Reti semplici (per davvero)

Rovesti Gabriel

Attenzione

Il file non ha alcuna pretesa di correttezza; di fatto, è una riscrittura attenta di appunti, slide, materiale sparso in rete, approfondimenti personali dettagliati al meglio delle mie capacità. Credo comunque che, per scopo didattico e di piacere di imparare (sì, io studio per quello e non solo per l'esame) questo file possa essere utile. Semplice si pone, per davvero ci prova.

Thank me sometimes, it won't kill you that much.

Gabriel

Sommario

Introduzione

Nelle reti si considerano molte cose; in primis i loro *protocolli*, quindi una serie di regole che devono essere seguite. In questo senso abbiamo come esempi i noti protocolli *ISO/OSI* e *TCP/IP*. Il primo presentava almeno 7 strati, partendo dal livello *fisico*, comunicando e prendendo i dati o bit dal media fisico di riferimento. Si passa quindi al livello di *collegamento*, che trasforma i dati in un segnale trasmissibile (header e relativi dati, nella tail) correggendo i singoli errori sui pacchetti/data frame, passando nel livello di *rete*, dove ogni dispositivo ha un proprio IP di riferimento e questi pacchetti vengono correttamente instradati. Il livello 4 è il livello di *trasporto*, che garantisce che i pacchetti arrivino nel giusto ordine, gestendo la connessione di rete e non congestionandola. Segue il livello di *sessione*, che si assicura venga mantenuta fluida la trasmissione e sincronizza le parti comunicanti, tramite l'uso di token. Il livello di *presentazione* invece crittografa e formatta il pacchetto, fornendolo all'applicazione di riferimento nel livello *applicazione*. TCP/IP poi ingloba i tre livelli finali di ISO/OSI in uno unico (application) e i livelli iniziali di ISO/OSI nel livello link; sin da subito TCP/IP fu il modello che venne nella pratica utilizzato.

Quindi anche la stessa Internet è definita da layers (strati come definiti sopra) e protocolli che ne permettono il funzionamento e ne articolano la complessità. Le reti in generale presentano varie classificazioni, per esempio in base al tipo di distanza (*MAN*, Metropolitan Area Network, installate in un'area estesa come una città, *LAN* o Local Area Netwrok, reti locali installate in un edificio, *WAN*, Wide Area Network come nazione o continente ed i singoli pezzi trasportano messaggi host ad host tramite linee ed elementi di commutazione/trasmissione, *PAN,* Personal Area Network, per interconnettere dispositivi in una piccola area circa raggio di una persona, tipo Bluetooth) o in base alle topologie (ad anello/ring, a stella/star, che determinano come sono i messi i dispositivi in una rete). Oltre a queste, parliamo dei singoli tipi di rete, quindi punto a punto (usata per reti grandi e connettono coppie di macchine) e reti broadcast (un pacchetto inviato a tutti i destinatari su una rete e processato solo se appartenente ad una certa macchina). In generale i layers permettono l'indirizzamento dei dati, il controllo degli errori e di flusso, la trasmissione a più dispositivi (multiplexing) e l'instradamento dei datagrammi o pacchetti (routing). Le connessioni possono essere inoltre connection-oriented, dipendenti da una connessione tra mittente e destinatario per effettuare successivi scambi di date oppure comunicazione estemporanea e non prestabilita, detta brutalmente "quando serve", quindi connection-less.

Nella trasmissione di rete è importante parlare anche dei relativi media wired (collegati), quindi in questo caso *l'UTP (Unshielded Twisted Pair)*, che limita l'interferenza reciproca grazie proprio al Twist. Negli altri punti non twisted potrebbe accadere un'interferenza, ridotta quindi dalle varie categorie di *doppino*, per esempio l'UTP-5, molto più twisted banalmente e con molta bandwidth (quanti dati possono essere trasmessi sul canale di riferimento). Possono essere soggetti ad interferenza elettromagnetica esterna e durano poco, devono essere mantenuti regolarmente; sono tuttavia facili da montare ed utilizzare. Altro cavo è quello *coassiale*, utilizzato principalmente nelle TV, cavo di rame rivestito da guaine e protezioni isolanti. Esso è decisamente meglio da un punto di vista di schermatura rispetto ai doppini precedenti e in generale performano bene a distanze corte, durando molto; hanno tuttavia lo svantaggio di perdere segnale dopo molto utilizzo e la loro ampiezza di banda varia a seconda della lunghezza, della qualità e del rapporto segnale/rumore.

Si passa poi al cavo in *fibra ottica*, a modalità singola o multipla. Esso è un filo di vetro in cui viene trasmessa una luce riflessa e protetta da una serie di guaine che permette l'avanzamento regolare del segnale fino all'altra estremità. È un media interessante e particolare, dato che è molto fragile e può perdere segnale (di solito di colore rosso quando capita). Per poter mantenere il segnale è quindi utile fare delle operazioni per esempio fondere i cavi (la migliore), usando dei connettori (perdendo dal 10 al 20% di luce) oppure con allineatori meccanici (perdendo circa il 10%). Il suo vantaggio principale è la tanta banda che trasporta e il fatto che è molto piccola e pesa meno; è anche difficile da intercettare (to tap). Se

confrontiamo altri tipi di luce, vediamo il vantaggio della fibra, rispetto per esempio a LED (economici, durano di più e piccoli, ma con poco data rate e a distanza corta) e laser (distanza lunga e alto data rate ma costosi e a vita corta).

Parlando delle onde elettromagnetiche (trasmissione wireless), possiamo utilizzare per scopi privati le bande ISM (Industrial/Scientifical/Medical), libera per utilizzi di questo tipo, ma ciascuna frequenza ha il suo scopo e il suo perché. Una piccolissima parte è visibile e diverse di queste sono dannose all'essere umano (tipo raggi gamma, raggi X, etc.) In generale, l'assegnazione di queste avviene tramite asta (vendita al miglior offerente di un certo intervallo di frequenze) o lotteria.

Trasmissione elettromagnetica

La trasmissione elettromagnetica prevede uno spostamento di elettroni tra i campi; il numero di oscillazioni al secondo di un'onda è chiamato *frequenza*, misurata in Hz e si parla di lunghezza d'onda quando si misura la distanza massima tra due picchi. In seguito vi è la suddivisione delle onde:

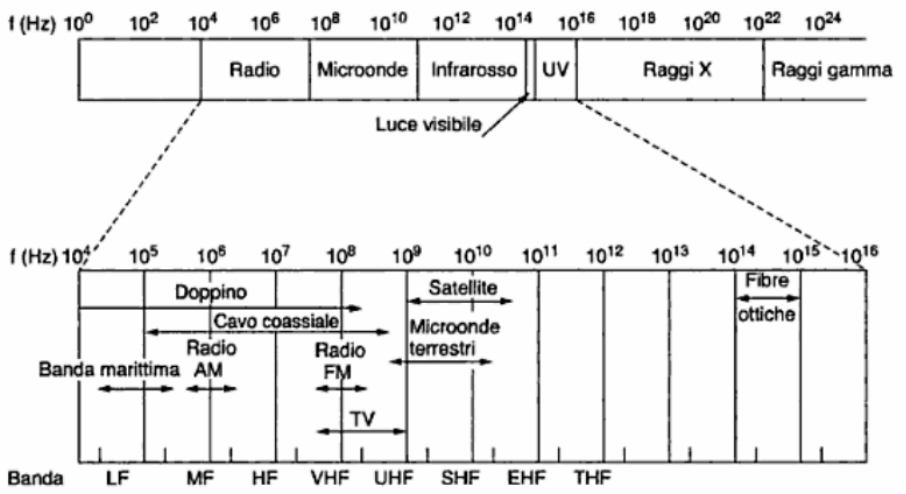


Figura 2: Spettro elettromagnetico.

