Routing

Introduzione

Il routing sulla rete avviene grazie all'indirizzamento *gerarchico*: ogni router non conosce tutte le possibili destinazioni. Routers di *livello basso* conoscono solo le sottoreti che riescono a raggiungere; salendo di livello, routers di uno stesso ISP conoscono tutta la rete interna in modo generale perchè informazioni dettagliate vengono *girate* ai routers più interni, di livello *inferiore*. Il meccanismo si ripete e, salendo via via di livello, si conoscono sempre meno informazioni specifiche, ma solo generali.

Routers e LAN aggregati in regioni prendono il nome di **Autonomous System** quando sono controllati dalla stessa amministrazione; in un AS:

. i routers interni usano protocolli di instradamento IGP (Interior Gateway Protocol);

. esistono routers gateway che effettuano collegamenti fra vari AS e usano protocolli di instradamento EGP (Exterior Gateway Protocol).

Poichè esistono due tipi di routing (*interno* con IGP ed *esterno* con EGP) i *confini* fra gli AS devono essere chiari. Inoltre le caratteristiche delle due tipologie sono diverse:

. se interni: metrica del percorso più breve;

. se esterni: metrica della **politica** migliore (può essere economica e gestionale, ma anche orientata alla sicurezza del traffico).

Interior Routing Protocols

I protocolli di routing interno sono principalmente: RIP, OSPF, IS-IS.
Questi protocolli garantiscono la possibilità di inviare ai routers della rete aggiornamenti automatici, sfruttando algoritmi:

. distance vector;

. link state.

A differenza dei protocolli EGP in cui lo standard è unico, gli Interior Routing Protocols sono molto diversi fra loro:

. RIP implementa distance vector;

. OSPF implementa link state;

. IS-IS implementa link state con indirizzamento OSI.

RIP

Questo protocollo, nato presso Xerox PARC, presenta due versioni compatibili fra loro:

. RIP v1;

- RIP v2.

RIP (Routing Information Protocol):

- implementa distance vector con metrica numero di hop;
- divide host attivi e passivi (host attivi sono routers);
- un router informa gli altri routers periodicamente (ogni 30 s) annunciando le proprie tabelle di routing;
- se un router identifica un percorso a costo minore rispetto all'intradamento che fa in quel momento, aggiorna la entry.

In generale, le proprietà degli algoritmi distance vector sono:

- scambio con i nodi (routers) vicini di:
  - destinazioni raggiungibili;
  - costo con cui raggiungere le destinazioni;
- quando un nodo (router) riceve un aggiornamento:
  - aggiunge destinazioni che non conosceva;
  - aggiorna la entry per cui il costo originario era maggiore;
  - cambia instradamenti se sono più brevi (e a costo minore);
  - ignora gli altri casi.

É fondamentale propagare updates periodici anche se il costo non varia altrimenti, se ci fossero due percorsi per raggiungere la stessa sottorete uno sarebbe ignorato: nell'ipotesi in cui quello usato nelle routing tables cadesse, senza aggiornamenti periodici quella sottorete non sarebbe più raggiungibile. Invece con updates periodici i routers si accorgono del nuovo instradamento possibile.

Quando cade un link, infatti, i nodi provvedono ad impostare un costo infinito per raggiungere tutte le destinazioni attraverso quel percorso.
Si crea un problema se cad un link e i costi delle topologie sono diversi; ipotizzando che:

- la stazione X sia raggiungibile con costo 1 attraverso Y e costo 20 attraverso Z;
- le stazioni Y e H sono direttamente collegate con un link a costo 5;
- la stazione Z è collegata con Y con costo 1.

Se cade il link che collega X a Y si crea un loop fra Y e H che prende il nome di *bouncing effect*: Y crede di poter raggiungere X attraverso H mentre H crede di poter raggiungere X attraverso Y.
Questo *palleggiamento* accade perchè i costi dei link sono tutti inferiori rispetto al costo del link X-Z: quando il bouncing effect termina tutti i nodi della rete possono raggiungere X attraverso Z perchè i costi di tutti i link saranno superiori a quello fra X-Z.

