Reti di Calcolatori

4 ottobre 2022

Rosso Carlo

Contents

1 Introduzione

Computer network un insieme di computer connessi, in un modo per cui altrimenti sarebbero disconnessi o autonomi.

client-server model coinvolge due processi: un processo è il client, l'altro è il server. Tutti i client si connettono al medesimo server ed interagicono con esso o mediante esso.

peer-to-peer ogni utente mantiene un database locale e una lista degli utenti. I dati sono trasferiti da un utente all'altro.

PAN Personal Area Network sono le comunicazioni che avvengono tra dispositivi appartenenti alla stessa persona (e.g. apple watch, airpods, smartphone).

LAN Local Area Network una rete privata che opera in un singolo edificio: per esempio il wifi.

MAN Metropolitan Area Network una rete che si estende per un centro abitato: per esempio i cavi della TV.

WAN Wide Area Network una rete che si estende per un continente (tipo le reti satellitari).

Protocol Layering I protocolli sono delle regole delle comunicazioni: mediante i protocolli i computer sanno come interagire tra loro. I layer servono per astrarre la tecnologia ed essere in grado di controllare le comunicazione. In realtà esiste solo il livello hardware. Ciascun layer controlla le informazione inerenti al proprio layer e le passa al layer sottostante. Ogni layer comunica con un layer del proprio livello mediante un protocollo. Per passare da un layer a quello superiore o sottostante si utilizzano le interfacce (il layer sottostante da origine, compone quello sovrastante).
Una serie di layer e protocolli si chiama architettura del network.

2 Il layer fisico

Il livello fisico è quello mediante cui sono passati i dati. Ci sono tre tipologlie di trasmissioni:

- con cavo (wired);
- senza cavo (wireless);
- satellitari.

2.1 Wired

Lo scopo del layer fisico è quello di trasportare o trasferire bit da un computer ad un altro. Le reti di comunicazione più comuni sono formate da cavi in rame o dalla fibra ottica.
Ciascun tipo di comunicazione ha le proprie caratteristiche: frequenza, larghezza di banda (bandwidth), ritardo (delay), costo, semplicità nell'installazione e manutenibilità. La larghezza di banda indica la velocità di trasmissione di bit di un medium, si misura in Hertz (Hz).

Persistent Storage

Per portare i bit da un computer ad un altro possiamo salvarli in una memoria esterna e trasportare fisicamente la memoria fino al secondo computer dove gli saranno resi accessibili. Dal momento che la densità con cui sono memorizzati i bit in un dispositivo fisico aumenta in modo esponenziale (nel tempo, scala bene) questo metodo rimane uno dei più economici.
NB le trasmissioni da un computer ad un altro richiedono circa un giorno: il tempo che impiega la memoria a raggiungere il secondo computer partendo dal primo. D'altro canto con questo metodo è possibile spostare enormi quantità di dati nell'arco di una sola giornata; la velocità media di queste trasmissioni è di 70 Gbps per ciascuna scatola $60 \cdot 60 \cdot 60 cm^3$.

In questo caso la larghezza di banda è di circa 70 Gbps; questa tecnologia è molto economica, ma ha anche un ampio delay, i dati ci mettono una giornata ad arrivare dalla "richiesta".

Twisted Pairs

Twisted Pair è una connessione mediante due cavi in rame torti (torcere) tra di loro, come il DNA, in questo modo i campi magnetici prodotti dai cavi si annullano tra loro e l'informazione arriva più lontano. Un bit è rappresentato dalla differenza di carica dei due cavi: se i due cavi trasportano la medesima carica il segnale è 0 altrimenti è 1. I doppini intrecciati sono in grado di percorrere chilometri senza bisogno di ripetitori. Se sono utilizzati più doppini intrecciati, allora sono protetti da uno scudo che diminuisce le interferenze dall'esterno. Le twisted pairs trasmettono abbastanza dati: qualche centinaio di Mbps su un tragitto di qualche chilometro. Sono semplici da produrre e costano poco, per questo motivo sono molto diffusi a livello globale.
Diverse LAN utilizzano questa tipologia di cavi.

full-duplex twisted pairs che trasmettono i bit in entrambe le direzioni nello stesso momento.

half-duplex twisted pairs che trasmettono i bit in entrambe le direzioni, ma in momenti diversi.

simplex twisted pairs che trasmettono i bit in una solo direzione.

Più i due cavi sono torti tra loro, minore sarà l'interferenza che producono, migliore sarà il segnale nelle lunghe distanze. Per aumentare la velocità delle comunicazioni è sufficiente aumentare lo spessore dei cavi. Un ulteriore distizione consiste nell'UTP: unshielded twisted pair, in questo caso i cavi non sono protetti.

Coaxial Cable

Il cavo coassiale è più schermato e ha una larghezza di banda maggiore rispetto ai doppini intrecciati. Il cavo coassiale sta venendo sostituito con la fibra ottica. Fu utilizzato per le comunicazioni telefoniche ed è ancora adoperato per la televisione e per le comunicazioni a livello MAN.

Power Lines

Le connessioni per alimentare gli elettrodomestici di casa possono essere utilizzate per trasmettere dati. Poichè accendere e spegnere la luce causa interferenze, queste connessioni sono utilizzate in ambito domestico, per cui la perdita di dati nella connessione è poco rilevante. Permette di trasmettere fino a 500 Mbps.

Fiber Optic

Le connessioni mediante la fibra ottica potrebbero arrivare a 50,000 Gbps, ma non siamo in grado di convertire i segnali luminosi così trasmessi alla medesima velocità. Ad oggi, il limite della larghezza di banda è di circa 100 Gbps. In generale, è simpatico notare che è più costoso trasportare i bit piuttosto che calcolarli, ma è più veloce trasportare i bit invece di calcolarli.
Un sistema di trasmissione ottico è formato da tre componenti: la sorgente, dove viene prodotto il segnale luminoso; il media, il materiale attraverso cui passa la luce (si tratta di qualcosa di molto trasparente); il detector, lo strumento che assorbe la luce e la trasforma in segnale elettrico. Il media consiste in un cavo in vetro molto sottile. Se il diametro del cavo è inferiore ai 10 microns allora si dice che la fibra è single-mode; un media così sottile permette alla luce di viaggiare solamente in linea retta: sono diminuite le dispersioni. Una fibra single-mode permette di percorrere maggiori distanze senza ripetitori: arriva fino a 100 Gbps per una distanza di 100Km senza ripetitore.
Alla fibra del tipo single-mode, si affianca quella multimode, in questo caso il cavo può essere anche più spesso di 50 microns. Il cavo multimode è attraversato da diversi raggi di luce nello stesso momento. La fibra multimode è più economica, ma la larghezza di banda diminuisce con l'aumentare della distanza.

Due fibre possono essere connesse in tre modi:

- gli estremi terminano in un cottore che è connesso ad un fiber socket, in questo caso è perso dal 10 al 20% del segnale;

- possono essere accostate meccanicamente; ad un tecnico richiede 5 minuti unire due fibre in questo modo e si perde il 10% del segnale;
- i due estremi sono fusi, saldati, assieme. In questo caso si ottiene una fibra che è praticamente come una fibra unica, ma è un tipo di collegamento più complesso.

Ci sono due tipi di sorgente di luce: LED (light emitting diodes) e laser semiconduttori;

Item	**LED**	**Semiconductor laser**
data rate	low	high
fiber type	multi-mode	multimode o single-mode
distanza	breve	lunga
durabilità	lunga	breve
sensibilità della temperatura	nessuna	sostanziale
costo	economica	costosa

Table 1: differenze tra una sorgente led e una laser

I raggi di luce sono assorbiti da un fotodiodo che trasforma la luce in segnali elettrici. Il tempo di risposta del fotodiodo limita la velocità di banda a 100 Gbps.