Tra le varie onde distinguiamo le *onde radio*, descrivendo quelle a *bassa frequenza*, che tendono a disperdere energia più facilmente e sono portate a trasmettere energia meno frequentemente ma raggiungono distanze più lunghe e ad *alta frequenza*, che non passano bene gli ostacoli e vengono più facilmente assorbite, in parte rotte da ostacoli nel percorso. In generale, le frequenze rimbalzano seguendo la curvatura terrestre, ad esempio le frequenze FM, perché sono onde più solide. Ad esempio, sulla ionosfera questo tipo di frequenze rimbalza e viene ritrasmessa in vari punti, selettivamente. Tutt'oggi è utilizzata, principalmente per scopi bellici. Frequenze migliori: 10^0 fino a 10^4 (clock troppo lento per normale trasmissione dati, tuttavia usate da un punto di vista bellico/militare). Questo tipo di frequenze permette di raggiungere dei punti altrimenti meno raggiungibili e basta una sola stazione per poter coprire lunghe distanze.

La *trasmissione a micro-onde* tende ad essere direzionale (viaggiando secondo una certa direzione). Il loro vantaggio principale è di essere focalizzate, pertanto meglio utilizzate nella trasmissione dati. È stato per molto tempo lo standard di comunicazione, in particolare per le lunghe distanze (MCI, Microwave Communications Inc.), usato per distribuire il segnale televisivo, telefonico ma anche il segnale Internet. In ambienti urbani, non funzionano più molto bene, in quanto necessitano di un grande ambiente possibilmente vuoto per poter funzionare (ad esempio, il forno a micro-onde, tali da poter scaldare i cibi e

questi dispongono generalmente di un piatto). Questo è stato fatto poiché questo tipo di onde è direzionale e rimbalzano a seconda della superficie, distribuendo uniformemente il calore.

Dopo le microonde, parliamo di *onde ad infrarossi*, usate nei telecomandi. Sono direzionali, ma presentano gli stessi problemi pratici delle micro-onde (non passando gli oggetti solidi). Il loro vantaggio, anche qui, è quello di essere economiche ma anche essere sicure, perché non riescono a superare gli ostacoli (per esempio se comunico all'interno di una stanza, le onde di questo tipo di comunicazione non escono dal punto in cui sono applicate). Andando oltre, si parla di *trasmissioni luminose* (primo tipo di comunicazione prima della corrente elettrica), tramite anche banalmente dei falò per comunicare rapidamente un determinato tipo di notizia. Questo tipo di comunicazione è fatta con la luce visibile, torna in auge con il *laser*, focalizzato a buona distanza. Non sempre il bel tempo è indice di buona telecomunicazione, perché il sole scalda il terreno orizzontalmente, rilasciando delle gocce che formano una nebbiolina (micropioggia dal basso verso l'alto) che rappresenta un ostacolo per il laser.

Satelliti e tipi + space debris

Parliamo ora dei *satelliti*, in particolare si distinguono in tre tipi, che sono LEO/GEO/MEO (Low, 0-5000 Km d'altezza/Medium, 5000/15000 Km d'altezza/Geostationary Earth Orbit, a 35000 Km d'altezza). Non tutte le fasce possono essere utilizzate, ad esempio le fasce di Van Allen, fasce energetiche cariche che devono essere evitate possibilmente dai satelliti, in quanto a chiazze (alimentate dalle tempeste solari, perché fasce protettive). Essi proiettano un cono per poter comunicare (questo è il suo raggio d'azione). Per fare ciò ha bisogno di potenza che è circa il quadrato dell'altitudine (un satellite il doppio più alto necessita del quadruplo della potenza). Questi satelliti funzionano con delle batterie (costa meno rimpiazzarlo che cercare di ripararlo, chiaramente) molto costose ed apposite (a ioni al litio). Il costo del lancio cambia in base agli ordini di grandezza. Tipicamente ciascun satellite utilizza delle frequenze alte o molto alte, incrementando di molto la potenza al fine di superare gli ostacoli. Ciò tuttavia costa molto a livello di batteria. Essi trasmettono segnali tramite i trasponder, che sono dei ricetrasmettitori. Distinguiamo i satelliti in base all'orbita di riferimento (MEO, LEO, GEO).

Orbita LEO, meno costosi da mandare in orbita perché non attraversano le fasce di Van Allen e con bassa latenza. Di questo tipo citiamo due reti satellitari, per esempio *Iridium*, con 77 satelliti (poi diminuiti a 66) e copriva all'incirca l'intera superficie terrestre. Questi formano 6 collane che si spostano continuativamente, in cui ciascuno di questi trasmette con la propria energia e riflettono la luce solare, risultando quindi avvistabili ad orbite basse (sfruttando i fuochi di iridio, da cui il nome della rete). Era un sistema costoso intersatellitare, quindi comunicazione tra i singoli satelliti, che dichiarò quasi subito bancarotta per poi essere riattivato e presente ancora oggi in limitati campi applicativi (militari e simili). Il data rate era molto basso (4 KB di dati, misurata in baud, misura delle frequenze), bastevoli comunque al tipo di comunicazione Altro esempio primario nei satelliti è il sistema *Globalstar*, dove vengono usati dei satelliti usati come sorta di ripetitori, usati quindi per far rimbalzare più in là possibile il segnale (bent pipe satellites, usati come trasponders). Costava molto meno rispetto al servizio Iridium, ma ha fatto la stessa fine (bancarotta), poi ricomprata e tornata operativa.

Orbita MEO, situata tra le due vasche di Van Allen. Essa era tipica anche dei primi satelliti creati, ad esempio lo Sputnik, che aveva un suono caratteristico, cambiando a seconda dell'effetto Doppler (cambia l'intensità del suono a seconda della distanza). In generale, è tutto grazie alle frequenze che adotta. Servizio utile di satelliti ad orbita media è il GPS, in cui ogni satellite invia un segnale specifico che viene poi triangolato e si ricava la posizione esatta di ciò che interessa, a seconda di come viene tarato. Ci sono vari tipi di GPS usati in varie parti del mondo, ad esempio il GLONASS (usato in Russia) o l'A-GPS (usando Internet come strumento di aiuto per geolocalizzare). Esisteva già un sistema simile al GPS, già dagli anni Sessanta, sempre a scopi militari. La tecnologia si chiamava NAVSAT, che utilizzava l'effetto Doppler (spostamento delle onde) ricavandosi una posizione esatta per triangolazione dallo spazio fisico.

Scritto da Gabriel

Orbita GEO, satelliti geostazionari. Questo tipo di tecnologia sembrerebbe la migliore, in quanto i più alti e migliori, perlomeno da un punto di vista ricettivo. Essi stanno sull'Equatore, ruotando in orbita circolare o più in generale, stanno sempre sopra una stessa area geografica. Il numero di questi satelliti, per poter evitare interferenze, è al massimo 180. Questo per un motivo semplice: i coni di azione su questa zona rischiano di interferire l'uno con l'altro nel loro campo d'azione. Essi hanno una grande velocità di azione, ma è molto difficile inserirli esattamente nell'orbita giusta, per essere precisi nella trasmissione all'interno della loro orbita di riferimento. Vengono tipicamente utilizzati come satelliti spia, stando fermi (muovendosi ovviamente) rispetto all'asse di riferimento terrestre. Altri usi possibili: televisione via satellite con frequenze ad altissima potenza (ad esempio SKY).