Considerando invece l'ipotesi di rottura di due link si assite al fenomeno del *counting to infinity*. Se:

. le stazioni X e Y sono collegate;

. X è collegata a Z;

. Y è collegata a H;

. si rompono questi due link;

ciò che accade è un loop fra X e Y che arriva fino ad ∞ (il massimo numero possibile nel conteggio).
Le soluzioni a questo problema sono:

. split horizon: se X usa Y per raggiungere Z, X non comunica a Y che raggiunge Z attraverso di lui;

. split horizon with poisonous reverse: X comuncica a Y che raggiunge Z a costo infinito spezzando un eventuale loop fra le due stazioni.

Tali soluzioni non rompono anelli formati da più di due nodi.

RIP v1 In questa versione:

. gli indirizzi sono su 32 bit (classico indirizzo ip);

. indirizzamento a classi;

. metrica in base al numero di hop:

  . $0 - 15$ normalmente;
  . 16 per indicare il costo ∞;

. sono supportati canali punto-punto e broadcast;

. si usa UDP come livello trasporto;

. i distance vector sono emessi ogni 30 s normalmente;

. se una entry non riceve aggiornamenti dopo 180 s si imposta il costo a ∞ e successivamente viene rimossa dalla routing table.

Formato PDU Il formato è composto da:

. 8 bit usati per specificare *request code* oppure *response code*: la request sollecita i routers ad annunciare le loro routing tables ad un router che si è appena acceso; la response corrisponde al normale update inviato periodicamente;

. 16 bit di *Address Family Identifier* per specificare il tipo di rete;

. 32 bit di indirizzo ip;

. 32 bit di metrica;

. diversi campi *Must be zero* usati nella versione successiva.

Funzionamento La trasmissione viene fatta in broadcast ip con response message e la sorgente specifica il proprio indirizzo; ogni messaggio contiene i distance vector ed è inviato ogni 30 s. In ricezione si elaborano tutte le entry ricevute.

Se durante la ricezione viene cambiata la routing table si manda un update a tutti i nodi dopo un tempo causale $[1, 5]$ s; si è deciso di aspettare un tempo casuale per:

. poter eventualmente sommare altri update;

. limitare il traffico in rete.

Le entry a costo infinito vengono segnalate una sola volta, poi vengono eleminte dalla routing table.

I messaggi di request sono inviati quando un nuovo nodo attivo si accende e come unica entry nella propria routing table possiede 0.0.0.0; i messaggi di risposta ad una request vengono inviati in unicast.

RIP v2 É un aggiornamento della versione 1 che apporta queste modifiche e aggiunte:

. routing con netmask anzichè classi;

. autenticazione;

. trasmissione in multicast dei distance vector.

Esiste compatibilità con la prima versione perchè le nuove funzionalità sono implementate nei campi che in RIP v1 sono *Must be zero*: perciò un pacchetto inviato da un router con RIP v2 con questi campi a zero può essere decodificato da un router con RIP v1.

Per ridurre il problema di sincronizzazione di RIP v1 dovuto agli update (si aspetta un tempo casuale molto piccolo) RIP v2 li invia dopo un tempo di attesa distribuito con probabilità uniforme fra 15 e 45 s.

OSPF

OSPF (Open Shortest Path First Protocol) è basato su link state con le seguenti funzionalità di base:

. definizione di adiacenze;

. compilazione del database attraverso flooding;

. distribuzione di informazioni realtive link esterni;

. supporto di routing gerarchico.

Negli algoritmi link state:

. ogni nodo invia le informazioni dei costi a tutti gli altri nodi;

- ogni nodo si costruisce la topologia di rete;
- ogni nodo, data la topologia, calcola i percorsi a costo minimo con l'algoritmo di Dijkstra.

In OSPF, per evitare un'enorme quantità di messaggi scambiati per la creazione della topologia, si elegge un *router designato* e ogni altro nodo stabilisce le adiacenze verso di esso.
I messaggi, che informano la rete di cambiamenti, vengono inviati periodicamente, con flooding, ogni 30 minuti: OSPF è infatti un algoritmo fortemente centralizzato, a differenza di RIP, per cui richiede meno aggiornamenti.

I nodi che ricevono un messaggio:

- controllano se hanno già ricevuto un messaggio con quel numero di sequenza da quel router;
- in caso affermativo scartano tale pacchetto;
- in caso negativo effettuano flooding;
- in entrambi i casi inviano un ACK alla sorgente.

