directly connected, Ethernet0
151.117.0.0/24 is subnetted, 1 subnets
S 151.117.6.0 [1/0] via 192.168.30.2
S* 0.0.0.0/0 [1/0] via 192.168.30.2
```

Secondo quanto affermato un riferimento alla rete 151.117.7.0 dovrebbe, per la regola del longest-match fare match con la route di default. Ma 151.117 e' una rete di classe B, che il router ritiene gia' di conoscere come indicato nella riga "151.117.0.0/24 is subnetted". Ecco cosa succede:

```
Router#ping 151.117.5.0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 151.117.5.0, timeout is 2 seconds:

01:43:11: IP: s=192.168.30.1 (local), d=151.117.5.0, len 100, unroutable
01:43:11: ICMP type=8, code=0.
01:43:13: IP: s=192.168.30.1 (local), d=151.117.5.0, len 100, unroutable
01:43:13: ICMP type=8, code=0.
01:43:15: IP: s=192.168.30.1 (local), d=151.117.5.0, len 100, unroutable
01:43:15: ICMP type=8, code=0.
01:43:17: IP: s=192.168.30.1 (local), d=151.117.5.0, len 100, unroutable
01:43:17: ICMP type=8, code=0.
01:43:19: IP: s=192.168.30.1 (local), d=151.117.5.0, len 100, unroutable
01:43:19: ICMP type=8, code=0.
```

Insomma la route di default non funziona per la rete 151.117.5 anche se questa non e' esplicitamente presente nella tabella di routing. La regola del longest-match

funziona quando la classe di destinazione non e' conosciuta dal router. Siamo in un ambiente "classfull" ovvero si utilizzano le regole che limitano le classi A, B, C, D. Se un indirizzo appartiene ad una classe sconosciuta si va verso il default route ma se una subnet di una classe e' conosciuta le altre subnet della stessa, anche se non presenti, non usano la longest-match.

Per svincolarci dall'ambiente "classfull" ed entrare in un ambiente "classless" dove i concetti di classe non hanno piu' significato si utilizza il comando (ormai attivo di default in molte versioni di IOS) **"ip classless"**. Dopo averlo inserito, la tabella di routing non cambia.

gianrico.fichera@itesys.it

```
Router#ping 151.117.5.0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 151.117.5.0, timeout is 2 seconds:

01:47:40: IP: s=192.168.30.1 (local), d=151.117.5.0 (Ethernet0), len 100, sending
01:47:40: ICMP type=8, code=0.
01:47:42: IP: s=192.168.30.1 (local), d=151.117.5.0 (Ethernet0), len 100, sending
01:47:42: ICMP type=8, code=0.
01:47:44: IP: s=192.168.30.1 (local), d=151.117.5.0 (Ethernet0), len 100, sending
01:47:44: ICMP type=8, code=0.
01:47:46: IP: s=192.168.30.1 (local), d=151.117.5.0 (Ethernet0), len 100, sending
01:47:46: ICMP type=8, code=0.
01:47:48: IP: s=192.168.30.1 (local), d=151.117.5.0 (Ethernet0), len 100, sending
01:47:48: ICMP type=8, code=0.
```

Con "**ip classless**" e un "**ip route 0.0.0.0 0.0.0.0 A.B.C.D**" siamo sicuri che il router provera' sempre a consegnare i pacchetti giunti da una delle sue interfacce.

Attenzione che:

```
Router#show ip route 18.181.0.31
% Network not in table
```

non vuol dire necessariamente che il router scarta il pacchetto ma solo che la rete non e' definita esplicitamente in tabella di routing.

1.5 Routing di default

La gestione del routing di default e' abbastanza articolata. Inserire un "ip route 0.0.0.0 0.0.0.0 serial0" (se abbiamo una interfaccia seriale nel nostro router) non sempre porta ai risultati sperati. Un esempio e' stato analizzato nel paragrafo precedente in un ambiente classfull. Il principio di funzionamento del default route si basa sulla regola del longest match: se una destinazione non fa match non nulla lo fara' con la classe 0.0.0.0. Si basa pertanto sull'analisi della tabella di routing che tuttavia, in contesti particolari dove il motore di routing IP di IOS e' disattivato, non funziona. Ad esempio se ci troviamo nell'IOS in boot mode (ad esempio sulla vecchia serie Cisco 2500) o se stiamo utilizzando un router come bridging il motore di routing potrebbe non essere attivabile (con il comando "ip routing" che normalmente di default e' inserito nel file di configurazione di fabbrica) impedendo il raggiungimento magari di un server TFTP. Il comando "ip default-gateway xx.xx.xx.xx" non e' basato sul motore di routing di IOS e risolve casi come questo.

Anche in circostanze ordinarie questo comando puo' essere utile. L'unica alternativa a "ip route 0.0.0.0 0.0.0.0 A.B.C.D" e' infatti il comando "ip default-network xx.xx.xx.xx". Indicando una rete di default e non un indirizzo ip, l'effettivo gateway di

default cambiera' dinamicamente insieme alla route che conduce alla rete A.B.C.D . Bisogna fare molta attenzione al fatto che questo comando in generale e' classful (questo dipende dalla versione di IOS). Ad esempio "ip default-network 151.97.0.0" e' corretto ma "ip default-network 151.97.10.0" potrebbe essere errato. Tuttavia normalmente quando si introduce una sottorete, come nel secondo esempio, il router automaticamente crea una routing statica per la rete classless. Il router poi cerchera' nella sua tabella di routing come raggiungere 151.97.10.0 e, non appena trovata la migliore subnet, considerera' quella come default.

Questo metodo e' il piu' robusto per configurare il routing di default in quanto, a parte che con piu' comandi di questo genere si possono scegliere dei percorsi di back-up, la caduta di un gateway in una rete a maglia consente ancora di raggiungere comunque 'l'uscita'. Consiglio di iniziare ad utilizzare questo comando per le route di default in presenza di reti complesse. Il motivo principale, oltre a quello della robustezza, e' il comportamento dei protocolli di routing in presenza della route di default: "ip route 0.0.0.0 0.0.0.0 A.B.C.D" viene propagato senza problemi solo dal RIP ma attenzione, dalle versioni 12.0T dell'IOS anche il RIP non lo rileva e bisogna utilizzare il comando "default-information originate". EIGRP e IGRP la comprendono se si effettua redistribuzione delle statiche o se si usa "ip default-network" (con il comando "redistribute static" presente oltretutto anche in RIP, IGRP, OSPF, EIGRP).

Con IGRP e EIGRP, con il comando "ip default-network A.B.C.D", si ha la propagazione purche' A.B.C.D partecipi al routing dinamico. In presenza di piu' comandi 0.0.0.0 con la stessa AD il router fa bilanciamento di carico. In presenza di piu' comandi default-network sceglie quello con AD piu' bassa. In caso di entrambi i metodi a precedenza quello con AD piu' bassa (in genere 0.0.0.0 perche' e' una statica) che capita per default-network se viene associato con una statica.

In ogni caso per far funzionare correttamente il default gateway usare anche IP CLASSLESS. Questo perche' quando in un router si cerchera' nella tabella di routing una net non presente, ma comunque subnet della classe propagata, ci sara' sempre una direzione per il pacchetto piuttosto che uno scarto. Senza IP CLASSLESS il routing di default funzionerebbe solo all'esterno della classi routate e quindi ci vorra' continuita' di subnet per non lasciare mai nessuno isolato (cosa che puo' essere spiacevole in casi di redistribuzione da VLSM a FSLM e viceversa).

1.6 IP MULTIPLI

Per assegnare un indirizzo ip ad un'interfaccia si utilizza il comando "*ip address <ipaddress> <mask> [secondary]*". Per assegnare piu' ip alla stessa interfaccia si aggiungono comandi "ip address" da terminare con la keyword "secondary" ad esempio:

```
ip address 192.168.30.1 255.255.255.0
ip address 192.168.40.1 255.255.255.0 secondary
```

Come si comporta un router con piu' ip sulla stessa interfaccia? La risposta e' che gestisce gli indirizzi indipendentemente l'uno dall'altro e nella tabella di routing

appaiono entrambi. E' opportuno consultare la documentazione nel caso in cui vi siano protocolli di routing dinamico. L'indirizzo IP primario e' quello usato dal router quando si inviano pacchetti che partono da quell'interfaccia. Cosi' ad esempio quando si effettua un ping dall'interno del router l'ip primario viene utilizzato come mittente. Se si vogliono utilizzare gli ip secondari la soluzione sta nell'uso del ping esteso dove si puo' specificare l'ip del mittente.

Piu' IP nella stessa interfaccia indicano piu' reti IP sovrapposte di cui il router puo' esserne il gateway. Interessante e' che come conseguenza in alcuni casi i pacchetti entreranno e riusciranno dalla stessa interfaccia perche' gli indirizzi multipli indicano reti distinte. Nel caso in cui i pacchetti entrano ed escono dalla stessa interfaccia per la stessa rete (caso che si puo' avere nel caso di netmask errate) il router manda un ICMP redirect indicando che non ha senso entrare e uscire dalla stessa interfaccia per la stessa subnet, tanto vale andare direttamente. I 'redirect' che giovano per indicare dei percorsi ottimali in molti casi sono indice di cattiva configurazione di routing. I redirect non vengono inviati se si entra ed esce dalla stessa interfaccia ma per due reti distinte cioe' nel caso di ip multiplo.

Gli algoritmi di routing in genere supportano solo parzialmente gli IP secondari e in modo differente.

1.65 NAT, PAT e firewall

Da realizzare

1.7 Conclusioni

Storicamente, fino alla fine degli anni '80, i router interpretavano solo singoli ip, reti classful e routing di default. Le subnet furono utilizzate dalla fine degli anni ottanta. Verso il 1993 venne introdotto il supernetting col CIDR. Oggi e' possibile operare sulle route quasi completamente svincolati dalle restrizioni dovute ai vincoli di classe di indirizzo.

2.0 Il routing dinamico

Nel caso di piccole LAN/WAN la configurazione di IOS e' piuttosto semplice nella definizione delle route. Normalmente poche righe di statiche consentono di risolvere tutte le problematiche presenti.

Col crescere della rete l'idea di inserire in ogni configurazione solo route statiche non fa altro che aumentare i tempi per le operazioni di manutenzione e soprattutto non consente di gestire percorsi multipli. In caso di reti gia' di dimensione media, magari con struttura magliata e quindi con piu' percorsi alternativi, l'impossibilita' di scalare del routing statico e' una seria limitazione.

Un protocollo di routing consente la propagazione automatica delle route a tutti senza la necessita' dell'uso delle statiche, ovvero di comandi "*ip route <network>*

<mask> <nexthop>". Ogni router annuncia, con le specifiche dell'algoritmo di routing prescelto, i percorsi di routing di sua conoscenza provenienti normalmente da reti direttamente connesse, come le ethernet e le seriali.

Esistono due famiglie di algoritmi di routing: *distance-vector* e *link state*. I primi periodicamente inviano l'intera tabella di routing ai propri vicini e conservano solo informazioni di 'distanza' e 'vettore' cioe' metrica e next-hop. I secondi creano un grafo topologico della rete dal quale determinano i percorsi ottimali utilizzando algoritmi come *Dijkstra shortest path first (SPF)* nel caso di OSPF. Il metodo link state e' il piu' efficiente ma computazionalmente il piu' esigente. Sono distance-vector RIP, IGRP e BGP; OSPF e' link-state.

EIGRP ha funzionalita' di entrambe le famiglie il che ne fa un caso particolare. Si chiama "balanced hybrid protocol". Invece di usare SPF utilizza l'algoritmo DUAL (Diffuse Update Algorithm). Si tratta di un algoritmo che conserva informazioni riguardanti solo i vicini di un router (invece dell'intera rete come nei link-state) ma e' piu' leggero computazionalmente rispetto Dijkstra (SPF).

Gli algoritmi di routing si differenziano anche come *classful* e *classless*. I primi operano nei classful boundaries, cioe' non gestiscono le subnet (detti anche FLSM, fixed lenght subnet mask) e si tratta di RIP e IGRP. I secondi invece (detti anche VLSM cioe' variable lenght subnet mask) gestiscono pienamente le subnet quindi operano senza le limitazioni date dalle classi di appartenenza. Esempio sono EIGRP, OSPF e BGP. Questa differenza e' vitale per la corretta configurazione degli apparati di rete.

2.1 RIP

Il protocollo RIP e' il piu' datato tra i protocolli di routing dinamico oggi in uso in una rete IP. Le specifiche sono indicate in RFC 1058 datato Giugno 1988. In seguito, nel 1995, e' stato pubblicato l'RFC 1388 che specifica il successore di RIP, il RIPv2, che presenta una valida alternativa al RIPv1 in ambienti dove non si possono ignorare le subnet.

RIP e' un protocollo IGP, "Interior Gateway Protocol", ovvero pensato per essere usato in piccole parti di Internet o in WAN isolate da internet (dal punto di vista del protocollo di routing), ma non per connettere tra loro diversi AS di Internet (per i quali e' necessario un protocollo External di tipo EGP)

RIPv1 e' l'ideale per piccole reti WAN con indirizzamento IP privato e omogenee in termini di larghezza di banda dei link dati presenti. Anche se il piu' datato e' presente su molti apparati di rete anche di fascia bassa ed e' quindi molto diffuso.

RIP utilizza la porta 520 dell'UDP.

2.2 Configurazione

Partiamo subito da un esempio concreto. Consideriamo la seguente configurazione IOS:

- router rip
  network 116.0.0.0
  network 192.168.30.0

Nota: Introducendo 116.30.40.0 invece di 116.0.0.0 in automatico IOS inserisce 116.0.0.0 poiche' RIP e' classful e 116 e' una classe A.

La configurazione presentata e' completa e attiva il protocollo RIP per le due network indicate. Le due reti sono state scelte in quanto presenti sulle interfacce fisiche del router. Ad esempio la ethernet di questo router ha ip 116.30.40.1 mentre la seriale 192.168.30.2.

RIP e' cosi' attivo sulle interfacce definite col comando *"network"*. Si faccia attenzione al fatto che affinche' RIP sia operativo su un'interfaccia questa deve avere un indirizzo IP che ricade dentro un comando *"network"*. Se ad esempio una seriale S ha indirizzo N e la classe di N non viene indicata nella configurazione RIP allora non si accetteranno route in ingresso su S e non si faranno annunci RIP in uscita su S. Attraverso le altre interfacce verra' comunque annunciata la network N. Si ricordi che le informazioni inviate sono prive di dati relativi alla netmask (non supportata da RIPv1) ma comprensive di metrica. La metrica per RIP e' basata sull'hop-count e puo' variare tra 0 e 15 (con 16 la route viene scartata). L'hop-count rappresenta il numero di router da attraversare per raggiungere una rete. Poiche' il valore massimo e' 15 si evince che, in reti con diametro superiore a 15, non si puo' utilizzare RIP.

RIP manda l'intera tabella di routing ogni 30 secondi con broadcast 255.255.255.255 su tutte le interfacce su cui e' attivo. Se vi e' un cambiamento in una tabella di routing, ad esempio perche' abbiamo cambiato la classe IP di una interfaccia ethernet, questo si propaga subito, come nel caso della rete 192.168.30.0 che potete vedere sotto, indicata come FLASH-UPDATE (magari prima l'indirizzo era 192.168.20.X). Ecco l'output del comando "debug ip rip" attivo sul router con la configurazione di cui sopra:

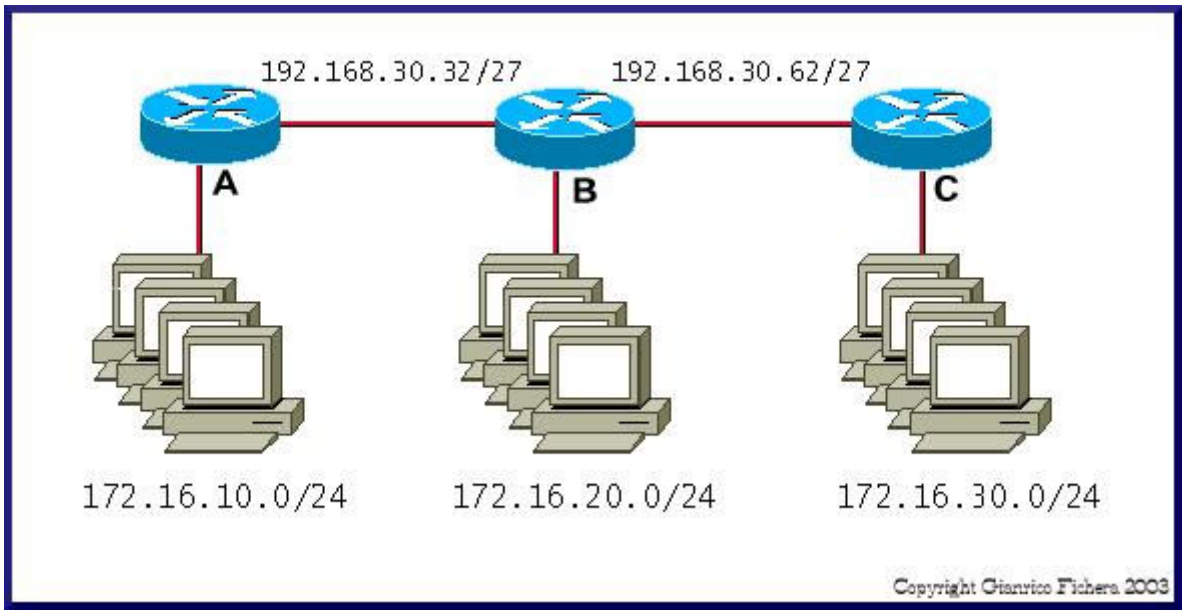
```
Router#
02:18:05: RIP: sending request on Ethernet0 to 255.255.255.255
02:18:07: RIP: sending v1 flash update to 255.255.255.255 via ATM0 (116.30.40.1)
02:18:07: RIP: build flash update entries
02:18:07: network 192.168.30.0 metric 1
02:18:07: RIP: sending v1 flash update to 255.255.255.255 via Ethernet0
(192.168.30.1)
02:18:07: RIP: build flash update entries - suppressing null update
02:18:11: RIP: sending v1 update to 255.255.255.255 via ATM0 (116.30.40.1)
02:18:11: RIP: build update entries
02:18:11: network 192.168.30.0 metric 1
02:18:11: RIP: sending v1 update to 255.255.255.255 via Ethernet0 (192.168.30.1)
02:18:11: RIP: build update entries
02:18:11: network 116.0.0.0 metric 1
02:18:39: RIP: sending v1 update to 255.255.255.255 via ATM0 (116.30.40.1)
```

```
02:18:39: RIP: build update entries
02:18:39: network 192.168.30.0 metric 1
02:18:39: RIP: sending v1 update to 255.255.255.255 via Ethernet0 (192.168.30.1)
02:18:39: RIP: build update entries
02:18:39: network 116.0.0.0 metric 1
02:19:07: RIP: sending v1 update to 255.255.255.255 via ATM0 (116.30.40.1)
02:19:07: RIP: build update entries
02:19:07: network 192.168.30.0 metric 1
02:19:07: RIP: sending v1 update to 255.255.255.255 via Ethernet0 (192.168.30.1)
02:19:07: RIP: build update entries
02:19:07: network 116.0.0.0 metric 1
```

vedete come, ogni 30 secondi (con piccole discrepanze dovute all'output del log) viene inviata la tabella di routing in broadcast sulle due interfacce presenti in questo router, ovvero ATM0 e Ethernet0. La periodicita' non e' valida per il "flash update".

2.3 Il problema delle subnet

Nell'esempio precedente abbiamo utilizzato una network di classe A e una di classe C. Questo non stupisce in quanto RIP e' un algoritmo di routing classful. Certo c'e' da domandarsi cosa si possa fare ai giorni nostri con un algoritmo di routing che lavori solo nei class boundaries visto che le reti attuali utilizzano fortemente il netmask spesso senza considerare la classificazione A, B, C, D. Probabilmente converrebbe usare solo RIPv2. Ma puo' non essere disponibile nei nostri apparati. In realta' con RIPv1 ci sono dei margini all'interno dei quali si possono usare le subnet. RIP consente di far uso di subnetting anche se con pesanti limitazioni. Tecnicamente diciamo che RIP e' un algoritmo FLSM (Fixed Lenght Subnet Mask) ovvero *consente di utilizzare subnetting ma mantenento la netmask fissa su tutta la WAN* (primo vincolo). Poi RIP e' un algoritmo che richiede la continuita' delle subnet nella rete ovvero *ogni router deve avere una subnet della classe che propaga, la quale deve proseguire nella direzione di propagazione*.



Osservate la figura. Supponiamo di avere tre router A e B e C collegati linearmente tramite seriali con indirizzi del tipo 192.168.30.0/27. Ogni router ha una interfaccia ethernet con indirizzi del tipo 172.16.0.0/24. Il router A inviera' e ricevera' informazioni di routing da B secondo i seguenti criteri:

In invio:

- Da ogni interfaccia con RIP attivo inviera' tutte le sottoreti conosciute della major network di appartenenza dell'interfaccia stessa;
- Da ogni interfaccia con RIP attivo inviera' solo le major network di appartenenza nel caso di major network differente (ovvero 172.16.0.0 invece di 172.16.10.0) ;
- Se una network da inviare ha stessa major network ma netmask differente viene scartata .

In ricezione:

- Se un aggiornamento e' della stessa major class della interfaccia da cui proviene gli da lo stesso netmask dell'interfaccia (nel caso di 192.168.30.62);
- Se un aggiornamento e' di major class differente lo summarizza alla major class MA se nella tabella di routing vi e' un'altra sottorete qualsiasi della stessa major class scarta l'aggiornamento (cosi' 172.16.0.0 proveniente da B viene scartata).

Cosi' *la configurazione in figura non funziona*. Il router A sara' in grado di raggiungere tutte le subnet di 192.168.30.0 ma non le subnet di 172.16.0.0 perche' per queste viene meno la *continuita'.* Per far funzionare una configurazione del genere avremmo dovuto usare tre classi B differenti sulle tre reti Ethernet.

Per dirla in breve conviene utilizzare in una WAN tutte sottoreti della stessa major network per le seriali. Se si utilizza per una ethernet una major network accertarsi di non usare altre sottoreti della stessa (con lo stesso o differente netmask) da

nessun'altra parte altrimenti non appena arrivera' l'aggiornamento questo verra' scartato. Se non si puo' evitare usare delle statiche. E' questo che si intende per *continuita' nel subnetting.* Tutto il discorso fatto vale per RIP v1. Il RIP v2 consente di evitare queste limitazioni in quanto non e' FLSM ma VLSM (Variable Lenght Subnet Mask) cioe' propaga anche le subnet masks.

Facciamo un esempio:

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

116.0.0.0/24 is subnetted, 1 subnets
C   116.30.40.0 is directly connected, ATM0
151.99.0.0/24 is subnetted, 2 subnets
C   151.99.100.0 is directly connected, Ethernet0
S   151.99.105.0 is directly connected, ATM0
```

ecco cosa si propaga:

```
02:44:12: RIP: sending v1 update to 255.255.255.255 via ATM0 (116.30.40.1)
02:44:12: RIP: build update entries
02:44:12: network 151.99.0.0 metric 1
02:44:12: RIP: sending v1 update to 255.255.255.255 via Ethernet0 (151.99.100.1)

02:44:12: RIP: build update entries
02:44:12: network 116.0.0.0 metric 1
02:44:12: subnet 151.99.105.0 metric 1
02:44:40: RIP: sending v1 update to 255.255.255.255 via ATM0 (116.30.40.1)
02:44:40: RIP: build update entries
02:44:40: network 151.99.0.0 metric 1
02:44:40: RIP: sending v1 update to 255.255.255.255 via Ethernet0 (151.99.100.1)

02:44:40: RIP: build update entries
02:44:40: network 116.0.0.0 metric 1
02:44:40: subnet 151.99.105.0 metric 1
```

Secondo le regole di cui sopra dall'interfaccia ethernet viene inviato l'update relativo alla subnet, fermo restando che la destinazione avra' netmask di classe B perche' la netmask non viene inviata.

