

BRIDGE E SWITCH

Bridge

Come dice il nome, il bridge è un dispositivo *ponte*, esso costituisce un precursore dello switch e il suo lavoro è stato oggi ereditato integralmente dagli switch. Collega tra loro due o più segmenti di una rete dello stesso tipo oppure due o più reti di tipo diverso, regolando il passaggio delle trame da una all'altra sulla base dell'indirizzo di destinazione contenuto in queste ultime. Esso riceve tutte le trame che circolano all'interno di ciascuna delle reti o dei segmenti di rete a cui si collega per mezzo delle proprie porte di ingresso/uscita. Legge l'indirizzo di provenienza di queste trame e, così facendo, costruisce una tabella di mappatura delle varie macchine collegate a ciascun segmento (detta *forwarding table*). Questa mappa gli servirà per ripartire il traffico così che passino da un segmento all'altro solo i pacchetti (trame) che ne hanno effettivamente bisogno, bloccando il traffico interno di ciascun segmento affinché non si riversi inutilmente negli altri (ricordiamo che Ethernet, per costituzione propria, tende a distribuire tutto a tutti, anche quelli che non ne hanno necessità). Quando il bridge riceve una trama su una delle proprie porte o interfacce, ha quattro alternative. Se l'indirizzo di destinazione appartiene a una macchina che si trova sullo stesso segmento da cui la trama arriva, il bridge scarta la trama. Se l'indirizzo di destinazione si trova su un segmento diverso rispetto a quello da cui la trama arriva, il bridge la invia sul segmento di destinazione. Se l'indirizzo di destinazione indicato non compare in alcun modo all'interno della tabella d'invio (*forwarding table*), la trama viene spedita a tutte le reti oppure a tutti i segmenti a cui il bridge è collegato, con la sola eccezione di quello da cui è arrivata. Quest'ultima operazione si chiama *flooding* (allagamento). Nel caso in cui la trama sia indirizzata a una rete che usa un sistema diverso nel confezionare le trame il bridge dovrà anche eseguirne la traduzione. Se infine la trama è destinata al bridge medesimo, viene intercettata e consegnata ai circuiti interni che dovranno interpretarla.

Switch

Adesso che abbiamo visto come funziona un bridge passiamo alla sua evoluzione più moderna: lo switch o commutatore. Si tratta di un'apparecchiatura che, alla pari di un bridge, collega tra loro diversi segmenti logici di una rete (diversi domini di collisione) e che consente il passaggio d'informazioni dall'uno all'altro, impedendo tuttavia che l'intero traffico presente su uno di essi si riversi negli altri, e viceversa, come invece accadrebbe se la lan Ethernet non disponesse di alcun filtro al proprio interno. Lo switch deve disporre almeno di due porte, anche se nelle configurazioni più comuni ne troviamo almeno 8.

La sola differenza rispetto al bridge è che lo switch esegue tutte le proprie elaborazioni in hardware e non via software, perciò non rallenta il fluire del traffico tra i segmenti. In gergo tecnico si dice che la connessione sia *wire speed* cioè lasci transitare i pacchetti alla velocità massima consentita dal tipo di conduttore usato per il cablaggio. Nella realtà, un rallentamento esiste sempre, anche se marginale, e la sua entità dipende dal modo in cui lo switch funziona. La primissima tecnica di switching, che eredita in toto la modalità operativa dei bridge, si chiama *store-and-forward*. Ogni trama che arriva su una delle porte dello switch viene incamerata per intero in una speciale porzione di memoria (buffer) e quindi scartata o trasferita a un altro segmento a seconda dell'indirizzo di destinazione (mac address) indicato al suo interno. L'operazione è velocissima, ma comporta in ogni caso un certo rallentamento perché la trama deve arrivare per intero nel buffer dello switch prima di

cominciare a essere ritrasmessa su un'altra porta (a cui corrisponde un altro segmento, appunto). È la tecnica di commutazione più affidabile, poiché prima di rispedito il pacchetto ci si accerta di averlo per intero e se ne verifica la correttezza attraverso il calcolo del *crc* (*Cyclic Redundancy Check*), ed è l'unica utilizzabile quando si collegano segmenti funzionanti a velocità diverse, come Ethernet e Fast Ethernet, per esempio. Tuttavia su impianti molto veloci, come nel caso di una dorsale che funziona tutta a 100 Mbps o più, il numero di trame in circolazione è molto elevato e il ritardo che si accumula per la registrazione di ciascuna si fa sentire.

