CAPITOLO 5: Livello di collegamento dati

**5.1 Introduzione**

Nel livello di collegamento ogni dispositivo, che sia un host, un router o uno switch, verrà indicato come un nodo. Diversi nodi comunicheranno tramite canali di comunicazione detti link, che possono essere guidati o non guidati. Le unità di dati (datagrammi) che vengono scambiati vengono dette frame. Inoltre, il data-link layer ha il compito di gestire la comunicazione tra nodi attraverso un link.

Il servizio base del livello di collegamento è il trasporto di datagrammi da un nodo a quello adiacente ma sono presenti altri servizi che possono variare da un protocollo all’altro, come per esempio:

-Framing: incapsulare i dati del livello superiore in un frame prima di trasmetterlo. I frame sono formati da un campo dati, dove è inserito il datagramma, e da vari campi di intestazione.

-Accesso al collegamento: gestire l’accesso al canale, nel caso in cui ci siano più nodi, tramite il protocollo MAC, che gestisce le regole con cui immettere i frame all’interno del collegamento.

-Consegna affidabile: fornire un servizio di consegna affidabile, garantendo il trasporto dati senza errori. Come nel livello di trasporto, anche qui la consegna affidabile può avvenire attraverso ACK e ritrasmissioni.

-Controllo di flusso: consiste nel far combaciare la velocità con cui si inviano i dati due nodi a quella a cui vengono letti.

-Rilevazione errori: permette la rilevazione di errori.

-Correzione errori: permette di determinare il punto preciso del frame in cui è presente l’errore e di correggerlo.

-La comunicazione può essere half-duplex e full-duplex: nella full-duplex un nodo può ricevere e trasmettere contemporaneamente mentre nell’half-duplex o riceve o trasmette ma non può fare entrambe le azioni contemporaneamente.

**Implementazione del livello di collegamento**

Il protocollo del livello di collegamento è realizzato da un adattatore di rete detto NIC (scheda di rete). In una scheda vengono gestite sia la parte di livello 1 che 2 e dunque sono presenti componenti sia hardware che software. Nella parte del lato mittente, il controller incapsula un datagramma in un frame riempendo i vari campi dell’intestazione e lo trasmette sul canale di comunicazione. Dall’altra parte, il lato ricevente verifica la presenza di errori, estrae il contenuto del datagramma e lo passa al livello superiore.

**5.2 Rilevamento di errori**

Viene fatto in hardware quindi è più sofisticato. Ai dati D che devono essere protetti da errori vengono aggiunti dei bit chiamati EDC ed entrambi vengono inviati in un frame al nodo ricevente, il quale legge una sequenza di bit D’ e EDC’ modificata dai bit in transito e il nodo ricevente deve capire se D’ coincide con D e nel caso di errori li rileva altrimenti se tutto va bene invia il datagramma al livello superiore. Una tecnica per rilevare errori nei dati trasmessi è il controllo di parità:

si può usare un solo bit di parità e l’unica cosa da fare è contare il numero di bit uguali a 1 tra quelli ricevuti: se il numero di bit è pari è tutto ok se invece è dispari so che si è verificato un errore anche se non so dove. Nel caso in cui si è verificato un numero pari di errori ci troviamo nel caso di un errore non rilevato ma ci si concentra sul fatto che è più probabile avere un errore su un singolo bit piuttosto che averne due, la cui probabilità è più bassa.

Se si usa una tecnica di parità bidimensionale, i bit di parità indicheranno la i-esima riga e la j-esima colonna in modo che il ricevente non solo si accorge dell’errore ma può anche identificarlo e correggerlo. Questa capacità di rilevare e correggere errori è detta forward error correction (FEC).

**5.3 Protocolli e collegamenti ad accesso multiplo**

Esistono due tipi di collegamento di rete: punto a punto e broadcast. Il primo è costituito da un trasmittente e da un ricevente mentre il secondo può avere più nodi trasmittenti e riceventi e quando un nodo trasmette un frame lo inoltra a tutto i nodi. Come si coordina l’accesso di più nodi trasmittenti e riceventi in un canale broadcast condiviso? Quando due o più nodi trasmettono un frame, i nodi riceventi ricevono contemporaneamente più frame creando delle collisioni e per risolvere occorre coordinare le trasmissioni dei nodi trasmittenti e questo è il compito dei protocolli ad accesso multiplo. Un protocollo di accesso multiplo per un canale broadcast con velocità R bit al secondo dovrebbe avere le seguenti caratteristiche:

-quando un solo nodo deve inviare dei dati, questo ha un throughput pari a Rbps.