2.4 Tempo di convergenza

RIP e' un algoritmo distance-vector ovvero propaga l'intera tabella di routing ai propri vicini. Per default ogni 30 secondi (broadcast time) un nodo RIP invia la propria tabella di routing ai suoi vicini. Supponiamo che il router R2 non riceva piu' aggiornamenti dal router R1. Invece di scartare subito le route, che R1 aveva inviato fino ai 30 secondi precedenti, R2 attende fino a 180 secondi conservando in tabella di routing le network di R1. Questo tempo si chiama "invalid time". Superati i 180 secondi si puo' affermare con sufficiente certezza che R1 e' nello stato di down e che le sue route quindi non vanno piu' utilizzate. A questo punto inizia il processo di cancellazione per queste route. La loro metrica viene posta a 16 (irraggiungibile) e vengono propagate con tale metrica per un periodo di altri 180 secondi. In questi 180 secondi le router sono nello stato di "hold-down" e figurano ancora nella tabella di routing come "possibly-down". Anche in questo stato la network e' utilizzata per il forward dei pacchetti in quanto e' nella tabella di routing.

Se un router R3 riceve questo aggiornamento da R2 mette la network direttamente nello stato di "hold-down".

Superato il tempo per l'hold-down la route viene scartata dalla tabella di routing. Questo non avviene subito ma considerando il tempo indicato nel "flush-time" che vale 240 secondi di default. Durante l'hold-down le informazioni riguardanti nuovi percorsi per raggiungere la rete provenienti da altri router vengono scartate (per evitare possibili routing loops). Il tempo di "flush-time" e' ottenuto dalla somma del tempo di hold-down piu' un periodo di tempo che in questo caso e' 60 secondi.

Possiamo quindi osservare che in caso di guasto nella WAN il percorso alternativo viene attivato dopo oltre 7 minuti. Non e' sicuramente un tempo breve ma e' motivato dalla necessita' di evitare i routing loops. Questo si chiama "tempo di convergenza" cioe' il tempo necessario affinche' tutte le tabelle di routing di tutti i router si aggiornino dopo un cambiamento di topologia (ad esempio dovuto a un guasto in un link)

Se abbiamo sufficiente banda possiamo ridurre tale tempo aumentando la frequenza degli aggiornamenti RIP. Tramite il comando "timers basic" e' possibile variare i timer RIP:

timers basic *update invalid holddown flush*

2.5 Split-horizont e poison-reverse

Insieme ai timers sono i metodi per evitare il problema dei routing loop. Per default se RIP riceve una route da una interfaccia S non invia verso S l'aggiornamento relativo alla network ricevuta, e questo e' split-horizont. Questa funzionalita' e' per default disabilitata nel caso di collegamenti come Frame Relay o ATM. In questi casi infatti una interfaccia potrebbe portare verso differenti network (tramite VC o DLCI) e lo split-horizon renderebbe incompleta la propagazione di tali informazioni. E' possibile attivare o disattivare lo split-horizon col comando "*ip split-horizont*". RIP utilizza split-horizon con poison reverse ovvero invece di non ripropagare verso il mittente di una route la invia con metrica 16.

2.6 Redistribuzione RIP

La redistribuzione, ovvero l'iniezione di routes RIP in altri protocolli di routing, non viene trattata in questo articolo. Basti sapere che in caso di redistribuzione in altri algoritmi di routing bisogna specificare una metrica altrimenti questa sara' 16.

2.7 Monitoraggio

Col comando "show ip protocols" si possono vedere tutti i settaggi RIP:

```
Router#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 15 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is
Incoming update filter list for all interfaces is
Redistributing: rip
Default version control: send version 1, receive any version
Interface Send Recv Triggered RIP Key-chain
ATM0 1 1 2
```

```
Ethernet0 1 1 2
Automatic network summarization is in effect
Routing for Networks:
116.0.0.0
151.99.0.0
Routing Information Sources:
Gateway Distance Last Update
Distance: (default is 120)
```

Col comando "debug" e' possibile visualizzare in tempo reale i dati RIP in ingresso e uscita dalle interfacce:

```
router#debug ip rip ?
database   RIP database events
events     RIP protocol events
trigger    RIP trigger extension
```

2.8 Route di default

Il tipico comando utilizzato per la route di default e' il seguente:

ip route 0.0.0.0 0.0.0.0 A.B.C.D

Questo viene propagato senza problemi dal RIP ma attenzione, dalle versioni 12.0T dell'IOS il RIP non lo rileva e bisogna utilizzare il comando "default-information originate". Ecco cosa succede con queste modifiche al nostro esempio, dove il router ha una versione di IOS superiore alla 12.0T:

```
router rip
  network 116.0.0.0
  network 151.99.0.0
  default-information originate
```

Notate che i router vicini ricevono adesso la riga di default che appare in tabella di routing con la notazione "S*"

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

116.0.0.0/24 is subnetted, 1 subnets
C 116.30.40.0 is directly connected, ATM0
151.117.0.0/24 is subnetted, 1 subnets
S 151.117.6.0 [1/0] via 192.168.30.2
151.99.0.0/24 is subnetted, 2 subnets
C 151.99.100.0 is directly connected, Ethernet0
```

```
S 151.99.105.0 is directly connected, ATM0
S* 0.0.0.0/0 is directly connected, ATM0

02:53:04: RIP: sending v1 update to 255.255.255.255 via ATM0 (116.30.40.1)
02:53:04: RIP: build update entries
02:53:04: network 151.99.0.0 metric 1
02:53:04: RIP: sending v1 update to 255.255.255.255 via Ethernet0 (151.99.100.1)

02:53:04: RIP: build update entries
02:53:04: subnet 0.0.0.0 metric 1
02:53:04: network 116.0.0.0 metric 1
02:53:04: subnet 151.99.105.0 metric 1
```

2.9 Esercitazione RIP

L'esercitazione si basa su una configurazione di laboratorio con due router Cisco collegati tra loro tramite la seriale e cavo V.35 in back-to-back. Ogni router Cisco ha una sua ethernet con un suo switch.
Qui ci focalizziamo su uno dei due router essendone la configurazione del secondo l'esatto simmetrico.

Ecco la condizione iniziale:
***Il router 'remoto'** ha una ethernet 192.168.0.1/24 e una seriale 192.168.40.2/30*
***Il router 'locale'** ha una ethernet 192.168.30.1/24 e una seriale 192.168.40.1/30*

Sia assegnata questa configurazione:

```
interface FastEthernet0/0
 ip address 192.168.30.1 255.255.255.0
 no ip directed-broadcast
 duplex auto
 speed auto
!
interface Serial0/0
 ip address 192.168.40.1 255.255.255.252
 no ip directed-broadcast
 no ip mroute-cache
!
router rip
 network 192.168.30.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/0
no ip http server


locale#
locale#
00:20:48: RIP: sending v1 update to 255.255.255.255 via FastEthernet0/0 (192.168
.30.1)
00:20:48: RIP: build update entries - suppressing null update
locale#
locale#
```

come si puo' il comando network definisce non cosa propagare ma bensi' dove. Infatti con questa configurazione nulla viene inviato attraverso la seriale 0/0 al router remoto. E, importante, gli aggiornamenti RIP non sono inviati verso la seriale0/0:

```
00:23:36: RIP: ignored v1 packet from 192.168.40.2 (not enabled on Serial0/0)
remoto#      .
```

La configurazione si corregge aggiungendo la network della seriale:

```
router rip
 network 192.168.30.0
 network 192.168.40.0
```

subito il rip si attiva. Notate i "flash update":

```
00:24:51: RIP: sending request on FastEthernet0/0 to 255.255.255.255
00:24:51: RIP: sending request on Serial0/0 to 255.255.255.255
00:24:53: RIP: sending v1 flash update to 255.255.255.255 via FastEthernet0/0 (1
92.168.30.1)
00:24:53: RIP: build flash update entries
00:24:53:       network 192.168.40.0 metric 1
00:24:53: RIP: sending v1 flash update to 255.255.255.255 via Serial0/0 (192.168
.40.1)
00:24:53: RIP: build flash update entries
00:24:53:       network 192.168.30.0 metric 1
```

dopo i 30 secondi l'intera tabella di routing viene ripropagata (tranne la default):

```
00:25:24: RIP: sending v1 update to 255.255.255.255 via FastEthernet0/0 (192.168
.30.1)
00:25:24: RIP: build update entries
00:25:24:       network 192.168.40.0 metric 1
00:25:24: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (192.168.40.1)
00:25:24: RIP: build update entries
00:25:24:       network 192.168.30.0 metric 1
```

agendo su entrambi i router la tabella di routing si completa ed e’ quella finale corretta:

```
locale#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0
```

```
C    192.168.30.0/24 is directly connected, FastEthernet0/0
     192.168.40.0/30 is subnetted, 1 subnets
C       192.168.40.0 is directly connected, Serial0/0
R    192.168.0.0/24 [120/1] via 192.168.40.2, 00:00:12, Serial0/0
S*   0.0.0.0/0 is directly connected, Serial0/0
locale#
```

Osserviamo, dal debug, questa riga:

```
00:32:44: RIP: sending v1 update to 255.255.255.255 via FastEthernet0/0 (192.168
.30.1)
```

se non vogliamo mandare gli update sulla fasteth0/0, ad esempio perche' in essa non vi sono altri router con RIP:

```
router rip
 passive-interface FastEthernet0/0
 network 192.168.30.0
 network 192.168.40.0
```

adesso:

```
00:35:18: RIP: received v1 update from 192.168.40.2 on Serial0/0
00:35:18:      192.168.0.0 in 1 hops
00:35:32: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (192.168.40.1)
00:35:32: RIP: build update entries
00:35:32:       network 192.168.30.0 metric 1
```

Scolleghiamo l'interfaccia ethernet dal router remoto per simulare un guasto:

```
00:37:20: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (192.168.40.1)
00:37:20: RIP: build update entries
00:37:20:       network 192.168.30.0 metric 1
00:37:23: RIP: received v1 update from 192.168.40.2 on Serial0/0
00:37:23:      192.168.0.0 in 16 hops (inaccessible)
00:37:25: RIP: sending v1 flash update to 255.255.255.255 via Serial0/0 (192.168
.40.1)
00:37:25: RIP: build flash update entries
00:37:25:       network 192.168.0.0 metric 16
```

```
locale#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C    192.168.30.0/24 is directly connected, FastEthernet0/0
     192.168.40.0/30 is subnetted, 1 subnets
C       192.168.40.0 is directly connected, Serial0/0
S*   0.0.0.0/0 is directly connected, Serial0/0

locale#sh ip rip database
192.168.0.0/24 is possibly down
192.168.30.0/24    auto-summary
192.168.30.0/24    directly connected, FastEthernet0/0
192.168.40.0/24    auto-summary
192.168.40.0/30    directly connected, Serial0/0
```

Dopo qualche secondo sparisce dal database. Praticamente subito dalla tabella di routing. Il tutto avviene entro pochi secondi, sia come cancellazione che come ripristino. Riproviamo ma col "debug ip rip database" stavolta:

```
locale#sh ip rip database
192.168.0.0/24 is possibly down
192.168.30.0/24    auto-summary
192.168.30.0/24    directly connected, FastEthernet0/0
192.168.40.0/24    auto-summary
192.168.40.0/30    directly connected, Serial0/0

00:41:03: RIP-DB: garbage collect 192.168.0.0/24

locale#sh ip rip database
192.168.30.0/24    auto-summary
192.168.30.0/24    directly connected, FastEthernet0/0
192.168.40.0/24    auto-summary
192.168.40.0/30    directly connected, Serial0/0
```

Alcuni dati sul rip che stiamo testando:

```
locale#sh ip protocol
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 17 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface         Send  Recv  Triggered RIP  Key-chain
    Serial0/0         1     1 2
  Automatic network summarization is in effect
```

```
  Routing for Networks:
    192.168.30.0
    192.168.40.0
  Passive Interface(s):
    FastEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.40.2         120      00:00:54
  Distance: (default is 120)
```

Simuliamo un secondo guasto sulla rete. Il router remoto non invia piu' aggiornamenti rip per malfunzionamento:

```
locale#
locale#
locale#
locale#
locale#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C    192.168.30.0/24 is directly connected, FastEthernet0/0
     192.168.40.0/30 is subnetted, 1 subnets
C       192.168.40.0 is directly connected, Serial0/0
R    192.168.0.0/24 [120/1] via 192.168.40.2, 00:00:20, Serial0/0
S*   0.0.0.0/0 is directly connected, Serial0/0
locale#
locale#
locale#sh ip rip database
192.168.0.0/24    auto-summary
192.168.0.0/24
    [1] via 192.168.40.2, 00:00:25, Serial0/0
192.168.30.0/24    auto-summary
192.168.30.0/24    directly connected, FastEthernet0/0
192.168.40.0/24    auto-summary
192.168.40.0/30    directly connected, Serial0/0
locale#
locale#
locale#
```

la route rimane anche nella tabella di route. Dopo tre minuti (non fase caso all'imprecisione dei secondi dovuta alle condizioni di test) scade l'hold-down e parte il poison:

```
locale#

...snip...

locale#
01:05:41: RIP-DB: flush route of 192.168.0.0/24 via 192.168.40.2
01:05:41: RIP-DB: Remove 192.168.0.0/24, (metric 4294967295) via 192.168.40.2, S
erial0/0
01:05:41: RIP-DB: hold down 192.168.0.0/24
01:05:41: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (192.168.40.1)
```

```
01:05:41: RIP: build update entries
01:05:41:       network 192.168.0.0 metric 16
01:05:41:       network 192.168.30.0 metric 1
01:05:43: RIP: sending v1 flash update to 255.255.255.255 via Serial0/0 (192.168
.40.1)
01:05:43: RIP: build flash update entries
01:05:43:       network 192.168.0.0 metric 16
locale#
```

adesso la route non verra' piu' usata ma e' nel database RIP nello stato di possibly down:

```
locale#
locale#sh ip rip database
192.168.0.0/24 is possibly down
192.168.0.0/24 is possibly down
192.168.30.0/24    auto-summary
192.168.30.0/24    directly connected, FastEthernet0/0
192.168.40.0/24    auto-summary
192.168.40.0/30    directly connected, Serial0/0
locale#
locale#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C    192.168.30.0/24 is directly connected, FastEthernet0/0
     192.168.40.0/30 is subnetted, 1 subnets
C       192.168.40.0 is directly connected, Serial0/0
S*   0.0.0.0/0 is directly connected, Serial0/0


01:06:08: RIP-DB: garbage collect 192.168.0.0/24
01:06:08: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (192.168.40.1)
01:06:08: RIP: build update entries
01:06:08:       network 192.168.30.0 metric 1
locale#
locale#
```

dopo all'incirca altri 60 secondi dai tre minuti (e siamo a 180+60=240) la route viene rimossa dal database (non fate caso al tempo qui sopra di 01:06:08 che e' relativo alla summary):

```
01:06:34: RIP-DB: garbage collect 192.168.0.0/24
01:06:34: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (192.168.40.1)
```

```
01:06:34: RIP: build update entries
01:06:34:       network 192.168.30.0 metric 1

locale#

locale#sh ip rip database
192.168.30.0/24    auto-summary
192.168.30.0/24    directly connected, FastEthernet0/0
192.168.40.0/24    auto-summary
192.168.40.0/30    directly connected, Serial0/0
locale#
```

Continuita' della subnet. Supponiamo che la ethernet del router remoto diventi 192.168.40.5/30. Vediamo che succede:

```
locale#
01:21:00: RIP: sending request on Serial0/0 to 255.255.255.255
01:21:00: RIP: received v1 update from 192.168.40.2 on Serial0/0
01:21:00:      192.168.40.4 in 1 hops
01:21:02: RIP: sending v1 flash update to 255.255.255.255 via Serial0/0 (192.168
.40.1)
01:21:02: RIP: build flash update entries
01:21:02:       network 192.168.30.0 metric 1
locale#
01:21:06: RIP: received v1 update from 192.168.40.2 on Serial0/0
01:21:06:      192.168.40.4 in 1 hops
locale#
locale#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C    192.168.30.0/24 is directly connected, FastEthernet0/0
     192.168.40.0/30 is subnetted, 2 subnets
C       192.168.40.0 is directly connected, Serial0/0
R       192.168.40.4 [120/1] via 192.168.40.2, 00:00:05, Serial0/0
S*   0.0.0.0/0 is directly connected, Serial0/0
locale#sh ip route 192.168.40.5
Routing entry for 192.168.40.4/30
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 192.168.40.2 on Serial0/0, 00:00:10 ago
  Routing Descriptor Blocks:
  * 192.168.40.2, from 192.168.40.2, 00:00:10 ago, via Serial0/0
      Route metric is 1, traffic share count is 1

locale#sh ip rip database
192.168.30.0/24    auto-summary
192.168.30.0/24    directly connected, FastEthernet0/0
192.168.40.0/24    auto-summary
```

```
192.168.40.0/30    directly connected, Serial0/0
192.168.40.4/30
    [1] via 192.168.40.2, 00:00:08, Serial0/0
```

mettiamo ora nel router remoto 192.168.50.1/25 e vediamo cosa succede:

```
locale#sh ip route 192.168.50.1
Routing entry for 192.168.50.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 192.168.40.2 on Serial0/0, 00:00:08 ago
  Routing Descriptor Blocks:
  * 192.168.40.2, from 192.168.40.2, 00:00:08 ago, via Serial0/0
      Route metric is 1, traffic share count is 1

locale#sh ip rip database
192.168.30.0/24    auto-summary
192.168.30.0/24    directly connected, FastEthernet0/0
192.168.40.0/24    auto-summary
192.168.40.0/30    directly connected, Serial0/0
192.168.50.0/24    auto-summary
192.168.50.0/24
    [1] via 192.168.40.2, 00:00:11, Serial0/0
locale#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C    192.168.30.0/24 is directly connected, FastEthernet0/0
     192.168.40.0/30 is subnetted, 1 subnets
C       192.168.40.0 is directly connected, Serial0/0
R    192.168.50.0/24 [120/1] via 192.168.40.2, 00:00:14, Serial0/0
S*   0.0.0.0/0 is directly connected, Serial0/0
```

notate che mentre nel primo esempio la netmask e' stata propagata nel secondo no e questo perche' nel secondo esempio viene meno la continuita' della subnet

Attiviamo ora ripv2 sulla configurazione di sopra (quella col 192.168.50.1/25):