Differenze tra fibra ottica e cavo in rame

Item	**Fibra ottica**	**Cavo in rame**
data rate	fino a 100 Gbps	centinaia di Mbps
distanza	fino a 50 km	5 km
interferenze	temperatura	tutte, meno temperatura
diametro	molto fino	molto grosso
peso	leggera (100 kg)	pesante (8000 kg)
costo di installazione	basso	elevato
direzione	unidirezionale o due frequenze	duplex
tap	impossibile	molto semplice
personale	pochi la sanno maneggiare	molti elettricisti
costo dell'interfaccia	elevato	economico

Table 2: differenze tra fibra ottica e cavo in rame

2.2 Wireless

Le telecomunicazioni wireless avvengono tutte con la medesima tecnologia: sono sfruttate le onde elettromagnetiche per propagare le informazioni (proprio come avviene per le comunicazioni wired). In questo caso le onde sono propagate in tutte le direzioni (in realtà dipende dalla frequenza delle onde utilizzata). La quantità di informazioni trasferibile dipende dalla potenza che arriva e dalla larghezza di banda. Per mandare le informazioni sono utilizzate le onde: più onde sommate assieme riescono a descrivere una funzione che rappresenta i bit. In particolare, se la derivata della funzione descritta è maggiore di 0 allora stiamo mandando un bit di valore 1, se la derivata della funzione è minore di 0 allora stiamo mandando un bit di valore 0. Per capire il segno della derivata è sufficiente memorizzare il valore dell'onda, misurarlo di nuovo e confrontare i due valori. Avendo più frequenze è possibile modificare la somma delle frequenze d'onda in modo più preciso: ogni volta che viene aggiunta una frequenza in più è possibile raddoppiare la quantità di informazioni che sono mandate.

Le connessioni wireless si distinguono per il range delle lunghezze d'onda, ma anche per le singole frequenze utilizzate: produrre alcune onde elettromagnetiche è più economico di altre. Per esempio produrre le onde elettromagnetiche che trasmettono i canali della radio è piuttosto economico. Comunicare mediante

onde ultraviolette è ben più costoso (inoltre queste onde sono nocive alle forme di vita). Minore è la frequenza d'onda maggiore è la distanza che quell'onda riesce a percorrere, minore è il costo per produrla. Ci sono dei pro e dei contro a questo: un'onda che non si ferma mai, che passa attraverso la Terra, è intercettabile ovunque, per cui deve trasmettere informazioni non segrete, altrimenti ha bisogno di essere criptata. Questo tipo di onda trasmette meno dati di un onda più costosa.

Il numero di oscillazioni al secondo di un'onda è chiamato frequenza e si misura in Hertz (Hz). La distanza tra due massime (oppure tra due minime) è chiamata lunghezza d'onda. I bit sono codificati modificando la portata (ampiezza), la frequenza o la fase dell'onda. Gli ultravioletti e i raggi gamma sono in grado di trasferire più informazioni, perchè la loro frequenza è maggiore; ma sono difficili da produrre e da modulare, non passano attraverso gli oggetti (ogni tanto anche il tempo atmosferico li blocca) e sono nocivi per gli animali (noi).

Frequency hopping spread spectrum

Questa tecnologia permette di diminuire le interferenze e rende le onde elettromagnetiche più difficili da rintracciare: le frequenze per trasmettere le informazioni cambiano centinaia di volte al secondo. viene utilizzato soprattutto in ambito militare.

direct sequence spread spectrum

Sono utilizzate più frequenze d'onda che si possono sovrapporre a quelle di altri. La connessione funziona anche se il dato è corrotto, perchè solo una frazione viene corrotta. Il ricevitore del segnale avrà un "codice" per decifrarlo. Genera interferenze con altre onde sulla medesima frequenza, ma non risulta un problema per chi riceve le informazioni grazie alla cifratura.

Ultra-WideBand

Viene adottata moltissima banda, almeno $500MHz$, e sono trasmessi una serie di impulsi a bassa intensità molto velocemente. Dal momento che il messaggio è molto diluito tra le frequenze le interferenze hanno poca importanza. Non trasmette tutto il tempo per cui non crea interferenze per altre tecnologie.

2.3 Analizziamo le frequenze

Radio Transmission

Le frequenze radio (RF) sono semplici da generare, viaggiano per grandi distanze e penetrano gli edifici; sono unidirezionali. A basse frequenze, il segnale passa attraverso gli ostacoli, ma l'intensità diminuisce rapidamente. Ad alte frequenze, il segnale viaggia dritto, ma rimbalza sugli ostacoli. Le interferenze con gli altri utenti sono un problema.

Microwave Transmission

Le onde viaggiano dritte, per cui il trasmettitore e l'antenna devono essere meticolosamente allineati. Si possono allineare più ricevitori in fila. La distanza a cui devono essere piazzati i ripetitori è il quadrato dell'altezza. Il segnale trasmesso da una torre alta 100m viaggia 80 km. Le microonde non passano attraverso gli ostacoli.
Aumentando la frequenze aumenta il data rate: oggi connessioni a 10GHz sono comuni. Oltre i 4GHz le onde sono assorbite dall'acqua, umidità compresa.
Le microonde sono economiche: non serve chiedere il permesso per piazzare i cavi, è sufficiente un ripetitore ogni 50 km.

Infrared Transmission

Le comunicazioni ad infrarossi sono molto utilizzate: sono economiche, direzionali e semplici da produrre; ma non passano attraverso gli oggetti. Sono utilizzate per le comunicazioni all'interno di una sola stanza (per esempio il telecomando della TV). Non sono intercettabili: non si propagano al di fuori della stanza.

Light Transmission

Sono molto semplici da utilizzare, ma hanno molte interferenze: il vento, la temperatura, la pioggia o la nebbia.

2.4 Codificare bit con le onde

Grazie alle serie di Fourier siamo in grado di descrivere matematicamente le onde. Grazie ad esse riusciamo a codificare i bit utilizzando le onde elettromagnetiche. Purtroppo, a seconda della frequenza utilizzata un'onda (elettroagnetica) si attenua in modo diverso.

Bandwidth la larghezza del range di frequenze che sono trasmesse senza che siano eccessivamente attenuate.

Baseband i segnali che utilizzano le onde che partono dalla frequenza 0 e arrivano ad una frequenza massima.

Passband i segnali che utilizzano un range di onde che non partono da 0 (sono traslate le baseband).

Data rate massimo $2B\log_2(1+S/N)$ bits/s, dove B sono le onde che non sono filtrate e S/N è il rapporto segnale/interferenza.

2.5 Digital Modulation

Le informazioni sono trasmesse modificando costantemente il voltaggio, l'intensità della luce o l'intensità del suono. In genere un canale è utilizzato nello stesso momento da più trasmettitori, questa tecnologia è chiamata multiplexing (l'abbiamo già vista!). Il multiplexing è ottenuto in diversi modi: trasmettendo segnali diversi in istanti diversi, utilizzando varie frequenze oppure codificando l'informazione.

Baseband transmission

NRZ Non-Return-to-Zero: il metodo più semplice per codificare i bit fa corrispondere 1 al voltaggio positivo e 0 a quello negativo; oppure 1 alla presenza di luce e 0 all'assenza.
In realtà non si utilizza una sola onda per trasmettere i bit: se si utilizzano due onde si possono mandare due bit in una volta sola, in particolare ogni onda rappresenta un bit. Per questo motivo, invece di usare il data rate è comodo introdurre il symbol rate (anche chiamato baud rate).

Clock Recovery

Con il metodo NRZ, il ricevitore ha bisogno di scandire il tempo in modo efficacie: se una serie di 0 o di 1 si susseguono si rischia di perdere informazioni. Sono mandati centinaia, talvolta migliaia, di bit al secondo, i sistemi per misurare il tempo in modo così preciso sono costosi, per questo in genere si ricorre a metodi di codifica differenti.

Manchester ecoding Sono trasferiti la metà dei bit, ma il tempo è scandito bene: un uno è codificato come segnale positivo e segnale nullo, mentre lo 0 è codificato come segnale nullo e segnale positivo. In questo modo non si sussegue più volte il medesimo segnale, ma i bit sono ben scanditi.

NRZI Non-Return-to-Zero Inverted: 1 è codificato come transizine dallo stato precedente ad uno diverso, se lo stato rimane invariato allora viene codificato lo 0. Questa tecnologia viene utilizzata per connettere le porte USB. In questo modo lunghe serie di 1 non causano alcun problema (lunghe serie di 0 sì).

4B/5B Ad ogni sequenza di 4 bit da trasmettere è aggiunto un bit. Il bit aggiuntivo è utilizzato per intervallare troppi 0 di seguito. In questo caso c'è un 25% di overhead (meglio del 100% del Manchester encoding).

2.6 Balanced signals

Alcuni componenti che trasmettono i messaggi elettrici sfruttano la tecnologia DC. Questo crea un problema: si spreca energia a mandare segnali non bilanciati. Un segnale è bilanciato se ci sono lo stesso numero di segnali positivi e negativi in brevi istanti di tempo.

