

RIPv1 nei router Cisco

Livello di preparazione: l'articolo presume una conoscenza di base della configurazione dell'IOS Cisco e delle problematiche di base del routing dinamico

Revisione: Aprile 2004

Copyright: 2003-2004 Gianrico Fichera

Il routing dinamico

Nel caso di piccole 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.

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, di cui non si parlera' in questo documento, ma 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, 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.

RIP utilizza la porta 520 dell'UDP.

Configurazione

Partiamo subito con 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' classfull 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.

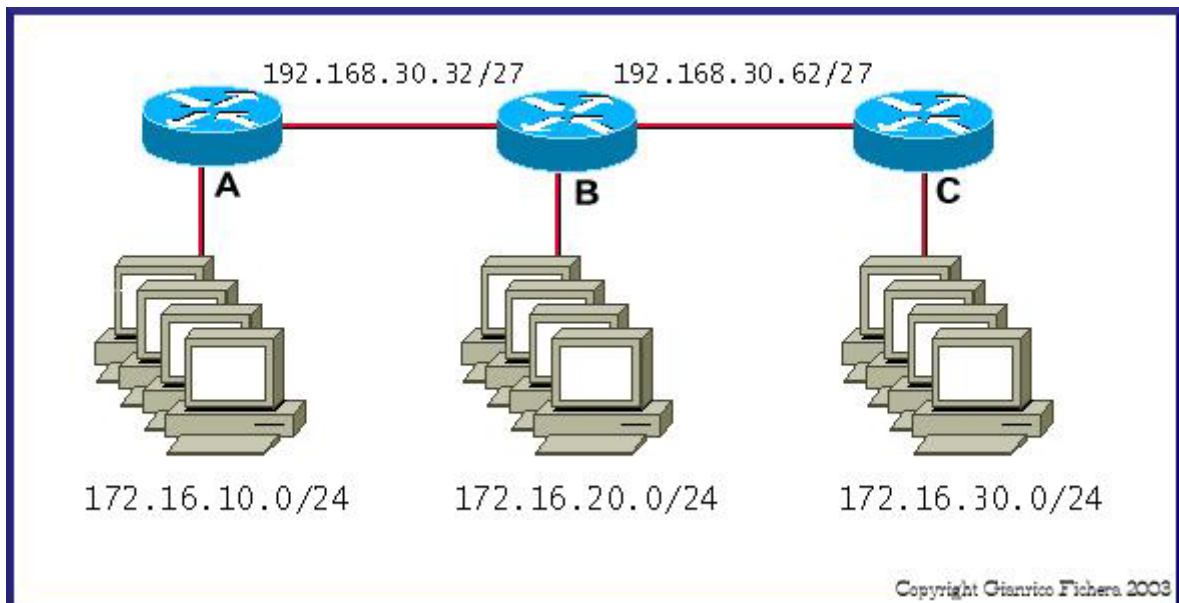
RIP manda l'intera tabella di routing ogni 30 secondi con broadcast 255.255.255.255 su tutte le interfacce su cui e' attivo. Se vi e' un cambiamento in una tabella di routing, ad esempio perche' abbiamo cambiato la classe IP di una interfaccia ethernet, questo si propaga subito, come nel caso della rete 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 periodicit  non e' valida per il "flash update".

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 ... La risposta piu' semplice e' usare RIPv2. Ma puo' non essere disponibile nei nostri apparati. In effetti ci sono dei limiti all'interno dei quali si possono usare subnet. RIP consente di far uso di subnetting anche se con pesanti limitazioni. Tecnicamente diciamo che RIP e' un algoritmo FLSM (Fixed Length Subnet Mask) ovvero consente di utilizzare subnetting ma mantenendo la netmask fissa per 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 e questa 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 e' corretta. 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. Possiamo cosi' risparmiare un buon numero di indirizzi IP sulle 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: non appena arrivera' l'aggiornamento questo verra' scartato. In tali casi quantomeno utilizzare delle statiche per queste reti. E' questo che si intende per *continuita' nel subnetting*. Tutto il discorso fatto vale per RIP v.1. Nel RIP v.2 che consente di evitare queste limitazioni in quanto RIP v.2 non e' FLSM ma VLSM (Variable Length 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.

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 routes 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 routes quindi non vanno piu' utilizzate. A questo punto inizia il processo di cancellazione per le routes giunte da R1. Per queste la 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 nella tabella di routing come "possibly-down". Anche in questo stato ancora 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 7 minuti. Non e' sicuramente un tempo breve ma e' motivato dalla necessita' di evitare i routing loops. Questo si chiama "tempo di convergenza" ovvero il tempo necessario affinche' la rete ritorni stabile dopo un guasto.

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
```

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.

Redistribuzione

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.

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
```

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