Questi satelliti banalmente vengono lasciati in orbita (*space debris*) al posto di essere "buttati" o abbattuti; nel 2007, tuttavia, si sono testate delle armi anti-satellite laser (che non ha fatto altro che sparpagliare tutti i pezzi dappertutto). Si sono generati tantissimi pezzi di grandezza 1-10 cm; questi possono essere decisamente distruttivi per quanto piccoli. Sono milioni e milioni di detriti che a loro volta ne distruggono altri a cascata e di fatto tutt'oggi ne ruotano tantissimi tuttora in atmosfera che hanno addirittura formato delle loro orbite. Di fatto questi danni si propagano in varie orbite (ovviamente passando per le orbite basse, a rischio quindi di scontro in ogni momento con la maggior parte dei satelliti), dando luogo alla cosiddetta sindrome di Kessler (effetto a cascata). Per poterli quindi smaltire, si potrebbe farli deviare nell'orbita terrestre (bruciandosi), ovviamente pericoloso per i frammenti che potrebbe rilasciare. Altra soluzione: spingere il satellite nell'orbita cimitero (*graveyard orbita*), dove lo stesso sarà spinto senza fare troppi danni in quanto è un'orbita che, come suggerisce il nome, non viene molto utilizzata.

La situazione di satelliti e relativi rischi viene gestita da un comitato mondiale, che si occupa di tracciare i frammenti dello space debris, comunicando ai proprietari dei satelliti quanto potenzialmente sono a rischio collisione. Vengono ricevuti diversi avvisi (alcune decine/centinaia a settimana) indicando la possibile probabilità che un satellite possa essere colpito. La strategia da seguire sarebbe complicata, nel caso fosse parte di una rete di satelliti. Naturalmente, si può solo sperare che il satellite non venga colpito, sarebbe naturalmente molto costoso (esempio del satellite Kosmos, che si schianta in pieno con un satellite già presente). Storicamente, i satelliti sembravano il futuro della comunicazione, in quanto di fatto la rete terrestre non ha poi avuto dei grandi progressi fino al 1984, quando si è introdotta una legge per favorire la concorrenza e l'innovazione della comunicazione. Tutto ciò a discapito dei satelliti, che hanno subito l'influenza della fibra o del coassiale. Quindi la soluzione ibrida risulta vincente, quindi la soluzione cavo + wireless, che ha dal punto di vista della normale comunicazione vinto. Risulta tuttavia essere la soluzione migliore nel caso di comunicazioni impervie oppure in broadcast (abbiamo anche la comunicazione per lo studio di fenomeni atmosferici).

Basi delle comunicazione, frequenza e Fourier

A questo punto passiamo alle *basi della comunicazione*. Si parla chiaramente di *frequenza* e *periodo*, l'intensità di energia trasmessa in un certo tempo precisato oppure la *lunghezza d'onda* (ogni quanto si ha un picco di energia nello spazio). Parliamo quindi di *misure*. Per esempio, la *larghezza di banda* (misura dello strato fisico), cioè il numero di bit effettivamente trasmissibili all'interno di un determinato canale. La misura determina tutte le frequenze utilizzabili all'interno di un canale, pertanto si misura in "battiti al secondo" (hertz). Altra misura vicina a noi, sarà il *data rate*, quindi quanta informazione passa in un secondo. Più formalmente, possono esservi due misure intermedie, parlando quindi di *bit rate* (quanti bit al secondo) e baud rate (quanti simboli al secondo vengono trasmessi, dove per *simbolo* si intende uno strato di energia o forma dello spazio usato come mezzo di trasmissione. L'informazione in bit in un alfabeto a V simboli usa come funzione ($\log_2(V)$). Citiamo la relazione tra bit rate e baud rate, quindi: $bit\ rate = baud\ rate * (log2(V))$

Intuitivamente, più frequenze abbiamo, maggiore sarà il data rate; naturalmente serve più potenza per poter ottenere questo. La potenza per creare informazioni e trasmissioni cresce col quadrato della frequenza (quindi se voglio raddoppiare quante frequenze ho, mi servirà il quadruplo dell'energia, per dare un esempio). Viene quindi fissato un limite di banda, quindi quanta potenza andrò ad impiegare.

Citiamo quindi la *trasformata di Fourier*, che studia l'oscillazione di tutte le possibili onde che compongono la realtà. L'obiettivo è quindi quello di ricreare le frequenze fatte in una forma univoca e rappresentabile. Quindi Fourier determina l'ordine di utilizzo delle frequenze, per ordine di importanza (andando verso il basso, diminuisce la qualità e la possibile utilità dell'onda). Essendo una somma infinita di seni e coseni, riesce facilmente a localizzare il set di cicli di velocità, ampiezze e fasi per ricombinare le onde tra di loro in ogni periodo di tempo considerato (quindi sia per uso di sampling/campionamento sia per costruire delle forme d'onda). Le onde gradualmente, grazie a Fourier, sono potenzialmente approssimate in maniera tale che con queste si possa studiare l'ordine usabile delle frequenze possibili che, nel corso del tempo, ha cercato di accogliere sempre più frequenze. Esempio pratico: Shazam, l'app di Apple per trovare il titolo di una canzone. Le frequenze si sporcano per possibili interferenze esterne; viene quindi usato la fourierizzazione per poter capire il brano dalle frequenze più importanti della registrazione e partendo da un sottoinsieme di queste calcolato confrontando il database di tutte le frequenze.

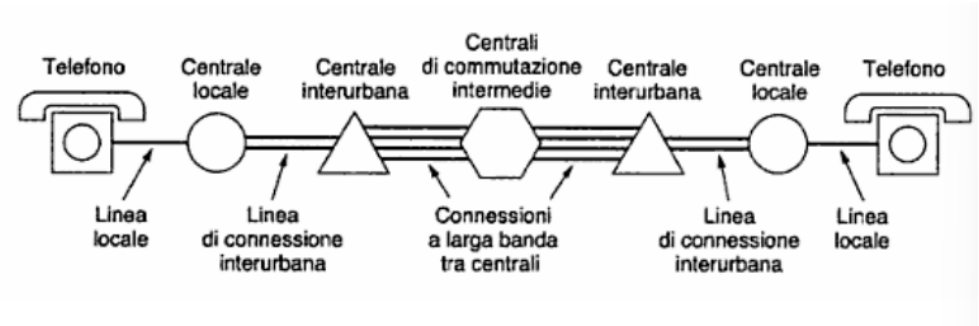
Nyquist, Shannon, Sistema telefonico e ampiezze di segnale

Ogni impulso energetico presenta *un'attenuazione* misurata in decibel, a seconda dei mezzi fisici che deve attraversare e capire quanta e quale energia viene effettivamente persa. $10\log_{10}$ è il simbolo matematico che rappresenta il rapporto tra la potenza trasmessa e la potenza ricevuta. Il logaritmo viene utilizzato per rappresentare la potenza richiesta per poter percepire ad orecchio gli impulsi, comprimendo l'impulso sonoro da rumori troppo forti (da cui l'uso dei decibel). Nelle bande di frequenza ci sono delle zone di attenuazione, in cui i singoli pezzi subiscono più o meno la stessa attenuazione (mantenendo la forma dell'onda, magari cambiando semplicemente la dimensione). Il *teorema di Nyquist* stabilisce che livelli di banda e potenza servono per poter campionare un segnale a seconda della banda disponibile (all'interno però di un canale senza rumore). Il data rate massimo è $2B \log_2 L$ (intendendo con B la banda massima e con L i livelli del segnale utilizzati). Se per esempio si volessero aumentare le frequenze disponibili, non è assolutamente facile, in quanto servirebbe il quadruplo della potenza ad andamento logaritmico per aumentarne la quantità. Shannon poi definì il *rapporto segnale rumore* (S/N, S per Signal ed N per Noise), per poter rappresentare la stessa idea all'interno di canali reali. La formula del *teorema di Shannon* è quindi la seguente: $(B \log_2 (1+S/N)$.

