

1 Lezione del 24-10-25

1.1 Reti a commutazione di pacchetto

Iniziamo ad entrare nel dettaglio delle reti a commutazione di pacchetto, e quindi ad anticipare i concetti che saranno alla base del cosiddetto livello **network**. Per fare ciò introdurremo nuovi tipi di dispositivo, col compito di smistare pacchetti, fra cui **switch** e **router** (per adesso parliamo principalmente dei primi).

1.1.1 Hub Ethernet

Riprendiamo come esempio lo standard Ethernet.

Avevamo introdotto l'idea di **hub** nel moderno Ethernet dove tutti i dispositivi sulla rete si collegano in modalità punto-punto. Questo ci permetteva di implementare la cosiddetta topologia a *stella*.

Questi hub possono essere intesi come *ripetitori "stupidi"* a livello fisico, che si limitano a prendere i bit in entrata dal dispositivo ed inoltrarli a tutti gli altri dispositivi, allo stesso bitrate.

Questo significa che tutti i nodi connessi all'hub possono collidere con gli altri, non si ha nessun tipo di bufferizzazione dei frame inviati, e non si implementa nessun tipo di CSMA/CD a livello hub: il compito di rilevare le collisioni è assegnato alle NIC dei dispositivi.

1.2 Switch Ethernet

Gli hub Ethernet sono stati oggi soppiantati da dispositivi più intelligenti: gli *switch*.

Gli switch sono dispositivi di livello datalink, e hanno quindi un ruolo *attivo* nella rete. Possono infatti:

1. Immagazzinare i frame Ethernet ricevuti dai dispositivi;
2. Esaminare l'indirizzo MAC di destinazione del frame, e selettivamente inoltrarlo a uno o più link in uscita;
3. Implementare protocolli di rilevamento collisioni come CSMA/CD.

Questo però non nega che lo switch deve essere comunque **trasparente** agli host sulla rete: questi non devono dover essere al corrente della sua esistenza.

Un'altra caratteristica degli switch è che sono *plug-and-play* e non richiedono configurazione.

Il compito dello switch è quindi quello di **commutare** le linee di ingresso con le linee di uscite: questo significa che gli host riescono ad avere una connessione dedicata e diretta con lo switch, e attraverso questo l'illusione di una connessione dedicata e diretta con l'host con cui stanno parlando.

1.2.1 Switch table

Per realizzare la funzionalità di instradamento degli switch si sfrutta una tabella, detta **switch table**. Ogni entrata della switch table contiene:

- L'indirizzo MAC dell'host;
- L'interfaccia (semplicemente la linea di uscita) su cui si trova tale host;

- Un *time stamp* che rappresenta il tempo di vita dell'interfaccia.

Notiamo che propriamente dovremmo parlare di *nodi*, e non di *host*, in quanto i primi sono a livello datalink e i secondi sono a livello network. Ci fidiamo di capirci.

La popolazione della switch table avviene attraverso l'**apprendimento automatico** dello switch: questo infatti *impara* man mano che si inviano frame Ethernet quali host possono essere raggiunti attraverso quali interfacce.

Vediamo come si tramette il primo frame quando la switch table è vuota:

1. Quando il frame arriva allo switch, questo può associare l'interfaccia del mittente al suo indirizzo MAC (contenuto nell'header del frame);
2. Per inoltrare il frame, controlla la sua switch table (che è attualmente vuota): non può quindi fare meglio che inoltrarlo a tutti (in quanto non sa a quale interfaccia corrisponde il MAC destinatario). Chiamiamo questo processo *flooding*;
3. Questo processo si ripete: man di mano che gli host inviano frame, lo switch riesce a farsi un'idea della struttura della rete popolando la switch table. Questa è una pretesa ragionevole in quanto immaginiamo che gli host che ricevono pacchetti prima o poi ne invieranno di altri, rendendosi disponibili all'interno della switch table.

In pseudocodice, il comportamento è simile al seguente:

```
1 on <frame_received>:  
2   record(<link>, <sending_MAC>);  
3   entry = switch_table.index(<receiving_MAC>);  
4   if(entry) {  
5     forward(entry.link);  
6   } else {  
7     forward(all); // flood  
8   }
```

1.2.2 Gerarchie di switch

Gli switch Ethernet possono essere interconnessi fra di loro: uno switch che collega più host può a sua volta connettersi (assieme ad altri switch) ad un singolo switch principale, e questo processo può ripetersi ricorsivamente.