```
01:32:45: RIP: sending request on Serial0/0 to 224.0.0.9
```

```
01:32:45: RIP: received v2 update from 192.168.40.2 on Serial0/0
01:32:45:      192.168.50.0/25 via 0.0.0.0 in 1 hops
01:32:46: RIP: received v2 update from 192.168.40.2 on Serial0/0
01:32:46:      192.168.50.0/25 via 0.0.0.0 in 1 hops
01:32:47: RIP: sending v2 flash update to 224.0.0.9 via Serial0/0 (192.168.40.1)
01:32:47: RIP: build flash update entries
01:32:47:       192.168.30.0/24 via 0.0.0.0, metric 1, tag 0
locale#
01:32:52: RIP: sending v2 update to 224.0.0.9 via Serial0/0 (192.168.40.1)
01:32:52: RIP: build update entries
01:32:52:       192.168.30.0/24 via 0.0.0.0, metric 1, tag 0

locale#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C    192.168.30.0/24 is directly connected, FastEthernet0/0
     192.168.40.0/30 is subnetted, 1 subnets
C       192.168.40.0 is directly connected, Serial0/0
     192.168.50.0/25 is subnetted, 1 subnets
R       192.168.50.0 [120/1] via 192.168.40.2, 00:00:15, Serial0/0
S*   0.0.0.0/0 is directly connected, Serial0/0

locale#sh ip rip database
192.168.30.0/24    auto-summary
192.168.30.0/24    directly connected, FastEthernet0/0
192.168.40.0/24    auto-summary
192.168.40.0/30    directly connected, Serial0/0
192.168.50.0/24    auto-summary
192.168.50.0/25
    [1] via 192.168.40.2, 00:00:01, Serial0/0
```

attenzione: per default e' attivo l'auto-summary. Mettere "no auto-summary" per vedere la propagazione della /25

Ed ecco un riepilogo di quanto messo a disposizione da ripv2:

```
remoto(config)#router rip
remoto(config-router)#?
Router configuration commands:
  address-family          Enter Address Family command mode
  auto-summary            Enable automatic network number summarization
  default                 Set a command to its defaults
  default-information     Control distribution of default information
  default-metric          Set metric of redistributed routes
  distance                Define an administrative distance
  distribute-list         Filter networks in routing updates
  exit                    Exit from routing protocol configuration mode
  flash-update-threshold  Specify flash update threshold in second
  help                    Description of the interactive help system
```

```
  maximum-paths             Forward packets over multiple paths
  neighbor                  Specify a neighbor router
  network                   Enable routing on an IP network
  no                        Negate a command or set its defaults
  offset-list               Add or subtract offset from IGRP or RIP metrics
  output-delay              Interpacket delay for RIP updates
  passive-interface         Suppress routing updates on an interface
  redistribute              Redistribute information from another routing
                            protocol
  timers                    Adjust routing timers
  traffic-share             How to compute traffic share over alternate paths
  validate-update-source    Perform sanity checks against source address of
                            routing updates
  version                   Set routing protocol version
```

3.0 Configurazione IGRP

IGRP e' FLSM e opera seguendo gli stessi principi di RIP. Quanto detto per RIP per la propagazione delle subnet continua a valere. IGRP e' distance-vector. Le differenze di IGRP rispetto RIP sono fondamentalmente nell'algoritmo di calcolo della metrica e nella gestione della stessa. IGRP infatti calcola la metrica utilizzando la somma dei delay e delle larghezze di banda presenti per raggiungere una network remota. Delay e bandwidth sono parametri introdotti manualmente nella configurazione con i comandi "delay" e "bandwidth" e rappresentano il ritardo di propagazione dei pacchetti e la larghezza di banda a disposizione. E' possibile anche l'utilizzo di una metrica composta piu' estesa comprendente "reliability", "load" e "MTU".

Per intenderci IGRP e' molto piu' efficiente di RIP nel calcolare i percorsi migliori (non a caso ha una AD di 100 contro i 120 di RIP) infatti RIP darebbe priorità anche a percorsi con pochi hop ma congestionati o con poca banda disponibile.

```
routing igrp 12
  network 172.18.0.0
  network 192.168.30.0
  network 10.0.0.0
```

Si faccia attenzione al fatto che affinche' IGRP sia operativo su un'interfaccia questa deve avere un indirizzo IP che ricade dentro un comando "*network*". In caso contrario non si accetteranno route in ingresso su una interfaccia e non si propaghera' nulla attraverso quell'interfaccia.

Una route IGRP con metrica –1 viene scartata. In caso di redistribuzione bisogna indicare la metrica altrimenti sara' –1. Quando si redistribuisce IGRP in OSPF di default sara' assegnata una metrica pari a 20 di default.

3.1 Tempo di convergenza

IGRP propaga la tabella di routing ogni 90 secondi (broadcast time). Se per tre volte questo tempo (invalid time pari a 270 sec.) non riceve un aggiornamento per una route la dichiara "possibly down". Quindi attende un periodo di 280 secondi (holddown time pari per default a tre volte invalid time piu' 10) nel quale non accetta nessun aggiornamento riguardante la route "possibly down". Superato anche questo intervallo di tempo la route viene cancellata dalla tabella di routing dopo altri 350 secondi. Complessivamente il tempo e' 10 minuti e 30 secondi (flush time pari a 7 volte il broadcast time). Durante l'holddown la route viene annunciata con metrica infinita pari a -1.

Complessivamente si tratta di oltre 10 minuti, un tempo non indifferente motivato dalla necessita' di evitare routing loops. Per aumentare l'efficienza IGRP utilizza come RIP i "flash updates" che consistono nell'inviare un aggiornamento dovuto a un cambiamento di stato di un'interfaccia (e quindi di una route) non appena questo avviene invece di completare un ciclo di 90 secondi. Questo funziona bene nel caso di modifiche agli IP nella nostra WAN ma non nel caso di caduta di link.

Una rete IGRP puo' avere diametro massimo pari a 255 ma per default e' settato a 100. Se si ha un hop-count superiore al diametro massimo le route non si propagano. I parametri per il calcolo della metrica: "bandwidth" e "delay", sono valori inseriti manualmente dall'operatore in ogni router per ogni link.

3.2 Split-horizont e poison-reverse

Per evitare i routing loops come per il RIP si utilizza split-horizon. Split-horizon previene routing loop tra router adiacenti. Per evitare la possibilita' di loop coinvolgenti un gran numero di router, detti grandi router loop, si utilizza il route poisoning. L'idea consiste nel controllare il tasso di crescita nell'aumento di metrica delle route. Un tasso eccessivo puo' indicare la presenza di un loop e quindi si mette in hold-down la route che ne e' coinvolta. Il fattore consigliato di crescita e' 1.1. Una route in hold-down verra' annunciata con metrica infinita (pari a -1).

3.3 Route di default

Il tipico comando utilizzato per la route di default e' il seguente:

ip route 0.0.0.0 0.0.0.0 A.B.C.D

IGRP non propaga di default questa route. Affinche' venga propagata la route di default bisogna utilizzare il comando "ip default-network A.B.C.D".

4.0 Configurazione EIGRP

Gli algoritmi distance-vector come RIP e IGRP prendono la loro denominazione dal fatto che ogni scelta di routing e' basata su un valore di distanza (che prende il nome di metrica) e dal fatto che il next-hop e' dato da un vettore. Tali algoritmi prendono le loro decisioni esclusivamente in base a quanto presente nella tabella di routing in quanto non conservano informazioni aggiuntive. Tutto cio' che proviene dai vicini e che non viene inserito nella tabella di routing non viene utilizzato (ad eccezione di RIPV2 dalla 12.0).

EIGRP si basa proprio su queste informazioni aggiuntive per migliorare il proprio tempo di convergenza. RIP e IGRP non possono evitare routing loop se non con hold-down e flush timers. Ricordiamo che cio' porta, di default, ad un tempo di convergenza di circa 10,5 minuti per IGRP e un po' meno per RIP. EIGRP conserva informazioni sulla topologia della rete fino ai router direttamente connessi. Questo vedremo che consente di evitare routing loops senza l'uso di hold-down e flush-timers. EIGRP si basa su un algoritmo chiamato DUAL.

Gli algoritmi link-state come EIGRP e OSPF non propagano l'intera tabella di routing ai loro vicini ma soltanto le sue variazioni. Pertanto e' necessario un nuovo metodo per poter stabilire quando un vicino non e' piu' raggiungibile (con RIP e IGRP basta verificare l'arrivo periodico la tabella di routing dai vicini). La soluzione consiste nell'utilizzo di speciali pacchetti, detti di HELLO, periodicamente inviati per segnalare il proprio stato di on-line (e altre informazioni aggiuntive).

EIGRP invia i pacchetti di HELLO ogni 5 secondi sui collegamenti piu' affidabili e a velocita' maggiori (come ethernet, token ring, NBMA superiori a T1), ogni 60 secondi per reti meno affidabili o a velocita' minore (ISDN, NBMA minori o uguali a T1). I pacchetti di HELLO sono inviati al multicast 224.0.0.10. Questo tempo si chiama "hello-interval" ed e' modificabile col comando "*ip hello-interval eigrp* NN". Dopo tre volte questo tempo, valore di hold-time, se non si ricevono HELLO il vicino sara' considerato DOWN. Questo valore e' modificabile con il comando "*ip hold-time eigrp NN*".

Una volta che la sessione tra due router e' attiva ognuno dei due conservera' tutte le informazioni inviate dal vicino. Queste gli consentono di determinare un valore di metrica, chiamato "REPORTED-DISTANCE", che indica la metrica con cui il router mittente raggiunge una certa destinazione. A questa il router affianca sempre la "FEASIBLE-DISTANCE" ovvero la metrica che lui adotta per raggiungere una determinata destinazione. Al momento in cui una destinazione non e' piu' raggiungibile un router RTA determinera' il nuovo percorso dal confronto tra REPORTED e FEASIBLE. Se FEASIBLE < REPORTED allora utilizza il percorso in suo possesso. Altrimenti RTA entra nello stato di ACTIVE e interroga i router adiacenti con una QUERY chiedendo per percorsi alternativi che gli consentano di soddisfare la regola di cui sopra. Il router vicino, diciamo RTB, dara' una risposta se possiede tale percorso altrimenti girera' la QUERY a monte. Si inviano percio' una serie di richieste

di QUERY in cascata che, se senza risposta affermativa, fara' si che RTA dichiarera' la rete remota come irraggiungile. Se per qualche ragione RTA non riceve risposta alla sua QUERY entrera' nello stato di SIA (Stuck in Active). Per sbloccare la situazione, dopo un certo tempo, 3 minuti di default, (modificabile col comando "*timers active-time NN*") si considerera' la rete remota irraggiungibile.

Grazie a questo meccanismo EIGRP evita routing loops ed ha un basso tempo di convergenza. Poiche' gestisce la VLSM e' l'alternativa naturale a RIP e IGRP.

Quando EIGRP ha piu' path per una medesima destinazione fa bilanciamento di carico di default. E' possibile fare bilanciamento di carico anche tra path con metriche differenti. Dati ad esempio tre percorsi A, B, C per la medesima destinazione e con metrica 100, 200, 300, col comando "*variance 3*" io affermo che i path con metrica da quella piu' bassa alla stessa moltiplicata per tre devono partecipare ad un load-balancing. In questo caso con "*variance 3*" i tre path 100, 200, 300 parteciperanno. La distribuzione dei pacchetti tra i tre link sara' calcolata dividendo la metrica piu' alta, che si indica con MAXMET, per i valori di metrica dei singoli path. Questa forma di bilanciamento quindi NON consiste nel mandare a ruota un pacchetto su ogni link (non e' round-robin).

Nel caso di collegamenti punto-multipunto EIGRP considera come metrica del singolo VC la metrica sull' interfaccia divisa per il numero di virtual-link nella stessa. Si consiglia allora di dare col comando "*bandwidth*" il valore pari al CIR (nel caso di FR) piu' basso tra quelli forniti dall'operatore TELCO per i VC. Questo evita un uso eccessivo di banda da parte di EIGRP che utilizza fino ad un max del 50% della banda disponibile per i suoi pacchetti di protocollo. Tale valore e' settabile col comando "*ip bandwidth-percent eigrp ASNUMBER VALORE*".

4.1 Redistribuzione

Per quanto riguarda la redistribuzione qualsiasi routing proveniente da altri algoritmi (RIP, IGRP etc.) per default entra in EIGRP con una metrica pari a -1. Cio' rende obbligatorio l'uso del comando "*default-metric BANDWIDTH(in k) DELAY RELIABILITY LOAD MTU*" . La distanza amministrativa e' posta invece a 170 per le redistribuzioni.

4.2 Compattamento delle route

EIGRP fa summarization in automatico. E' possibile disabilitare la summarization con il comando "*no auto-summary*". E' possibile fare manual summarization con "*ip summary-address eigrp AS NETWORK NETMASK*". Nell'ipotesi in cui questa feature non sia voluta, come quando vi e' discontinuita' nell'uso delle subnet in una rete, basta inserire il comando 'no auto-summary' nella propria configurazione di EIGRP.

4.3 Configurazione

La configurazione di base e come RIP e IGRP:

```
router eigrp 10
network 116.0.0.0
network 151.99.0.0
```

Con "show runn" vediamo:

```
router eigrp 10
network 116.0.0.0
network 151.99.0.0
auto-summary
no eigrp log-neighbor-changes
```

poiche' l'auto summary e configurato per default. Cosi' la tabella di routing originaria:

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

116.0.0.0/24 is subnetted, 1 subnets
C 116.30.40.0 is directly connected, ATM0
151.99.0.0/24 is subnetted, 2 subnets
C 151.99.100.0 is directly connected, Ethernet0
S 151.99.105.0 is directly connected, ATM0
```

diventa, dopo l'introduzione di EIGRP:

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

116.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 116.30.40.0/24 is directly connected, ATM0
D 116.0.0.0/8 is a summary, 00:00:03, Null0
151.99.0.0/16 is variably subnetted, 3 subnets, 2 masks
C 151.99.100.0/24 is directly connected, Ethernet0
S 151.99.105.0/24 is directly connected, ATM0
D 151.99.0.0/16 is a summary, 00:00:03, Null0
```

4.4 Route di default

Il tipico comando utilizzato per la route di default e' il seguente:

ip route 0.0.0.0 0.0.0.0 A.B.C.D

EIGRP propaga in automatico la route di default 0.0.0.0 con la redistribution. Se si vuole propagare una rete di defalt differente bisogna specificarla col comando "*ip default-network NETWORK*". Questo a differenza di IGRP che vuole sempre "*ip*

default-network ...".

gianrico.fichera@itesys.it

4.5 Conclusione

Per concludere una serie di brevi su EIGRP:

- EIGRP funziona solo per gli ip primari di un collegamento. Se ci sono indirizzi secondari su un'interfaccia non saranno usati per identificare un vicino;

- EIGRP usa il numero di protocollo IP 88;

- Per default l'hop-count massimo di EIGRP e' posto a 100, e' settabile a max 255;

- EIGRP ha distanza amministrativa pari a 90;

- La metrica di EIGRP e 256 volte quella di IGRP. Se in un router girano entrambi gli algoritmi la redistribuzione tra loro si ha di default.

5.0 Laboratorio II – RIP, IGRP, EIGRP

Nell'esempio a seguire due router, "itesys1" e "itesys2" sono collegati tra loro tramite WAN. Il router "itesys2" ha accesso ad internet tramite un gateway in LAN, con IP 192.168.0.2. L'esempio mostra come propagano questa route di default RIP, IGRP e EIGRP

Ecco il router "itesys2" con una corretta e funzionante configurazione con RIP:

```
... snip ...

!

interface FastEthernet0/0
ip address 192.168.0.2 255.255.255.0
no ip directed-broadcast
duplex auto
speed auto
!
interface Serial0/0
ip address 128.10.10.2 255.255.255.252
no ip directed-broadcast
no ip mroute-cache
no fair-queue
!
interface FastEthernet0/1
no ip address
no ip directed-broadcast
shutdown
```


gianrico.fichera@itesys.it

```
duplex auto
speed auto
!
router rip
network 128.10.0.0
network 192.168.0.0
default-information originate
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.0.1
```

Sovrapponiamo al RIP di "itesys2" una configurazione IGRP:

```
itesys2#conf t
itesys2(config)#router igrp 10
itesys2(config-router)#network 128.10.10.0
itesys2(config-router)#network 192.168.190.0
```

Vediamo cosa succede al router adiacente (che ha sia RIP che IGRP):

```
itesys1#sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 128.10.10.2 to network 0.0.0.0

C 192.168.94.0/24 is directly connected, FastEthernet0/0
128.10.0.0/30 is subnetted, 1 subnets
C 128.10.10.0 is directly connected, Serial0/0
I 192.168.0.0/24 [100/8486] via 128.10.10.2, 00:00:00, Serial0/0
R* 0.0.0.0/0 [120/1] via 128.10.10.2, 00:00:25, Serial0/0
```

IGRP non propaga la route di default cosi' non la sovrascrive. Le altre route RIP sono sovrascritte a causa della minore AD di IRGP. Adesso togliamo il RIP. Ad 'itesys2' facciamo propagare tramite IGRP la route di default. "default-information originate" non funziona cosi' come non funziona "redistribute static" o "ip default-gateway". E' necessario l'uso di "ip default-network":

```
itesys2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
itesys2(config)#ip default-network ?
A.B.C.D IP address of default network
itesys2(config)#ip default-network 192.168.0.0
```

```
itesys2(config)#exit

itesys2#sh run
Building configuration...
...snip...
!
router igrp 10
network 128.10.0.0
network 192.168.0.0
!
ip classless
ip default-network 192.168.0.0
ip route 0.0.0.0 0.0.0.0 192.168.0.1
ip route 192.168.0.0 255.255.255.0 192.168.0.1

...snip...

itesys2#sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 192.168.0.1 to network 0.0.0.0

I 192.168.94.0/24 [100/8486] via 128.10.10.1, 00:00:05, Serial0/0
128.10.0.0/30 is subnetted, 1 subnets
C 128.10.10.0 is directly connected, Serial0/0
C* 192.168.0.0/24 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 192.168.0.1
```

Notate come la Fast0/0 abbia adesso C* nella tabella di routing. Questa informazione sara' propagata da IGRP:

```
itesys1#sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 128.10.10.2 to network 192.168.0.0

C 192.168.94.0/24 is directly connected, FastEthernet0/0
128.10.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 128.10.10.0/30 is directly connected, Serial0/0
D 128.10.0.0/16 is a summary, 00:01:42, Null0
D* 192.168.0.0/24 [90/2172416] via 128.10.10.2, 00:01:16, Serial0/0
```

Adesso aggiungiamo EIGRP ad 'itesys2' e togliamo "ip default-network":

```
itesys1#sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 128.10.10.2 to network 192.168.0.0

C 192.168.94.0/24 is directly connected, FastEthernet0/0
128.10.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 128.10.10.0/30 is directly connected, Serial0/0
D 128.10.0.0/16 is a summary, 00:02:23, Null0
D 192.168.0.0/24 [90/2172416] via 128.10.10.2, 00:00:25, Serial0/0

itesys1#exit
```

togliamo "ip default-network" e, per propagare la route di default con EIGRP usiamo "redistribute static":

```
itesys2(config)#router eigrp 15
itesys2(config-router)#redistribute static
itesys2(config-router)#exit
itesys2(config)#exit



itesys1#sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 128.10.10.2 to network 0.0.0.0