-quando M nodi devono inviare dati, questi hanno un throughput pari a R/M bps.

-il protocollo è decentralizzato ossia non ci sono nodi principali e non ci sono sincronizzazioni di clock.

-il protocollo è semplice.

I protocolli ad accesso multiplo posso essere divisi in protocolli a suddivisione del canale, che si basa sull’idea di dividere la risorsa in parti più piccole e allocare le porzioni ai singoli nodi, protocolli ad accesso casuale, che si basa sull’idea che la risorsa non è condivisa e permetto ai nodi di comunicare quando vogliono accettando le collisioni, ed infine protocolli a rotazione, che si basa sull’idea che vengono assegnati dei turni ai nodi per trasmettere eliminando le collisioni.

**Protocolli a suddivisione del canale**

La suddivisione della risorsa in parti più piccole può essere a divisione di tempo (TDMA) e a divisione di frequenza (FDMA). Nel primo viene suddiviso il canale in intervalli di tempo, ciascuno suddiviso in N slot temporali. Ogni slot è quindi assegnato ad uno degli N nodi e ogni volta che un nodo deve inviare un pacchetto, trasmette i bit durante il suo slot di tempo che gli viene assegnato. Nel secondo viene suddiviso il canale in frequenze differenti e assegna ogni frequenza ad un nodo.

**Protocolli ad accesso casuale**

Nell’accesso casuale, un nodo trasmette sempre alla massima velocità R bps e quando si verifica una collisione, i nodi coinvolti ritrasmettono i loro frame fino a che non raggiungono la destinazione senza collisioni. La ritrasmissione non è immediata ma avviene dopo un random delay e ogni nodo seleziona un ritardo casuale indipendente da quello degli altri nodi. Esempi di protocolli ad accesso casuale solo lo slotted ALOHA, ALOHA e CSMA.

**Slotted ALOHA**

Assumiamo che:

-tutti i frame hanno la stessa dimensione

-il tempo è diviso in slot di L/R secondo ossia il tempo di trasmissione di un pacchetto

-i nodi cominciano la trasmissione dei frame solo all’inizio degli slot

-i nodi sono sincronizzati in modo che sanno quando iniziano gli slot

-se in uno slot due o più frame collidono allora tutti i nodi della rete rilevano questo evento prima della fine dello slot.

Le operazioni dei nodi sono:

-quando un nodo ha un frame da inviare, attende fino all’inizio dello slot successivo per poi trasmettere l’intero frame

-se non c’è collisione, l’operazione è andata a buon fine

-se avviene una collisione, il nodo la rileva prima della fine dello slot e ritrasmette con probabilità p compresa tra 0 e 1 il frame durante gli slot successivi fino a che l’operazione non va a buon fine.

I vantaggi dello slotted ALOHA sono, per esempio, il fatto che consente ad un singolo nodo di trasmettere alla massima velocità nel caso in cui esso sia l’unico nodo attivo. Inoltre, è decentralizzato perché ogni nodo rileva le collisioni e decide quando ritrasmettere ed è semplice. Gli svantaggi sono che le collisioni comportano uno spreco di slot e di energia, le collisioni vengono rilevate solo alla fine della trasmissione dell’intero frame e richiede di sincronizzare i clock.

L’efficienza di un protocollo ad accesso multiplo è definita come la frazione di slot riusciti in presenza di tanti nodi attivi con tanti pacchetti da inviare. Supponiamo che ogni nodo, in questo caso N nodi, ha sempre un frame da spedire e che il nodo trasmette con probabilità p. La probabilità che uno slot sia vincente è data dalla probabilità che un solo nodo trasmette (p) e che N-1 nodi siano inattivi ((1-p)^N-1). Quindi la probabilità che un dato nodo abbia successo è p(1-p)^N-1 e la probabilità che un nodo arbitrario abbia successo è Np(1-p)^N-1. Dunque, con N nodi attivi, l’efficienza è Np(1-p)^N-1 e per trovare la massima efficienza con N nodi attivi devo trovare p\* che massimizza questa espressione ossia il lim per N che tende a infinito dell’espressione e troviamo p\*=1/N e che la massima efficienza è 1/e=0.37 ossia solo il 37% degli slot nel caso migliore compie lavoro utile.

