Piano di controllo 2

sabato 26 agosto 2023 21:43

LS vs LD

La prima differenza è che nell'algoritmo distance-vector ciascun nodo dialoga unicamente con i propri vicini, link state dialoga con l'intera sottorete di appartenenza via broadcast. Confrontiamo gli algoritmi in 3 macro-aree:

- **Complessità dei messaggi:** LS richiede che ciascun nodo conosca il costo di ogni collegamento nella rete, quindi l'invio dei messaggi avviene in $O(|N| \cdot |E|)$. Ad ogni iterazione l'instradamento DV richiede lo scambio di messaggi tra i nodi adiacenti, la complessità varia quindi in base a molti fattori;
- Velocità di convergenza: LS è un algoritmo $O(|N|^2)$ che richiede $O(|N| \cdot |E|)$ messaggi da mandare. L'algoritmo DV può convergere lentamente e presentare cicli di instradamento, oppure anche il problema del conteggio infinito;
- Robustezza: nel caso in cui un router si guasti, funzioni male o venga sabotato, LS può comunicare via broadcast un costo sbagliato per uno dei suoi collegamenti connessi, ma dato che ogni nodo calcola solo le proprie tabelle di inoltro si crea un certo grado di isolamento che si traduce in *robustezza*. Per DV, invece, un calcolo errato può significare una propagazione dell'errore nell'intera rete.

Finora abbiamo trattato la rete come un sistema che ha al suo interno una collezione di router… **Internet è una rete di reti**, abbiamo più ISP interconnessi tra loro. Sorgono quindi due problemi: la scalabilità, il numero di router cresce sempre di più e cresce anche il tempo del calcolo d'instradamento, l'autonomia amministrativa, ogni ISP desidera amministrare i propri router come meglio crede.
Questi due problemi si risolvono organizzando i router in sistemi autonomi, ogni ISP sceglie se raggruppare i propri router in un singolo **AS** *(sistema autonomo)* oppure crearne diversi, nel secondo caso ogni AS viene identificato tramite un numero univoco ICANN. Vi sono dei protocolli per gestire l'instradamento interno ad un sistema autonomo…

Instradamento interno ai sistemi autonomi (INTRA-ISP routing): Open Shortest Path First (OSPF)

Nota: gli ISP utilizzano il protocollo OSPF per determinare i percorsi ottimali per le coppie sorgente-destinazione interne a un sistema autonomo.

OSPF e un protocollo link-state che utilizza il *flooding* di informazioni per lo stato dei collegamenti e Dijkstra per il percorso a costo minimo.
Ogni router costruisce un grafo dell'intero AS in cui si trova, a questo punto esegue localmente Dijkstra per comprendere tutti i cammini minimi verso tutte le destinazioni nella sua sottorete. *I costi dei collegamenti vengono impostati dall'amministratore*. Se lo stato di un collegamento cambia, allora OSPF invia informazioni di instradamento via broadcast ai suoi router, questo inoltre viene fatto periodicamente anche se lo stato non è mutato.
I vantaggi di questo protocollo:

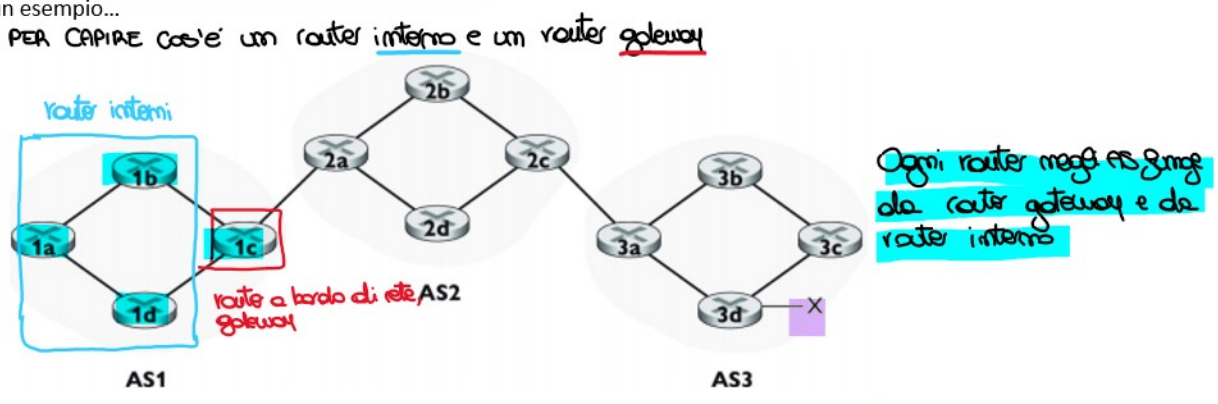
- **Sicurezza**: gli scambi tra router possono essere autenticati, quindi solo router appartenenti al dato AS possono partecipare al protocollo. *Si hanno due modalità di autenticazione: semplice o tramite MD5*;
- **Percorsi con lo stesso costo**: quando risultano più percorsi con il medesimo costo, OSPF permette di usarli tutti senza appesantirne uno solo;
- **Supporto per l'instradamento unicast e multicast**: per il multicast OSPF diventa **MOSPF** e sfrutta il database dei collegamenti, aggiunge una tipologia di annuncio nello stato dei collegamenti al meccanismo broadcast;
- **Gerarchie**: OSPF configura i router in aree, una delle più importanti è l'**Area di Dorsale** costituita dai propri router di confine (gateway), che instradano i pacchetti verso l'esterno. Un AS determina che i pacchetti siano inviati prima ai router di confine e attraverso la dorsale raggiungano l'area esterna dell'AS di destinazione.

Instradamento esterno ai sistemi autonomi (INTER-ISP routing): Border Gateway Protocol (BGP)

È l'attuale standard dei protocolli di instradamento tra sistemi autonomi in Internet, il collante che tiene insieme i migliaia di ISP che formano Internet.
BGP non instrada i pacchetti verso uno specifico indirizzo, i pacchetti vengono instradati verso prefissi CIDR *(prefissi senza alcuna classe vedere cap. Piano dei dati 3)* che rappresentano una sottorete o una collezione di sottorete.
Tale protocollo mette a disposizione a ciascun router un modo per:

1. Ottenere informazioni sulla raggiungibilità dei prefissi di sottorete da parte di sistemi confinanti, ovvero BGP consente a ciascuna sottorete di comunicare la propria esistenza alle altre sottoreti;
2. Determinare i percorsi "ottimi" verso le altre sottoreti.

Facciamo un esempio…



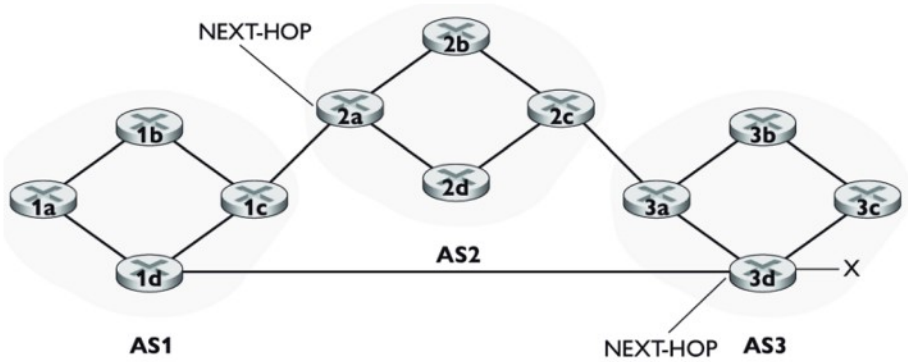
Vediamo lo schema generale con cui BGP distribuisce le informazioni su come raggiungere il prefisso X:

1. AS3 invia un messaggio BGP ad AS2 con l'annuncio dell'esistenza in AS3 di X, denotiamo questo messaggio con "AS3 X";
2. A questo punto AS2 invia un messaggio BGP ad AS1 con l'annuncio che X esiste ed è raggiungibile passando prima da AS2 e poi per AS3, il messaggio lo denotiamo con "AS2 AS3 X", **specularmente a prima aggiungendo un AS**.