Bipolar encoding lo 0 è rappresentato dal segnale nullo, mentre l'1 è rappresentato prima da un segnale positivo e poi da un segnale negativo, e così da capo. In questo modo il segnale sarà sempre bilanciato.

8B/10B si dividono 8 bit in due gruppi una da 3 bit e uno da 5 bit. I primi 3 bit sono codificati in 4 bit bilanciati rispetto ai 5 bit che sono mappati su 6 bit. Dal momento che si aggiungono bit lo stesso codice può essere mappato in modi diversi, la mappatura è scelta per bilanciare le cariche del segnale mandato.

Passband transmission

Questa tecnologia è utilizzata principalmente nelle comunicazioni wireless: in questo tipo di comunicazione è scomodo far partire la prima frequenza da 0 (le antenne dovrebbero essere gigantesche). Per trasmettere le informazioni si modula l'ampiezza (capacità), la frequenza o la fase del segnale.

ASK Amplitude Shift Keying, l'ampiezza di un'onda indica il segnale: ampiezza nulla codifica lo 0, ampiezza non nulla codifica 1. Più lughezze d'onda sono sovrapposte per rappresentare più bit nello stesso momento.

FSK Frequency Shift Keying, è cambiata la frequenza della portata dell'onda. Le frequenze sono predeterminate per cui il client e il server hanno prestabilito la codifica delle frequenze della portata.

Phase Shift Keying l'onda è traslata di 0 o di 180 gradi sul suo periodo. Per esempio traslare di 0 gradi corrisponde allo 0, mentre traslare di 180 gradi corrisponde all'1. In questo caso si tratta di Binary PSK. In realtà, si trasla di 45, 135, 225 o 315 gradi per trasmettere un simbolo. In questo modo ad ogni cambio di fase sono mandati due simboli contemporaneamente. Inoltre c'è sempre una traslazione, per cui siamo più tranquilli riguardo ad eventuali interferenze.

Si può cambiare solo la fase o la frequenza nello stesso momento, perchè le due variabili sono legate tra loro. Di solito, l'amplitude e la fase sono modificate nello stesso momento, in questo modo una sola onda è in grado di codificare otto simboli in una volta sola. Per comodità del programmatore la fase e l'amplitude sono codificate come cordinate sul piano complesso: la fase rappresenta l'angolo e l'amplitude il modulo. Ora siamo in grado di rappresentare le onde con due coordinate. Punti vicini tra loro codificano sequenze di bit simili, per diminuire la quantità di errori di chi riceve.

2.7 Multiplexing

Multiplexing vuol dire mandare più messaggi nello stesso momento. Ci sono diversi metodi per implementare il multiplexing:

- tempo;
- frequenze;
- codifica.

FDM Frequency division multiplexing: le frequenze sono divise in slot, ciascun canale utilizza tutte le frequenze dello slot a lui assegnato.
Ciscun canale è diviso da una guard band per evitare interferenze. Veniva usato per le linee telefoniche, oggi si preferisce dividere il tempo a disposizione per mandare il segnale, piuttosto che la fascia di onde utilizzabili.

OFDM Orthogonal Frequency Division Multiplexing: come nel caso precedente le frequenze sono divise in canali. Ciascun canale è a sua volta diviso in sottocanali che mandano dati in modo indipendente, in questo modo anche se un sottocanale ha qualche interferenze il segnale arriva mediante un sottocanale diverso.

TDM Time division Multiplexing: ad ogni utente è assegnato un arco di tempo nel quale può mandare informazioni, in quell'arco di tempo ha a disposizione l'intera banda. Perchè questo sistema funzioni tutti gli utenti devono essere sincronizzati.

STDM Static time division multiplexing: aggiungendo l'aggettivo statico, vuol dire che il tempo assegnato ha ciascun utente è deciso staticamente: a run-time.

CDM Code division multiplex (access): il segnale è diviso in una banda più larga e tutti gli utenti comunicano contemporaneamente. La stazione assegna ad ogni utente il vettore di una base (algebra). La base ha come dimensione il numero di utenti. Ciascun utente trasmette il proprio vettore per codificare un 1 e la negazione del proprio vettore per codificare uno 0. Poichè tutti i vettori appartengono ad una base, i vettori sono linearmente indipendenti e conoscendo il vettore assegnato all'utente che vuoi ascoltare sei in grado di decodificare le onde. Per riuscire ad instaurare questo tipo di comunicazione c'è bisogno che i due utenti che vogliono comunicare si mettano d'accordo prima.

3 Cellular Networks

Cell una regione geografica è divisa in celle, ogni cella ha delle frequenze, che non sono utilizzate da nessuna delle celle ad essa adiacente, ma sono usate da celle vicine. La divisione in celle permette il riutilizzo delle frequenze. Le celle hanno la medesima dimensione e sono raggruppate in gruppi di 7, tutte le frequenze utilizzate dalle connessioni cellulari sono spartite tra le 7 celle che formano i gruppi nello stesso modo indipendentemente dal gruppo. Minore è lo spazio che una cella è tenuta a ricoprire più piccola è la cella che si necessita per ricoprire l'area in questione e più è economica. Sono anche distribuite delle microcelle, con uan divisione analoga a quella delle celle. La stazione di base consiste in un computer connesso ad un trasmettitore e un ricevitore con un'antenna.

MSC/MTSO ogni stazine di base è connessa a un Mobile Switching Center o Mobile Telephone Switching Office. In un sistema piuttosto grande sono necessari più MSC connessi ad un MSC di secondo livello. Un MSC comunica con le stazioni di base, con altri MSC e con il PSTN mediante un packet-switching network.

Handoff In ogni istante un telefono è connesso ad una sola stazione di base (si trova in una sola cella), quando un telefono si sposta ed esce dalla ricezione della cella in cui si trovava, il computer nella cella si accorge che il segnale si sta per interropere e chiede alle celle vicine quale sia quella che riceve meglio il segnale del dispositivo. Così la stazione di base assegnata al telefono cambia. Il telefono è informato del cambio di cella. Questo processo richiede circa 300 ms.

Le reti telefoniche hanno quattro canali: il canale di controllo: per gestire il sistema; il canale di paging: per gli avvisi di chiamate; il canale di accesso (bidirezione) per comunicare al telefono quale canale utilizzare; il canale dei data per trasmettere chiamate, fax e altri data.

3.1 1G

IMTS Improved Mobile Telephone System: utilizza un trasmettitori ad alta potenza (200 watt) in cima ad una collina. Utilizza due frequenze: una per mandare e una per ricevere. In questo modo gli utenti che mandano più informazioni contemporaneamente non si sentono a vicenda. L'IMTS supporta 23 canali, per cui ogni tanto gli utenti sono tenuti ad aspettare che un canale si liberi prima di riuscire ad usarlo. Inoltre ricoprono un'area molto estesa perché la potenza del segnale è elevata, impedendo di installare sistemi simili nelle vicinanze. ITMS sono a centinaia di chilometri distanti.

AMPS L'Advanced Mobile Phone System incrememta la capacità delle reti cellulari: si gestiscono più dispositivi. Celle di AMPS sono distanti dai 10 ai 20 chilometri. L'AMPS riesce a gestire 10-15 chiamate sulla stessa frequenza. AMPS utilizza l'FDM per separare i canali. L'AMPS utilizza 832 canali full-duplex, ciascuno composto da una paio di canali simplex, che si dice FDD, Frequency Division Duplex.

Call Management ciascun dispositivo in una rete AMPS ha un numero seriale di 32 bit e un numero di telefono di 10 cifre, nella propria ROM. In molti stati il numero di telefono è diviso in 3 cifre per rappresentare lo stato e 7 cifre per rappresentare il numero rappresentate in 24 bit. Quando un telefono si connette ad una cella scansiona 21 canali di controllo per trovare il segnale più forte. Poi manda i bit del numero seriale e 34 bit che rappresentano il numero di telefono. L'MSC registra il dispositivo e informa il dispositivo della sua posizione. In genere questa operazione è ripetuta ogni 15 minuti.
Per effettuare una chiamata il dispositivo trasmette il numero da chiamare e la propria identità (numero seriale e numero di telefono) nel canale di accesso. L'MSC identifica un canale adatto alla chiamata e lo inoltra al telefono chimante, che aspetta che il dispositivo chiamato si connetta.
Tutti i telefoni controllano continuamente il canale di paging per eventuali messaggi o chiamate a loro diretti. Quando si effettua una chiamata per un dispositivo, è mandato un pachetto all'MSC che lo gestisce per trovare il dispositivo chiamato. Poi è mandato un pacchetto sul canale di paging della stazione di base (cella) del dispositivo. Il dispositivo risponde e allora gli viene indicato in quale canale connettersi. Il dispositivo si connette e suona.