C 192.168.94.0/24 is directly connected, FastEthernet0/0
128.10.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 128.10.10.0/30 is directly connected, Serial0/0
D 128.10.0.0/16 is a summary, 00:02:57, Null0
D 192.168.0.0/24 [90/2172416] via 128.10.10.2, 00:00:59, Serial0/0
D*EX 0.0.0.0/0 [170/2172416] via 128.10.10.2, 00:00:14, Serial0/0

itesys1#
```

Per EIGRP avremmo potuto utilizzare anche "ip default-network"

6.0 Cenni di OSPF

OSPF e' un protocollo di routing in cui ogni router della rete mantiene il grafo topologico dell'intera rete. Pertanto ogni router della rete ha visibilita' su tutta la struttura di rete. In RIP, EIGRP, IGRP un router non ha idea dell'intera struttura topologica della rete in cui si trova ma soltanto di quella relativa ad esso e ai suoi suoi vicini. Per questa ragione OSPF si chiama protocollo "link-state". La presenza del grafo dell'intera rete consente ad ogni router, applicando l'algoritmo di Dijkstra (shortest-path), di estrarre un albero dal grafo cosi' da determinare il percorso ottimale per raggiungere ogni destinazione evitando loop e senza fare uso di hold-down timer che sono causa della lenta la convergenza in RIP e IGRP. Ogni router estrae un albero di cui lui stesso e' la radice. Se due percorsi hanno lo stesso costo OSPF distribuisce il carico tra i due. Nella configurazione di OSPF non si fa uso di AS number ma di process-id che e' cosa differente in quanto si puo' avere differente process-id in router differenti. OSPF ha pieno supporto della netmask, invia solo aggiornamenti e non l'intera tabella di routing ai suoi vicini, e' piu' CPU intensive di RIP, IGRP ed EIGRP, ed e' scalabile.

Poiche' ogni router ha un grafo dell'intera rete col crescere della stessa il ricalcolo con Dijstra puo' impiegare eccessive risorse di CPU e memoria. Per questa ragione sono state introdotte le AREE. Ogni router ha in realta' consapevolezza della struttura dell'area a cui appartiene. Alcuni router, gli ABR, hanno interfacce su aree diverse. L'area 0 e' speciale ed e' sempre presente in una rete OSPF. Gli ASBR sono invece quei router che ridistribuiscono in OSPF informazioni provenienti da statiche o da altri algoritmi di routing.

Tutte le informazioni di routing vengono scambiate tramite messaggi chiamati LSA. RFC1247definisce gli LSA 'link-state advertisements' e vengono inviati all'indirizzo multicast 224.0.0.5. Vi sono 5 tipi di LSA:

type 1

Propaga informazioni relative a tutte le reti collegate ad un router e appartenenti alla sua stessa area. Gli LSA type 1 vengono inviati ai router di un'area e non ne superano i confini;

type 2

Quando piu' router hanno un'interfaccia sullo stesso segmento di LAN ne viene eletto uno, chiamato DR (Designated Router), che ha il compito di inviare informazioni per conto di tutti i router connessi al segmento. Gli LSA type 2 vengono inviati ai router di un'area e non ne superano i confini;

type 3,4

Questi messaggi passano da un'area ad un'altra adiacente. Sono originati dagli ABR (Area Border Router). ABR invia ad un'area informazioni relative ad un'altra area, sempre di cui fa parte. I type 3 contengono informazioni relative a route verso reti presenti nelle aree. I type 4 contengono informazioni di routing in direzione degli ASBR, ovvero router che inseriscono in OSPF route di altri protocolli come BGP;

type 5

Originati dagli ASBR e contenti informazioni per route relative a destinazioni su AS esterni. Sono di due tipi: E1 e E2. Di default E2. Le route esterne (external routes) vengono propagate dagli ASBR tramite gli LSA Type 5. Le route esterne si classificano a loro volta in:

- external type 1 (E1)
- external type 2 (E2)

La tipologia E1 propaga la route nella nuvola OSPF aggiornandone il costo con la propagazione. La tipologia E2 propaga la route nella nuvola OSPF lasciandone inalterato il costo.

Per default gli LSA sono propagati attraverso tutte le interfacce su una stessa area con esclusione dell'interfacce da cui sono arrivati. Due neighbor, anche adiacenti, si scambiano pacchetti Hello ogni 10 secondi (Hello interval). Se dopo quattro volte questo tempo (Dead interval) non arrivano Hello la vicinanza col neighbor passa dallo stato "FULL" allo stato "DOWN". Hello e' 30 secondi per NBMA (Non Broadcast Multi Access come ATM e Frame Relay). Col comando "show ip ospf neighbor" e' possibile individuare i router OSPF vicini con i quali si ha una sessione attiva:

```
RTA#sh ip ospf nei
Neighbor ID     Pri   State           Dead Time   Address         Interface
172.121.139.253   1   FULL/  -        00:00:36    172.121.139.90  Serial2/0
```

Gli LSA hanno un aging time di 1h oltre la quale sono cancellati. In ogni caso i router inviano ogni 30 minuti dei refresh indipendentemente da eventuali cambiamenti di topologia.

C'e' da considerare che nelle versioni di OSPF meno recenti si inviano refresh indiscriminatamente per tutti gli LSA generati. In seguito si e' adottato un TIMER per ogni LSA. Cio' consente di inviare solo gli LSA effettivamente vecchi. Comunque questi ultimi vengono raggruppati per intervalli di 4 minuti di default (pacing timer) per generare 1 solo pacchetto di aggiornamento per piu' LSA e meno spreco di banda e CPU.

OSPF funziona in modalita' diverse a seconda della rete su cui opera. Si distingue:

point-to-point

E' il caso dei collegamenti punto-punto tra router come un seriale con PPP o HDLC o un ISDN. L'interfaccia puo' essere unnumbered. Se c'e' l'ip il link e' trattato come una stub network;

broadcast networks

E' il caso in cui il router e' collegato ad esempio ad una rete ethernet. Se nella ethernet vi e' un solo router con OSPF questa e' trattata come stub network altrimenti si procede con l'elezione di un DR e la rete e trattata come transit network;

point-to-multipoint non-broadcast

Si tratta delle reti Non Broadcast Multi Access come ATM e Frame Relay. Col point-to-multipoint la rete e vista come un insieme di collegamenti point-to-point. E' il meno efficiente per l'elevata bandwidth utilizzata per gli aggiornamenti. Con il non-broadcast invece la rete viene vista come una rete ethernet. Si elegge un DR. In questo caso pero' ogni router nella NBMA deve vedere gli altri cioe' ci vuole una configurazione full-mesh;

6.1 Esempi

Per resettare il processo OSPF e rigenerare la tabella di routing si usa "clear ip ospf process". Ecco l'invio di un comando clear ad un router con OSPF. La sequenza mostra tutto il processo di avvio di una sessione OSPF e scambio di LSA:

```
prova-ct# clear ip ospf process
Reset ALL OSPF processes? [no]: yes

prova-ct#

*Mar 10 16:07:06.810: OSPF: Flushing External Links
*Mar 10 16:07:06.814: OSPF: Inc retrans unit nbr count index 1 (0/1) to 1/1

#
#  10.121.139.253 e' il nostro unico Neighbor ID
#

*Mar 10 16:07:06.814: OSPF: Set Nbr 10.121.139.253 1 first flood info from 0 (0)
 to 62D48788 (792)
*Mar 10 16:07:06.814: OSPF: Init Nbr 10.121.139.253 1 next flood info to 62D48788

#
#  192.13.19.80 sono gli ip per la navigazione internet assegnati, definiti
#  mediante statica e sono type 5 in quanto provengono da rete esterna
#

   # Quando si manda un LSA questo viene conservato in attesa dell'ACK
   # Se questo non arriva entro il "retransmit-interval", di default 5 sec.,
   # e selezionabile tra 1 a 65535 con "ip ospf retransmit-interval T"
   # Gli LSA sono in una retransmission-list finche' non arriva ACK

*Mar 10 16:07:06.814: OSPF: Add Type 5 LSA ID 192.13.19.80 Adv rtr 10.121.139.89
 Seq 80000023 to Serial2/0 10.121.139.253 retransmission list
*Mar 10 16:07:06.814: OSPF: Start Serial2/0 10.121.139.253 retrans timer
*Mar 10 16:07:06.814: OSPF: Set idb next flood info from 0 (0) to 62D48788 (792)
*Mar 10 16:07:06.814: OSPF: Add Type 5 LSA ID 192.13.19.80 Adv rtr 10.121.139.89
 Seq 80000023 to Serial2/0 flood list
*Mar 10 16:07:06.814: OSPF: Start Serial2/0 pacing timer for 33 msecs
*Mar 10 16:07:06.814: OSPF: Flushing Opaque AS Links
```

```
# Si trasmette LSA type 5 (si noti che siamo nello stesso istante di prima)

*Mar 10 16:07:06.814: OSPF: Flooding update on Serial2/0 to 224.0.0.5 Area 0
*Mar 10 16:07:06.814: OSPF: Send Type 5, LSID 192.13.19.80, Adv rtr 10.121.139.8
9, age 3600, seq 0x80000023 (0)
*Mar 10 16:07:06.814: OSPF: Create retrans unit 0x62D47E14/0x62D46CD4 1 (0/1) 1
*Mar 10 16:07:06.814: OSPF: Set nbr 1 (0/1) retrans to 4968 count to 1
*Mar 10 16:07:06.814: OSPF: Set idb next flood info from 62D48788 (792) to 0 (0)
*Mar 10 16:07:06.814: OSPF: Remove Type 5 LSA ID 192.13.19.80 Adv rtr 10.121.139
.89 Seq 80000023 from Serial2/0 flood list
*Mar 10 16:07:06.814: OSPF: Stop Serial2/0 flood timer

#
# LSA type 5 per la rete 192.13.90.80 e' stato inviato
#
# Finiti gli LSA type 5 adesso manda quello della seriale in questo caso 10.121.139.89
# che e' LSA type 1
#

*Mar 10 16:07:06.854: OSPF: Flushing Link states in area 0
*Mar 10 16:07:06.854: OSPF: Inc retrans unit nbr count index 1 (0/1) to 1/1
*Mar 10 16:07:06.854: OSPF: Set Nbr 10.121.139.253 1 first flood info from 0 (0)
 to 62D49024 (1000)
*Mar 10 16:07:06.854: OSPF: Init Nbr 10.121.139.253 1 next flood info to 62D49024
*Mar 10 16:07:06.854: OSPF: Add Type 1 LSA ID 10.121.139.89 Adv rtr 10.121.139.89 Seq 80000036 to Serial2/0
10.121.139.253 retransmission list
*Mar 10 16:07:06.854: OSPF: Set idb next flood info from 0 (0) to 62D49024 (1000)
*Mar 10 16:07:06.854: OSPF: Add Type 1 LSA ID 10.121.139.89 Adv rtr 10.121.139.89 Seq 80000036 to Serial2/0 flood list
*Mar 10 16:07:06.854: OSPF: Start Serial2/0 pacing timer for 33 msecs
*Mar 10 16:07:06.854: OSPF: Flooding update on Serial2/0 to 224.0.0.5 Area 0
*Mar 10 16:07:06.854: OSPF: Send Type 1, LSID 10.121.139.89, Adv rtr 10.121.139.89, age 3600, seq 0x80000036 (0)
*Mar 10 16:07:06.854: OSPF: Create retrans unit 0x62D47E44/0x62D46B94 1 (0/1) 1
*Mar 10 16:07:06.854: OSPF: Set nbr 1 (0/1) retrans to 4736 count to 1
*Mar 10 16:07:06.854: OSPF: Set idb next flood info from 62D49024 (1000) to 0 (0)
*Mar 10 16:07:06.854: OSPF: Remove Type 1 LSA ID
10.121.139.89 Adv rtr 10.121.139.89 Seq 80000036 from Serial2/0 flood list

#

*Mar 10 16:07:06.854: OSPF: Stop Serial2/0 flood timer
*Mar 10 16:07:06.894: OSPF: Interface Serial2/0 going Down
*Mar 10 16:07:06.894: %OSPF-5-ADJCHG: Process 10, Nbr 10.121.139.253 on Serial2/0 from FULL to DOWN, Neighbor
Down: Interface down or detached
*Mar 10 16:07:06.894: OSPF: Dec retrans unit nbr count index 1 (0/1) to 0/0
*Mar 10 16:07:06.894: OSPF: Free nbr retrans unit 0x62D47E44/0x62D46B94 0 total
0. Also Free nbr retrans block
*Mar 10 16:07:06.894: OSPF: Set Nbr 10.121.139.253 1 first flood info from 62D49024 (1000) to 0 (0)
*Mar 10 16:07:06.894: OSPF: Adjust Nbr 10.121.139.253 1 next flood info to 0
*Mar 10 16:07:06.894: OSPF: Remove Type 1 LSA ID 10.121.139.89 Adv rtr 10.121.139.89 Seq 80000036 from
10.121.139.253 retransmission list
*Mar 10 16:07:06.894: OSPF: Dec retrans unit nbr count index 1 (0/1) to 0/0
*Mar 10 16:07:06.894: OSPF: Free nbr retrans unit 0x62D47E14/0x62D46CD4 0 total0. Also Free nbr retrans block
*Mar 10 16:07:06.894: OSPF: Set Nbr 10.121.139.253 1 first flood info from 62D48788 (792) to 0 (0)
*Mar 10 16:07:06.894: OSPF: Adjust Nbr 10.121.139.253 1 next flood info to 0
*Mar 10 16:07:06.894: OSPF: Remove Type 5 LSA ID 192.13.19.80 Adv rtr 10.121.139.89 Seq 80000023 from
10.121.139.253 retransmission list
*Mar 10 16:07:06.894: OSPF: Stop nbr 10.121.139.253 retransmission timer
```

*Mar 10 16:07:06.894: OSPF: Interface Loopback1 going Down
*Mar 10 16:07:06.962: OSPF: Interface Serial2/0 going Up
*Mar 10 16:07:06.966: OSPF: Interface Loopback1 going Up
*Mar 10 16:07:07.394: OSPF: Build router LSA for area 0, router ID 10.121.139.89, seq 0x80000001

#
Si passa allo stato 2WAY dopo che arriva hello dal vicino
#
Si passa allo stato EXSTART quando si stabilisce chi e' master e chi slave
Il router con il piu' alto IP address diviene il master
E si manda un numero di sequenza che consente di determinare i vecchi LSA
#
DBD sta per DataBaseDescriptor

EXCHANGE: In questo stato il router scambia pacchetti DBD, che
descrivono l'intero link-state database

*Mar 10 16:07:08.274: OSPF: Rcv hello from 10.121.139.253 area 0 from Serial2/0
10.121.139.90
*Mar 10 16:07:08.274: OSPF: 2 Way Communication to 10.121.139.253 on Serial2/0,state 2WAY
*Mar 10 16:07:08.274: OSPF: Send DBD to 10.121.139.253 on Serial2/0 seq 0x2573 opt 0x42 flag 0x7 len 32
*Mar 10 16:07:08.278: OSPF: End of hello processing
*Mar 10 16:07:08.294: OSPF: Rcv DBD from 10.121.139.253 on Serial2/0 seq 0x14AF opt 0x42 flag 0x7 len 32 mtu 1500
state EXSTART
*Mar 10 16:07:08.294: OSPF: NBR Negotiation Done. We are the SLAVE
*Mar 10 16:07:08.294: OSPF: Send DBD to 10.121.139.253 on Serial2/0 seq 0x14AF opt 0x42 flag 0x2 len 72
*Mar 10 16:07:08.334: OSPF: Rcv DBD from 10.121.139.253 on Serial2/0 seq 0x14B0 opt 0x42 flag 0x3 len 1472 mtu 1500
state EXCHANGE
*Mar 10 16:07:08.334: OSPF: Send DBD to 10.121.139.253 on Serial2/0 seq 0x14B0 opt 0x42 flag 0x0 len 32
*Mar 10 16:07:08.338: OSPF: Database request to 10.121.139.253
*Mar 10 16:07:08.338: OSPF: sent LS REQ packet to 10.121.139.90, length 864

#
#NOTARE L'INVIO DELLA MTU:

Note: Cisco IOS ® Software Release 12.0(3) introduced interface MTU mismatch detection. This detection involves OSPF advertising the interface MTU in the DBD packets, which is in accordance with the OSPF RFC 2178 , appendix G.9. When a router receives a DBD packet advertising a MTU larger than the router can receive, the router ignores the DBD packet and the neighbor state remains in exstart. This prevents an adjacency from forming. To fix this problem, make sure the MTU are the same on both ends of a link.

*Mar 10 16:07:08.370: OSPF: Rcv DBD from 10.121.139.253 on Serial2/0 seq 0x14B1 opt 0x42 flag 0x3 len 1472 mtu 1500
state EXCHANGE
*Mar 10 16:07:08.374: OSPF: Send DBD to 10.121.139.253 on Serial2/0 seq 0x14B1 opt 0x42 flag 0x0 len 32
*Mar 10 16:07:08.390: OSPF: received update from 10.121.139.253, Serial2/0
*Mar 10 16:07:08.390: OSPF: Rcv Update Type 1, LSID 192.13.21.241, Adv rtr 192.13.21.241, age 1656, seq 0x80000879
*Mar 10 16:07:08.390: OSPF: Rcv Update Type 1, LSID 192.13.21.209, Adv rtr 192.13.21.209, age 958, seq 0x80001220
*Mar 10 16:07:08.390: OSPF: Rcv Update Type 1, LSID 192.13.21.197, Adv rtr 192.13.21.197, age 497, seq 0x80000FE3
*Mar 10 16:07:08.394: OSPF: Rcv Update Type 1, LSID 192.13.21.194, Adv rtr 192.13.21.194, age 1133, seq 0x8000017E
*Mar 10 16:07:08.394: OSPF: Rcv Update Type 1, LSID 192.13.21.189, Adv rtr 192.13.21.189, age 304, seq 0x80000E57

```
 ...

*Mar 10 16:07:08.546: OSPF: Received same lsa
*Mar 10 16:07:08.546: OSPF: Sending direct ACK on Serial2/0 to 10.121.139.253
*Mar 10 16:07:08.546: OSPF: Ack Type 1, LSID 192.13.21.209, Adv rtr 192.13.21.209, age 959, seq 0x80001220

 ...

*Mar 10 16:07:08.558: OSPF: Rcv Update Type 1, LSID 192.13.2.166, Adv rtr 192.13.2.166, age 1801, seq 0x800034BC
*Mar 10 16:07:08.558: OSPF: Received same lsa
*Mar 10 16:07:08.558: OSPF: Rcv Update Type 1, LSID 192.13.2.100, Adv rtr 192.13.2.100, age 237, seq 0x800060DA
*Mar 10 16:07:08.558: OSPF: Received same lsa
*Mar 10 16:07:08.558: OSPF: Rcv Update Type 1, LSID 192.13.2.15, Adv rtr 192.13.2.15, age 350, seq 0x800003DA

#
#  router che formano un loop si inviano LSA tra loro finche' non c'e' un
#  "Received same lsa"
#

*Mar 10 16:07:08.562: OSPF: Received same lsa
*Mar 10 16:07:08.562: OSPF: Rcv Update Type 1, LSID 192.13.2.8, Adv rtr 192.13.2.8, age 1076, seq 0x80001805

 ...

*Mar 10 16:08:01.894: OSPF: Rcv Update Type 3, LSID 10.121.131.4, Adv rtr 10.121.131.2, age 7, seq 0x80000679
*Mar 10 16:08:01.894:       Mask /30
*Mar 10 16:08:01.894: OSPF: Rcv Update Type 3, LSID 10.121.131.8, Adv rtr 10.121.131.2, age 7, seq 0x80000EDC
*Mar 10 16:08:01.894:       Mask /30
*Mar 10 16:08:01.894: OSPF: Rcv Update Type 3, LSID 10.121.131.12, Adv rtr 10.221.131.2, age 7, seq 0x800002F8

 ...

*Mar 10 16:08:04.410: OSPF: Ack Type 3, LSID 192.13.21.237, Adv rtr 10.121.131.2, age 7, seq 0x80000B01
*Mar 10 16:08:04.410: OSPF: Ack Type 4, LSID 192.13.21.237, Adv rtr 10.121.131.2, age 7, seq 0x80000B83
*Mar 10 16:08:04.410: OSPF: Ack Type 5, LSID 10.121.156.39, Adv rtr 10.121.136.252, age 5, seq 0x80000080
*Mar 10 16:08:07.238: OSPF: received update from 10.121.139.253, Serial2/0
*Mar 10 16:08:07.238: OSPF: Rcv Update Type 1, LSID 192.13.29.94, Adv rtr 192.13.29.94, age 6, seq 0x80000623
*Mar 10 16:08:08.274: OSPF: Rcv hello from 10.121.139.253 area 0 from Serial2/0 10.121.139.90
*Mar 10 16:08:08.274: OSPF: End of hello processing
*Mar 10 16:08:09.738: OSPF: Sending delayed ACK on Serial2/0
*Mar 10 16:08:09.738: OSPF: Ack Type 1, LSID 192.13.29.94, Adv rtr 192.13.29.94, age 6, seq 0x80000623