Altro problema è la *dispersione* che cambia la forma dell'onda e l'onda attuale viene completamente sostituita da un'altra onda; assieme all'attenuazione il problema è grave. Per risolvere questo problema possiamo usare dei ripetitori che aggiustano e trasmettono il segnale corretto. La chiave sono i *solitoni*, studiati da Scott, il quale comincia a disegnare e studiare un'onda che non si fermava, letteralmente. Queste onde hanno un solo tono (da cui il nome) e sono localizzate (agiscono solo nel punto dove esse vengono effettivamente applicate), hanno forma permanente, sono bidirezionali, e passano anche sopra ad altri solitoni e simili forme d'onda. Esse arrivano anche oltre 10000 Km senza dispersione. I solitoni spingono verso il basso con una spinta d'acqua e i raggi solari vengono deviati a causa del cono formato dalla collisione multidimensionale dei solitoni, facendo vedere quell'ombra da loro generata. Di fatto però sono costosi da creare, perché necessitano di condizioni particolari e parecchia energia.

Il grande *sistema telefonico* fu un esempio primario di trasmissione energetica intercontinentale, considerando i numerosi sforzi che si fecero comunicando a partire dal Canale della Manica fino a raggiungere gli USA (vicenda Cyrus Field).

Questa linea era realizzata *punto a punto*, tra un dispositivo ed un altro. Il centralino, infrastruttura di secondo livello quindi passaggio successivo al precedente, era quello che permetteva di selezionare e cambiare la linea di utilizzo. In generale, lo schema di rete ha una serie di loop intermedi, passando per vari trunk (cavi di collegamento) tra i singoli pezzi della rete, mandando l'informazione tramite dei cavi UTP categoria 3 nel caso del local loop. Gli altri livelli, per maggiori necessità di banda e performance, sono composti da cavi di tipo coassiale o fibra. Sotto la rappresentazione:



Come accennato sopra, nel 1984 l'AT&T viene smembrata in una serie di sottocompagnie indipendenti per poter rinnovare il mercato telefonico e/o di conunicazione, sancendo il diritto che ogni compagnia telefonica può dare il proprio pezzo di rete per ogni parte di utente/comunicazioni, arrivando vicino agli utenti finali; caso di esempio gli operatori virtuali, sfruttando reti pre-esistenti e usando parte della rete altrui. In quest'epoca si sancisce anche il diritto di portabilità dei numeri, per liberalizzare ulteriormente il mercato. Come viene dunque trasmesso il segnale? Sono simboli analogici, poi successivamente convertiti al digitale. Per poter passare da analogico e digitale e viceversa, si introduce l'utilizzo del *modem*, quindi un modulatore per poter fare effettivamente questa cosa. Le onde digitali hanno seri problemi di attenuazione e distorsione; in generale quella che funziona meglio è la corrente alternata per teletrasmettere.

Per poter trasmettere il segnale digitale, posso modulare in ampiezza (*ASK*, Amplitude Shift Keying) usando due ampiezze per rappresentare 0 e 1, modulare in frequenza (*FSK*, Frequency Shift Keying, aumentando o diminuendo la frequenza a seconda di quanti sono i simboli trasmessi) o modulazione di fase (*PSK*, Phase Shift Keying), tecnica più utilizzata, quindi la traslazione dell'onda di 0 o 180 gradi e funziona bene con pochi sfasamenti; a livello pratico significa che "ribalto" l'onda in un senso o in un altro). Si hanno varie tecniche derivate dalla modulazione di fase, ad esempio *QAM*, che combina la modulazione in ampiezza con PSK oppure la *QPSK* (Quaternary Phase Shift Keying), facendo 4 traslazioni e trasmettendo 2 bit di informazione al secondo, aumentando esponenzialmente la potenza per i dati trasmessi. Citiamo anche le varianti di QAM, come QAM-16, che usa 16 combinazioni di ampiezza e fase trasmettendo 4 bit per simbolo o QAM-64, con 64 combinazioni e 6 bit per simbolo.

Tipi di QAM

Esistono inoltre le *QAM circolari*, sfruttando più spazio possibile attraverso tutti i possibili simboli. La disposizione non è quella ottimale, per quanto meglio di QPSK. Nella linea telefonica il limite fisico, considerando il teorema di Shannon, è di circa 35000 bps (35K in trasmissione). Usiamo quindi una serie di standard di trasmissione all'interno dei modem dette le precedenti modulazioni; ad oggi, per esempio, citiamo V.32/V.32 bis e V.34/V.34 bis, che utilizzano rispettivamente 4, 6, 12 e 14 bit a parità di 2400 baud per ciascuno di questi. Col tempo la complessità aumenta, arrivando negli anni 90 ad avere molti simboli in più di trasmissione. E arrivando circa al limite fisico a metà anni 90 (ricordiamo circa 35000 bps). La barriera fisica è poi superata grazie a modem a 56 Kbps. Avere un suono pulito significa arrivare all'incirca a 4000 Hz, con una linea duplex con un limite fisico intorno ai 35000 bps. Se invece uno dei due riceventi si connettesse ad un servizio digitale, il limite fisico raddoppia, arrivando a circa 70000 bps. Sarebbe possibile arrivare a circa 64K di trasmissione, ma per il fatto che gli USA usano 7 al posto di 8 bit per i dati, si usa

questa trasmissione con data rate ridotto ancora adesso. Tornando alle tecnologie di trasmissione si parla del *fax*, che trasmette solo informazioni testuali e che sfruttava le linee telegrafiche ed usato principalmente per trasmissione di immagine. Esistono vari gruppi di trasmissione dei fax e anche supergruppi, aumentando sempre più la risoluzione, dalla standard alla ultrafine. Si sono poi unificate le linee dati tra fax e telefono, condividendo la stessa linea tra telefoni e fax e sfruttando gli standard esistenti. Il fax si è fermato a circa 35000 bps con una connessione punto-punto.

DSL e Tipi

Le *DSL (Digital Subscriber Lines)* nascono per seguire la spinta di Internet e sono servizi di trasmissione a banda larga, quindi con più esigenza di data rate/trasmissiva. Vi era un importante limite fisico da superare nella creazione di queste *DSL*, poiché soppiantare il cavo *UTP3* avrebbe avuto dei costi decisamente grandi se non enormi. L'idea è stata quella di utilizzare la già esistente banda telefonica, dove la trasmissione parallela era ottenuta tramite filtraggio; di fatto si passa da 4000 Hz ad 1.1 MHz, semplicemente togliendo un filtro. Una volta tolto il filtro alla centralina, verrà rimesso per l'utente finale tramite lo *splitter*, che fa deviare le giuste frequenze telefoniche oltre la ADSL, separando le due linee dati e telefonica. Ogni splitter genera delle interferenze, potenzialmente rovinando la qualità del segnale, pertanto la cosa indicata è di averne uno solo. Con questo viene anche messo un *NID (Network Interface Device)*, che è un dispositivo a forma di armadietto che fa da tramite tra local loop del gestore e le reti domestiche dei singoli utenti. Ora grazie al wireless la soluzione ottimale è di collegare un solo telefono al cordless e collegare tutti gli altri al wireless. Di fatto il maggiore impedimento è dato dal decadimento dovuto al cavo UTP-3, con un grossissimo decadimento e la grossa perdita di dati si ha entro il Km. Si intende con distanza la banda garantita in uscita dalla centralina e la nostra effettiva distanza da essa determina la qualità della banda/segnale.