3.2 2G

La seconda generazione è digitale. Aumenta ulteriormente la capacità permettendo alla voce di essere digitalizzata e compressa, inoltre permette la criptazione delle comunicazioni e del segnale di controllo. Aggiunge i messaggi testuali. Non c'è ancora una standardizzazione globale delle reti.

GSM Il Global System for Mobile Communications è il single European standard. Riutilizza il design a celle, il riuso delle frequenze e la gestione handoff dei dispositivi che si muovono. Il dispositivo è separato dalla SIM card (Subscriber Identity Module). Le stazioni di base sono connesse a un **BSC** (Base Station Controller) che controlla le cellle e gesisce lo handoff. Il BSC è connesso ad un *MSC*, come nel sistema AMPS. L'MSC fa da router per le telefonate ed è connesso ad un *PSTN* (Public Switched Telephone Network).
L'MSC mantiene un database che memorizza i dispositivi connessi alle celle che gestisce, il VLR (Visitor Location Register). C'è un uteriore database, l'HLR (Home Location Register), che è utilizzato per mandare le chiamate in entrata nella cella corretta.

Il GMS dispone di 124 canali full-duplex, ciascuno composto da 2 canali simplex, divisi per frequenza. Ciascun canale è dotato di 200 kHz di larghezza di banda e supporta 8 chiamate. Ciascuna stazione ha assegnato uno slot di tempo in una coppia di canali. Il dispositivo trasmette e riceve su slot ti tempo diversi. I canali di controllo sono di 3 tipologie:

- canale di controllo della trasmissione: si tratta di un segnale continuo, dalla stazione di base al dispositivo, e contiene l'identità della stazione di base e lo stato del canale;
- canale di controllo dedicato: usato per aggiornare la posizione, la registrazione e impostare le chimate dei dispositivi;
- canale di controllo comune: diviso in tre canali logici: il canale di paging, il canale di accesso casuale, che permette ai dispositivi di chiedere l'accesso al canale di controllo dedicato, e il canale di accesso garantito.

L'invio di informazioni è limitato ad un ottavo del tempo: il GMS mantiene più dispositivi dividendoli in gruppi chiamati *time slots*. In questo modo, il dispositivo, quando non manda o riceve, può verificare quale sia la cella con potenza di segnale maggiore. L'handoff viene gestito in questo modo e il design è chiamato *MAHO* (Mobile Assisted Handoff).

3.3 3G

Vedi appunti kara, mi sembra che nel libro non ci sia l'argomento riguardante il GPRS, l'evoluzione del GSM.

CDMA Il CDMA (Code Division Multiple Access) sfrutta il CDM 2.7 per permettere a più celle di utilizzare le medesime frequenze. Al posto di trovare basi indipendenti e ortogonali, tuttavia sono utilizzate sequenze pseudo-casuali che con alta probabilità non si sovrappongono. Le celle sono sincronizzate anche dal punto di vista della potenza del segnale. Più le celle mobili sono lontane dalla stazione di base, maggiore è la

potenza del segnale che gli mandano indietro, in questo modo alla stazione di base arrivano sommate diverse onde che però sono ortogonali e facilmente disgregabili.
Inoltre il CDMA riesce a sfruttare i momenti in cui una persona non parla e quindi non trasmette i dati: circa il 40% del tempo. Quando non sono mandati dati, le interferenze diminuiscono e la qualità del segnale aumenta.
Con il CDMA sono utilizzate le medesime frequenze per tutti i canali, che vuol dire che serve meno pianificazione per la gestione delle frequenze e aumenta la capacity della rete.
In terzo luogo il CDMA facilita il *soft handoff* tra le celle, in quanto tutte le celle utilizzano le medesime frequenze e quindi celle differenti sono sempre in grado di gestire anche le chiamate delle celle vicine.

4 Link Layer

Il link layer si occupa di ottenere comunicazioni affidabili ed efficienti per unità di informazioni chiamate *frame* tra due dispositivi adiacenti. Con adiacenti intendiamo due dispositivi che sono collegati e i cui bit giungono al secondo dispositivo nello stesso ordine in cui sono mandati dal primo.
Il link layer raffina le funzioni del layer fisico, in particolare si occupa di:

1. fornire un'interfaccia ben definita al layer superiore (network layer);
2. assimilare più bit restituendo dei segmenti di informazione maggiori;
3. trovare e correggere gli errori;
4. regolare il flusso di informazioni tra i due dispositivi.

Frame Il link layer prende i pacchetti del livello superiore e li incapsula in frame: ogni frame condiene un *header*, un *payload* e un *trailer*. Il frame è l'unità di trasmissione del link layer.

La funzione principale del link layer è quella di trasmettere un pacchetto del livello superiore nella macchina attuale ad un'altra macchina sul livello superiore, funge da ponte.
Analizziamo le tre tipologie di connessione che può gestire un link layer:

1. unacknowledged connectionless service: nessuna connessione logica è effettuata prima di mandare i dati e nessuna connessione logica è chiusa dopo. Semplicemente il dispositivo A manda i dati al dispositivo B; per esempio connessione ethernet;
2. acknowledged connectionless service: per ogni frame mandato dal dispositivo A al dispositivo B, il dispositivo B riferisce al dispositivo A l'esito della ricezione del frame; per esempio connessione wifi;
3. acknowledged connection-oriented service: è preparata una connessione prima che si cominci a mandare i dati ed è chiusa dopo che si sono mandati tutti i dati. I frame sono numerati; per esempio connessioni satellitari.

In realtà non è detto che queste tre tipologie siano fornite dal link layer, ma se sono fornite a questo livello sono più ottimizzate. Per esempio, per connessioni come la fibra ottica, il link layer potrebbe non essere implementato, perchè si tratta di una connessione affidabile.

Framing Il link layer sfrutta i servizi del layer fisico, il layer fisico accetta delle stringhe di bit grezze e prova a spendirle a destinazione. In qualche caso il layer fisico aggiunge della rindondanza per favorire il ricevimento dei dati corretti. Con framing intendiamo l'elaborazione della stringa di bit grezza in frame, per riorganizzare i dati. Quindi bisogna dividere la stringa di bit grezza in modo tale da occupare poca nella larghezza di banda disponibile. Approfondiamo quattro modi:

1. Byte count: un campo nello header specifica quanti byte ci sono nel frame attuale;
2. flag bytes with byte stuffing: è utilizzato un byte speciale per segnalare l'inizio di un frame e la fine dello stesso. Questo byte potrebbe essere presente anche all'interno del frame, per questo motivo se è questo il caso, il byte è preceduto da un altro carattere speciale. Anche questo carattere speciale potrebbe essere presente all'interno del frame, per questo motivo, ogni volta che si usa il secondo carattere speciale all'iterno del frame di per sè, viene ripetuto;

3. flag bits with bit stuffing: è utilizzata una sequenza di bit per segnalare l'inizio e la fine di un frame. Se questa sequenza è presente nel frame di per sè, allora viene spezzata in modo tale che il dispositivo ricevente possa elaborare il frame correttamente;

4. physical layer coding violation: sono utilizzati dei bit che non possono essere generati dal layer fisico, si pensi alla codifica 4B/5B;

Error Control Questo sotto branch del link layer si preoccupa di mandare al dispositivo inteso sul network layer tutti i frame e nell'ordine corretto. Nel caso si utilizzi l'unacknowledged connectionless service, questo problema non si pone. Altrimenti, in genere si fa in modo che il link layer del mittente abbia un feedback sui frame che spedisce. Un segnale positivo vuol dire che il frame è arrivato con successo, altrimenti il frame deve essere rimandato. Ogni tanto, può succedere che, per qualche problema hardware per esempio, il link layer non riceva alcun feedback. Per ovviare ad un deadlock, il link layer del mittente ha un timer che gli indica entro quando il feedback dovrebbe essere arrivato. Se non arriva alcun feedback, il mittente rimanda il frame in questione. Dal momento che chi riceve potrebbe non sapere se un frame è stato mandato due volte oppure se si tratta di due frame distinti, i frame sono numerati. Ulteriori raffinamenti dei pacchetti mandati avvengono al livello del network layer o sopra.