prova-ct#undebug all
```

7.0 La necessita' per il BGP

L'allocazione degli indirizzi della rete Internet mondiale e' responsabilita' di quattro enti, chiamati RIR (Regional Internet Registry): RIPE, l'ARIN, APNIC e il LACNIC. Il RIPE si occupa di Europa e Africa del Nord, l'ARIN dell'America del Nord, l'APNIC dell'Asia e del Pacifico, il LACNIC dell'America Latina.

I restanti paesi sono suddivisi tra questi enti (http://www.arin.net/library/internet_info/ARINcountries.htm). Anche l'Africa ha il suo RIR (http://www.afrinic.org).

Chiunque utilizzi un indirizzo ip pubblico con cui riceve o fornisce servizi internet lo ha ottenuto direttamente dal RIR che rappresenta la sua regione o da un ente (normalmente un ISP) che a sua volta lo ha ottenuto da un RIR

I RIR assegnano gli indirizzi internet a chiunque ne chieda in numero "sufficiente" a fronte di un canone annuale. I RIR normalmente assegnano indirizzi con mask da /20 a scendere. Se non si dimostra di impiegare da subito almeno 8 classi C in modo efficiente difficilmente la propria richiesta va avanti, specialmente se non si dimostra di essere in un ambiente multihomed. Se cosi' non e' per ottenere da un RIR delle classi puo' essere necessario un utilizzo di indirizzi consistente, come 16 classi C. Ma il condizionale e' d'obbligo. La crescita delle tabelle di routing mondiali e il prossimo esaurimento delle risorse IPV4 puo' portare ad un improvviso cambiamento delle regole in modo piu' restrittivo.

Insomma dai piccoli utenti fino all'enterprise difficilmente si contatta direttamente il RIPE (visto che siamo in europa). Infatti chi ci assegna gli indirizzi IP e' normalmente il nostro fornitore di banda Internet, che dovra' comunque comunicare al RIPE l'intestatario degli stessi, a meno che non si tratti semplicemente di un singolo o di una coppia di indirizzi. Basta avere un ADSL con 8 indirizzi IP assegnati per figurare nel database del RIPE e quindi essere anche facilmente identificabili a partire dagli indirizzi ip in uso.

Normalmente avere in uso degli indirizzi assegnati dal nostro provider invece che direttamente dal RIPE non e' un problema, anzi la maggior parte degli utenti non ne e' neppure consapevole.

Nel momento in cui forniamo servizi verso la rete internet, da un certo punto in poi, la questione che si puo' porre e' se restare dipendenti dal provider internet o meno. Cambiare fornitore infatti vorrebbe dire perdere i propri indirizzi IP e cambiarli con altri, assegnati dal nuovo provider. Se abbiamo parecchie macchine che forniscono servizi all'esterno, magari con certificati digitali o server DNS, un'indipendenza dal fornitore di connettivita' puo' essere preferibile. Cio' si puo' ottenere richiedendo gli indirizzi IP direttamente al RIPE, quando possibile, in quanto si ricade nelle problematiche di cui sopra.

La tanto agognata indipendenza puo' essere fondamentale in contesti in cui vi sono requisiti di ridondanza e affidabilita' che trovano l'anello piu' debole della catena proprio nell'avere un unico provider fornitore di banda internet. In questi casi cio' che va fatto e' dividere la banda tra due differenti fornitori, soluzione sicuramente piu' costosa, ma in grado di consentire alti livelli di up-time e, come vedremo, l'indipendenza dal singolo provider anche per gli indirizzi IP. Infatti per gestire una situazione del genere, chiamata di multihoming, si dovrebbero avere necessariamente degli indirizzi IP propri, e non di proprieta' di uno dei due provider in quanto, in tal caso, i pacchetti in ingresso giungerebbero sicuramente da un link solo, consentendo un bilanciamento del traffico solo in uscita.

In casi di multihoming come questo, e' necessario ottenere degli IP propri, che per l'Europa chiederemo al RIPE. Si puo' anche richiedere al RIPE solo l'autonomous system (AS), da associare alle classi fornite dal fornitore di connettivita'. Si noti che questa soluzione garantisce un corretto multihoming ma conserva una dipendenza per gli indirizzi IP dal fornitore di connettivita'.

Ma cosa vuol dire essere Autonomous System? Il RIPE non assegna solo indirizzi /20, /19 o altro (fino a /13) ma fornisce anche un numero AS. Gli AS suddividono la internet mondiale in aree, ognuna delle quali e' sotto una comune amministrazione. Il numero di AS non e' un concetto astratto e burocratico ma piuttosto un valore numerico che i router stessi utilizzano quando si scambiamo le informazioni di routing tramite il protocollo BGP. AS e' un numero a 32 bits.

Chi possiede un AS puo' comunicare ai router degli AS vicini gli indirizzi assegnati dal RIPE. Questi si propagheranno nella internet mondiale. Cosi' la nostra definizione grezza di "spazio IP utilizzabile come personale" si traduce in "possedere un numero di AS". Se non si ha un AS il traffico entrante proviene sempre dal provider che possiede l'AS associato agli IP che si stanno utilizzando quindi niente ridondanza e/o bilanciamento del traffico. RIPE assegna il numero di AS a chi ha un ambiente multihoming (a tal proposito vedi http://www.ripe.net/ripe/docs/asn-assignment.html). Il protocollo di routing che agisce tra gli AS e consente la propagazione delle route a livello mondiale e' il BGP (Border Gateway Protocol).

Le politiche di bilanciamento del traffico non verranno trattate in questo articolo, e lo stesso vale per una trattazione esaustiva del protocollo BGP o sulla gestione amministrativa nel database del RIPE. Si tratta di argomenti piuttosto complessi che richiedono un libro piu' che una pagina web. In ogni caso ecco qualche link:

http://www.cisco.com/en/US/tech/tk648/tk365/technologies_case_study09186a00800949ea.shtml
http://www.arin.net/policy/ipv4.html#requirements
http://www.ripe.net/ripe/docs/asn-assignment.html
http://www.ripe.net/ripe/docs/new-lir.html

o, per vedere qualcosa di piu' nuovo e interessante:

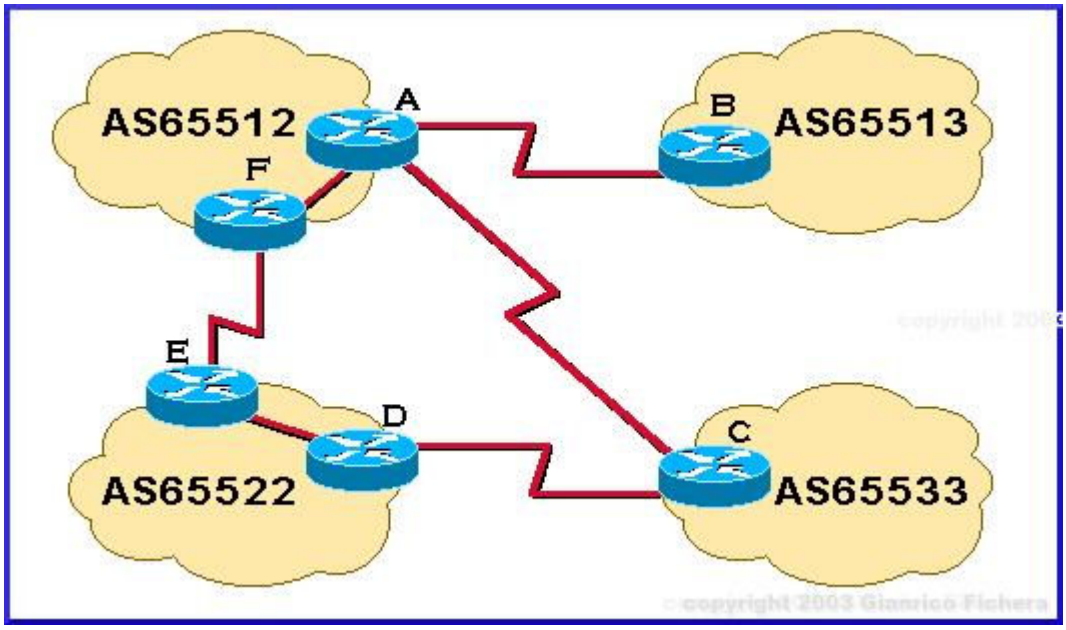
http://www.ripe.net/ripe/docs/ipv6policy.html

7.1 Il protocollo BGP

I vari protocolli di routing descritti sino ad ora, quali RIP, IGRP, EIGRP, OSPF, sono in grado di gestire reti di dimensione da piccola a media, se vogliamo medio/grande, come nel caso di OSPF. Ma una caratteristica comune a tutti questi protocolli e' che sono pensati per funzionare all'interno di un'unica amministrazione. Sostanzialmente sono pensati per essere gestiti da una stessa organizzazione (basta pensare al comportamento delle distribute-list in OSPF). Ad esempio un router di una rete OSPF, se mal configurato, puo' creare problemi alla rete intera. Ogni router OSPF contiene nella sua tabella informazioni topologiche su tutti i router della sua area. Ma la rete

internet mondiale e' tutt'altra cosa. Due router possono appartenere a due amministrazioni differenti ed essere gestiti da persone di enti differenti. Non attiverei mai l'OSPF del mio ISP all'interno dei router dei clienti, a meno che essi siano privi della possibilita' di amministrarli. Se c'e' necessita' di amministrazione configurerei la macchina del cliente con delle route statiche, oppure, usando il BGP.

BGP interviene ai confini del nostro AS, scambiando le informazioni di routing con gli altri AS. Con BGP vi e' la possibilita' di definire chiaramente cosa annunciare ai router degli AS vicini e cosa accettare da loro. Una buona configurazione BGP rifiuta dagli AS vicini cio' che gli AS vicini non sono amministrativamente tenuti ad annunciare. Ma ecco un esempio:



All'interno dell'AS numero 65512 vi e' un'organizzazione unica, che ad esempio puo' essere un ISP, oppure una enterprise. All'interno dell'organizzazione vi sara' una rete con decine di router, con protocollo EIGRP, o OSPF ad esempio. Ma a noi interessano le relazioni di routing tra questa organizzazione e le altre, che potrebbero essere altri ISP. Cosi' se AS65512 e' un provider nazionale allora AS65513, AS65522 e AS65533 potrebbero essere i tre provider che gli forniscono la banda internazionale. Oppure AS65533 e' una enterprise che prende banda internet dai due provider nazionali AS65512 e AS65513. Poi il provider AS65512 prende la sua banda internazionale da AS65513. AS65522 non ha banda internazionale diretta ma la compra da AS65512.

BGP consente l'interfacciamento tra queste diverse entita' in modo sicuro ed efficiente. Ecco come:

- estesa possibilita' di controllo su quali route annunciare ai vicini e cosa accettare dai vicini; agli altri AS arrivano solo le route prescelte e dagli AS vicini si accettano solo le route volute;
- algoritmo di scelta della route migliore pensato per gestire le macroregioni dell'internet mondiale; non si usano delay, bandwidth, MTU, reliability, hop

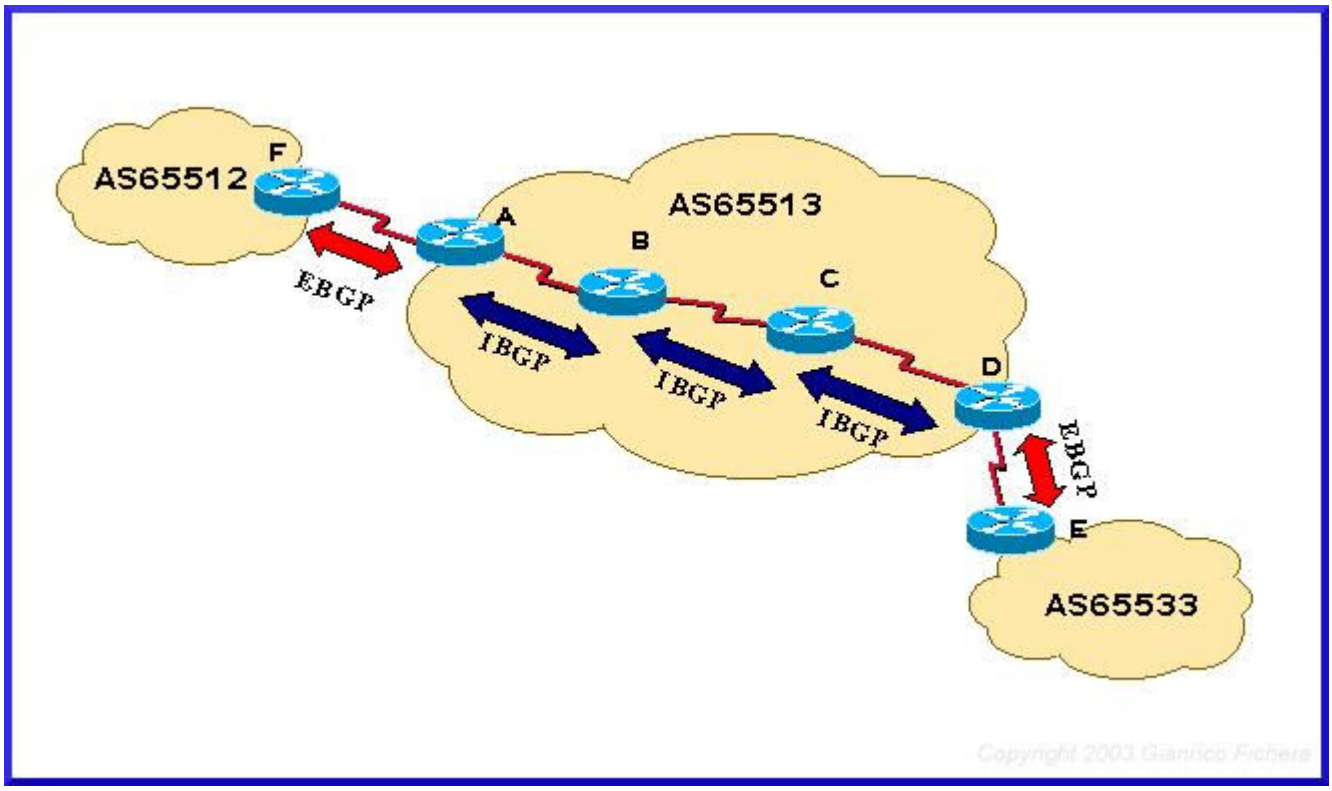
count. Si utilizzano l'AS-PATH (numero di AS per raggiungere una destinazione), il peso, l'eta' delle route, la provenienza della route;

Tra due router con BGP sono possibili due tipi di sessioni: I-BGP, E-BGP. Il primo, internal BGP, si ha quando i due router appartengono allo stesso AS, come A ed F in figura. Il secondo, external BGP, e' tra AS differenti, come A e B, o A e C in figura.

Nel caso di I-BGP vale la seguente importante regola:

- Le route provenienti da un link IBGP vengono propagate solo attraverso links EBGP

La figura mostra sicuramente meglio il concetto:



Il router B riceve la tabella dal router A ma non la propaga al router C. Il router D riceve la tabella dal router E e la propaga al router C. Il router C non la propaga al router B. Se il nostro scopo e' di avere entrambe le tabelle di routing su B e C in modo da poter scegliere il percorso migliore allora e' necessario creare le sessioni BGP formando una topologia FULL-MESH: dobbiamo aggiungere delle sessioni BGP tra A e C e tra D e B.

7.2 Esempio di configurazione

router bgp 65300	In questo caso il RIPE ha assengnato al nostro provider il numero di AS 65300
no synchronization	A BGP router con la synchronization abilitata non annuncera' le route acquisite tramite iBGP se non le puo' validare in IGP. Se nella vostra configurazione BGP il router non annuncia nulla probabilmente avete bisogno di "no synchronization"
network 194.242.61.0	Questa e' la rete che ci e' stata assegnata dal RIPE e che il router deve annunciare al mondo tramite i neighbor definiti nelle righe successive
neighbor 217.223.243.9 remote-as 65400	217.223.243.9 e' un vicino. Si tratta del router dell'ISP da cui prendiamo banda internet normalmente. Questo IP deve essere indicato dall'ISP. In questo esempio il nostro ISP 'BOBNETWORKS' ha as 65400
neighbor 217.223.243.9 description -- PROVIDER A --	Un semplice commento
neighbor 217.223.243.9 ebgp-multihop 255	Il router di 'BOBNETWORKS' non e' ad un hop di distanza dal nostro ma vi sono dei router intermedi in numero non precisato
neighbor 217.223.243.9 update-source Loopback0	Il router di 'BOBNETWORKS' deve inviare al nostro la sua tabella di routing. Cosi' a 'BOBNETWORKS' dobbiamo indicare in quale ip inviare questi dati. Conviene indicare l'ip di una interfaccia di Loopback. Cosi', in questa riga, indichiamo che gli aggiornamenti dal router vicino provengono attraverso la Loopback0
neighbor 217.223.243.9 route-map filtricubeIN in	Non tutta la tabella di routing inviata da 'BOBNETWORKS' dev'essere presa in considerazione. Alcune route vanno scartate, probabilmente perche' vogliamo che si esca attraverso un altro ISP fornitore di banda per queste destinazioni. Nel nostro caso 216.133.69.2 definito sotto.
neighbor 217.223.243.9 route-map localonly out	Qui si decide cosa annunciare attraverso 'BOBNETWORKS'. Di tutti gli IP che il RIPE ci ha assegnato normalmente vogliamo che una parte venga annunciata verso 'BOBNETWORKS' e un'altra verso il secondo fornitore di banda, che chiamiamo 'JOHNSNET'
neighbor 216.133.69.2 remote-as 65500	Ecco 'JOHNSNET'. Questo e' il nostro secondo fornitore di banda internet. Ha numero di AS 65500. Le righe successive riflettono per 'JOHNSNET' le considerazioni

	di sopra.
neighbor 216.133.69.2 description --- PROVIDER B ---	
neighbor 216.133.69.2 ebgp-multihop 255	
neighbor 216.133.69.2 route-map filtriPROVIDERA in	
neighbor 216.133.69.2 route-map localonly out	
ip as-path access-list 4 permit .*	
ip as-path access-list 5 permit _65400 1267_	
ip as-path access-list 5 permit _65400 3257_	
ip as-path access-list 5 permit _65400 12876_	
ip as-path access-list 6 deny _1299_	
ip as-path access-list 6 permit .*	.* fa matching con tutto. Si usa come "permit any" al termine di una serie di route-map. Infatti queste, come le access-list, hanno un deny any di default alla fine.
ip as-path access-list 10 permit ^$	^$ vuol dire "originate da questo AS". Questa ci serve per evitare di annunciare al provider B le route provenienti dal provider A e viceversa.
ip as-path access-list 11 permit _65500 3269_	
ip as-path access-list 11 permit _65500 3257_	
ip as-path access-list 11 permit _65500 12874_	
route-map filtricubeIN permit 10	
match as-path 5	Si controlla l'as-path per ogni riga di routing che proviene dal vicino. Se c'e' un matching con l'espressione regolare 5, che vediamo piu' sopra in colore arancio, allora gli si associa un peso di 200 (tutte le altre avranno peso 0, che e' il default). Le route con peso piu' alto sono le preferite. Il nostro obiettivo qui e' di uscire con "BOBNETWORKS" quando le destinazioni passano per gli AS 1267, 3257 e 12876
set weight 200	Vedi anche la sezione in giallo
route-map filtricubeIN permit 20	
match as-path 4	
route-map filtriPROVIDERA permit 10	JOHNSNET verra' scelto per i pacchetti di uscita quando la destinazione passa per 3269, 3257, 12874 secondo le espressioni regolari di cui sopra. JOHNSNET e' anche di default ma stiamo esplicitando le destinazioni principali in questa configurazione.

match as-path 11	
set weight 200	
route-map filtriPROVIDERA permit 20	
match as-path 4	
route-map altopeso permit 10	
set weight 200	
route-map pesoPROVIDERA permit 10	Dando peso 20 a tutto cio' che proviene da JOHNSNET, in accordo con quanto visto nella sezione arancione concludiamo che di default si esce con JOHNSNET perche' il peso e' piu' basso.
set weight 20	Vedi anche la sezione in arancio
route-map localonly permit 10	
match as-path 10	

7.3 Debugging e comandi

gw-3640-A# **sh ip bgp summ**
BGP router identifier 10.70.80.13, local AS number 65300
BGP table version is 6824365, main routing table version 6824365
112585 network entries and 224554 paths using 19004689 bytes of memory
40195 BGP path attribute entries using 2413080 bytes of memory
35837 BGP AS-PATH entries using 997492 bytes of memory
9 BGP community entries using 280 bytes of memory
20268 BGP route-map cache entries using 324288 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 343026/5982225 prefixes, 1190775/966221 paths, scan interval 15 sec

s

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
217.223.243.9 4 65400 849309 3765 6824365 0 0 17:21:09 112084
216.133.69.2 4 65500 541631 2893 6824351 0 0 1d15h 112469

gw-3640-A#**sh ip bgp nei**
BGP neighbor is 217.223.243.9, remote AS 65400, external link
Description: -- CUBECOM --
BGP version 4, remote router ID 217.223.243.9
BGP state = Established, up for 17:21:12
Last read 00:00:12, hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
Route refresh: advertised and received(new)
Address family IPv4 Unicast: advertised and received
Received 849309 messages, 0 notifications, 0 in queue
Sent 3765 messages, 2 notifications, 0 in queue
Route refresh request: received 0, sent 13
Default minimum time between advertisement runs is 30 seconds

```
For address family: IPv4 Unicast
BGP table version 6824365, neighbor version 6824365
Index 1, Offset 0, Mask 0x2
Inbound path policy configured
Outbound path policy configured
Route map for incoming advertisements is filtricubeIN
Route map for outgoing advertisements is localonly
112084 accepted prefixes consume 4035024 bytes
Prefix advertised 5, suppressed 0, withdrawn 0

Connections established 5; dropped 4
Last reset 17:22:08, due to Peer closed the session
External BGP neighbor may be up to 255 hops away.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 10.70.80.13, Local port: 179
Foreign host: 217.223.243.9, Foreign port: 49922
Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)

Event Timers (current time is 0xD8EC8AC):
Timer Starts Wakeups Next
Retrans 1082 32 0x0
TimeWait 0 0 0x0
AckHold 13841 3337 0x0
SendWnd 0 0 0x0
KeepAlive 551 0 0x0
GiveUp 0 0 0x0
PmtuAger 0 0 0x0
DeadWait 0 0 0x0

iss: 2781263762 snduna: 2781283769 sndnxt: 2781283769 sndwnd: 16194
irs: 991799584 rcvnxt: 1005201028 rcvwnd: 16084 delrcvwnd: 300

SRTT: 300 ms, RTTO: 303 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 12 ms, maxRTT: 1696 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs

Datagrams (max data segment is 536 bytes):
Rcvd: 32455 (out of order: 4314), with data: 31599, total data bytes: 13401443
Sent: 32969 (retransmit: 32), with data: 1049, total data bytes: 20006

