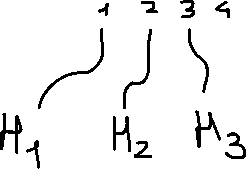
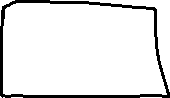
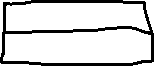
La volta scorsa abbiamo introdotto il protocollo Ethernet e abbiamo visto l’implementazione storica e quella recente, basata su dispositivi di tipo switch. I dispositivi di tipo switch permettono di avere un incremento notevole nella banda di comunicazione, poiché i canali diventano full duplex e perché non ci sono più collisioni. Questo vantaggio in termini di banda, però, si paga con un aumento della latenza dovuto alle operazioni di Store and Forward dello switch.   
Questo miglioramento della banda funziona solo per le comunicazioni punto-punto, non ha nessun effetto sulle comunicazioni broadcast, in questo caso lo switch si comporta esattamente come l’hub (ma mantenendo un ritardo maggiore dovuto allo S&F).

Prendiamo in considerazione una rete locale in cui ci sono un certo numero di host, uno switch con un certo numero di porte di comunicazione con doppini telefonici doppi e i cavi che li collegano.   
Se H1 vuole inviare un messaggio a H2, questo specifica come destinazione l’indirizzo MAC di H2 e lo invia allo switch. Il messaggio viene inserito in un buffer dello switch, dove questo può analizzarlo leggendo l’header di livello 2: in questo modo lo switch potrà inviarlo sulla porta che conduce all’indirizzo MAC di H2.



Ma come viene implementato l’instradamento del frame ethernet verso il destinatario nello switch?   
Uno dei requisiti delle porte di comunicazione è che siano di tipo plug and play (quindi per far funzionare lo switch deve essere sufficiente che gli venga inserito un cavo in una porta). Per fare in modo che lo switch funzioni sempre, viene usato un algoritmo di auto-apprendimento. Si può quindi pensare che le porte siano numerate e che il nostro dispositivo si costruisca una tabella, in cui a ogni indirizzo MAC viene messo in relazione un numero di porta. Come fa lo switch a riuscirci? Per farlo guarda gli indirizzi MAC sorgenti degli host man mano che questi gli mandano messaggi: se l’host H1 manda un messaggio destinato ad H2, lo switch prende nota dell’indirizzo MAC di H1 e lo inserisce nella tabella con associata la porta 1. Per poter inviare il messaggio ad H2 servirà quindi che H2 abbia già un mandato un messaggio allo switch. Per poter recapitare il primo messaggio, però, come si può fare? Beh, quando lo switch non trova l’indirizzo MAC del destinatario nella sua tabella si comporta come se il messaggio fosse di tipo broadcast e lo invia in tutte le uscite tranne quella da cui è arrivato. Lo switch è quindi come un hub finché la tabella delle porte non è inizializzata completamente.

La tabella è gestita come se fosse una cache, in modo da supportare anche eventuali modifiche fisiche alla rete (se per esempio si stacca H1 e lo si riattacca alla porta 4 è necessario che si possa riassociare l’indirizzo di H1 alla porta giusta). Per questo motivo ai valori della tabella è associato un timeout (inoltre ogni nuovo messaggio può potenzialmente portare a un aggiornamento della tabella).

Si noti che questi algoritmi di auto apprendimento non sono definiti dallo standard ethernet; pertanto, sono i costruttori a scegliere di implementarlo e, anche se al giorno d’oggi lo fanno tutti, ognuno sceglie arbitrariamente l’algoritmo di auto apprendimento da usare. Perché noi ci preoccupiamo di questo? Perché le cose sono più complicate di quanto sembrano: Le NIC sono infatti realizzate dai costruttori e il loro indirizzo MAC è scelto da questi e deve essere unico a livello mondiale. Le NIC però hanno una memoria che NON è NON modificabile, quindi, è possibile riprogrammare un dispositivo in modo da cambiare il suo indirizzo MAC: tipicamente questa operazione di riprogrammazione delle interfacce di rete è riservata a un amministratore di sistema. Questo vuol dire che se uno ha il controllo amministrativo sulla propria macchina può andare a cambiare l’indirizzo MAC della scheda. Un’altra cosa che è programmabile è il fatto che la scheda filtri i messaggi in ingresso rispetto all’indirizzo MAC della scheda stessa oppure no: disattivare il filtraggio (mettendo la scheda in modalità Promiscua) può essere utile per monitorare i messaggi che viaggiano sulla propria rete, ammesso che tale rete sia di tipo CSMA/CD (e quindi che ci sia un hub che effettua la ripetizione dei messaggi). Unendo però la modalità promiscua alla possibilità di cambiare l’indirizzo MAC della propria scheda - e al fatto che gli algoritmi di auto apprendimento non sono standardizzati - è possibile però ricevere tutto il traffico della rete: infatti inviando un messaggio con come indirizzo mittente quello di un altro router nella rete, lo switch aggiornerà la sua tabella, associando all’indirizzo del router in questione la porta corrispondente alla nostra macchina.

Ciò può dare problematiche di sicurezza, se si tiene conto che gli algoritmi di auto apprendimento non sono standardizzati: su una rete locale contenente uno Switch è possibile effettuare tutta una serie di attacchi che sono identici a quelli che si potrebbero fare su una rete contenete un HUB. Un attacco può essere ti tipo passivo o attivo.