Nota: ogni AS verrà a conoscenza non solo dell'esistenza di x, ma ad ogni messaggio BGP riceve un AS ed indica il percorso aggiornato

Cosa sono questi annunci? *Sono messaggi BGP*...e come vengono prodotti?
I messaggi BGP vengono inviati dai router, quest'ultimi si scambiano informazioni di instradamento su connessioni TCP semi-permanenti usando la porta 179; ciascuna sessione TCP dove vengono inviati tali messaggi è detta sessione BGP. Quest'ultima può essere interna o esterna: nel caso in cui questa coinvolga due sistemi autonomi viene detta **sessione** BGP esterna (sessione **eBGP**). Invece una sessione BGP tra router dello stesso sistema autonomo è chiamata **sessione** BGP interna (sessione **iBGP**).
Quando BGP annuncia le *rotte* si avvale di attributi, che abbiamo descritto, una rotta è l'unione del prefisso a dei suoi attributi. I due più importanti sono :

- As-Path: elenca i sistemi autonomi attraverso i quali è passato l'annuncio del prefisso, quello che succede è che quando un prefisso attraversa un sistema autonomo, quest'ultimo aggiunge il proprio AS*N* all'attributo As-Path. Ad esempio, prendiamo la rete di prima aggiungendogli un collegamento tra AS1 e AS3, vi sono quindi ora due rotte da AS1 al prefisso x, con rispettivamente l'**As-Path una settato ad "AS2 AS3"**, l'altra "AS3";
- Next-Hop: ha ruolo nel fornire il collegamento tra i protocolli di instradamento intra-AS e inter-AS. In **Next-Hop** è riportata l'interfaccia del router che inizia l'As-Path, verso il primo router gateway esterno che il sistema autonomo incontra...



Vediamo ora un algoritmo semplicistico: **hot potato**, per introdurre poi l'algoritmo che BGP adotta...

Hot Potato

1. Dal protocollo inter-AS(BGP) si apprende che una data sottorete AS*N* è raggiungibile attraverso più router gateway;
2. Si usa l'informazione d'instradamento proveniente dal protocollo intra-AS (ad esempio OSPF) per determinare i costi dei percorsi minimi verso i router gateway;
3. Instradamento a "patata bollente": si sceglie di instradare verso il gateway che il costo minimo di raggiungimento inferiore all'interno della mia sottorete;
4. Dalla tabella d'inoltro si determina l'interfaccia I che conduce a tale gateway a costo.

Si scrive quindi (AS*N*, I) nella tabella di inoltro.
Nota come quando si aggiunge un prefisso fuori dalla mia sottorete, ad una tabella di inoltro vengono utilizzati entrambi i protocolli inter-AS(BGP) e intra-AS (es. OSPF).

Algoritmo usato da BGP

L'input del processo di selezione è l'insieme di tutte le rotte apprese e accettate dal router, se ne esistono due o più verso lo stesso prefisso, BGP invoca in sequenza le seguenti regole di eliminazione fino ad individuare un'unica possibilità:

1. Alle rotte viene assegnato un attributo valore di "preferenza locale" che viene impostato dal router stesso o appreso da un altro nello stesso AS (questa scelta è lasciata all'amministratore di rete del sistema autonomo); successivamente si selezionano le rotte con i più alti valori di preferenza locale;
2. Tra le rotte con lo stesso valore di preferenza locale, si seleziona quella con As-Path più breve.
3. Tra le rotte con lo stesso valore di preferenza locale e la stessa lunghezza As-Path, si seleziona quella il cui router di Next-Hop presenta il percorso con costo minore, determinato dall'algoritmo intra-AS *(instradamento a patata bollente)*;
4. Se rimane ancora più di una rotta, il router si basa sugli identificatori BGP.