BGP neighbor is 216.133.69.2, remote AS 65500, external link
Description: --- PROVIDERB ---
BGP version 4, remote router ID 216.133.69.2
BGP state = Established, up for 1d15h
Last read 00:00:24, hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
Route refresh: advertised and received(new)
Address family IPv4 Unicast: advertised and received
Received 541631 messages, 1 notifications, 0 in queue
Sent 2893 messages, 2 notifications, 0 in queue
Route refresh request: received 0, sent 8
Default minimum time between advertisement runs is 30 seconds

For address family: IPv4 Unicast
BGP table version 6824365, neighbor version 6824365
Index 2, Offset 0, Mask 0x4
Inbound path policy configured
```

Outbound path policy configured
Route map for incoming advertisements is pesoPROVIDERA
Route map for outgoing advertisements is localonly
112469 accepted prefixes consume 4048884 bytes
Prefix advertised 5, suppressed 0, withdrawn 0

Connections established 5; dropped 4
Last reset 1d15h, due to BGP Notification sent, hold time expired
External BGP neighbor may be up to 255 hops away.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 213.92.14.23, Local port: 11083
Foreign host: 216.133.69.2, Foreign port: 179

Enqueued packets for retransmit: 0, input: 0 mis-ordered: -225 (0 bytes)

Event Timers (current time is 0xD8ED14C):
Timer Starts Wakeups Next
Retrans 2707 301 0x0
TimeWait 0 0 0x0
AckHold 29417 8472 0xD8ED1C0
SendWnd 0 0 0x0
KeepAlive 0 0 0x0
GiveUp 0 0 0x0
PmtuAger 0 0 0x0
DeadWait 0 0 0x0

iss: 3173509963 snduna: 3173555775 sndnxt: 3173555775 sndwnd: 16251
irs: 1534877445 rcvnxt: 1559531265 rcvwnd: 16341 delrcvwnd: 43

SRTT: 355 ms, RTTO: 526 ms, RTV: 171 ms, KRTT: 0 ms
minRTT: 4 ms, maxRTT: 1168 ms, ACK hold: 200 ms
Flags: higher precedence, nagle

Datagrams (max data segment is 536 bytes):
Rcvd: 63450 (out of order: 8606), with data: 59865, total data bytes: 24654031
Sent: 64241 (retransmit: 301), with data: 2405, total data bytes: 45811

gw-3640-A#**sh ip bgp path**
Address Hash Refcount Metric Path
0x65C58938 0 1 0 65500 7176 286 517 15772 21112 i
0x638D4BC0 0 1 0 65400 1299 2914 25626 i
0x639336B4 0 1 0 65400 1299 3561 7501 2525 i
0x638B0B58 0 1 0 65400 1299 702 15814 i
0x65C553C8 0 6 0 65400 1299 1239 11183 i
0x657E519C 0 6 0 65400 1299 7018 22026 25625 i
0x6252E284 0 1 0 65500 5400 1239 9237 10030 i
0x65EF565C 0 1 0 65500 1267 20993 i
0x62DAA910 0 1 0 65500 5400 1239 21818 i
0x639996E8 0 2 0 65500 7176 1 701 4230 13353 21741 i
0x65C60AA8 0 1 0 65500 5400 7018 724 5872 2721 i
0x65C25214 0 1 0 i
0x62527920 1 3 0 65400 1299 15742 i
0x639E281C 1 2 0 65500 7176 1 701 11329 i
0x639E13EC 1 1 0 65500 7176 6461 10530 18059 18059 4832 i
0x64D26EE0 1 1 0 65500 7176 1299 15855 i
0x642AEDA0 1 2 0 65500 7176 1 3561 17173 i
0x65C5F7A8 1 2 0 65500 7176 1 9942 9942 17982 9408 i

0x62615328 1 1 0 65500 5400 209 568 721 1494 i
0x65335368 1 5 0 65500 7176 3549 1313 i
0x62E0AD30 1 1 0 65500 7176 1 10912 10912 10912 18526 i
0x65B1FF00 1 1 0 65500 7176 1 14589 i

Address Hash Refcount Metric Path
0x639D6124 2 3 0 65400 6461 14679 14679 1467
....

gw-3640-A#**sh ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

B 208.221.13.0/24 [20/0] via 217.223.243.9, 06:42:59
B 206.51.253.0/24 [20/0] via 217.223.243.9, 01:51:01
B 205.204.1.0/24 [20/0] via 217.223.243.9, 06:43:10
B 204.255.51.0/24 [20/0] via 217.223.243.9, 06:43:11
B 204.238.34.0/24 [20/0] via 216.133.69.2, 06:45:04
B 204.17.221.0/24 [20/0] via 216.133.69.2, 06:45:06
B 203.238.37.0/24 [20/0] via 217.223.243.9, 06:43:16
B 203.34.233.0/24 [20/0] via 216.133.69.2, 06:45:07
B 200.68.140.0/24 [20/0] via 216.133.69.2, 06:45:10
B 198.17.215.0/24 [20/0] via 217.223.243.9, 06:43:37
B 192.68.132.0/24 [20/0] via 216.133.69.2, 06:45:19
170.170.0.0/16 is variably subnetted, 3 subnets, 3 masks
B 170.170.0.0/19 [20/0] via 217.223.243.9, 03:35:21
B 170.170.224.0/20 [20/0] via 216.133.69.2, 06:45:21
B 170.170.254.0/24 [20/0] via 217.223.243.9, 03:35:23
B 216.239.54.0/24 [20/0] via 216.133.69.2, 06:44:56
B 216.220.5.0/24 [20/0] via 216.133.69.2, 06:44:56
B 216.103.190.0/24 [20/0] via 216.133.69.2, 06:44:57
B 213.239.59.0/24 [20/0] via 217.223.243.9, 06:42:48
B 213.152.76.0/24 [20/0] via 217.223.243.9, 03:15:30
B 212.205.24.0/24 [20/0] via 216.133.69.2, 06:44:58
B 207.254.48.0/24 [20/0] via 217.223.243.9, 06:43:04

....

gw-3640-A#**sh ip bgp**

BGP table version is 6667706, local router ID is 10.70.80.13
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

```
* 3.0.0.0 217.223.243.9 0 65400 1299 1239 80 i
*> 216.133.69.2 0 65500 5400 1239 80 i
* 4.0.0.0 217.223.243.9 0 65400 1299 1 i
*> 216.133.69.2 0 65500 7176 1 i
*> 4.2.86.128/26 216.133.69.2 0 65500 7176 1 i
*> 4.2.88.48/28 216.133.69.2 0 65500 7176 1 i
> 4.2.88.128/26 216.133.69.2 0 65500 7176 1 i
*> 4.2.101.0/24 216.133.69.2 0 65500 7176 1 i
*> 4.2.102.128/28 216.133.69.2 0 65500 7176 1 i
*> 4.2.102.144/28 216.133.69.2 0 65500 7176 1 i
*> 4.2.102.160/28 216.133.69.2 0 65500 7176 1 i
*> 4.2.102.176/28 216.133.69.2 0 65500 7176 1 i
*> 4.2.102.192/28 216.133.69.2 0 65500 7176 1 i
*> 4.2.102.208/28 216.133.69.2 0 65500 7176 1 i
*> 4.18.247.40/29 216.133.69.2 0 65500 7176 1 18509 i
*> 4.18.251.128/25 216.133.69.2 0 65500 7176 1 18509 i
*> 4.22.240.0/21 217.223.243.9 0 65400 1299 1 7843 i
*> 216.133.69.2 0 65500 7176 1 7843 i

Network Next Hop Metric LocPrf Weight Path

*> 6.1.0.0/16 216.133.69.2 0 65500 7176 1 7170 14

.....

gw-3640-A#sh ip bgp regexp _65500 3257_

BGP table version is 6411071, local router ID is 10.70.80.13
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

* 62.10.0.0/15 216.133.69.2 0 65500 3257 8612 i
* 62.24.226.0/23 216.133.69.2 0 65500 3257 8785 8785 24765 i
* 62.24.228.0/23 216.133.69.2 0 65500 3257 8785 8785 24765 i
* 62.24.230.0/23 216.133.69.2 0 65500 3257 8785 8785 24765 i
* 62.26.0.0/15 216.133.69.2 0 65500 3257 12312 i
* 62.40.0.0/19 216.133.69.2 0 65500 3257 8469 i
* 62.64.128.0/17 216.133.69.2 0 65500 3257 9105 i
* 62.79.0.0/16 216.133.69.2 0 65500 3257 8807 i
* 62.111.0.0/17 216.133.69.2 0 65500 3257 12312 20968 20968 20968 20968 20968 20968 20968 i
* 62.116.128.0/19 216.133.69.2 0 65500 3257 12312 15456 i

gw-3640-A#sh ip bgp regexp _65500 3257_

BGP table version is 6411091, local router ID is 10.70.80.13
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

*> 62.10.0.0/15 216.133.69.2 200 65500 3257 8612 i
*> 62.24.226.0/23 216.133.69.2 200 65500 3257 8785 8785 24765 i
*> 62.24.228.0/23 216.133.69.2 200 65500 3257 8785 8785 24765 i
```

```
*> 62.24.230.0/23 216.133.69.2 200 65500 3257 8785 8785 24765 i
*> 62.26.0.0/15 216.133.69.2 200 65500 3257 12312 i
*> 62.40.0.0/19 216.133.69.2 200 65500 3257 8469 i
*> 62.64.128.0/17 216.133.69.2 200 65500 3257 9105 i
*> 62.79.0.0/16 216.133.69.2 200 65500 3257 8807 i

gw-3640-A#
gw-3640-A#sh ip bgp regexp _9034_

BGP table version is 6411741, local router ID is 10.70.80.13

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

*> 62.98.0.0/17 216.133.69.2 0 65500 9034 i
* 217.223.243.9 0 65400 9034 9034 9034i
*> 62.98.128.0/17 216.133.69.2 0 65500 9034 i
* 217.223.243.9 0 65400 9034 9034 9034 i
*> 62.149.128.0/17 216.133.69.2 0 65500 9034 i
* 217.223.243.9 0 65400 9034 9034 9034

gw-3640-A#sh ip bgp regexp _20959_

BGP table version is 6411946, local router ID is 10.70.80.13
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete



Network Next Hop Metric LocPrf Weight Path

*> 80.204.0.0/16 216.133.69.2 200 65500 3269 20959 i
* 217.223.243.9 0 65400 3269 20959 i
*> 80.205.0.0/16 216.133.69.2 200 65500 3269 20959 i
* 217.223.243.9 0 65400 3269 20959 i
*> 80.206.0.0/16 216.133.69.2 200 65500 3269 20959 i
* 217.223.243.9 0 65400 3269 20959 i

gw-3640-A#sh ip bgp 195.210.91.83

BGP routing table entry for 195.210.64.0/19, version 5824765
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Not advertised to any peer

65500 1267
216.133.69.2 from 216.133.69.2 (216.133.69.2)
Origin IGP, localpref 100, valid, external, best
Community: 217120770 217120869

gw-3640-A#sh ip route 207.254.48.0

Routing entry for 207.254.48.0/24
```

```
Known via "bgp 65300", distance 20, metric 0
Tag 65500, type external
Last update from 216.133.69.2 00:00:07 ago
Routing Descriptor Blocks:
* 216.133.69.2, from 216.133.69.2, 00:00:07 ago
Route metric is 0, traffic share count is 1
AS Hops 5

gw-3640-A#sh ip route 207.254.48.0 longer

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

B 207.254.48.0/24 [20/0] via 217.223.243.9, 00:00:07

gw-3640-A#clear ip bgp 217.223.243.9 soft in
```

8.0 STP

8.1 Cos'e' lo Spanning Tree Protocol

Poiche' gli switch sono pensati per lavorare efficacemente su reti di qualsiasi dimensione ci si puo' domandare se supportano configurazioni ridondate tipiche delle reti piu' grandi. In effetti e' possibile ed auspicabile che vengano utilizzate delle topologie a grafo nella progettazione di reti dotate di switch. Ma un grafo implica dei loop, improponibile in quanto un ciclo puo' determinare una saturazione o blocco della rete per la presenza dei pacchetti imprigionati al suo interno (cosa che avverrebbe in particolare con i broadcast generati per popolare la CAM).

Fortunatamente tutti gli switch Cisco sono dotati di algoritmo STP (Spanning Tree Protocol) che estrae un albero dal grafo di switch da noi progettato e implementato disattivando le porte che generano cicli. STP dinamicamente gestira' le varie porte sfruttando quelle inizialmente disattivate in caso di failure di switch o di porte attive.

La trattazione che segue e' basata sul funzionamento degli switch Cisco 'set' based. Si tenga presente che i valori e i range di priorita', la struttura del portID ed altri valori possono variare in funzione del modello di switch e della versione di sistema operativo presente.

8.2 Algoritmo di STP

Sia assegnato il grafo costituito di switch ai suoi nodi. L'algoritmo STP costruisce il suo albero gestendo due problematiche:

- determinazione della root dell'albero, che si chiama 'root bridge'
- determinazione del percorso migliore verso il 'root bridge'

Le informazioni tra gli switch vengono scambiate con broadcast tramite dei pacchetti di livello 2 che si chiamano BPDU (Bridge Protocol Data Unit). Per il momento accontentiamoci di sapere che ogni BPDU contiene un valore di BridgeID. Quest'ultimo e' un numero di 10 byte consistente in un valore di priorita', nei 16 bits piu' significativi, e un MAC address dello switch nei restanti 8 byte. La funzione del BridgeID e' quella di dare una priorita' ad uno switch, che come si vede ha un valore in parte casuale (MAC) e in parte configurabile (i 16 bits di default hanno il valore 32768).

8.3 Determinazione della root dell'albero

Nel momento iniziale ogni switch dichiara di essere il 'root bridge' ed invia BPDU a tutti i suoi vicini. Questi ultimi confrontano il BridgeID ricevuto con quello loro. Poiche' nello STP la regola e' che "cio' che ha valore piu' basso vince" se il BridgeID ricevuto ha un valore piu' basso il ricevente rinuncia al titolo di 'root bridge' e propaga il BridgeID ricevuto ai suoi vicini.

Poiche' i MAC address sono unici alla fine uno ed un solo switch non avra' rinunciato al titolo di 'root bridge'. Il campo priorita' e' impostato per default a 32768. Essendo posizionato nei 16 bits piu' significativi del BridgeID se modificato sara' determinante nell'elezione del root bridge. In effetti e' proprio questo il motivo della sua esistenza: consentire al progettista di poter intervenire nella scelta del root bridge con un parametro settabile. Questo perche' lo switch root dell'albero sara' probabilmente uno switch abbastanza trafficato: e' bene che sia in posizione baricentrica nella rete e sufficientemente performante, non e' opportuno che sia uno switch nella rete di accesso. Senza configurazione di priorita' la scelta del root dipende dal MAC e quindi e' *totalmente casuale*.

8.4 Determinazione del percorso migliore verso il root bridge

Consideriamo lo switch S1 nella nostra rete con STP. S1 non e' root bridge. Supponiamo che S1 riceva BPDU provenienti dal root bridge dalle porte P1 e P2. Si puo' concludere che e' stato determinato un loop e che va eliminato. Per fare questo una delle due porte va bloccata e posta in quello che si chiama 'blocking state'. L'altra fara' passare il traffico e verra' posta in quello che si chiama 'forwarding state'. Una porta disabilitata dal progettista volontariamente si mette in 'disabled state'. In una rete stabile ogni porta e' in uno di questi tre stati ma il passaggio da blocking a forwarding non e' immediato ma avviene attraverso gli stati 'listening' e 'learning'.

Cosi' complessivamente una porta puo' stare in ogni istante in uno di 5 stati in una rete con STP. Nello stato di listening la porta non fa altro che analizzare le BPDU per scegliere la nuova root port. Nello stato di learning la porta impara i MAC address della rete senza fare forwarding di pacchetti.

Resta da chiarire tra le porte P1 e P2 quale vada bloccata e quale no. Basare la scelta sul caso non e' un'alternativa valida in quando la giusta idea e' quella di scegliere la porta associata al percorso piu' valido verso il root bridge. Parametri validi sono la priorita' degli switch lungo questo percorso e la capacita' dei link (per es. se P1 ha un percorso con Fast Ethernet e P2 un percorso con GigaEthernet e' meglio scegliere P2).

L'algoritmo di scelta per le nostre due porte P1 e P2 si basa sul valore di BridgeID gia' introdotto, su un valore di costo che e' settabile per porta e su un valore di PortID che e' anch'esso settabile per porta.

Per default ad ogni porta e' associato un costo che e' inversamente proporzionale alla banda a disposizione sulla stessa. Questo valore non e' lo stesso per tutti gli switch e per tutte le versioni di SO. Al di la del valore specifico l'idea e' che, ad esempio, una porta GigaEthernet avra' priorita' inferiore rispetto una porta Ethernet in quanto la prima e' preferibile (ricordiamo che il valore piu' basso vince). Tale parametro e' modificabile dall'operatore per porta.

Per default ad ogni porta viene associato un PortID. Questo e' un valore a 16 bit di cui 6 sono riservati alla priorita' della porta (variabile da 0 a 63 con 32 di default). I restanti 10 bit sono derivati dal numero logico della porta sullo switch. Tale parametro e' modificabile dall'operatore per porta.

Avendo introdotto tutti i concetti necessari ecco l'algoritmo di scelta che permette di comprendere cosa viene scelto tra le porte P1 e P2 e cosa viene scelto in tutti gli altri casi dove piu' porte hanno path validi per raggiungere il root bridge:

- si presume che il Root BridgeID sia uguale per entrambe le porte in quanto la root della rete e' unica
- la porta che ha il 'path cost' verso il root bridge piu' in basso vince
- la porta che ha il vicino con BridgeID piu' basso vince
- la porta con PortID piu' basso vince

Una BPDU, oltre che al valore di BridgeID dello switch mittente, contiene anche il BridgeID dello switch che lo sta ritrasmettendo nonche' il valore di path cost verso il root switch e il PortID della porta che sta ritrasmettendo la BPDU. Infine la BPDU contiene anche i timers associati allo STP di cui al prossimo paragrafo. Se la porta P1 ha un path cost piu' basso della P2 quest'ultima andra' in blocking. Se i path cost di P1 e P2 sono uguali vincera' il BridgeID piu' basso tra i vicini collegati alle porte P1 e P2. Se anche questi sono uguali vince il PortID piu' basso. Sostanzialmente, solo nella peggiore delle ipotesi la scelta sara' casuale e cioe' determinato dal numero della porta nello switch.

Nel caso in cui piu' switch si trovano nello stesso segmento di rete, ad esempio nel caso di piu' switch collegati ad un hub, la scelta piu' saggia e' eleggere un

rappresentante, chiamato 'designed switch' che si occupi di inviare i dati verso il root bridge. Vince chi ha il costo piu' basso verso il root Bridge.

8.5 Ricostruzione dello Spanning Tree

Non appena raggiunta la stabilita' il traffico passera' sui path dell'albero determinato dallo spanning tree. Resta da chiarire cosa succede se un guasto determina il blocco del traffico su un link. Poiche' le BPDU inviate dal root bridge non arriveranno piu' a tutti gli switch dell'albero quelli interessati si accorgeranno della failure. Poiche' le porte in stato di Blocking comunque ricevono le BPDU dal root sara' possibile capire se si tratta di una indisponibilita' dell'intero root bridge o se vi e' solo un problema in un link. Se uno switch, che per comodita' indichero' con S1, non riceve piu' le BPDU del root da nessuna porta allora dichiarera' di essere il nuovo root switch e partira' un ricalcolo dell'albero. Se invece riceve la BPDU dal root da una porta messa in blocking allora la porta cambiera' il suo stato fino a giungere in forwarding.

Tutto quanto indicato segue precise temporizzazioni:

- Le BPDU vengono inviate per default ogni 2 secondi. Questo tempo si chiama 'hello time' ed e' modificabile dall'operatore tra 1 e 10 secondi
- Se non si ricevono BPDU per un tempo massimo pari per default a 20 secondi si considera perso lo switch mittente. Questo tempo si chiama 'max age' ed e' modificabile dall'operatore tra 6 e 40 secondi
- Se una porta deve passare da blocking a fowarding permane per default per 15 secondi negli stati intermedi di listing e learning. Questo tempo si chiama 'forwarding time' ed e' modificabile dall'operatore tra 4 e 30 secondi

Facendo qualche calcolo si vede che il tempo medio di recovery da un guasto varia tra i 30 e i 50 secondi. Questo si chiama 'tempo di convergenza' ed e' un valore di primaria importanza in quanto indica un tempo in cui la rete in parte non funziona correttamente. Modificando i settaggi e' teoricamente possibile ottenere dei tempi di recovery da fault molto piu' bassi ma l'operazione manuale in questo caso e' da sconsigliare in quanto dipendente da fattori legati alla struttura della rete. Piuttosto vanno seguite le strategie alternative indicate nel paragrafo successivo.

8.6 Ridurre il tempo di convergenza

E' molto piu' comodo e sicuro lasciare al Catalyst il settaggio opportuno indicando esclusivamente il diametro della rete. Tale parametro, per default a 7 che e' il valore massimo, indica il numero di switch che un pacchetto deve attraversare per raggiungere una destinazione nel caso peggiore. Poiche' per default e' pari al valore massimo va corretto senza esitazioni per ottenere tempistiche migliori.

Oltre a questo e' possibile selezionare tre modalita' differenti per migliorare i tempi di convergenza dipendenti dalla topologia e quindi tutte disabilitate per default. Queste sono: port-fast, uplink-fast, backbone-fast.

- PORT-FAST va utilizzato esclusivamente su porte con collegati server e workstation. Non va usato su porte con altri hub o switch collegati. Consente una transizione immediata da 'block state' a 'forwarding state' senza passare per gli stati intermedi. Questi ultimi giovano per prevenire loops che non possono tuttavia verificarsi in porte collegate a PC;
- UPLINK-FAST consente di unificare tutte le root-ports lasciandone una sola attiva. In caso di fail un'altra porta si attivera' e cosi' a seguire. In ogni istante solo una porta e' in forwarding state. Cio' consente di evitare loop pertanto si puo' accelerare il tempo di convergenza con lo stesso criterio del PORT-FAST; in una rete si puo' utilizzare per collegare gli switch dell'access-layer al distribution-layer. Non va usato nel core layer; questo switch non deve diventare root e di default la sua priorita' va a 49152 e il portcost aumenta di 3000; lavora VLAN based;
- BACKBONE-FAST consente di accelerare il tempo di convergenza quando un Designed Switch perde il contatto col root switch; andrebbe sempre utilizzato in una rete;

8.7 Esempio

Un router Cisco, anche di fascia bassa, e' in grado di effettuare bridging tra le sue porte e comportarsi esattamente come uno switch. Un esempio tipico di applicazione si ha nel caso di utilizzo di protocolli che non possono essere gestiti da routers quali Netbios. Ecco un esempio (qui siamo in un Cisco 827):