Flow Control Un dispositivo che riceve, potrebbe non essere in grado di ricevere tanti dati quanti sono quelli che il mittente è in grado di spedire. Per questo motivo, oltre al feedback che segnala se un frame è arrivato o meno, il ricevitore rimanda al mittente un segnale che indica se può ricevere altri frame o no. Questo design si chiama *feedback-based flow control*.
Un altro design è chiamato *rate-based flow control*, questo protocollo ha un meccanismo che limita la quantità di dati che il mittente può spedire, senza bisogno di un feedback; non approfondiremo ulteriormente questo metodo.

4.1 Error Detection and Correction

Una strategia che hanno sviluppato i designer di sistemi di comunicazione è la rindondanza: rimandare lo stesso frame più volte, in modo tale che il ricevente sia in grado di correggere eventuali errori; oppure in modo tale che il ricevente sia in grado di capire se ci sono stati errori nella comunicazione e chiedere quindi al mittente di rimandare il frame. La prima strategia sfrutta dei codici di correzione degli errori, anche chiamata **FEC** (Forward Error Correction), il secondo utilizza dei condici di rilevamento degli errori. Per connessioni come la fibra ottima è meglio utilizzare il secondo sistema, perchè lgi errori capitano molto raramente. Per connessioni come il wifi, è meglio utilizzare il primo sistema, perchè gli errori capitano molto spesso.
In generale sono distinte due tipologie di errori: gli errori singoli e gli errori in blocco e le metodologie per arginarli sono diverse.
Notiamo che codici di correzione degli errori possono essere trovati anche al livello fisico, soprattutto per connessioni che sfruttano canali poco affidabili. Mentre codici di rilevamento degli errori possono essere trovati nei layer link, network e transport.

Error Correction

Hamming Code L'Hamming Code lo spiego nel particolare negli appunti di Architettura degli Elaboratori. In breve sia (n, m) code, un codice di m dati che contengono informazioni e $n - m$ bit di rindondanza. La codeword è definita come n; il code rate è definito come m/n; la distanza di Hamming è il numero di bit diversi tra due stringhe di bit. Per individuare d bit di errore, è necessario che il codice di correzione sia di almeno $d + 1$ bit; per correggere d bit di errore, è necessario che il codice di correzione sia di almeno $2d + 1$ bit.
La codeword originale è univocamente determinata un numero maggiore di errori è più improbabile, viene quindi scelta la codeword con distanza di Hamming minima.

Convolutional Code Un codice di convoluzione è un codice che utilizza una sequenza di bit per generare una sequenza di bit di output. L'output dipende dalla sequenza di bit dell'input e dalle sequenze in input precedenti.
Il codice di convoluzione è parecchio utilizzato perchè tiene in considerazione della probabilità di un errore

di un bit: se arriva un segnale disturbato che è difficile da tradurre in bit, si tiene conto del fatto che è più probabile che quel bit sia sbagliato piuttosto di un altro.

Reed-Solomon Code Reed-Solomon è un codice che corregge gli errori in modo lineare nei blocchi. L'algoritmo si basa sul fatto che ogni polinomio di grado n è univocamente determinato con $n+1$ punti; i punti in più che sono dati risultano rindondanti e sono utili per la correzione degli errori. Sia un simbolo composto da m-bit, allora la codeword è lunga $2^m - 1$ simboli. Reed-Solomon è utile perchè è in grado di rivelare errori in blocco, che sia un singolo bit ad essere sbagliato o un intero simbolo, il meccanismo per correggerlo prevede la correzione dell' intero simbolo. Se sono trasmessi $2t$ simboli rindondanti, allora questo algoritmo è in grado di corregge fino a t simboli. Se gli errori constano nel fatto che si sa che c'è un bit, ma non si conosce il valore del bit in questione, allora riesce a correggere $2t$ errori. In genere l'algorimo Reed-Solomon è utilizzato con i codici convoluzionali: gli ultimi correggono i singoli bit, e questo interi simboli sbagliati.

Low-Density Parity Check LDCP è un codice che corregge gli errori in modo lineare nei blocchi. Viene praticamente zippato il codice e poi viene ricostruito il codice nel modo che più probabilmente rappresenta quello che è stato mandato. Solo una porzione dell'input è utilizzato per formare l'output e l'output è poi rappresentato come una matrice.
LDPC è utilizzato per blocchi di dati molto grandi ed è eccellente per quanto riguarda la correzione degli errori.

Error Detecting

Su connessioni come la fibra ottica o cavi in rame di alta qualità gli errori sono poco frequenti e risulta più efficiente capire se un errore è stato commesso e quindi proseguire richiedendo il frame nuovamente.

Parity Viene aggiunto un singolo bit di parità alla fine del frame. Un codice con un solo bit di parità ha distanza di Hamming pari a 2. La probabilità di rilevare errori per errori in blocco è del 50%. Un sistema più affidabile consiste nel dividere il codice per n e così formare una matrice di dimensione $n \cdot k$, dove $n \cdot k$ è il numero di bit che forma il frame. A questo punto per ogni colonna si trova il bit di parità. Risulta più opportuno trovare un bit di parità per rilevare con più accuratezza errori in blocco, infatti una stringa di bit invertiti dovrebbe essere lunga $2n$ perchè l'errore venga completamente celato, oppure devono essere presenti due errori nella stessa colonna. La probabilità che un errore non sia rilevato è di 2^{-n}

checksum Si esegue una sommatoria di una funzione di n bit e alla fine del frame, in genere, è aggiunto il risultato. Per esempio, la funzione che somma modulo 2^8 di interi rappresentati in 16 bit, avrà $n = 16$ e il risultato sarà un intero di 16 bit; se il risultato della somma è uguale alla checksum, finale allora non si è verificato alcun tipo di errore. Notiamo che questo algoritmo è vulnerabile nel qual caso il risultato della somma non cambi.
Un algoritmo migliore risulta essere quello di Fletcher, che somma i prodotti tra il valore dei 16 bit e la posizione nel frame della stringa di 16 bit.

Cyclic Redundancy Check CRC è anche conosciuto come codice polinomiale. In questo caso una stringa di bit viene identificata come se fosse un polinomio. Praticamente viene concordato un polinomio, $G(x)$ tra i dispositivi che comunicano. Vengono aggiunti tanti 0 alla fine del frame quanto vale il grado del polinomio. Infine viene mandato il maggiore polinomio, minore o uguale al frame aumentato degli 0, che è divisibile per $G(x)$.

$$T(x) = \frac{x^r M(x) - x^r M(x) \% G(x)}{G(x)} \tag{1}$$

dove r è il grado del polinomio $G(x)$, $M(x)$ è il frame e $T(x)$ è il valore che è mandato.
Questo algoritmo trova tutti gli errori in stringhe di bit di lunghezza massima r. Il CRC è un algorithm molto efficiente e facilmente implementabile in hardware.

4.2 Protocolli del data link notevoli

Innanzitutto ripetiamo che un frame è formato da tre field: un header, il payload e il trailer; come abbiamo già visto l'header potrebbe contenere informazioni del tipo, dimensione del frame, ack (acknowledged), tipi di dato che sono mandati, il numero del frame; il trailer tendenzialmente contiene il risultato della checksum. Al livello superiore non arrivano queste informazioni: layer diversi devono essere il più indipendenti possibili, per facilitare lo sviluppo dei layer. Ovviamente duplicati o frame sbagliati non sono passati al livello superiore.

stop-and-wait Un dispositivo A manda un frame al dispositivo B. A fa partire un timer. Ora ci sono due cose che possono accadere: il pacchetto arriva su B intatto oppure no. Se il pacchetto arriva su B intatto B manda un segnale ad A. Altrimenti B non fa nulla, il timer scade e A rimanda il pacchetto. Notiamo che se il segnale che B manda ad A viene perso A rimanda a B lo stesso frame. Per evitare che B scambi questo frame per un nuovo frame, il frame nel suo header ha il numero del frame. Se viene mandato solo un frame per volta allora il numero del frame sarà o uno 0 o un 1 e mano a mano che sono mandati i pacchetti questi valori si alternano. D'altro canto B per dire ad A che gli è arrivato il pacchetto rimanda ad A il numero del pacchetto che ha ricevuto. Questo tipo di protocolli sono spesso chiamati *ARQ* (Automatic Repeat reQuest) o *PAR* (Positive Acknowledgement with Retransmission).