Esempio di applicazione dell'algoritmo alla nostra rete: consideriamo il router 1b, sappiamo che ci sono due rotte BGP verso il prefisso X: una che passa da AS2, l'altra che lo evita, che va direttamente nel sistema autonomo che contiene tale prefisso...
L'algoritmo hot potato sceglierebbe la prima perchè seleziona tra i due percorsi a costo minimo, quello a costo minimo. Invece l'algoritmo di selezione delle rotte mostrato qui sopra applicherebbe la regola 2 prima della 3 e quindi verrebbe scelta la seconda rotta avete As-Path più breve.

Il piano di controllo SDN

L'architettura e SDN ha quattro caratteristiche fondamentali:

- Inoltro ho basato sui flussi: questo approccio è opposto all'approccio tradizionale dei router, dove l'inoltro del datagramma viene effettuato basandosi esclusivamente sull'indirizzo IP di destinazione; il piano di controllo SDN in un'architettura SDN, ovvero tramite degli switch SDN che occupano la funzione di inoltro dei pacchetti, ha il compito di calcolare gestire e installare le tabelle di flusso all'interno degli switch;
- Separazione piano dei dati e piano di controllo: in questa architettura è netta, gli switch eseguono regole match-action occupandosi quindi del piano dati mentre i *server SDN* gestiscono il piano di controllo determinando e amministrando le tabelle di flusso;
- Funzioni di Controllo Esterne al Piano Dati (Switch): il piano di controllo viene implementato su server distinti e remoti rispetto agli switch, tramite due componenti: il **controller SDN** e un **insieme di applicazioni di controllo di rete**.
  Il controller SDN mantiene informazioni di stato della rete e le fornisce alle applicazioni di controllo, attraverso le quali quest'ultima monitora e controlla i dispositivi di rete sottostanti;
- Una rete programmabile: le applicazioni sono i cervelli del piano di controllo SDN usando le API fornite dal controller SDN per determinare e controllare il piano dei dati negli switch, artraverso queste applicazioni la rete diventa programmabile.

Controllo SDN ed Applicazioni di Controllo

Queste due componenti sono suddivise su tre livelli bottom-up di funzionalità:

1. Livello di comunicazione, si occupa per l'appunto della comunicazione tra controller SDN e i dispositivi di rete; le comunicazione passano attraverso l'interfaccia South-Bound e il protocollo più utilizzato per gestirle è Open Flow;
2. Livello di gestione globale della rete: il controller deve mantenere informazioni aggiornate sullo stato dei link degli host, degli switch, ecc. solo in questo modo si possono prendere decisioni finali coerenti. Tutte queste informazioni costituiscono lo stato globale della rete e sono contenute in tabelle di flusso;
3. Interfaccia con il livello di applicazione di controllo della rete: le API dell'interfaccia North-Bound permettono al controller di comunicare con le applicazioni di controllo che possono quindi leggere e/o scrivere lo stato della rete e le tabelle di flusso del livello di gestione di stato globale della rete -> *questo permette un pronto intervento delle applicazioni ai cambiamenti di stato della rete.*
   Le API più comunemente utilizzate sono di tipo REST.

Open Flow

Tale protocollo opera tra un controller ed un dispositivo sottostante (host, router o quel che è), comunica tramite API Open Flow con connessione TCP sulla porta 6653.

Elenchiamo i più famosi ed utilizzati messaggi tra un controller e uno switch:

- **CONFIGURATION**, il controller configura i parametri dello switch;
- MODIFY-STATE: il controller imposta proprietà sulle porte ed aggiunge, modifica o cancella occorrenze della tabella di flusso dello switch;
- **READ-STATE**: il controller acquisisce statistiche e valori di contatori dalla tabella di flusso dello switch;
- **SEND-PACKET**: il controller invia un pacchetto specifico tramite una specifica porta dello switch.

Viceversa uno switch comunica ad un controller:

- **FLOW-REMOVED**: lo switch informa il controller che un occorrenza della propria tabella di flusso è stata cancellata, per un avvenimento di timeout oppure tramite MODIFY-STATE;
- **PORT-STATUS**: lo switch informa il controller che una sua porta ha cambiato stato;
- **PACKED-IN**: Lo switch invia un pacchetto al controller perché necessita di un'ulteriore elaborazione.