**ALOHA**

Il primo protocollo ALOHA era privo di slot e decentralizzato e appena arrivava un frame il nodo lo trasmetteva subito al canale broadcast. Se c’era collisione il nodo ritrasmetteva subito con probabilità p. Quindi per determinare l’efficienza di ALOHA analizzo un singolo nodo e come unità di tempo prendo il tempo di trasmissione di un frame. La probabilità che un nodo stia trasmettendo è p e, supponendo che la trasmissione inizi a t0, nessuno deve cominciare un’altra trasmissione nell’intervallo (t0-1,t0) affinché vada a buon finela trasmissione. La probabilità che nessuno cominci un’altra trasmissione è (1-p)^N-1 e analogamente la probabilità che nessuno trasmetti mentre il nodi i sta trasmettendo è sempre (1-p)^N-1. Quindi la probabilità che un nodo trasmetta con successo è p(1-p)^2(N-1) e facendo il limite trovo che l’efficienza massima è la metà di quella di slotted ALOHA ossia 0.18 ossia il 18% degli slot nel caso migliore compie lavoro utile.

**CSMA**

In ALOHA i nodi trasmettono un frame a prescindere dal fatto che un altro nodo potrebbe star trasmettendo un altro frame. CSMA si comporta in maniera più “educata”, in particolare ascolta il canale prima di trasmettere e se il canale sta già trasmettendo, il nodo aspetta finché non si è liberato. Inoltre, rimane sempre in ascolta del canale e se un altro nodo sta trasmettendo un frame che interferisce con il proprio, si ferma, attende un intervallo di tempo e poi riprova.

Nonostante ciò, è possibile che avvengano delle collisioni. Infatti, se per esempio un nodo A ad un certo istante vede il canale libero, comincia la trasmissione e i suoi bit iniziano a propagarsi. Se ad un instante successivo un nodo B deve trasmettere un frame ma i bit del nodo A non lo hanno ancora raggiunto, il nodo B pensa che il canale sia libero e comincia a trasmettere e si creerà una collisione. Quindi, il ritardo di propagazione ha un ruolo fondamentale per determinare l’efficienza del protocollo e maggiore è il ritardo, maggiore è la probabilità che un nodo non si accorga che in realtà qualcuno prima di lui ha già cominciato una trasmissione.

Per questo motivo viene aggiunta la collision detection. Grazie al CSMA/CD i nodi, se si accorgono di una collisione, smettono di trasmettere e dunque grazie al sensing del canale capiscono se avvengono delle collisioni.

**Protocolli a rotazione**

Sono stati creati poiché ALOHA e CSMA non funzionano bene quando sono attivi M nodi e i più importanti sono il protocollo polling e il protocollo token-passing.

**Protocollo Polling**

In questo protocollo uno dei nodi è detto Master e interpella a turno gli altri, inviando un messaggio al nodo 1 dicendogli che può trasmettere fino ad un numero massimo di frame, successivamente inviando un messaggio al nodo 2 dicendogli che può trasmettere fino ad un numero massimo di frame e così via per tutto gli altri nodi. In questo modo vengono eliminate le collisioni e gli slot vuoti ma si presentano alcuni svantaggi come l’introduzione del ritardo di polling, ossia il tempo necessario per avvisare un nodo che può trasmettere, e il fatto che se il Master si guasta, l’intero canale diventa inattivo.

**Protocollo token-passing**

IN questo protocollo non esiste un Master ma un messaggio di controllo detto token che viaggia tra i nodi con un ordine prefissato. In pratica il nodo con il token trasmette un numero massimo di frame consentito e poi passa il token al nodo successivo che farà la stessa cosa. Se un nodo non ha frame da trasmettere, passa direttamente il token al nodo successivo. Questo protocollo è decentralizzato ed efficiente ma, nel caso in cui un nodo si guasti, l’intero canale si blocca oppure se un nodo dimentica di inoltrare il token bisognerà recuperarlo.