Un esempio di attacco passivo è semplicemente l’osservazione di quello che succede in rete, è una tecnica di spionaggio (si costringe lo switch a funzionare come un hub e si guarda tutto il traffico che circola sulla rete redirigendolo verso il nostro dispositivo).  
L’attacco attivo è invece una forma più grave, un esempio particolarmente pericoloso è il man-in-the-middle, la cui idea è quella di mettersi in mezzo a un canale di comunicazione che dovrebbe essere privato tra due host, così da poter leggere i messaggi che circolano e eventualmente modificarli prima che arrivino a destinazione (per poi consegnarli fingendo che non sia successo nulla).

Naturalmente, costringere uno switch a comportarsi sempre come un hub, porterebbe a un rallentamento notevole della connessione, per via della diminuzione a cui andrebbe incontro la banda, fino a rendere la rete inservibile; anche questo è un potenziale tipo di attacco.

Per difendersi da questi attacchi, una soluzione sarebbe di spegnere l’algoritmo di auto-apprendimento e compilare a mano la tabella di associazione indirizzo MAC-porta: questo evita le falle di sicurezza, ma è molto oneroso nei confronti dell’amministratore di rete, che avrebbe la responsabilità di tenere aggiornata a mano la tabella ogni volta che si effettuano modifiche alla rete.  
Esistono situazioni meno drastiche; switch più costosi permettono di attivare forme di monitoraggio della rete, rendendo questo capace di riconoscere situazioni anomale e potenziali attacchi (in tal caso lo switch può mandare un’allerta all’amministratore di rete). Questa è in genere la soluzione preferenziale.

Possiamo passare al problema che avevamo lasciato in sospeso quando avevamo parlato del routing: l’instradamento finale, dell’ultimo hop su una rete locale, è compito del router che connette la LAN a internet. Il modo in cui viene effettuato l’ultimo hop è traducendo l’indirizzo IP di destinazione dell’host finale in un indirizzo MAC. Per realizzare ciò viene introdotto un altro protocollo di comunicazione, l’ARP (Address Resolution Protocol).   
ARP è un protocollo di livello applicativo che però serve per coadiuvare il livello 3 e il livello 2 di comunicazione nel momento in cui si passa darete internet a rete locale. Questo protocollo prevede quindi una comunicazione solo tra host di una rete locale.

Ipotizziamo che uno qualunque degli host della LAN voglia sapere quali sono gli indirizzi MAC delle macchine associate di cui conosce l’indirizzo IP (in particolare di cui conosce l’Host Number, visto che la sezione Network dell’indirizzo è uguale per tutte le macchine della LAN). Questo host manda una richiesta (che a livello di trasporto è di tipo UDP -> il motivo di ciò è che la richiesta viene mandata in broadcast), che avrà un header di livello 4, uno di livello 3, uno di livello 2 e il payload. Dentro al payload ci sarà l’indirizzo IP della macchina di cui si vuole conoscere l’indirizzo MAC, mentre dentro l’IP Dest dell’header 3 ci sarà l’indirizzo broadcast (corrispondente al Network giusto e a un host number = tutti 1) e sempre un indirizzo broadcast sarà il contenuto del MAC Dest dell’header 2. La richiesta arriva a tutte le macchine della rete locale, ma risponde soltanto quella che riconosce il proprio indirizzo IP nel payload della richiesta: la macchina risponderà specificando qual è il proprio indirizzo MAC.   
Da quel momento in poi la comunicazione non avviene più in broadcast, ma come comunicazione punto-punto, dunque, il messaggio di risposta avrà come destinazione l’host mittente della richiesta e avrà come payload l’associazione IP, MAC.

Questo è il modo in cui l’ultimo router può riuscire a instradare il messaggio su una rete locale: ottiene l’indirizzo MAC e invia il messaggio all’host corrispondente.

Questo sistema di comunicazione, però, è un po’ costoso e lento perché prevederebbe di lanciare un broadcast ogni volta che arriva un messaggio a un host di una LAN, per questo motivo si è progettato il protocollo ARP in modo che comprenda una cache (che avrà un timeout entro il quale la traduzione dell’indirizzo si può considerare affidabile): ciò permette di evitare di lanciare un broadcast ogni volta.

Una delle caratteristiche principali di questo protocollo è che è senza memoria: dopo aver mandato una richiesta, chi l’ha mandata si dimentica di averlo fatto. Se viene ricevuta una risposta il dispositivo scriverà l’associazione IP,MAC all’interno della cache. Ciò causa però problematiche di sicurezza: cosa succede se rispondono due o più host alla richiesta? Dipende a seconda dell’implementazione; la versione più semplice prevede che ogni volta che è ricevuta una risposta si aggiorni la cache. Questo può portare ad attacchi di sicurezza di tipo (ARP) cache poisoning: per ingannare un utente basta inviare una risposta sbagliata (con una associazione che non è corretta) e aspettare che l’host che la riceve aggiorni la propria cache ARP. Questo rende facile effettuare un attacco di tipo Man-In-The-Middle.

L’unico modo per evitare questo genere di attacchi consiste nel disabilitare il protocollo ARP (ciò è possibile nei sistemi UNIX attraverso la configurazione manuale del file /etc/hosts associando indirizzi IP a indirizzi MAC -> così facendo il contenuto del file viene preso per buono e il protocollo ARP non viene utilizzato).

In assenza di questa contromisura, occorre non fare troppo affidamento su quello che succede sulla rete locale: bisogna evitare che sulla LAN ci siano macchine che possono essere controllate da un attaccante malintenzionato.