A seconda della topologia di rete:

- se è broadcast:
  - un router che deve mandare un aggiornamento lo trasmette solo all'indirizzo multicast dei routers designati (224.0.0.6);
  - un router designato può notificare tale aggiornamento a tutti i routers OSPF inviando il pacchetto all'indirizzo multicast 224.0.0.5;
- se è punto-punto:
  - un router che deve inviare un aggiornamento lo invia solo al router designato;
  - il router designato invia copie separate dell'aggiornamento a tutti i routers.

Il routing avviene a due livelli con l'uso di diverse *aree* OSPF; questo comporta:

- una riduzione della dimensione delle routing tables;
- la topologia di un'area è nascosta alle altre;
- ogni area ha un proprio database;
- la necessità di avere dei routers che connettono le varie aree (questi routers prendono il nome di ABR, Area Border Router).

I messaggi, chiamati LSA (Link State Advertisement), contengono i LS record che formano il database e rappresentanon la topologia.
La trama di un LS record è formata da:

- LS age: tempo trascorso dall'ultimo aggiornamento;

- Options: fra gli altri compiti, può impostare routing TOS (Type Of Service), la priorità in ip;
- LS type: identifica il tipo di link annunciato; esistono:
  - LS1: link del router (tutti lo possono memorizzare);
  - LS2: indirizzi dei routers che hanno adiacenze con lui (lo possono memorizzare solo i routers designati);
  - LS3: può essere memorizzato solo da ARB e indica quale reti conosce l'ARB in questione;
  - LS4: anche questo può essere memorizzato solo da ARB e rappresenta tutti gli ARB che conosce l'ARB in questione;
  - LS5: memorizzato solo da ARB identifica le reti esterne raggiungibili.
- link state ID: indirizzo ip del link annunciato;
- advertising router: indirizzo ip del router che ha inviato l'annuncio;
- LS sequence number: aggiornato ad ogni annuncio;
- LS checksum: controllo d'errore.

Quando tutti i database sono riempiti e sincronizzati vengono calcolati i percorsi migliori con l'algoritmo di Dijkstra; le metriche sono intere e incrementative: valori bassi corrispondono ai percorsi migliori.
Possono essere calcolate tenendo conto di costi, banda e ritardi e, per link bidirezionali, possono essere diverse.

Il protocollo OSPF è composto da:

- protocollo HELLO;
- protocollo EXCHANGE;
- protocollo FLOODING.

Il protocollo HELLO:

- è usato per controllare che i link siano operativi;
- è usato per eleggere il router designato e il suo backup;
- invia pacchetti ogni n secondi;
- specifica:
  - indirizzo del router designato e del suo backup;
  - indirizzi dei routers vicini da cui ha ricevuto un pacchetto HELLO nell'ultimo *dead interval*.

Il protocollo EXCHANGE:

- è usato per la sincronizzazione iniziale dei database dei nodi;

- imposta lo scambio di messaggi in modalità *master/slave*, ossia chi invia un pacchetto diventa master mentre chi lo riceve è lo slave;
- ogni messaggio si diversifica a seconda del flag usato:
  - I: initialize;
  - M: more;
  - MS: master/slave.
- i messaggi regolano le procedure di sincronizzazione;
- ogni messaggio, inoltre, include:
  - gli LS record del database;
  - un ack dei pacchetti di EXCHANGE ricevuti dalla direzione opposta.

Il protocollo FLOODING:

- è usato per:
  - annunciare cambi di stato dei links;
  - eseguire il refresh periodico dei LSA (che avviene ogni 30 minuti);
- ogni messaggio di FLOODING contiene più LSA ed è numerato con un numero di sequenza;
- il nodo, se riceve un messaggio di FLOODING dallo stesso router e con lo stesso numero di sequenza:
  - lo scarta;
  - se il numero di sequenza è diverso inoltra il messaggio su tutte le porte meno quella sorgente (procedura di flooding tradizionale);
  - comunque invia un ack alla stazione mittente.

Protocolli di routing esterno

Il primo protocollo introdotto fu EGP (Exterior Gateway Protocol), ma quando dorsali e cammini multipli ebbero forte diffusione andò in crisi perchè, tra gli altri motivi, il suo funzionamento presupponeva topologie ad albero.

Le funzionalità di EGP erano tre principali:

- *neighbour acquisition*: capacità di un router di accordarsi con i vicini per comunicare;
- *neighbour reachability*: capacità di un router di controllare se i suoi vicini sono attivi;
- *network reachability*: routers vicini si scambiano informazioni di raggiungibilità.