Ad esempio, in un palazzo potremmo immaginare che ogni piano ha il suo switch (che collega i dispositivi di tale piano), e questi switch di piano vengono connessi da un singolo switch per l'intero palazzo.

1.2.3 Differenze fra switch e router

Possiamo quindi anticipare le differenze fra switch e router: sappiamo che gli switch sono dispositivi di livello 2 (datalink), che guardano agli header di frame (e quindi indirizzi MAC), mentre i router sono dispositivi di livello 3 (network), che guardano agli header di pacchetto (e quindi indirizzi IP).

Rifacendoci alla pila protocollare, abbiamo che gli switch implementano i livelli 1 e 2 (fisico e link), mentre i router implementano i livelli 1, 2 e 3 (fisico, link e network).

Sia switch che router usano delle tabelle per governare l'instradamento dei messaggi ottenuti (frame per gli switch e pacchetti per i router), che negli switch sono dette *switch table* e nei router *routing table*.

1.3 Datacenter

I grandi fornitori di servizi su Internet sfruttano strutture dette **datacenter** per organizzare le loro macchine server. I datacenter rappresentano quindi l'altro "*nodo*", assieme a quello utente, dove si concentra la potenza di calcolo (e quindi in qualche modo l'"*intelligenza*" che ci fornisce il servizio).

Chiaramente i datacenter devono assicurare un servizio affidabile, continuativo nel tempo, e assicurare che il carico sulle macchine sia equamente distribuito.

1.3.1 Reti datacenter

All'interno dei datacenter si vanno a creare gerarchie di switch piuttosto complesse. I rack server sono composti da 20-40 *server blade*, che rappresentano gli host veri e propri. Ogni rack ha poi un suo switch detto **TOR** (*Top Of Rack switch*). I TOR sono connessi da switch di livello 1 (circa 16 per volta), e questi a loro volta sono connessi da switch di livello 2 (ancora, circa 16 per volta).

Infine, l'intera rete viene esposta all'esterno usando router (più di uno per assicurare la ridondanza del servizio).

1.4 VLAN

Quando le reti LAN diventano molto grandi, si iniziano ad avere diversi problemi dati dal fatto che si ha a disposizione un **singolo dominio di broadcast**: tutti i nodi devono ricevere tutto il traffico di livello 2 (ARP, DHCP, frame con MAC broadcast, ecc...). Questo porta chiaramente a problemi di efficienza, sicurezza e privacy.

Notiamo che in Ethernet moderno non è invece vero dire che si ha un **singolo dominio di trasmissione**: abbiamo visto come gli switch permettono di instradare il traffico fra singoli host, risparmiandolo agli altri host sulla rete.

Un'altra problematica è data dal fatto che il collegamento **fisico** agli switch è anche un collegamento **logico**: si potrebbe volere che un host collegato *fisicamente* ad un certo switch, debba essere collegato *logicamente* ad un altro switch.

Le **VLAN** (*Virtual Area Network*) sono la soluzione a questi problemi. Gli switch che implementano funzionalità VLAN possono essere configurati per definire più LAN *virtuali* su una singola infrastruttura LAN fisica.

1.4.1 VLAN a porte

Si possono avere **VLAN basate a porte**: le porte degli switch vengono raggruppate (da un certo software di configurazione dello switch), così che un singolo switch si comporti come più switch virtuali. Le switch virtuali sono confinate fra di loro: anche se è il solito switch fisico a gestire il traffico, questo non verrà instradato da una VLAN all'altra.

Invece delle porte, si potrebbero usare anche i MAC di destinazione dei frame Ethernet.

Per inoltrare i pacchetti fra più VLAN, si ha quindi bisogno di un vero e proprio router (stiamo effettivamente simulando più reti locali, e quindi più switch separati). Nella pratica, i produttori di switch VLAN spesso vendono combinazioni di switch e router.

Le VLAN possono essere distribuite fra più switch: in questo modo più switch fisici rappresentano una singola unità logica (o parte di essa, o di più unità logiche). Questo si implementa sfruttando una porta (**trunk port**) degli switch coinvolti al trasporto di frame da uno switch all'altro.