```
bridge-group 1 protocol ieee
interface Ethernet0
 ip address 192.168.30.1 255.255.255.0
 bridge-group 1
!
interface ATM0
 ip address 151.99.200.10 255.255.255.0
 no atm ilmi-keepalive
 bundle-enable
 dsl operating-mode auto
 bridge-group 1
!
```

Router#show bridge group
Bridge Group 1 is running the IEEE compatible Spanning Tree protocol
Port 3 (ATM0 RFC 1483) of bridge group 1 is forwarding
Port 2 (Ethernet0) of bridge group 1 is forwarding

Router#show spanning brief
Bridge group 1
Spanning tree enabled protocol ieee
Root ID Priority 32768
Address 0004.27fd.41d4
This bridge is the root

```
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 32768
Address 0004.27fd.41d4
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
Interface Designated
Name             Port ID    Prio    Cost    Sts    Cost    Bridge ID    Port ID
---------------- ---------- ----    -------  ---    -----   -------------    -------
Ethernet0        128.2      128     100     FWD    0       32768 0004.27fd.41d4 128.2
ATM0             128.3      128     1562    FWD    0       32768 0004.27fd.41d4 128.3
```

```
Router# show spanning

 Bridge group 1 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 0004.27fd.41d4
  Configured hello time 2, max age 20, forward delay 15
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set
  Number of topology changes 1 last change occurred 00:00:49 ago
        from Ethernet0
  Times:  hold 1, topology change 35, notification 2
        hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0, aging 300

 Port 2 (Ethernet0) of Bridge group 1 is forwarding
   Port path cost 100, Port priority 128, Port Identifier 128.2.
   Designated root has priority 32768, address 0004.27fd.41d4
   Designated bridge has priority 32768, address 0004.27fd.41d4
   Designated port id is 128.2, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 41, received 0

 Port 3 (ATM0) of Bridge group 1 is forwarding
   Port path cost 1562, Port priority 128, Port Identifier 128.3.
   Designated root has priority 32768, address 0004.27fd.41d4
   Designated bridge has priority 32768, address 0004.27fd.41d4
   Designated port id is 128.3, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 41, received 0
```

8.8 Comandi utili

set spantree root [*vlan*] [dia *diameter*] Nota: la priorita' viene settata a 8192
set spantree secondary [*vlan*] [dia *diameter*] Nota: la priorita' viene settata a 16242
set spantree priority *priority* [*vlan*]
set spantree hello *time* [*vlan*] Nota: default 2, range 1-10
set spantree fwddelay *time* [*vlan*] Nota: default 15, range 4-30
set spantree maxage *time* [*vlan*] Nota: default 20, range 6-40

set spantree backbonefast [enable|disable]
set spantree uplinkfast [enable|disable]
set spantree portfast *m/n* [enable|disable]
set spantree portvlanpri *m/n priority vlans* Nota Settabile solo nelle porte trunk (range 0-63)
set spantree portpri *m/n priority* Nota: default a 32 range 0-63
set spantree portvlancost *m/n priority vlans* Nota: Settabile solo nelle porte trunk
set spantree portcost *m/n cost*
show spantree

8.9 Altre funzionalita'

Sfruttando il meccanismo di priorita' per vlan consentito sui trunk e' possibile creare piu' link paralleli funzionanti tra switch con un carico distribuito in funzione della vlan di appartenenza.

9.0 Trunking

In ogni rete ben progettata si ha normalmente l'esigenza di utilizzare piu' VLAN. Una volta definite piu' VLAN sugli switch della nostra LAN resta il problema della comunicazione tra VLAN corrispondenti di switch diversi. Se abbiamo ad esempio due switch, con tre VLAN in ciascuno di essi, potremmo pensare di dover riservare una porta per VLAN per l'aggancio con la VLAN corrispondente dell'altro switch. Una tale soluzione e' assolutamente non scalabile e dispendiosa nell'uso delle porte. In totale infatti dovremmo occupare 6 porte complessive per l'aggancio degli switch. Il VLAN TRUNKING e' la soluzione al problema in quanto consente di far transitare su un unico link fisico i dati di piu' VLAN pur mantenendole separate tra loro logicamente e quindi garantendone l'isolamento reciproco. Ecco allora che i nostri due switch avrebbero un unico collegamento fisico tra loro in cui vengono raccolti e quindi suddivisi logicamente i dati per ogni VLAN. In una topologia con decine di switch e collegamenti TRUNK introdurremo anche il VTP, o Virtual Trunk Protocol. VTP si occupa di propagare tra gli switch le modifiche alla configurazione su una o piu' VLAN inserite in un unico switch, con un ruolo privilegiato di server, che si preoccupera' di comunicarle agli altri (VTP ha anche altre funzionalita' che non tratteremo qui, come la traduzione di VLAN tra Ethernet, TokenRing, ATMLANE).

Un legame tra VTP e TRUNKING sta nel fatto che gli aggiornamenti VTP si propagano tra gli switch nelle porte trunk attraverso multicast.

9.1 Protocolli di trunking

Cisco supporta due protocolli di trunking: ISL e 802.1q. Il primo e' proprietario Cisco mentre il secondo e' standard. Al di la delle differenze nei pacchetti dei due protocolli, ad esempio ISL incapsula i pacchetti ethernet mentre 802.1q inserisce le

sue informazioni all'interno di essi, cio' che c'e' di sostanziale e' che ISL supporta PVST (Per Vlan Spanning Tree) mentre 802.1q non lo supporta.

Gli switch Cisco eseguono una istanza di STP per VLAN. Pertanto esistera' un root switch per ogni VLAN e una topologia di Spanning Tree differente per ogni VLAN. Tale soluzione permette di ottimizzare la rete significativamente specialmente quando le VLAN non sono distribuite uniformemente nella LAN. Questa metodologia si chiama PVST. 802.1q supporta un'unica istanza di STP che prende il nome di CST (common spanning tree). Da cio' si deduce che se abbiamo una rete di switch Cisco non conviene adottare 802.1q. Se abbiamo switch di marca diversa, non essendo ISL standard, dobbiamo usare 802.1q almeno nei collegamenti tra Cisco e switch non ISL. Gli switch Cisco si occuperanno di gestire le disomogeinita' nella rete 'isolando' le parti 802.1q e creando delle nuvole con PVST. ISL non e' supportato nelle interfacce a 10mbps.

Quando si definisce una porta come trunk questa puo' essere settata in una delle modalita' seguenti:

on	forza la porta in trunk mode
off	forza la porta in non-trunk mode
auto (default)	la porta diventa trunk se la porta vicina e' **on** oppure **desirable**
desiderable	la porta diventa trunk se la porta vicina e' **on**, **desirable** o **auto**
nonegotiate	la porta e' trunk ma non si invia alla porta vicina alcuna indicazione. Pertanto la porta vicina va configurata manualmente

Per default tutte le porte sono in **auto**. Pertanto possiamo facilmente attivare il trunk su una porta usando **on**. La comunicazione tra porte corrispondenti si ha col protocollo DTP. In caso di switch di marca diversa i frame DTP potrebbero generale qualche problema. In tal caso usare **nonegotiate** per le porte che devono diventare trunk e **off** le altre porte.

Naturalmente e' possibile configurare trunks anche senza VTP. Alcune funzionalita' pero' non saranno disponibili come ad esempio quelle di negoziazione.

9.2 VTP

Quando si utilizza VTP ogni switch della rete va configurato in uno dei seguenti tre modi:

- server: nello switch e' possibile creare, cancellare o modificare VLAN (condizione di default)
- client: nello switch non e' possibile creare, cancellare o modificare VLAN. Queste informazioni vengono caricare dallo switch dopo il boot da un server e aggiornate successivamente in accordo con le modifiche nello switch server
- transparent: lo switch non utilizza VTP e ignora gli eventuali multicast VTP. Le VLAN vanno configurate in locale come sul server ma non c'e' propagazione

E' possibile definire gruppi differenti di switch che si scambiano informazioni solo al loro interno creando domini differenti. Per default uno switch e' in modalita' server senza un dominio definito. Se riceve dei pacchetti VTP dal dominio assume il dominio presente negli stessi.

Al crescere del numero di switch nella LAN sono diversi i criteri che si possono adottare nella configurazione di VTP:

- In una rete molto piccola e con poche VLAN potrebbe non avere senso usare VTP e quindi gli switch potrebbero essere tutti posti in modalita' transparent
- Al crescere della rete, tutti gli switch possono essere messi in modalita' server, cosi' da poter operare su ognuno di loro ma con la certezza dell'uniformita' delle informazioni tra tutti gli switch
- Al crescere della rete e delle VLAN le informazioni relative alle stesse potrebbero superare una certa soglia, da valutare caso per caso, per cui la NVRAM occupata negli switch diventerebbe significativa. In questo caso andrebbero ridotti gli switch server e creati un numero crescente di client, che preservano la NVRAM in quanto scaricano le informazioni tramite VTP dopo il boot

Un importante parametro propagato con VTP e' il 'configuration number'. Si tratta di un numero progressivo che cresce ogni volta che cambia una definizione VTP. Gli switch della rete, ricevendo pacchetti con un numero piu' alto di quello attuale, si accorgono che le informazioni in arrivo sono piu' aggiornate e quindi ne valutano il contenuto. Non e' un parametro settabile ma piuttosto una potenziale fonte di seri problemi quando si svolgono certe attivita' di manutenzione sulla rete.

Da notare che il numero di configurazione viene memorizzato nella NVRAM e chi ha il numero piu' alto decide per il cambiamento della configurazione di tutti gli altri switch. Introdurre in una nuova rete uno switch con un numero di configurazione troppo elevato potrebbe far perdere la configurazione VLAN per tutta la rete! Ogni nuovo switch e' bene che abbia il configuration number azzerato (lo si puo' fare cambiando il dominio e riavviando).

Le informazioni VTP viaggiano nella VLAN 1 che pertanto non deve essere eliminata per garantirne il funzionamento. VTP aggiorna ogni 5 minuti con multicast.

Le informazioni VTP come detto si propagano nelle porte trunk. Per far arrivare le definizioni di VLAN a tutti gli switch questi devono essere collegati tra loro con trunks. Non si possono avere reti dove switch client o server non siano collegati tra loro da trunk. Paradossalmente anche se tutte le porte di uno switch sono nella stessa VLAN e' necessaria una porta trunk. Questo in linea di principio non e' un problema ma un vantaggio in quanto da flessibilita' e scalabilita' alla rete. Tuttavia:

"in un trunk viaggiano informazioni su tutte la VLAN disponibili anche su quelle non presenti nello switch di destinazione. Broadcast, multicast e informazioni VTP su tutte le VLAN arrivano a tutti"

Per risolvere il problema con VTP e' configurabile il 'pruning'. In questo modo automaticamente, ma in modo configurabile, ad ogni switch arrivera' solo il traffico relativo alle VLAN effettivamente in uso. In una trattazione successiva di questo documento daro' una descrizione piu' dettagliata.

9.3 Comandi

set vtp domain dominio
set vtp mode [server|client|transparent]
set vtp passwd password
set vtp v2 [enable|disable]
set vtp pruning [enable|disable]
set vtp pruneeligible vlan-range
set trunk mod/port [on|desirable|auto|off|nonegotiate] [isl|dot1q|negotiate]
set trunk mod/port vlans
clear trunk mod/port vlans
clear vtp pruneeligible vlan-range
show vtp domain
show vtp statistics
show trunk

9.4 Altre funzionalita'

Sfruttando il meccanismo di priorita' per vlan e trunk configurati opportunamente e' possibile creare piu' link paralleli tra switch con un traffico distribuito in funzione della vlan di appartenenza. Per una descrizione dettagliata di tale procedimento vedere nel sito Cisco.

Il VTP versione 2 di default e' disabilitato. Puo' essere utile fondamentalmente in presenza di reti token-ring. In ogni caso consente, a differenza della versione 1, la propagazione di informazioni VTP attraverso gli switch in 'transparent' mode, cosa che non avviene di default.

10.0 HSRP

Quando e' necessaria ridondanza e quindi alta affidabilita' la soluzione migliore e' quella di utilizzare coppie di apparati o di alcuni loro moduli. Poiche' tuttavia apparati diversi hanno indirizzi layer 3 differenti sarebbe assai utile un meccanismo che consentisse di utilizzare un unico indirizzo layer 3 in grado di 'spostarsi' sempre sull'apparato funzionante. HSRP consente di svincolare gli indirizzi IP dalle singole interfacce fisiche e di associarli invece a gruppi di interfacce fornendo cosi' protezione da fault su queste interfacce.

10.1 Principio di funzionamento

Un caso tipico di utilizzo di HSRP e' quando un router R1, gateway di una rete, essendo questa rete ad alta affidabilita' va ridondato. La presenza di due gateway nella rete, diciamo due Cisco 2600, in circostanze normali ci forza ad assegnare due indirizzi IP distinti alle loro interfacce ethernet. Questo implica che, per avere ridondanza, dovremmo assegnare due indirizzi IP di gateway ai server di rete. Il problema sta nel fatto che tale configurazione non funziona correttamente in quanto i sistemi operativi nella maggior parte dei casi non gestiscono efficacemente la configurazione di piu' gateway.

Lasciamo inalterata la configurazione delle ethernet dei nostri Cisco 2600 e aggiungiamo invece un elemento nuovo e cioe' HSRP. La configurazione HSRP ci consentira' di introdurre un nuovo indirizzo IP che verra' associato ad un nuovo MAC e che diventera' l'ip del gateway di riferimento per i client e i server che pertanto diventa unico. Se un 2600 fallisce dal lato ethernet per un guasto di rete o interno HSRP assegnera' IP e MAC al secondo 2600 garantendo la continuita' di servizio per la rete.

10.2 Configurazione

La configurazione di HSRP fa uso del comando "standby ip":

standby [group-number] **ip** ip-address [**secondary**]
standby [group-number] **mac-address** mac-address
standby [group-number] **priority** priority [**preempt** [**delay** delay]]
standby [group-number] **track** type number [priority]

Configuriamo il primo 2600, diciamo 2600-A, nel modo seguente:

```
description - router 2600A -
interface FastEthernet0/0
  ip address 192.168.30.10 255.255.255.0
  duplex auto
  speed auto
  standby 1 priority 105 preempt
  standby 1 ip 192.168.30.1
  standby 1 track Fa0/1
```

L'indirizzo fisico e' 192.168.30.10. Supponiamo che il 2600-B abbia indirizzo ethernet 192.168.30.11. Le tre righe con **standby** abilitano HSRP. Un valore di priorita' definisce il router che ha precedenza sugli altri per acquisire l'indirizzo virtuale, in questo caso il 192.168.30.1. Nel caso di un gruppo di router l'utilita' di una priority e' abbastanza ovvia. Il comando preempt fa si che, in caso di fault e passaggio al 2600-B, si torni sempre al 2600-A quando questo ritorna on-line. L'indirizzo 192.168.30.1 e' quello che va impostato come gateway sui client e server della rete.

L'ulima riga di configurazione contiene il subcomando **track.** Questo consente di variare la priorita' di un router a seconda dello stato delle sue interfacce. Ad esempio se la seconda ethernet del 2600-A (supponendo che ne abbia una) va in down allora il valore di priorita' del router scende di 10 punti (default) portandolo a 95. Questo determina il passaggio dell'ip virutale al 2600-B (che stiamo supponendo con priorita' 100). Insomma con HSRP possiamo determinare condizioni di fault non solo in base all'interfaccia su cui HSRP e' attivo ma anche in base al comportamento di altre interfacce presenti sul router, ad esempio una seriale. E' quindi facile pensare che se il 2600-A ha un collegamento seriale verso internet che passa nello stato di *protocol down* il gateway cambia pur avendo una interfaccia ethernet perfettamente funzionante. Infine 1 e' il numero di gruppo. E' possibile creare piu' gruppi HSRP ognuno dei quali gestisce in autonomia delle interfacce e degli ip virtuali.

Ma secondo quale criterio un router con HSRP classifica come off-line gli altri router del suo gruppo? Le porte di uno stesso gruppo si scambiano pacchetti di *hello* ogni tre secondi. Se per tre volte questo valore (holdtime) non arrivano i pacchetti di hello il router mittente viene considerato in fault. Questi parametri sono settabili. Si deduce che di default sono necessari almeno 10 secondi per passare da un router all'altro in caso di fault ma questo tempo puo' variare a seconda della configurazione di HSRP e della topologia di rete.

Per finire ecco dei comandi per il monitoring e il debugging:

```
Router#show standby
Ethernet0 - Group 1
Local state is Active, priority 100, may preempt
Hellotime 3 holdtime 10
Next hello sent in 00:00:02.272
Hot standby IP address is 192.168.30.5 configured
Active router is local
Standby router is unknown expired
Standby virtual mac address is 0000.0c07.ac01
2 state changes, last state change 00:01:23
Router#debug standby
HSRP debugging is on
Router#
00:40:33: SB1: Et0 Hello out 192.168.30.1 Active pri 100 ip 192.168.30.5
00:40:36: SB1: Et0 Hello out 192.168.30.1 Active pri 100 ip 192.168.30.5
00:40:39: SB1: Et0 Hello out 192.168.30.1 Active pri 100 ip 192.168.30.5

Router#sh run int eth0
Building configuration...