La DSL più comune e che utilizza proprio FDM è la *Asymmetric DSL o ADSL*, dando più banda in download che in upload, idea non più ottimale, per il semplice fatto che la rete si è adattata ad un diverso utilizzo da parte degli utenti, non solo scaricatori ma creatori attivi di contenuti. Si utilizza il *multitono* (divisione in 256 canali indipendenti dello spettro) per poter comunicare ed inviare dati negli standard DSL, ma non nella realtà comune, dato che causerebbe eccessive interferenze. La soluzione quindi è di spezzare questi canali in tanti sottocanali riservati per la voce, canali vuoti (che sono 5) e il resto in download/upload. Per fortuna, grazie a questa frammentazione, è possibile resistere mediamente bene alle interferenze mantenendo buone performance. Vi sono vari standard di ADSL, per esempio quello Lite (1.5 Mbit down, 0.5 Up), ADSL normale (8 Mbit down, 1 Mb up), ADSL2 (12 Mbit down, 1 Mb up), poi migliorando aumentando la banda in upload o anche aumentando le frequenze usandone il doppio (ADSL2+). Esistono delle varianti all digital, non sfruttando la banda voce ma solo la banda dati, guadagnando una certa percentuale in upload. Nel corso del tempo si cerca di fare l'upgrade di questa infrastruttura usando sempre meno cavi in rame e passando alla fibra, unendo entrambe le tecnologie (passando da ADSL a *VDSL*, quindi *Very High Speed DSL*). Piuttosto che cambiare l'infrastruttura esistente, si è deciso di cambiare banda, quindi semplicemente utilizzando altri tipi di annessi e sottostrutture della stessa tecnologia, variando a seconda della necessità la velocità di download/upload. Di queste esistono 3 standard principali, VDSL/VDSL2/VDSL2 V+, che hanno delle buone velocità massime in download rispetto ad ADSL, arrivando fino a 300 Mbps. Esse necessitano di un apposito modem compatibile a queste

FDM-TDM, Switching e Tipi

Per garantire corretta separazione tra canali dati e canali voce o evitare interferenze si usa la tecnica del multiplexing, in particolare *FDM, Frequency Division Multiplexing*, che divide lo spettro di frequenza in bande e ciascun utente ne possiede una. I canali voce vengono quindi divisi in gruppi che comprendono più utenti, ad esempio un gruppo, composto da 12 canali voce, uniti in multiplexing a un supergruppo e a loro

volta in un mastergroup. Altro tipo di multiplexing utilizzato è il *TDM, Time Division Multiplexing*, dove i canali voce frammentano i canali per frequenze e vengono scansionati temporalmente in round robin, facendo in modo ve ne sia sempre uno attivo in un dato momento in maniera bilanciata e poi riunendo ciascun canale in uno unico. Questa tecnologia viene usata all'interno dei CD, per esempio. Per controbilanciare l'effetto di rallentamento che si crea nella rilevazione delle frequenze, si raddoppia la velocità, anche per la musica registrata a più canali. Altri tipi di modulazione che vengono utilizzati alternativi a questo descritto sono la *differential pulse code modulation*, che riduce il numero di bit per impulso digitalizzando i dati trasmessi (passando per esempio da 7 a 5 bit utili per la trasmissione) e la *modulazione delta*, che digitalizza anch'essa la trasmissione e qui ogni valore campionato differisce dal precedente di +1 o -1; in base alla differenza, capendo chi è il minore e chi il maggiore, decide come modulare ed impostare correttamente la trasmissione.

Dal punto di vista della trasmissione, si fa una breve introduzione alla trasmissione TV, sia per il digitale terrestre che per la rete satellitare televisiva, che utilizza la trasmissione robusta in QPSK, con più potenza e meno interferenza (tutto dipende dalla zona, come si discuteva sopra). Incontrando il terreno, una parte delle frequenze viene assorbita e una parte rilasciata (sotto forma di ech I problemi distorsivi delle onde dipendono dalla frequenza del segnale. Il digitale terrestre viene suddiviso in tante trasmissioni indipendenti e quindi il segnale televisivo non arriva univocamente, ma con tante frequenze/canali diversi (esempi, 2K, con circa 2000 canali, 8K, circa 7000 canali). Nelle vecchie TV analogiche, con un'interferenza era tutto visibile sullo schermo (tv noise), in quanto era un'unica forma d'onda trasmessa su un canale e la sua azione all'interno dei cavi di trasmissione era di fatto visibile. La cosa diversa capita invece nel digitale, dove si vedono delle piccole strisce qualora ci siano interferenze, per via dei vari canali di trasmissione. Per poter risparmiare spazio nella trasmissione TV si usa una tecnica di compressione *MPEG2*, in cui ogni canale occupa molte meno frequenze e posso sfruttare questo risparmio di frequenze per creare ulteriori canali.

Ritorniamo alla struttura gerarchica della rete telefonica con il PSTN, sfruttata e rigirata per le varie bande con vari *switching*, compito una volta svolto dalle centraliniste, ora presenti tre tecniche automatizzate. Per esempio il *circuit switching* è il fatto di creare un collegamento fisico tra due canali di comunicazione, con un grosso delay fisico iniziale nella connessione, ma una volta stabilita si ha un collegamento fisso dedicato tra chiamante e chiamato. Vi è poi il *message switching*, che non prevede un collegamento fisico a priori ma si basa su un'idea step-by-step. Il messaggio viene inviato alla prima centrale di comunicazione e poi lo ritrasmette alla successiva, sequenzialmente fino al destinatario (tecnica dello *store and forward*, memorizza ed invia). Con il *packet switching*, si decide una dimensione fissa per i singoli messaggi/sottomessaggi, non solo occupando risorse ma anche perdendo delle parti di messaggio trasmesse in parallelo che devono essere mandate. Essendo una rete con una gestione più dinamica anche al traffico di rete in tempo reale, i pacchetti seguono rotte (routes) diverse, arrivando anche in ordine diverso da quello effettivo, riducendo infine i possibili bottleneck (colli di bottiglia).

Trasmissioni mobili: PTT, MITS, 0G, 1G

La standardizzazione delle comunicazioni è stata introdotta a livello europeo per cercare di imporre un solo tipo di comunicazione, cosa che è avvenuta solo in Europa (es. 2G con GSM, ma non negli USA). Nasce quindi il principio della conoscenza tariffaria, quindi conoscere le condizioni di un determinato servizio prima di pagarlo. Nel caso degli USA non c'è distinzione tra numerazione fissa e mobile (cosa che da noi si ha ad esempio) e i possessori di rete fissa pagano un surplus per l'utilizzo di queste. Ecco quindi che si comincia a parlare delle gerarchie/standard telefonici; l'infrastruttura stessa si basa sul problema della divisione del territorio, che deve essere coperto tramite la rete fissa. Il gestore della comunicazione mobile deriva dalla fissa, in particolare derivando il tutto da una determinata cella telefonica, che copre una certa porzione di territorio.

La *prima generazione di trasmissione mobile* è stata la trasmissione *PTT (Push To Talk)*, derivante dalla trasmissione radio e poi evolutasi nell'equivalente delle radiolinee walkie-talkie. Esso era half duplex, usando il bottone per parlare, quindi sfruttandolo in maniera unica parlando da una parte e tacendo dall'altra. Tra le altre cose questa comunicazione non era privata e il numero limitato di canali limitava il possibile numero di utenti collegati. Si ha una miglioria di questa comunicazione negli anni 60 con *IMTS (Improved Mobile Telephone System/0G),* passando a due frequenze e con trasmettitori ad alta potenza. Grazie alla frammentazione in celle e a loro volta in celle più piccole (microcelle), si aumenta il riutilizzo delle possibili frequenze. In mezzo a queste celle si ha un gestore della rete (conosciuto come MTSO, Mobile Telephone Switching Office), facendo colloquiare i singoli dispositivi nelle singole parti.

La successiva tecnologia è stata la *1G-AMPS* (in Europa nota come *TACS, Total Access Communication System*) con celle che coprivano circa 10/20 Km, aumentando la copertura e diminuendo la potenza richiesta per la trasmissione. Il problema è che celle più piccole fanno aumentare le interferenze e inoltre vi erano pochi canali, facendo anche aspettare molto gli utenti finali. La soluzione è quindi di separare la frequenza, perché così cerco di evitare i problemi tra le celle di interferenza; il problema è che parte della banda viene tagliata a matrice esagonale, per dare più potenza di rete in una determinata zona. Occorrerebbe quindi trovare meno frequenze tali da superare tutte le celle, quindi una soluzione ottimale. Nella maggioranza dei casi quindi non abbiamo controllo totale sul terreno che abbiamo, quindi vogliamo fare in modo separare ogni pezzo di terra evitando che ciascuna utilizzi la stesse frequenze (quindi terre che usano una frequenza hanno un colore e con la stessa frequenza non devono essere vicine. L'idea iniziale era la suddivisione di un territorio a colori, usandone 4, prendendo per esempio le aree geografiche del Regno Unito o degli USA. Il problema resta aperto per almeno per diversi anni e ne vengono elaborate diverse varianti, con più e meno colori, ma poi si riconferma l'idea iniziale dell'uso di usare almeno quattro colori.