L'alternativa ideata per eliminare quest'ultimo inconveniente si chiama commutazione *cut-through*. La parola significa "tagliare attraverso", "prendere una scorciatoia" e in effetti è proprio quello che accade. Non appena lo switch comincia a ricevere una trama su una qualsiasi delle sue porte, ne legge l'indirizzo di destinazione e, se questo corrisponde a un segmento collegato a un'altra porta, inizia immediatamente a trasmettere la trama senza aspettare che questa sia arrivata per intero. In questo modo, dopo aver letto l'indirizzo, la trasmissione in uscita avviene quasi in contemporanea con la ricezione, e il ritardo è minimo (fino a 20 volte inferiore a quello della tecnica store-and-forward). Benché molto efficace sotto il profilo della velocità, questa tecnica presenta il difetto di far passare anche porcheria. Infatti vengono ritrasmesse anche le trame difettose (troppo corte) risultanti da collisioni o da altri problemi nel segmento di provenienza. Lo switch si limita a controllare l'indirizzo e quindi fa passare tutto quel che segue senza controllo alcuno.

Inoltre il beneficio del sistema cut-through diminuisce quando il traffico diventa molto intenso e continuo. In questa circostanza è facile che una trama trovi la porta di uscita ancora occupata a trasmettere la trama precedente e perciò deve comunque essere memorizzata per intero nell'attesa di essere trasmessa (si torna alla modalità store-and-forward). Tale fenomeno è ancora più visibile con switch dotati di numerose porte visto che la quantità di traffico complessiva che si genera al loro interno è sempre elevata. Dal confronto di questi due approcci, ne è stato ideato un terzo, intermedio, che si chiama *fragment-free switching*.

Alla pari della modalità store-and-forward, anche qui si aspetta di ricevere l'intera trama prima di iniziare a trasmetterla, però ci si assicura che questa sia almeno lunga 64 byte (il minimo consentito dallo standard Ethernet) e si scarta qualsiasi frammento di trama che abbia dimensioni inferiori. In una rete Ethernet progettata con cura, gli unici errori ricorrenti sono le trame corte provocate da collisioni (*runt*) e per definizione queste hanno una dimensione inferiore ai 64 byte. Uno switch fragment-free è più veloce di un modello store-and-forward e al contempo non inonda il segmento di destinazione con frammenti di trama inutilizzabili. Il controllo, tuttavia, non è completo visto che continuano in ogni caso a passare le trame troppo lunghe (che sono peraltro il prodotto di malfunzionamenti sulla rete, come abbiamo visto sullo scorso numero, e che vengono risolte sostituendo i componenti difettosi).

Per filtrare queste ultime non esiste alternativa se non quella di utilizzare il sistema store-and-forward: si aspetta che la trama arrivi per intero, la si misura e la si fa proseguire solo se va tutto bene. Ciascuna di queste tre modalità offre vantaggi specifici e l'impiego dell'una piuttosto che dell'altra è anche determinato dallo stato contingente in cui si trova la rete: un'alta percentuale di collisioni e di errori oppure un traffico normale.

Di conseguenza gli switch più moderni ed evoluti offrono una funzione che alcuni chiamano *switching adattativo* e che consiste nel monitorare costantemente lo stato della lan e nel passare in modo dinamico a una delle tre modalità di commutazione che abbiamo prima visto. Questo genere di switch consente all'amministratore di regolare le soglie di errore oltre le quali l'apparato abbandona la modalità cut-through per scendere al sistema fragment-free oppure al sistema store-and-forward. Diventa così possibile definire un modello di comportamento che si adatta perfettamente al proprio impianto.