1.4.2 Frame VLAN

Dopo tutte le funzionalità di VLAN che abbiamo aggiunto agli switch, i frame che vengono inviati non potranno più essere a di tipo 802.1 Ethernet semplice, ma devono essere 802.1Q VLAN.

Quello che si fa è estendere il campo `type` del frame Ethernet, introducendo: qualcosa

1.5 WAN

Il principio dello switching può essere implementato su reti non solo locali, ma anche a grande copertura geografica. Le reti che si vanno a formare vengono dette **WAN** (*Wide Area Network*).

La topologia più adatta a questo punto non è più quella gerarchica che avevamo visto, ma una a *mesh*, dove ogni switch è collegato con un certo numero di altri switch, > 1 . Questo permette ridondanza nel caso i link, o addirittura interi switch, fuori operazione.

Avere ridondanza introduce il problema del **routing**: bisogna capire qual'è il percorso migliore da seguire per instradare frame da un host all'altro. Rimandiamo a dopo questo processo.

1.5.1 Forwarding

Parliamo invece del problema del **forwarding**, cioè come inoltrare effettivamente i pacchetti sul percorso desiderato.

Individuiamo innanzitutto i 2 tipi di servizio che potremmo offrire:

- **Connectionless**: in questo caso non si stabilisce alcuna connessione preliminare, e ogni pacchetto è gestito in maniera stateless come se fosse il primo. Questo tipo di servizio è detto anche **datagram** (in quanto è il modello di servizio usato in Internet).

In questo caso non prevediamo nessun tipo di connessione preliminare, e non chiediamo agli switch o router intermedi di mantenere alcuno stato sullo stato della connessione (in quanto non esiste un concetto di "connessione" in primo luogo).

I pacchetti vengono gestiti sulla base dell'indirizzo di destinazione: più pacchetti per la stessa destinazione potrebbero però prendere percorsi diversi sulla rete (magari per congestioni, ecc...).

Per gestire l'instradamento si fa uso di **forwarding table** simili alle switch table già viste: la differenza sta nel fatto che si possono mappare non singoli indirizzi, ma **range** di indirizzi, alle interfacce.

- **Connection**: prima che la comunicazione inizi, un *circuito virtuale* (o **VC**, *Virtual Circuit*) viene stabilito. Da questo punto in poi, ogni pacchetto inviato segue lo stesso percorso nella rete.

Un circuito virtuale è un modo per emulare la funzionalità delle reti a **commutazione di circuito** (ad esempio la vecchia rete telefonica). In questo caso si prevedono fasi esplicite:

1. Fase di **chiamata**, dove l'host mittente *chiama* l'host destinatario;
2. Fase di **accettazione** della chiamata, dove l'host destinatario inoltra un nuovo messaggio, appunto di accettazione della chiamata;

3. A questo punto la comunicazione può effettivamente avvenire. Chiaramente, prima o poi seguirà una fase di **chiusura** del circuito virtuale, dove con uno scambio di messaggi simile al precedente mittente e destinatario negoziano la chiusura delle comunicazioni;

Il circuito virtuale, dal lato implementativo, deve essere qualcosa di noto sia agli host che agli switch o router che appartengono alla rete.

Prevediamo quindi per ogni VC:

- Un **percorso** da mittente a destinatario;
- **Numeri VC**, uno per ogni link lungo il circuito;
- Le entrate delle **forwarding table** degli switch o router lungo il percorso (la trattazione è generica, vale sia al livello 2 che al livello 3).

I pacchetti che viaggiano su un VC portano il numero di VC. Questo può quindi essere cambiato da switch o router, consultando la loro forwarding table. Ciò significa che gli switch mantengono informazioni sullo stato della connessione.

1.6 Internetworking

Veniamo adesso ad introdurre il concetto di **internetworking**, cioè collegamento di più reti fra di loro.

Visto che più reti sfruttano standard diversi di comunicazione, bisognerà dotarci di una qualche "*lingua franca*" che ogni rete può usare per parlare con altre reti. Come abbiamo anticipato, questa sarà rappresentata dai **datagrammi** Internet.