EGP non permette ai nodi di propagare come informazione la distanza fra le reti, ma solo se sono raggiungibili o meno. Non supporta:

- strutture con dorsali multiple;
- load sharing;
- cammini alternativi;
- policy routing.

Inoltre, un problema importante, è dato dal fatto che, in presenza di routers malfunzionanti (inviano route non vere), il protocollo entra i crisi.

Con la crescita della rete, e quindi un aumento della complessità della topologia, EGP non era più sufficiente: è stato introdotto, allora, BGP (Border Gateway Protocol).
BPG è un protocollo che implementa *path vector* (non si creano anelli), e permette di determinare il cammino migliore, in un contesto di possibili scelte multiple, attraverso opportune politiche di instradamento. Tali politiche tengono conto di:

- fattori economici;
- fattori gestionali;
- sicurezza.

Path vector:

- implementa un routing update contenente un elenco di AS necessari per raggiungere una destinazione;
- se lo stesso AS compare un numero di volte maggiore di uno si riconosce la presenza di un anello;
- i vantaggi sono:
  - riconoscimento facile di anelli;
  - possibilità di avere metriche diverse per ogni router;
- lo svantaggio è rappresentato essenzialmente dall'elevato overhead: esiste, infatti, una entry per ogni AS annunciato.

Si distinguono tre tipi di AS:

- *stub AS*: sono connessi ad un solo AS esterno e accettano solo scambio di traffico locale;
- *multihomed AD*: sono connessi a più AS esterni e accettano solo scambio di traffico locale;
- *transit AS*: connessi a più AS esterni, accettano anche traffico in transito verso altri AS.

La comunicazione fra due routers che implementano BGP avviene con TCP e i routers prendono il nome di *BGP peers*.
In una prima fase i routers di bordo di un AS scambiano le tabelle di instradamento interne, stabilendo un *BGP peering interno*; in un secondo momento

vengono aperte connessioni con routers di bordo di altri AS, stabilendo così un *BGP peering esterno*.
Dopo queste fasi i peers inviano notifiche solo se ci sono modifiche sulla rete (periodicamente vengono solo inviati messaggi di KEEPALIVE).

L'intestazione del pacchetto BGP contiene informazioni come:

- autenticazione;
- lunghezza del messaggio;
- tipo di messaggio:
  - open;
  - update;
  - notification;
  - keepalive.

I messaggi di UPDATE vengono inviati solo pera nnunciare un nuovo percorso possibile e per annullare percorsi inammissibili:

- percorsi annullati sono identificati attraverso l'indirizzo ip della destinazione;
- percorsi nuovi sono identificati attraverso Path Attributes, che si riferiscono all'indirizzo ip della destinazione.

Le Path Attributes specificano inoltre diversi flags, tra cui AS_PATH che:

- identifica gli AS attraversati dal messaggio di UPDATE;
- vincola un update ad avere solo l'ID del router stesso che lo ha generato;
- impone l'inoltro di un update ricevuto da routers esterni solo dopo che il router attraversato ha aggiunto il proprio ID.

BGP memorizza i percorsi possibili in un RIB (Routing Information Base), secondo una suddivisione di questo tipo:

- *Adj_Rib_in*: percorsi appena annunciati dai vicini;
- *Loc_Rib*: percorsi predefiniti;
- *Adj_Rib_out*: percorsi appena annunciati ai routers vicini.

Il calcolo di un percorso ottimale avviene scegliendo fra gli Adj_Rib_in, in tre passi:

- calcolo del grado di preferenza (secondo politiche locali);
- scelta del percorso (in base al grado di preferenza);
- diffusione dei percorsi.

Il grado di preferenza è espressione di:

. numero e qualità di AS da attraversare;

. politiche locali;

. origine del percorso;

. carico dei links;

. QoS (Quality of Service).

Per percorsi sovrapposti le varie destinazioni sono identificate dal campo *NLRI overlap*.
Esistono due tipologie di percorsi sovrapposti:

. più specifici, se raggruppano un insieme piccolo di destinazioni;

. meno specifici, se raggruppano un insieme grande di destinazioni.

I percorsi sovrapposti possono essere gestiti:

. con aggregazione;

. accettandoli singolarmente.