Con un esempio in un foglio quadrettato, simuliamo il segnale che posso avere per una singola persona all'interno delle celle tracciando con delle linee il possibile cammino percorso da un certo dispositivo rispetto alle zone di confine delle celle, simulazione abbastanza realistica per dire che anche cambiando un punto oppure un altro, cambio completamente cella. Nella creazione di una rete mobile, cerco banalmente di spezzare le linee dritte creando una linea sfasata, per esempio con una struttura esagonale. Il problema è che la zona può potenzialmente andare in overload, risolvibile tramite la costruzione a microcelle e frammentando ulteriormente la gestione delle zone. Ci possono essere dei problemi nelle microcelle, per assurdo se uno fosse in pieno centro magari la rete non va perché ci sono troppe persone. L'idea dei colori era di usare 4 colori per frammentare la mappa, partendo da zero. Certo è che non è possibile dover ricostruire tutta la rete per una singola microcella. Quindi in questo caso è bene usare tra 3 e 7 colori, il cui numero comunque può variare potenzialmente. Ci potrebbe essere una perdita di parti della banda grazie alla frammentazione a colori e microcelle, perdendo da 2/3 a 6/7 di banda nella trasmissione.

Handoff, 2G, D-AMPS, GSM

Tornando alle varie tecniche di trasmissione, abbiamo vari tipi di *handoff*, la vecchia stazione molla il cellulare per poi riagganciarlo successivamente. Di questi distinguiamo il *soft handoff*, in cui l'acquisizione della nuova stazione avviene prima di interrompere il segnale precedente ed *hard handoff*, dove la vecchia stazione rilascia il telefono prima che la nuova lo acquisisca. Si utilizzano circa 45 canali effettivi per ogni singola cella e un cellulare possiede un numero seriale di 32 bit, avendo quindi un numero telefonico di 10 cifre; ogni 15 minuti il dispositivo si tiene attivo con un keep-alive, registrandosi alla cella più vicina (mandandole il proprio numero seriale e numero telefonico, appropriandosi esclusivamente della cella). Mentre si chiama, si usa un canale apposito per la richiesta del cellulare nella cella. In ricezione si ha un apposito canale di paging, dove i cellulari controllano la situazione, "sentendo" se la comunicazione li coinvolge. Nel qual caso, gli viene assegnato un canale esclusivo per la comunicazione.

Parliamo poi della tecnologia *2G*, spartiacque tra comunicazione analogica e digitale. Ci sono 4 standard diffusi, tra cui il principale è *D-AMPS* (negli USA), che funziona usando tutte le bande di AMPS solo in digitale, compatibile anche con PDC, tipo di standard giapponese. Si arriva ad una banda di frequenza di 1850/1910 MHZ, usando onde più corte (solide), subendo sempre più le interferenze degli ostacoli fisici. In questo caso basterà anche un'antenna più piccola per poter prendere questo tipo di onda. Cerchiamo quindi di sfruttare più canali possibile usando tecniche di compressione del flusso voce. Gli algoritmi utilizzati devono essere efficienti energeticamente, per il semplice fatto che su un dispositivo come un telefono, lo zip è un grosso limite fisico/pratico. La voce viene qui digitalizzata con l'ausilio del vocoder, comprimendo la qualità ma permettendo più utenti con le tecniche di multiplexing citate. Si comprime fino a 4/8 Kbps riuscendo ad avere 6 volte il numero di utenti di AMPS.

Si usa sempre TDM come tipo di multiplexing. Essendo che le telefonate sono compresse digitalmente ha peggiorato sensibilmente la qualità audio. In questo caso cambia tutta la gestione dell'handoff, in cui il control switch (centralino) diventa collo di bottiglia. Il telefono in una cella monitora la qualità del rapporto, misurando periodicamente la potenza del segnale. La disconnessione quindi avviene in automatico, per poi cercare di riacquisire successivamente il segnale prendendolo più potente. Si usa una tecnica assistita chiamata *MAHO (Mobile Assisted HandOff)*, usando i tempi morti durante il multiplex per il ricalcolo ed acquisizione del segnale, con un minimo dispendio di energia. L'introduzione della più famosa tecnologia 2G è il *GSM (Global System for Mobile Communications)*, simile ad AMPS e introdotto in Europa a partire dalla Finlandia. Qui i canali supportati sono circa 1000, la maggior parte dei quali usati come canali di controllo e sono molto più ampli rispetto ad AMPS, avendo 200 KHz rispetto a 30 che si hanno negli USA. Ogni singolo canale detiene un data rate più alto rispetto allo standard americano. Il multiplexing temporale si fa sentire letteralmente per esempio nelle interferenze tra microfono e casse o telefono, sentendo un rumore a tacche.

Vi sono vari tipi di celle, in cui ogni singola cella copre aree da più grandi a più piccole e i singoli gestori hanno grande flessibilità nella gestione del carico e distanza (*macro*, 35 Km, *micro*, celle più piccole, coprono fino ad un edificio, *pico*, aree simili a quelli indoor e *umbrella*, per coprire i buchi). Si usano le *SIM* (Subscriber Identity Module), che hanno pochissima dimensione, principalmente per memorizzare due numeri, che sono l'identificativo della SIM (IMSI) e la chiave di autenticazione Ki per l'autenticazione crittografica a chiave condivisa. Di fatto il telefono trasmette IMSI e Ki in broadcast, da cui l'operatore è in grado di generare un numero casuale, rimandato al mittente e firmato con Ki; l'operatore userà questo passaggio come conferma di avvenuta autenticazione.

CDMA e funzionamento

Abbiamo poi il sistema *CDMA* (Code Division Multiple Access), che permette ad ogni stazione di dividere l'intervallo in centinaia di canali a banda stretta, permettendo ad ogni stazione di trasmettere per tutto il tempo attraverso tutto lo spettro di frequenza. Il punto è usare una lingua di trasmissione diversa, cercando di trasmettere usando gli spazi dimensionali per poter trasmettere più informazioni usando gli assi pre-esistenti. Le onde si sommano tra loro nei singoli vettori dello spazio multidimensionale. Sarà bastevole semplicemente l'alfabeto binario, con i simboli "0" e "1", distinguendo con i segni il positivo e negativo come delle classiche coordinate e selezionando grazie a queste la forma d'onda giusta.

Con l'utilizzo della trasformata di Fourier è la creazione di un'onda rappresentativa di una certa informazione, minimizzando però il numero di simboli utilizzati per avere una rappresentazione efficiente. Nel CDMA ciascun dispositivo viene univocamente identificato all'interno della rete tramite un codice, che permette di poter distinguere e capire se un determinato dispositivo ha già trasmesso/comunicato all'interno di una rete. Richiede un controllo sulla potenza di trasmissione, dovuto dalla distanza rispetto alla cella (se più vicino alla stessa, posso potenzialmente rubare segnale a dispositivi più lontani) e quindi organizzando l'infrastruttura in maniera flessibile.

Scritto da Gabriel

Per distinguere le forme d'onda l'idea è quella di associare il picco alto a "1", mentre il picco basso a "0"; invece da un punto di vista energetico useremo 1 per il picco alto e -1 per picco basso. Per creare assi perpendicolari fatti da 1 e -1 vengono usate le *matrici di Walsh*, che riuniscono tabelle di coordinate unite, semplicemente ruotate e in queste vengono applicate apposite regole di composizione e proiezione.