**5.4 Indirizzo MAC e protocollo ARP**

Tutti i nodi hanno un indirizzo IP a 32 bit che è l’indirizzo di livello di rete, usato per gestire i datagrammi. L’indirizzo che invece caratterizza le singole schede è detto indirizzo MAC, formato da 48 bit, il cui ruolo è quello di garantire l’indirizzamento all’interno di una rete di livello due. Se considero una rete LAN, ogni nodo collegato ad essa ha un indirizzo MAC, gestito dalla IEEE. L’indirizzo MAC è di tipo flag ossia non è legato alla posizione del nodo nella rete ma individua quella scheda all’interno della rete mentre gli indirizzi IP devono essere aggiornati in base alla posizione del nodo nella rete. Il protocollo ARP ha il compito di permettere la risoluzione dei due indirizzi: dato un indirizzo IP voglio conoscere l’indirizzo MAC. Per fari ciò, ogni nodo IP ha una tabella ARP che contiene le informazioni dei mapping tra indirizzi IP e MAC e in aggiunta contiene un valore detto TTL (time to live) che indica quando bisogna eliminare una data informazione della tabella (tipicamente 20 minuti).

-Per capire il funzionamento del protocollo ARP, suppongo che A voglia inviare un datagramma a B, il quale sta nella stessa sottorete, e l’indirizzo MAC di B non è nella tabella ARP di A. Per cui A chiede, tramite un messaggio broadcast, l’indirizzo MAC del nodo B e B risponderà con l’informazione richiesta. Una volta ricevuta la risposta potrò comunicare con il nodo B ed inoltre mi salvo questa informazione nella tabella ARP. ARP è quindi un protocollo plug and play, ossia un nodo si collega alla rete e se vuole un indirizzo MAC di un nodo basta che lo chiede.

-Il messaggio ARP è formato da 28 byte che contiene le informazioni per fare la traduzione da indirizzo IP e indirizzo MAC. Nel messaggio di richiesta, il nodo richiedente invia le informazioni del suo indirizzo MAC in modo che tutti sanno chi ha fatto la richiesta e il suo indirizzo MAC.

-È un protocollo particolare perché funziona solo nell’ambito della stessa rete LAN e non posso chiedere indirizzi MAC di nodi all’esterno della mia rete ed inoltre un messaggio ARP si limita solo a chiedere l’indirizzo MAC associato ad un preciso indirizzo IP. Per questi motivi lo si può pensare come un protocollo di livello due ma in realtà si considera come protocollo intermedio, ossia un protocollo di livello tre perché risolve indirizzi tre in indirizzi due ma limitato all’interno di una sottorete.

**Routing ad un nodo esterno**

Immagino di voler inviare un datagramma ad un host esterno alla mia sottorete, quindi ho due sottoreti interconnesse da un router. Se A deve inviare un pacchetto a B deve capire se B si trova all’interno della propria sottorete, altrimenti deve inviare il pacchetto al router che collega le sottoreti e quindi si chiede quale è il next hop per il pacchetto. Quindi A deve inviare il pacchetto a B che sta in un’altra sottorete passando per il router però A, oltre a conoscere l’indirizzi IP di B e del router, deve conoscere l’indirizzo MAC del router e se non lo conosce lo richiede tramite un messaggio broadcast. A questo punto, grazie al servizio di livello due, il pacchetto viene inserito in un frame con indirizzo MAC del mittente e del destinatario (router). Una volta arrivato al router, il pacchetto viene analizzato e il router capisce che è destinato ad una rete collegata ad esso e dunque lo invia a B usando il livello due della rete B. Il pacchetto viene inserito in un frame e inviato a B.

**Ethernet**

Oggi il 99% delle reti LAN sono reti ethernet, sia perché funzionano bene sia perché sono economiche. La struttura di una rete ethernet è quella di una “stella”, ossia un componente centrale detto switch (inizialmente era un hub) a cui sono collegati tutti i nodi. Lo switch è privo di collisioni perché c’è un cavo che permette le trasmissioni da computer a switch e un altro che permette le trasmissioni da switch a computer.

La struttura di un frame ethernet è fatta da un’intestazione e un’informazione di coda: è presente un preambolo di 7 bytes, un campo di 6 byte di informazioni di destinazione e sorgente, un campo che indica il tipo