piggybacking Il dispositivo A manda un frame al dispositivo B. A fa partire un timer. Poniamo il caso che il pacchetto che è arrivato è intatto (altrimenti il timer scade e A rimanda il pacchetto). B fa partire un timer, che chimiamo timer di ack (acknowledgement). Se un pacchetto dal livello superiore arriva sul link layer prima dello scadere del tempo (e questo pacchetto è diretto verso A), allora il bit di ack (di conferma di ricezione) viene incorporato nel frame che viene mandato a A, nel campo dell'header ack. Altrimenti il bit di ack viene mandato in modo indipendente, sprecando la banda. Notiamo che questa tecnica permette di risparmiare un bel po' di banda, nel momento in cui il segnale di ack viene mandato insieme al pacchetto, al pacchetto mandato viene aggiunto un bit. Altrimenti bisogna costruire un frame ad hoc, con tanto di checksum, per mandare il segnale di ack.

sliding window Il dispositivo A e il dispositivo B si mettono d'accordo sul numero di frame che possono essere mandati senza ricevere un ack. Questo numero è chiamato *window size*, let's call it n. A e B si mettono d'accordo anche sul numero che identifica ciascun frame: un bit evidentemente non basta più.
Quello che viene fatto è implementare un buffer circolare, sia sul dispositivo A che sul dispositivo B. A e B cominciano con due indici, low e high, la loro differenza è n. n è anche il numero minimo di buffer che devono avere A e B. Ogni volta che A riceve un ack da B, sia A che B incrementano low di uno modulo n, analogo per high. Notiamo che B accetta solo frame nel range $(low, high)$. Nel momento in cui A riceve un ack più grande di low, $low = ack, high = high + ack - low$; questo design è chiamato cumulative acknowledgement. Ho scritto le operazioni in questo modo, in realtà viene anche considerato il modulo n. In effetti i buffer sono circolari e la somma applicata a low e high è pure circolare.

Quanto è opportuno che sia grande n? $2BD + 1$ dove B è la banda e D è la latenza. Notiamo che in questo modo in una situazione ideale otteniamo la medesima situazione di stop-and-wait, solo che A e B sono in qualche modo traslati l'uno rispetto all'altro. Se si applicasse solo stop-and-wait, verrebbe sprecato tempo a causa della latenza, oppure verrebbe sprecata banda, che si traduce in perdita di tempo (we don't like such things). Naturalmente non è detto che un dispositivo disponga della memoria necessaria per effettuare questo tipo di connessioni.

Go-Back-N Semplicemente, quando B riceve un frame corrotto, non manda nulla indietro ad A, ragione per cui A ricomincia a mandare i frame uno per volta a partire da quello successivo all'ultimo ack ricevuto. I frame successivi al frame corrotto non sono nemmeno memorizzati.

Selective repeat In questo caso, quando B riceve un frame che si accorge essere corrotto, manda un nak (negative acknowledgement) ad A. A rimanda solamente il frame corrotto e ricomincia a mandare i frame fino a che può. B dopo aver mandato un nak, riceve i frame fino allo high, come prima, e se riceve il frame in questione correttamente, manda un ack che ha come valore il numero del frame maggiore corretto che ha ricevuto, tale che non manchi nessun frame precedente ad esso.

Go-Back-N ha bisogno di meno memoria, ma spreca più banda. Selective repeat è più efficiente in termini di banda, ma ha bisogno di più memoria.

4.3 Multiaccess protocol

In questa sottosezione vediamo in quale modo il data link può gestire comunicazioni tra più dispositivi connessi attraverso la stessa rete. Prendiamo per dato che dividere la rete su n, il numero di dispositivi ed assegnare ciascuna porzione a ciascun dispositivo non è efficace: molto spesso la maggior parte dei dispositivi non comunica nulla, rendendo di fatto porzioni piuttosto grandi della rete inutilizzate. Questo implica che la divisione del canale non può avvenire staticamente, ma deve essere dinamica. Ecco le ipotesi di cui abbiamo bisogno per comprendere i protocolli di questo tipo:

1. independet traffic: ci sono N dispositivi indipendenti che possono generare frames;
2. single channel: è disponibile un solo canale, per tutti i dispositivi;
3. observable collitions: gli unici errori che possono avvenire sono dovuti a collisioni di pacchetti, due pacchetti diversi che si trovano nel canale di trasmissione nello stesso momento;
4. continuous or slotted time: il tempo è continuo o diviso in slot, a seconda del protocollo sarà specificato;
5. carrier sense or no carrier sense: i dispositivi hanno modo di capire se il canale sta venendo utilizzato.

Dato che c'è solo un canale, i dispositivi non possono segnalare di poter mandare un frame. In genere trasmissioni su canali wireless non permettono di capire se il canale viene utilizzato, le altre connessioni in genere hanno modo di capirlo (fibra e cavo in rame). Slotted time migliora le performace, ma i dispositivi devono sincronizzarsi in qualche modo.

ALOHA Inizialmente ALOHA è stato progettato per funzionare su connessioni wireless. Ciascun dispositivo nella rete trasmette appena può. Se il pacchetto arriva senza collisioni un computer centrale ritrasmette il pacchetto in modo tale che il mittente capisca che il pacchetto è arrivato con successo. D'altro canto se c'è stata una collisione il pacchetto non sarà ritrasmesso, per cui dopo un intervallo casuale di tempo il mittente riproverà a mandare il pacchetto. Ogni volta che due dispositivi provano ad occupare un canale nello stesso momento c'è una collisione. In questo caso il tempo è continuo.

Slotted ALOHA Le differenze rispetto al protocollo precedente c'è una differenze: il tempo è diviso in slot. Con questo cambiamento, se un frame è pronto per essere mandato, viene invece aspettato l'inizio dello slot successivo. Questo protoccolo ha un success rate di mandare con successo un frame del doppio rispetto al caso precedente (circa il 37%).

1-persistent CSMA Carrier Sense Multiple Access è un protocollo che funziona come slotted ALHOA, tranne che il tempo non è diviso in slot, e un dispositivo è in grado di vedere se il canale è occupato o meno. Il pacchetto è mandato se il canale è libero; altrimenti si aspetta che il canale si sia liberato e si prova a mandare il pacchetto. Se si verifica una collisione, viene aspettata una quantità di tempo casuale prima di ritentare.

nonpersistent CSMA Nel caso base, che il canale non è occupato il protocollo funziona come CSMA e slotted ALOHA. Se un dispositivo ha pronto un pacchetto, ma il canale è occupato, allora aspetta una quanità di tempo casuale e poi verifica se il canale si è liberato. Diverso da 1-persistent CSMA perchè il controllo non è effettuato continuamente. Questo protocollo comporta minori collisioni e un miglior utilizzo della banda, ma aumenta i tempi di delay.

p-persistent CSMA Quando un dispositivo è pronto a trasmettere un pacchetto fa un check del canale, se non stanno venendo trasmessi dati, allora con probabilità p comincia a trasmettere, ovvero con probabilità $q = 1 - p$ non trasmette.

CSMA/CD CSMA with collision detection è un'uteriore miglioramente del protocollo precedente: in questo caso le collisioni sono travate molto velocemente e quando avvengono i dispositivi smettono di mandare il pacchetto. Questo vuol dire che il canale viene utilizzato fondamentalmente in modo diverso: viene trasmesso un pacchetto; il trasferimento è concluso; ora i dispositivi che hanno un pacchetto pronto per essere inviato concorrono e provano a prendere possesso del canale. Se non ce la fanno riprovano dopo una quantità casuale di tempo, fino a che un dispositivo riesce a prendere il controllo del canale. Nel qual caso il canale è libero semplicemente viene saltato lo slot di contesa. Lo slot di contesa dura due volte il tempo massimo che un bit impiega a passare da un dispositivo ad un altro all'interno della rete; coincide con il tempo per cui siamo sicuri che tutti i dispositivi della rete capiscano che è successa una collisione.

4.4 Collision-free protocols

I protocolli precedenti contemplano il caso in cui avvenga una collisione, ora vedremo come evitare collisioni in toto. Il preambolo è che ciascun dispositivo nella rete è assegnato ad un valore, possiamo vederlo come un indirizzo.