Current configuration : 112 bytes
!
interface Ethernet0
ip address 192.168.30.1 255.255.255.0
standby 1 preempt
```

```
standby 1 ip 192.168.30.5
end
```

11.0 Configurazione di base degli switch Cisco

In questo documento trattero' della configurazione di base degli switch Cisco IOS-based. Il modello preso in esame e' un Catalyst 2900XL ma i concetti introdotti possono essere applicati con successo a molti altri switch Cisco della stessa fascia. Il modello preso in esame ha un IOS 12.0(5)2XU.

11.1 Start-up

Ecco la sequenza di boot e la configurazione di fabbrica del prodotto preso in esame:

```
Boot Sector Filesystem (bs:) installed, fsid: 3
Parameter Block Filesystem (pb:) installed, fsid: 4
Loading "flash:c2900XL-c3h2s-mz-120.5.2-
XU.bin"...##########################################################################
file "flash:c2900XL-c3h2s-mz-120.5.2-XU.bin" uncompressed and installed, entry point: 0x3000
executing...
Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco Internetwork Operating System Software
IOS (tm) C2900XL Software (C2900XL-C3H2S-M), Version 12.0(5.2)XU, MAINTENANCE INTERIM SOFTWARE
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Mon 17-Jul-00 17:35 by ayounes
Image text-base: 0x00003000, data-base: 0x00301F3C

Initializing C2900XL flash...
flashfs[1]: 108 files, 3 directories
flashfs[1]: 0 orphaned files, 0 orphaned directories
flashfs[1]: Total bytes: 3612672
flashfs[1]: Bytes used: 2776576
flashfs[1]: Bytes available: 836096
flashfs[1]: flashfs fsck took 7 seconds.
flashfs[1]: Initialization complete.
...done Initializing C2900XL flash.
C2900XL POST: System Board Test: Passed
C2900XL POST: Daughter Card Test: Passed
C2900XL POST: CPU Buffer Test: Passed
C2900XL POST: CPU Notify RAM Test: Passed
C2900XL POST: CPU Interface Test: Passed
```

```
C2900XL POST: Testing Switch Core: Passed
C2900XL POST: Testing Buffer Table: Passed
C2900XL POST: Data Buffer Test: Passed
C2900XL POST: Configuring Switch Parameters: Passed
C2900XL POST: Ethernet Controller Test: Passed
C2900XL POST: MII Test: Passed
cisco WS-C2924-XL (PowerPC403GA) processor (revision 0x11) with 8192K/1024K bytes of memory.
Processor board ID FOC0520Y0KB, with hardware revision 0x01
Last reset from power-on
Processor is running Enterprise Edition Software
Cluster command switch capable
Cluster member switch capable
24 FastEthernet/IEEE 802.3 interface(s)
32K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 00:06:28:74:40:40
Motherboard assembly number: 73-3382-08
Power supply part number: 34-0834-01
Motherboard serial number: FOC052000FZ
Power supply serial number: PHI051105F7
Model revision number: N0
Motherboard revision number: C0
Model number: WS-C2924-XL-EN
System serial number: FOC0520Y0KB
C2900XL INIT: Complete
00:00:28: %SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
IOS (tm) C2900XL Software (C2900XL-C3H2S-M), Version 12.0(5.2)XU, MAINTENANCE INTERIM SOFTWARE
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Mon 17-Jul-00 17:35 by ayounes
--- System Configuration Dialog ---
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
Continue with configuration dialog? [yes/no]: no
Press RETURN to get started.
```

```
Switch#sh run
Building configuration...
Current configuration:
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch
!
!
ip subnet-zero
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
```

```
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
!
interface FastEthernet0/14
!
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface VLAN1
no ip directed-broadcast
no ip route-cache
!
!
line con 0
transport input none
stopbits 1
line vty 5 15
!
end
```

11.2 Configurazione delle password

Nell'assegnazione delle password si utilizzano gli stessi comandi conosciuti per i router. Per l'accesso in modalita' privilegiata:

```
Switch(config)#enable secret cisco
```

Per l'accesso da telnet e' necessario l'impostazione della password sulle linee di terminale virtuale:

```
Switch(config)#line vty 0 4
              password cisco
              login
```

11.3 Amministrazione remota

Notate la configurazione di default delle VLAN nello switch:

```
mioswitch#sh vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4,
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12,
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16,
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20,
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        1002   1003
1002 fddi  101002     1500  -      -      -        -    -        1      1003
1003 tr    101003     1500  1005   0      -        -    srb      1      1002
1004 fdnet 101004     1500  -      -      1        ibm  -        0      0
1005 trnet 101005     1500  -      -      1        ibm  -        0      0
```

Tutte le porte sono nella VLAN1. Il 2900XL e' uno switch layer2 pertanto non e' in grado di fare routing. La configurazione layer 3 si limita all'assegnazione di un indirizzo ip e di un gateway che ne consentono l'amministrazione da remoto, tramite telnet, http o CiscoWorks. L'amministrazione e' possibile dagli apparati presenti nella VLAN1, la vlan di default, che per questo si chiama anche 'vlan di management'. In questo esempio assegnamo allo switch l'IP 192.168.30.6/24 e gateway 192.168.30.1:

```
mioswitch(config)#int vlan1
mioswitch(config-if)#ip address 192.168.30.6   255.255.255.0
```

```
mioswitch(config)#ip default-gateway   192.168.30.1
```

E' possibile cambiare la VLAN di management ove necessario. Creiamo una nuova VLAN, la due, configuriamo l'indirizzo ip e il gioco e' fatto. E' possibile una sola VLAN di management.

```
interface VLAN1
 no ip address
 no ip directed-broadcast
 no ip route-cache
 shutdown
!
interface VLAN2
 ip address 192.168.30.6 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
!
```

Adesso lo switch e' amministrabile da tutte le postazioni nella VLAN1 e anche al di fuori della LAN, se il gateway e' nella VLAN1. Il Catalyst 2900 e' dotato di server web ed e' configurabile nelle sue funzionalita' di base tramite browser quale Netscape Communicator o Internet Explorer. Per quest'ultima opzione e' necessario attivare il server web come segue:

```
mioswitch(config)#ip http server
```

L'accesso web e' protetto da nome utente e password. Lasciare il campo username in bianco se nella configurazione non sono stati definiti degli utenti. Utilizzare la password di enable. L'accesso web andrebbe protetto definendo gli indirizzi ip delle postazioni di management ed impedendone la gestione a tutti gli altri. A tale scopo si puo' utilizzare un'access-list tramite il comando "ip http access-class":

```
mioswitch(config)#ip http access-class ?
  <1-99>  Access list number
```

I file html del server web sono nella flash dello switch:

```
mioswitch#dir
Directory of flash:/

  2  -rwx     1645810   Jul 18 2000 01:26:29  c2900XL-c3h2s-mz-120.5.2-XU.bin
  3  -rwx      105970   Jul 18 2000 01:26:29  c2900XL-diag-mz-120.5.2-XU
  4  drwx        6784   Jul 18 2000 01:26:30  html
111  -rwx        1296   Mar 01 1993 00:48:59  config.text
112  -rwx         272   Jan 01 1970 00:00:26  env_vars

3612672 bytes total (834560 bytes free)
mioswitch#dir html
Directory of flash:/html/

  5  drwx           0   Jul 18 2000 01:26:29  Snmp
  6  -rwx         656   Jul 18 2000 01:26:29  ClusterBuilder.html.gz
  7  -rwx         613   Jul 18 2000 01:26:29  ClusterManager.html.gz
  8  -rwx        1413   Jul 18 2000 01:26:29  Graph.html.gz
```

```
  9  -rwx          211   Jul 18 2000 01:26:29  back.html.gz
 10  -rwx          253   Jul 18 2000 01:26:29  basiccfg.html.gz
 11  -rwx          636   Jul 18 2000 01:26:29  switchmgr.html.gz
 12  -rwx          185   Jul 18 2000 01:26:29  blank.html.gz
 13  -rwx          989   Jul 18 2000 01:26:29  cluster.html.gz
 14  -rwx          250   Jul 18 2000 01:26:29  menu.html.gz
 15  -rwx          347   Jul 18 2000 01:26:29  port.html.gz
 16  -rwx          331   Jul 18 2000 01:26:29  cv.html.gz
 17  -rwx          860   Jul 18 2000 01:26:29  popup.html.gz
 18  -rwx          343   Jul 18 2000 01:26:29  Detective.html.gz
 19  -rwx          787   Jul 18 2000 01:26:29  DrawGraph.html.gz
 20  -rwx          803   Jul 18 2000 01:26:29  GraphFrame.html.gz
... snip ...
```

Per l'amministrazione remota tramite snmp si utilizza il comando "snmp-server". Questo e' utile quando si utilizzano applicazioni per l'amministrazione o il monitoraggio da remoto come mrtg o CiscoWorks. Notate le password di sola lettura (RO) e di lettura/scrittura (RW) rispettivamente 'cisco' e cisco2'

```
snmp-server community cisco RO
snmp-server community cisco2 RW
snmp-server community pluto view v1default RO
snmp-server location catania
snmp-server contact gianrico
snmp-server host 192.168.30.45 trap pluto  tty config
```

11.4 Parametri di base: duplex e speed

Ad ogni porta e' associata una interfaccia di tipo FastEthernet. Lo stato di tale interfaccia si mostra col comando: 'show interface fasteth0/N' dove N e' il numero di porta dello switch:

```
 FastEthernet0/1 is up, line protocol is up
 Hardware is Fast Ethernet, address is 0006.2874.4041 (bia 0006.2874.4041)
 MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation ARPA, loopback not set
 Keepalive not set
 Auto-duplex (Full), Auto Speed (100), 100BaseTX/FX
 ARP type: ARPA, ARP Timeout 04:00:00
 Last input never, output 00:00:00, output hang never
 Last clearing of "show interface" counters never
 Queueing strategy: fifo
 Output queue 0/40, 0 drops; input queue 0/75, 0 drops
 5 minute input rate 5000 bits/sec, 10 packets/sec
 5 minute output rate 98000 bits/sec, 17 packets/sec
    4674 packets input, 383089 bytes
    Received 229 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog, 8 multicast
    0 input packets with dribble condition detected
```

```
     7451 packets output, 5149573 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Configuriamo velocita' e duplex per la porta 1. Di default le porte hanno abilitata l'autonegoziazione ovvero il riconoscimento automatico della velocita' della Ethernet e del duplex. Ecco come si forza rispettivamente il duplex e la velocita':

```
mioswitch(config)#int fast0/1
mioswitch(config-if)#description --- router gateway internet ---
mioswitch(config-if)#duplex ?
  auto  Enable AUTO duplex configuration
  full  Force full duplex operation
  half  Force half-duplex operation

mioswitch(config-if)#duplex auto
mioswitch(config-if)#speed ?
  10    Force 10 Mbps operation
  100   Force 100 Mbps operation
  auto  Enable AUTO speed configuration

mioswitch(config-if)#speed auto
```

11.5 Assegnazione delle VLAN alle porte

L'assegnazione di una porta in una VLAN si effettua col comando 'switchport'. La VLAN viene creata automaticamente con l'assegnazione della stessa ad una porta. Nell'esempio a seguire associamo la VLAN numero due alle porte 1 e 2 dello switch:

```
interface FastEthernet0/1
 description --- router gateway internet ---
 switchport access vlan 2
!
interface FastEthernet0/2
 switchport access vlan 2
```

In questo secondo esempio creiamo una terza VLAN, la numero 10. Ecco il risultato:

```
mioswitch# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
mioswitch(config)#int fast0/15
mioswitch(config-if)#switch access vlan 10
mioswitch#sh vlan brief
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/3, Fa0/4, Fa0/5, Fa0/6,
                                                Fa0/7, Fa0/8, Fa0/10, Fa0/11,
                                                Fa0/12, Fa0/13, Fa0/14, Fa0/16,
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20,
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
2    VLAN0002                         active    Fa0/1, Fa0/2, Fa0/9
10   VLAN0010                         active    Fa0/15
```

```
1002 fddi-default                       active
1003 token-ring-default                 active
1004 fddinet-default                    active
1005 trnet-default                      active
```

Lo spanning-tree e' abilitato di default. Nelle LAN senza la presenza di loop puo' anche essere disabilitato.Nelle LAN con loop e' indispensabile. In ogni caso conviene disabilitarlo sulle porte non collegate ad altri switch in quanto queste non possono generare loop. In quest'ultimo caso la porta iniziera' il forwarding dei pacchetti da subito invece di necessitare di circa 30 secondi dovuti alla presenza di STP. Se l'argomento non e' chiaro consiglio la lettura del tutorial sullo Spanning-Tree.

Per disabilitare STP si utilizza il comando 'spanning-tree portfast'. Nell'esempio a seguire utilizziamo il comando "debug spantree event" per analizzare il portfast:

```
Ecco cosa succede col portfast disattivo sulla fastethernet 0/20:

mioswitch#debug spantree event
Spanning Tree event debugging is on
mioswitch#debug spantree tree
Spanning Tree BPDU debugging is on
mioswitch#
01:28:08: ST: FastEthernet0/20 vlan 1 -> listening
01:28:08: %LINK-3-UPDOWN: Interface FastEthernet0/20, changed state to up
01:28:08: The port state changed due to the interface up/down on interface FastE
thernet0/20
01:28:09: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/20, chan
ged state to up
01:28:23: port = 0, old = 0, new = 0
01:28:23: ST: FastEthernet0/20 vlan 1 -> learning
01:28:38: port = 0, old = 0, new = 0
01:28:38: ST: FastEthernet0/20 vlan 1 -> forwarding
Ecco cosa succede col portfast attivo sulla fastethernet 0/2:
01:29:45: ST: FastEthernet0/2 vlan 2 ->jump to forwarding from blocking
01:29:45: port = 0, old = 0, new = 0
01:29:45: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
01:29:45: The port state changed due to the interface up/down on interface FastE
thernet0/2
01:29:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, chang
ed state to up
interface FastEthernet0/1
 description --- router gateway internet ---
 switchport access vlan 2
 spanning-tree portfast
!
interface FastEthernet0/2
 switchport access vlan 2
 spanning-tree portfast
```

Se lo switch e' l'unico della rete o se gli switch hanno una topologia priva di loop e' possibile disabilitare lo STP per tutte le porte. Poiche' il tipo di STP e' PVST (ovvero vi e' una istanza per VLAN) ecco l'esempio per disattivarlo nella VLAN numero 10:

```
!
no spanning-tree vlan 10
ip subnet-zero
!
```

Fare molta attenzione quando si disabilita lo Spanning-Tree. Accertarsi che la topologia lo consente.

11.6 Sicurezza

La sicurezza sulle porte consente la protezione della rete da intrusioni provenienti dalla LAN stessa piu' che dall'esterno.
Le intrusioni provenienti dall'esterno di una LAN si gestiscono tramite firewall.
Supponiamo per esempio di gestire la LAN dell'Universita' di Scienze dell'Informazione di Pennyville. Senza che nessuno ci avverta vi sono continui inserimenti di HUB nella LAN e la reale topologia della rete sfugge ormai al nostro controllo. Questo e' da evitare. Per correre ai ripari forziamo lo switch a mappare un solo MAC su ogni porta che dev'essere collegata ad un PC. Se si viola la regola la porta si disattiva automaticamente:

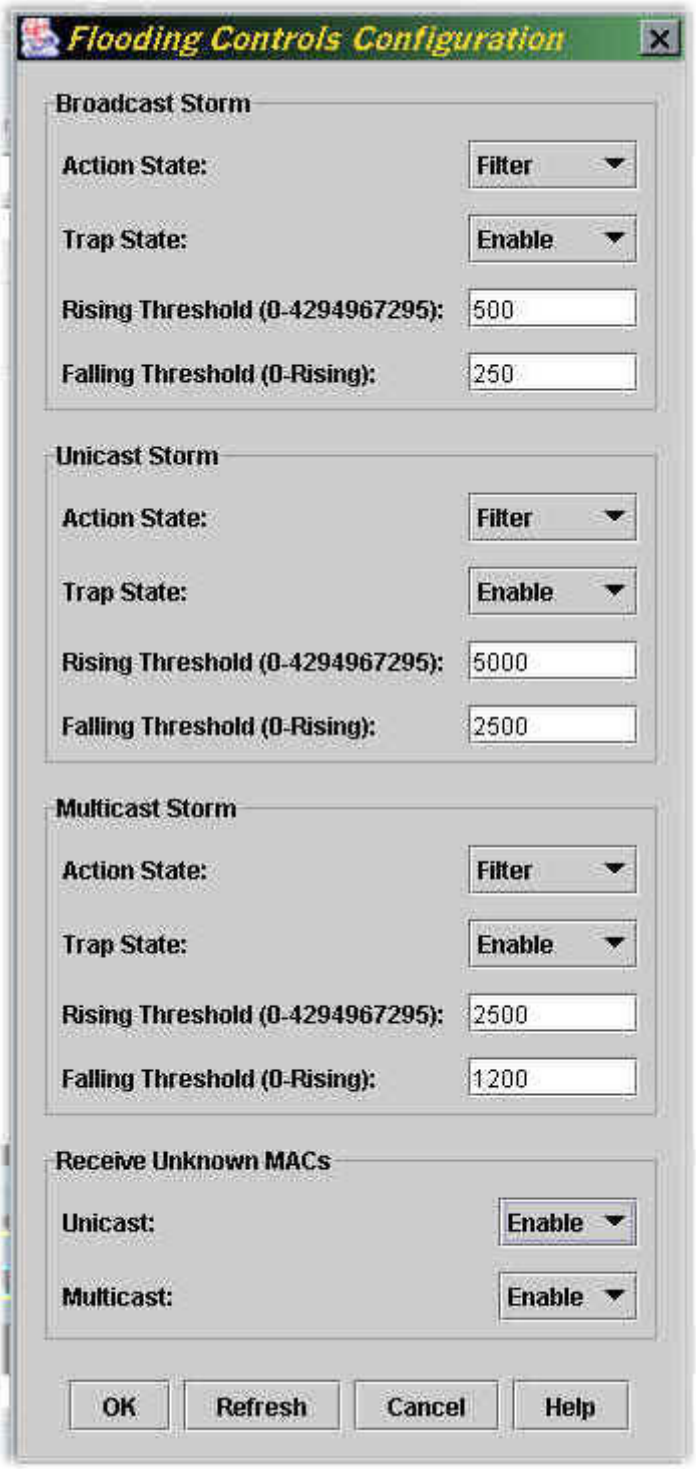
```
mioswitch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
mioswitch(config)#int fast0/20
mioswitch(config-if)#port security action ?
  shutdown  shut down the port from which security violation is detected
  trap      send snmp trap for security violaiton

mioswitch(config-if)#port security ?
  action         action to take for security violation
  max-mac-count  maximum mac address count
  <cr>

Current configuration:
!
interface FastEthernet0/20
 port security max-mac-count 1
 port security action shutdown
end
```

Proteggiamo adesso lo switch da flooding di traffico dovuti, ad esempio, da un attacco DDoS, o da un componente di rete malfunzionante. Nel caso in cui i pacchetti per secondo superano una soglia viene inviata una segnalazione tramite trap snmp. La segnalazione avviene sia nel caso di traffico unicast che multicast o broadcast. Ecco un esempio di configurazione realizzata tramite la comoda interfaccia web in dotazione:

```
interface FastEthernet0/13
 port storm-control filter
 port storm-control trap
 port storm-control broadcast action filter
 port storm-control broadcast trap
```

```
port storm-control multicast action filter
port storm-control multicast trap
port storm-control unicast action filter
port storm-control unicast trap
```

Nell'immagine che segue si possono interpretare i valori (qui sono quelli di default) relativi a questo esempio. I valori sono espressi in termini di pacchetti per secondo:



11.7 Risoluzione dei problemi e span

Se la rete non si comporta come dovrebbe e vogliamo monitorare esattamente tutti i frame in transito in una porta possiamo utilizzare lo "span". Si tratta di duplicare il traffico di una porta su una seconda, libera cioe' non collegata fisicamente cosi' da porter agganciare il nostro strumento di monitoring, normalmente uno sniffer.

Le due porte devono essere nella stessa VLAN. Ecco come duplicare il traffico della Fast0/4 sulla 0/3.

```
interface FastEthernet0/3
 port monitor FastEthernet0/4
!
```

11.8 Per chi usa i telefoni IP

Anche se non e' un argomento di base ecco, per chi utilizza nella propria rete i telefoni IP, come gestire la seconda VLAN, riservata al traffico voce, su una stessa porta.

```
IPphone collegato alla fasteth0/18 con VLAN 10. Traffico dati nella stessa
interfaccia nella VLAN di
default:
interface FastEthernet0/18
 switchport priority default 0
 switchport voice vlan 10        <---- traffico voce va qui
!
mioswitch(config-if)#switchport priority default ?
  <0-7>  Priority for untagged frames (7 is highest)
```

```
In questo secondo tipo di configurazione, tutto il traffico va nella vlan
nativa:
mioswitch(config-if)#switchport voice vlan ?
  <1-4094>  Vlan for voice traffic
  dot1p     Priority tagged on PVID
  none      Don't tell telephone about voice vlan
  untagged  Untagged on PVID

mioswitch(config-if)#switchport voice vlan dot1p
```

11.9 Etherchannel

Per Etherchannel si intende la possibilita' di raggruppare piu' porte al fine di creare un unico flusso di dati di maggior capacita'. Un uso tipico e' per il collegamento tra switch. Raggruppare piu' porte a 100mbps, per esempio, puo' voler dire avere un canale dati di 200, 300 o piu' mbps, ideale per una dorsale collegante piu' switch. Se non si hanno a disposizione porte gigaethernet e gli switch sono tutti 10/100 e' una buona soluzione per LAN di piccole dimensioni. Naturalmente ai due capi dell'Etherchannel gli switch devono essere configurati allo stesso modo. Ecco nell'esempio come raggruppare tre porte, dalla 0/10 alla 0/12 e creare il gruppo 1, che rappresenta l'Etherchannel:

```
interface FastEthernet0/10
 port group 1
```

```
!
interface FastEthernet0/11
 port group 1
!
interface FastEthernet0/12
 port group 1
!
```

11.10 Trunk

Per trunking si intende la possibilita' di far transitare piu' VLAN sullo stesso canale fisico. Cio' accade nelle reti con piu' VLAN che transitano nelle singole dorsali fisiche.

Laddove vi sono piu' switch e si usano le VLAN e' pressocche' indispensabile usare i trunk nei collegamenti switch-switch altrimenti si dovrebbe riservare una dorsale per ogni VLAN. Vi sono due protocolli che gestiscono l'incapsulamento delle VLAN sullo stesso canale fisico: ISL e 802.1q. Il primo e' proprietario Cisco, il secondo standard. Ecco la porta 0/22 configurata con trunk 802.1q:

```
mioswitch#sh run int fast0/22
Building configuration...

Current configuration:
!
interface FastEthernet0/22
 switchport trunk encapsulation dot1q
 switchport mode trunk
end
```

11.11 Configurazione da interfaccia WEB

Le funzionalita' di base dello switch possono essere configurate da web. Cio' e' molto utile per creare una configurazione iniziale con rapidita' e lasciare solo ad un uso successivo l'interfaccia caratteri. Abbiamo gia' spiegato come configurare lo switch per consentirne l'amministrazione da web. Lo switch in oggetto ha ip 192.168.30.6

12.0 Configurazioni per collegamenti ADSL

Esempi su come configurare i router ADSL Cisco per l'interfacciamento con gli operatori quali TELECOM, I.NET, EDISONTEL etc.

gianrico.fichera@itesys.it

12.1 Esempi di configurazioni per l'ufficio

Tutto cio' che serve per usare l'ADSL per dare Internet a tutto l'ufficio

12.2 Glossario dei termini piu' usati

Piccolo dizionario di termini

gianrico.fichera@itesys.it