Queste a livello pratico filtrano i rumori sul canale. Grazie alla proprietà di ortogonalità offerta dalle matrici, usando l'alfabeto binario siamo certi se un dispositivo abbia comunicato o meno, quindi tutti i prodotti tra le componenti dei dispositivi che non hanno comunicato devono annullarsi ad eccezione dell'ultimo che ha comunicato (banalmente quindi, seguendo gli esempi del prof, 1 se ha comunicato con successo, 0 altrimenti; questo sarà il risultato finale). CDMA sfrutta l'interferenza al meglio, non sprecando tempo in caso di silenzi, risparmiando celle e si dimostra adatta alla trasmissione locale.

2.5G, UMTS, EDGE, 3G, 4G

Questo tipo di trasmissione si presta meglio alla comunicazione vocale che è attiva circa il 35/40%; una tecnologia *2.5G* è la *GPRS*, che essenzialmente è un overlay del 2G ed introduce la gestione del traffico a pacchetti. Il suo problema principale è l'occupazione di un intero canale, sprecando molte risorse. Il GPRS utilizza il multiplexing risolvendo il problema e scompattando i dati in pacchetti. A questo punto non si spreca banda, tariffando quindi un utente in base al suo traffico. Esistono vari tipi di cellulari GPRS, ad esempio la classe C che è simile alla B, e qui decidono se settarsi come GPRS o GSM o la Pseudo Classe A, la quale può essere contemporaneamente GSM e GPRS, che usa una sola frequenza se la rete lo supporta o la vera Classe A, che usa due frequenze diverse e quindi sfrutta entrambi gli standard senza problemi. È una tecnologia buona per il tempo e l'idea, ma abbastanza lenta e in generale non funzionale per i problemi descritti. Si continua a spingere sull'aumento di banda e velocità e nasce *EDGE (Enhanced Data rates for GSM Evolution)*. Oltre alla modulazione di frequenza utilizza anche la modulazione multi-fase. Esistono varie versioni e la velocità effettiva varia anche a seconda di come viene implementata questa tecnologia.

Passiamo quindi al *3G* che usa frequenze più larghe ed utilizza due standard principali, senza dover più tagliare la banda come prima e introduce canali di comunicazione anche per il video, mail, Internet, ecc. oltre che testuali. Esso utilizza *CDMA*, ampiamente descritto sopra. Un primo standard è Il *W-CDMA* (CDMA Wideband, in Europa conosciuta come *UMTS*) utilizza una banda di canale molto larga, quindi ad esempio 5 MHz, dando circa 384 kb/s per ciascun utente come data rate. Esso ragiona con il principio di dare la stessa potenza di trasmissione a ciascun dispositivo, frammentando le velocità in base alle lunghezze dei chip degli utenti; in questo modo si aumenta la capacità, migliorando l'uso delle celle. Il *CDMA2000* è stato un altro standard di trasmissione degli USA, che puntava più sugli utenti per cella piuttosto che per la trasmissione dati, avendo quindi meno data rate, in questo caso 144 Kbit/s. Oltre a 3G c'è il 4G, in questo caso con *HSDPA* (divisione a pacchetti ad alta velocità), in particolare spingendo per quanto riguarda i download. Esso utilizza una combinazione CDMA e QAM per poter accelerare al meglio la rete e il suo data rate. Un'ulteriore accelerazione risulta essere il *3.5G – HSPA+*, con *HSUPA*, dove l'unica cosa che cambia è l'uplink, spingendo sull'upload e considerato 3.75G, oppure anche *HSDPA*, tecnologia retrocompatibile con UMTS. Altra tecnologia ancora è *4G-HSOPA/LTE*, con bande variabili 1.25 MHz a 20 MHz e con velocità massima di 1.2 Gbps/600 Mbps in down/up. Piccolo appunto conclusivo sono le varie categorie di cavi utilizzati nelle comunicazioni, citando ad es. CAT3, CAT4, CAT5, CAT6, tagliando nelle ultime la banda disponibile di upload o al contrario assegnando enormi quantitativi a disposizione degli utenti, come CAT8 3000 Mb download, 1500 Mb upload).

LTE, Trasmissione stereo, Radio digitale

La banda 4G/LTE interferisce con la banda del digitale terrestre, parte di frequenze che prima non veniva utilizzata. La necessità di frequenze e del loro utilizzo di fatto ha dovuto portare a usare quella parte di frequenze e si deve proteggere la trasmissione, aggiungendo un altro filtro di trasmissione televisiva. Gli standard di comunicazione sono tuttora in piena evoluzione, per esempio il 5G, che cerca di sfruttare il meno possibile risorse materiali (ad esempio i dispositivi mobili, per risparmiare batteria) dando un certo tipo di data rate a seconda dell'uso/fascia d'utenza che viene coinvolta nella trasmissione.

Dato che si ha bisogno di più frequenze si utilizza un nuovo standard che è il DVB-T2 (il nuovo digitale terrestre), dando anche accesso alla risoluzione 8K. Il primo 5G viene creato da Qualcomm nel 2017 con velocità 1 Gbps, dove la scelta fatta da Qualcomm è quella di usare da 28 a 39 GHz, pezzi di banda non utilizzati, nella zone tra le micro-onde e le onde infrarosse, per poi introdurre altri tipi di microchip, con molta banda in download/upload, anche in grado di funzionare in parallelo con il 4G, accelerano nel processore 2021 con 10 Gb in download e 7 Gb di upload. Le microonde recuperano molto più spazio, necessitando come già visto di antenne più piccole (ad esempio nelle varie tecnologie 1G, 2G, ecc. all'interno dei telefoni cellulari).

Essendo onde più spesse, il 5G arriva meno e non oltrepassa il corpo umano, così come gli ostacoli, non risultano quindi dannoso per la salute. L'ultima frontiera di comunicazione rimasta analogica è la *trasmissione stereo*, in cui ci sono due canali che devono essere trasmessi. Le trasmissioni stereo sono quindi venute dopo e dovevano essere compatibili con le trasmissioni precedenti monofoniche. Caso simile è stato anche nelle trasmissioni televisive, quindi passando da audio mono ad audio stereo. Nel caso della banda radiofonica, essa va dai 30 Hz ai 15 KHz. Nel qual caso esiste un segnale pilota, che qualifica che ci stiamo collegando alla banda radio stereofonica. Di fatto vengono creati due nuovi segnali, il primo che è la media dei segnali dx e sx ed il secondo che è la metà della differenza tra il segnale dx e sx. In questo modo, mentre i ricevitori mono riceveranno solo il primo dei segnali, quelli stereo useranno anche il secondo per ricostruirsi il segnale.

Ciò avviene ad un prezzo perché peggiora il rapporto segnale-rumore (almeno di un fattore tre rispetto allo stereo, minimo il 300% di rumore in più rispetto ad un canale mono). Forzando mono, non ho più un processing di quel tipo e, se non si sente molto bene, permette di migliorare la qualità dell'audio, non sfruttando il canale extra dei dati; ciò le radio le fanno in automatico. Un altro tipo di multiplexing è il tipo RDS (Radio Data System) che sfrutta dati digitali se presenti nel canale (sono quei dati che vengono scritti sui display, es. nome della radio o del pezzo attualmente in riproduzione). Il data rate è di poco più di 1 Kbps, bastevolissimo per il tipo di dati trasmessi; sarebbe possibile anche sfruttare le categorie di canzoni per poterle filtrare.

Esiste la *radio digitale (DAB)*, permettendo molte più frequenze radiofoniche e stazioni; esso non ha preso piede come il digitale terrestre, in quanto l'utenza della radio non è poi molta e con più distorsioni e meno qualità della radio analogica. Questo tipo di tecnologia utilizza una riconversione del segnale digitale che viene compresso e ritrasmesso sulla banda; possiede anche un segnale inferiore rispetto alla banda FM, dato che subisce molte più interferenze, oltre a quelle dovute dall'effetto Doppler. Ecco quindi che per risparmiare banda viene sfruttata la compressione MPEG2 dove non c'è video ma solo audio, musica in particolare, peggiorando la qualità sonora.