Bit-map Dopo che ciascun dispositivo è stato assegnato ad un valore comincia il contention slot: a turno in ordine per il numero assegnato ciascun dispositivo manda un bit: 1 se può mandare un pacchetto, 0 altrimenti. Poi sempre a turno a partire dal numero più piccolo si comincia a mandare un frame. Quando tutti i dispositivi che avevano mandato un 1 hanno mandato con successo il proprio frame ricomincia il contention slot.

Token passing Viene formato un ciclo virtuale dei dispositivi connessi; viene assegnato il token ad un dispositivo. Il dispositivo con il token può mandare un frame a tutti. Che abbia mandato il frame oppure no, passa il token al dispositivo successivo del ciclo.

Adaptive tree walk Tornano ad essere utilizzate le collisioni. Ciascun dispositivo diventa la foglia di un albero binario bilanciato. Partendo dalla radice, si permette a tutte le foglie sotto il nodo in questione di trasmettere un bit, per segnalare che può mandare un pacchetto. Viene controllato se ci sono state collisioni, se non ci sono state, allora la foglia che ha mandato il bit ha il permesso di mandare il proprio frame. Se ci sono state collisioni si passa al figlio sinistro e si ripete il processo. Dopo che un pacchetto è stato mandato con successo si passa al nodo successivo. Notiamo che sono possibili vari miglioramenti: per esempio se nella root ci sono collisioni e nell'albero sinistro nessun dispositivo può mandare un segnale allora si controlla il nodo sinistro dell'albero destro, saltando un controllo inutile.

Ethernet L'ethernet classico sfrutta frame così coposti:

1. 8 byte di preamble: permettono ai dispositivi nella rete di sicronizzarsi con il manchester encoding (metà della banda è sprecata);
2. 6 byte indirizzo del destinatario, il primo bit se vale 0 punta ad un solo dispositivo, se vale 1 punta ad un gruppo di dispositivi;
3. 6 byte indirizzo del mittente;
4. 2 byte per tipo di file mandato, ovvero spiega al destinatario che cosa fare dei dati ricevuti; oppure per la dimensione dei dati;
5. da 0 a 1500 byte di dati;
6. da 0 a 46 byte di padding, per essere sicuri di avvertire una eventuale collisione;
7. 4 byte di CRC (checksum), per essere sicuri di aver ricevuto il pacchetto correttamente.

CSMA/CD with Binary Exponential Backoff Il protocollo di CSMA/CD con Binary Exponential Backoff definisce il modo in cui è determinato il tempo randomico di attesa prima di riprovare a mandare un pacchetto. In primo luogo si prova a mandare il pacchetto, se ci sono collisioni viene assegnato un valore randomico $r \in (0, 2^k]$ dove k è un contatore che parte da 0 e viene incrementato ogni volta che si verifica una collisione. k assume valore massimo 10. Dopo il quale rimane costante. Se dopo 16 tentativi di trasmissione non si riesce a mandare il pacchetto, si è segnalato un errore che è passato al livello superiore. Nel caso del protocollo Ethernet non viene mandato un bit ack, se la checksum non è corretta ancora una volta il problema è segnalato al livello superiore.

Switch e hub Inizialmente le reti ethernet erano connesse in degli hub, fondamentalmente il cavo (all'epoca in rame) era saldato con gli altri cavi che arrivavano all'hub. Questo sistema è fallace perchè non scala bene: sempre più dispositivi sono connessi e bisogno gestire sempre e solo più dispositivi. A questa tecnologia si contrappongono gli switch, che come prima hanno più porte che arrivano allo switch, ma ciascuna porta è collegata attraverso uno switch a tutte le altre, in modo tale che un pacchetto possa passare da un dispositivo A ad un dispositivo B attraverso un numero arbitrario di switch, senza che nessun altro dispositivo lo sappia. Implementare gli switch permette un'altro miglioramento: sono inclusi dei buffer nello switch in modo tale che se ci sono due pacchetti che devono essere mandati ad un dispositivo un pacchetto aspetta nello switch che l'altro abbia finito di essere mandato. Introdurre gli swith ha un ulteriore vantaggio: gli switch possono sempre ricevere dati, perchè hanno i buffer, per cui un dispositivo non ha bisogno di aspettare prima di mandare alcun dato.
Gli switch col passare del tempo sono diventati sempre più economici, oggi tra implementare un hub e uno switch, si preferisce sempre l'ultimo.

L'evoluzione dell'Ethernet avviene sempre pensando alla backward compatibility, per cui non si può cambiare il protocollo di trasmissione, il modo in cui sono formati i paccheti che sono mandati rimane il medesimo.

5 Network Layer

Il livello di rete si occupa di trovare il percorso attraverso il quale mandare un pacchetto da un dispositivo ad un altro. Il livello di rete è il livello più basso che si occupa di gestire connessioni end-to-end. Trovare il percorso migliore è un processo chiamato routing. Per trovare il percorso adatto il dispositivo ha bisogno di conoscere la topologia della rete e da ciò è in grado di calcolare attraverso un algoritmo di routing il percorso migliore. Il percorso migliore potrebbe non essere sempre il medesimo tra due dispositivi, perché la tipologia potrebbe cambiare (lo fa spessissimo).
La topologia che rappresenta la rete è rappresentabile come un grafo, i cui archi hanno un peso per indicare la distanza tra due dispositivi che possono mandarsi pacchetti. In effetti, la distanza può essere rapprsentata da diversi dati: il delay, la banda, ...
Di seguito sono elencate le proprietà minime per soddisfare un protocollo di rete:

1. il servizio di rete deve essere indipendente dalla tecnologia di trasporto, in particolare dai router;
2. il livello superiore non deve sapere la cardinalità, il tipo e come si compone la topologia di rete;
3. l'indirizzo che è dato al livello superiore deve essere univoco e uniforme per tutti i dispositivi;

Queste proprietà permettono di sviluppare vari protocolli di rete diversi, ciò da spazio alla flessibilità. D'altro canto, i protocolli di rete si dividono in due categorie: i protocolli connectionless, che definisce pochi servizi, principalmente, manda un pacchetto e ricevi un pacchetto; e i protocolli connection-oriented, per cui viene stabilita una connessione tra i due dispositivi e tutte le comunicazioni tra i due dispositivi avvengono attraverso gli stessi canali. Se un router nella connessione crolla, i protocolli connectionless semplicemente trovano un altro percorso per raggiungere la destinazione; mentre i protocolli conneciton-oriented interrompono la connessione, che dovrà quindi essere ristabilita.
Di seguito sono elencati i pro e i contro dei protocolli connection-oriented e connectionless:

Routing Il routing è diviso in due processi: ogni volta che arriva un pacchetto, questo deve essere mandato verso il prossimo router, fino a che il pacchetto non giunge a destinazione. Questo processo è chiamato

Issue	Connection-oriented	Connectionless
setup	yup	nope
indirizzo	un breve numero di identificazione	indirizzo completo di mittente e destinatario
routing	routing statico	routing dinamico
crash di un router	la connesione è persa	un pacchetto è perso
qualità	alta, le risorse sono preallocate	bassa, le risorse sono allocate dinamicamente
traffico	basso, le risorse sono preallocate	alto, le risorse sono allocate dinamicamente

Table 3: Comparison between connection-oriented and connectionless protocols

forwarding. Il router guarda la tabella di routing per trovare il prossimo router.
Il secondo processo è quello di aggiornare la tabella di routing, in modo da trovare il percorso migliore, utilizzando un algoritmo di routing.

Notiamo che per quanto riguarda i protocolli connection-oriented, il routing è statico, per cui la tabella di routing è aggiornata manualmente.

Dijkstra Algorithm L'abbiamo studiato in Algebra Lineare, per dubbi su questo algoritmo lo si cerchi altrove.

Flooding Il flooding ha un meccanismo piuttosto banale: viene assegnato un numero a ciascun pacchetto, questa dovrebbe essere la distanza destinazione-mittente. Quando il pacchetto arriva ad un router, questo lo manda a tutti i router connessi, tranne quello da cui è arrivato, solo se il counter è maggiore di 0. Notiamo che questo sistema non è molto efficiente dal punto di vista di utilizzo della banda, ma è molto semplice da implementare. Ci sono alcuni accorgimenti per evitare di rimandare il medesimo pacchetto indietro al mittente inutilmente oppure per evitare di mandare il pacchetto a router che l'hanno già ricevuto. Banalmente viene assegnato un numero al pacchetto che è memorizzato. Se un router riceve due volte il medesimo pacchetto lo cancella e basta. Il flooding si assicura che il pacchetto arrivi a destinazione alla massima velocità, per cui è utile per compararlo con altri algoritmi di routing.

Distance Vector Routing Il distance vector routing memorizza la distanza minima per raggiungere ogni dispositivo a cui è connesso, utilizzando l'algoritmo di Dijkstra.
Notiamo che questo algoritmo è semplice da implementare, nel momento in cui un nodo si libera o viene aggiunto un nuovo nodo, le tabelle di routing si aggiornano molto velocemente.
Il problema è che se un nodo crasha invece le tabelle di routing non vengono aggiornate sufficientemente velocemente e quindi il pacchetto potrebbe arrivare a destinazione in ritardo.

Link State Routing Il link state routing memorizza la topologia della rete, ovvero il grafo che rappresenta la rete; questo algoritmo è più completo dell'algoritmo precedente, nel senso che ogni router ha una visione maggiore. D'altro canto, ciscun router deve avere maggior memoria. Funziona come di seguito:

1. trova i vicini e ne memorizza gli indirizzi;
2. comprende la distanza dai vicini;
3. costruisce una tabella di routing;
4. invia la tabella di routing a tutti i vicini e riceve le tabelle di routing dei vicini;
5. calcola la tabella di routing finale, per tutti i dispositivi connessi;

Le tabelle di routing sono condivise con il flooding, infatti in questo caso è utile che ogni router conosca la topologia della rete. Il link state routing è un algoritmo di routing dinamico che risolve il problema della stabilità rispetto ai nodi che crollano. Tuttavia consuma più banda, richiede più memoria e richiede più tempo per aggiornare la tabella di routing. Fondamentalmente richiede maggiori risorse. La quantità di memoria che richiede questo algoritmo è direttamente proporzionale al numero di router connessi alla rete. Non scala bene. Per questo motivo si sviluppano vari livello di link state, i router sono raggruppati in cluster, ciascun cluster si comporta come se fosse un router.
Il numero di livelli ottimali su n router è $e \cdot \ln n$.

5.1 Gestione del traffico

Provisioning Se per diverso tempo si nota che la connessione tra due router spesso è congestionata, allora si investe in una connessione più veloce.

Traffic-aware routing Quando i router si accorgono di una congestione, perché per esempio la distanza tra i router si allunga, i router si organizzano per evitare di mandare il pacchetto attraverso il percorso congestionato.

Admission control Prima che sia stabilita una connessione, per quanto riguarda le trasmissioni connection-oriented viene verificato di quali risorse ha bisogno la suddetta connessione. Se ci si accorge che i router non sono in grado di sostenere la connessione, allora questa è rifiutata.

throttling Nel momento in cui una connessione peer-to-peer è congestionata si richiede al mittente di mandare i pacchetti più lentamente.

Shedding Nel momento in cui una connessione peer-to-peer è congestionata, può succedere che alcuni pacchetti siano scartati.

Load Shedding Le aziende che vendono il servizio della connessione, ogni tanto propongono ai propri clienti (che sono le aziende) di segnalare una priorità su ciascun pacchetto. In questo modo, se la banda è congestionata, sono scartati prima i pacchetti a bassa priorità. Per esempio, per condividere un video, potremmo avere un frame e poi salviamo le modifiche tra quello e il successivo. Il frame sarà marcato ad alta priorità, mentre, le differenze tra il frame attuale e quello successivo sono marcate ad una priorità più bassa. Nel momento in cui si verifica una congestione, all'utente finale arriva il video sembrerà perdere qualche frame, ma sarà comunque fluido.
I dati in genere sono divisi in due categorie:

- wine: le informazioni più vecchie sono più importanti di quelle più recenti;
- milk: le informazioni più recenti sono più importanti di quelle più vecchie.

Un esempio del primo tipo sono i frame: se scarto il frame più vecchio, il destinatario, potrebbe non salvare i frame successivi, richiedendo che gli venga rispedito il pacchetto più vecchio. Un esempio del secondo tipo è una live: se scarto il frame più vecchio, la live sarà più aggiornata.
Per invogliare le aziende a segnalare i pacchetti per ordine di importanza, il contratto prevede che, se non c'è alcuna congestione, i pacchetti possono essere mandati anche su bande più veloci di quelle contrattualmente stipulate.

Leaky bucket algorithm Prima che la rete possa garantire delle prestazioni, ha bisogno di conoscere quali sono le prestazioni richieste. Per cui viene richiesta una larghezza di banda minima, inoltre si richiede che sia assegnato un buffer di memoria, nell'eventualità in cui l'output sia maggiore di quanto dichiarato. I dati che escono dal buffer, sono segnalati con minore priorità oppure sono eliminati.

Token bucket algorithm Ciò che è fatto è analogo a quello che è fatto con il leaky bucket algorithm; l'host genera dei token in un modo concordato con il router. Se l'host finisce i token non può più mandare pacchetti, fino a che non saranno generati nuovi token. I token possono essere accumulati fino ad un certo limite.

Choke algorithm Quando si nota che si sta formando una congestione, l'host è notificato e rallenta la trasmissione. L'host è notificato attraverso un choke packet. Questa notifica è mandata dal router che ha notato la congestione a ritroso fino all'host. Dato che le congestioni tendono a durare poco tempo. Questa tecnica potrebbe richiedere all'host di rallentare la trasmissione solo dopo che la congestione è stata superata.

Choke hop-by-hop algorithm Il choke packet viene mandato dal router che ha notato la congestione e a ritroso fino all'host, ciascun router che riceve il pacchetto si impegna a ralletare lui stesso la connessione. Questa tecnica riesce a risolvere prontamente il problema della congestione. Tuttavia richiede che siano presenti diversi buffer su ciascun router.

5.2 Quality of Service

Per evitare danni maggiori, quando è stabilita una connessione sono richiesti i servizi necessari per tale connessione. I servizi disponibili sono i seguenti:

- bandwidth: le email hanno bisogno di meno banda di un video;
- delay: il delay è meno importante per un video on demand, rispetto ad una live;
- jitter: la variazione del delay, è importante sia per un video on demand che per una live, decisamente di meno per una mail;
- loss: una live può permettersi di perdere qualche frame nel passaggio, mentre una mail non può permettersi di perdere alcun pacchetto;

5.3 Internet Control Protocols

Ci sono diversi protocolli che permettono di gestire la rete. Tutti i seguenti sono sempre presenti.

ICMP Il protocollo ICMP è un protocollo di controllo, che permette di comunicare al mittente eventuali errori. Di seguito sono riportati alcuni possibili errori:

- Destination unreachable: il destinatario non è raggiungibile;
- Time exceeded: il pacchetto è stato scartato perché il tempo di vita è scaduto;
- Parameter problem: il pacchetto è stato scartato perché il parametro non è valido;
- Source quench: il pacchetto è stato scartato perché la banda è congestionata, poco utilizzato (non utilizzato affatto);
- Redirect: il pacchetto è stato scartato perché il destinatario è diventato inaccessibile;
- Echo request/reply: ping, controlla che il destinatario sia raggiungibile;
- Timestamp request/replay: come echo, ma aggiunge il timestamp;
- Router advertisement/solicitation: trova un router;

ARP Converte un indirizzo ethernet in un indirizzo IP.
Praticamnte un dispositivo sulla stessa rete che vuole conoscere l'indirizzo ethernet di una rete, manda un pacchetto ARP, in broadcast. Il computer con l'indirizzo IP risponde con il proprio indirizzo ethernet. Possono essere fatti dei miglioramenti: un computer che ha ricevuto un indirizzo ethernet, la un mapping e lo memorizza in una cache locale. Oppure ogni volta che un dispositivo è configurato o riconfigurato, in broadcast trasmette il nuovo mapping; se non ci sono risposte è andato a buon fine, altrimenti ci sono due dispositivi con lo stesso indirizzo IP.

DHCP Quando un dispositivo si connette ad una rete la prima volta non ha un indirizzo IP. Fornische un indirizzo IP ad un dispositivo che ne è sprovvisto. Viene mandato in broadcast un pacchetto DHCP, il router risponde con un pacchetto DHCP, che contiene l'indirizzo IP. L'indirizzo IP così fornito è dato in prestito e ha una scadenza; l'indirizzo IP deve essere rinnovato prima che scada.