Inizio livello fisico, framing, stuffing, parity bit

Si parla (finalmente) dello strato successivo al livello fisico, quindi il *livello 2 (data link)*, che si occupa del controllo della trasmissione (*error control*), possibilmente evitando errori di trasmissione e regolando il flusso della rete e la capacità (*flow control*). Nel servizio di rete, i pacchetti vengono inviati senza aspettare conferma di ricezione (acknowledged), evitando di stabilire una connessione dedicata (connectionless). Se il

canale è ritenuto particolarmente affidabile, si evita la conferma di ricezione (tutto dipende dalla scelta dell'utente). Quindi varia tra acknowledged/unacknowledged e connection/connectionless. L'approccio più generale è di prendere i singoli pacchetti dati e codificarli in appositi frame, successivamente mandati all'interno della rete.

L'insieme dei dati (payload) viene inizializzato, possibilmente con un header (che inizializza la porzione di dati) più il trailer (che codifica la parte di frame che verrà successivamente trasmesso). In una trasmissione è importante saper determinare dove una determinata trasmissione inizi e finisca (*framing*); un primo esempio di strategia per provare a tracciare e risolvere questa problematica è stata l'utilizzo di sincronizzazione di orologi (clock), idea poi praticamente non affrontabile (perché impegnativo su piccole misure). Per esempio è possibile mettere nell'header un numero che codifica quanti caratteri sono presenti (*character count*), tecnica già utilizzata nei singoli linguaggi di programmazione.

Esso è stato uno dei primi metodi utilizzati ma è un ambiente dove ci possono essere in altri contesti parecchi errori. Andiamo quindi ad utilizzare dei byte speciali (flag bytes) per segnalare inizio e fine dei singoli frame (*byte stuffing*). È anche necessario eseguire l'escape del byte successivo al byte di controllo, facendo quindi capire che si tratta di un byte dati. Vi è poi una tecnica analoga al byte stuffing, eseguita sui bit (*bit stuffing*), che modifica la sequenza di bit trasmessa inserendo per esempio dei bit in più, separando bit in maniera preventiva a seconda della posizione del bit assunto come flag (quindi se io avessi 01111110, significa inserire uno 0 dopo il quinto 1), che sarà successivamente tolto in fase di decodifica perché si capisce che non è altro che un bit delimitatore.

Per poter fare il controllo degli errori si sviluppano tecniche che controllano i possibili errori, nel caso chiedendo anche ritrasmissione di dati, per esempio tramite feedback (segnalando il tutto all'altra stazione coinvolta nella trasmissione) o limitando la velocità, senza utilizzo di feedback. Trovare gli errori è utile quando il canale è molto affidabile e quando l'errore in questione non è critico. Si cerca di evitare la ritrasmissione dei pacchetti facendo direttamente correzione degli errori (facendo prima *encoding*, aggiungendo quindi la protezione dagli errori e poi *decoding*, dove si ha il controllo sugli errori, quindi error detection). Un esempio molto semplice è l'inserimento di un bit di parità ogni m bit, a seconda che la somma degli m bit precedenti sia pari o dispari (il famoso *parity bit*), efficace solo con gli errori di potenza 1, quindi che toccano solo un bit.

Checksum, repetition codes

Passiamo alla fase di correzione dell'errore, suddivisi in codici di correzione (descritti sotto) e di rilevazione dell'errore (descritti qui); prima di tutto si ha il semplice *parity bit*, la tecnica più semplice, che rileva errori con distanza 1 (cioè i bit differenti tra due messaggi e possibilmente errati). Un'altra semplice tecnica è l'utilizzo di una *checksum*, che calcola il complemento ad uno della somma dei dati inviati sul canale fino a quel momento e gli errori possono essere rilevati sommando l'intera parola contenente sia i bit dati che il checksum. Qualunque sia il parity bit "m", i messaggi codificati, diversi tra di loro, sono minimo a distanza 2. Per ogni errore (ad esempio 1 bit), un messaggio codificato non potrà mai diventare un altro messaggio; la scelta del parity bit dipende sicuramente dalla dimensione del messaggio trasmesso. Possiamo identificare quindi un errore ogni $\frac{1}{M-1}$ bit, ottenendo un data rate effettivo massimo di $\frac{M}{M+1}$ bit. Di conseguenza, aumentando la possibilità di rilevare gli errori, diminuiamo il data rate (vedi ogni 2 bit inserirne 1 di controllo) perché stiamo togliendo banda alla trasmissione.

È però possibile correggere gli errori di quantità doppia in trasmissione tramite i *repetition codes*, che ripetono N volte ogni singolo bit, facendo così error detection fino a potenza $N-1$. Un codice corretto riporta i messaggi corretti quando distanti minimo N, qualora intervengano K errori e portando il tutto a distanza K. Basta avere un numero *K* di errori tali che siano meno della metà di *N*, per esempio $\frac{N}{2}-$

1 avendo N pari oppure $\frac{N-1}{2}$ se N è dispari. In questo modo i codici error correction possono correggere i messaggi senza doverli ritrasmettere pur avendo dei bit arbitrari flippati (quindi 0 al posto di 1 e viceversa). Banalmente i codici di correzione sono usati ovunque. Molte volte le stesse informazioni di error detection vengono aggiunte all'inizio dei dati trasmessi e fanno parte di essi, indipendentemente dal metodo di trasmissione. Esempio di codice di correzione è il *codice di Luhn*, codice di potenza 10 e che utilizza una checksum di parità, trovando tutti gli errori anche nel caso di una cifra sbagliata o in caso di cifre invertite. In generale comunque gli *error correction codes* sono costosi, perché devono esaminare ogni singola parola, avendo quindi un alto costo di esecuzione.

Hamming e distanza, peso

In base a quanto appena detto e i difetti riscontrati, sempre Hamming crea un nuovo set di codici, chiamati codici lineari, le cui operazioni vengono espresse mediante combinazioni lineari. I *codici lineari* sono la famiglia di cui fanno parte questi sottospazi lineari che stiamo considerando. Essi utilizzano codici a blocco usando i bit di parità in modo efficiente. Uno dei primi esempi fu il *codice Hamming (7,4)* prendendo in input blocchi da 4 bit e mandando in output blocchi da 7 bit. La *distanza di Hamming* quindi è capace di considerare il numero X di bit dei blocchi di input e costruirvi delle figure geometriche di dimensione Y (in particolare, una distanza d corrisponde al numero di errori per convertire una sequenza nell'altra e i bit diversi dopo uno XOR tra due codeword). L'encoding viene realizzato linearmente con una matrice generatrice, che sostanzialmente imita i diagrammi di Venn (cioè i cerchi rosso, verde, blu che si intersecano e determinano quali bit di protezione agiscono su quali singoli bit dati) e in questi qualora vi siano intersezioni tra i bit, tra essi viene calcolato lo XOR.

Diciamo che il *peso* di un messaggio consiste nel numero di bit ad 1 presenti al suo interno e, secondo il *teorema del peso minimo*, se il peso minimo dei vettori della matrice generatrice dista almeno d, il codice di Hamming necessario è (X, Y, d) con error detecting di potenza $d-1$ ed error correcting di potenza $\frac{d-1}{2}$. Nei codici lineari di Hamming, si ha una matrice di parità, che controlla se alla fine ci sono stati particolari errori, verificando se sommando i bit di parità ci sono stati errori o meno una volta conclusasi la trasmissione.

Burst ed errori

La codifica data interviene solo su errori singoli, tuttavia gli errori possono avvenire in qualsiasi momento; in questo caso intervengono i *burst*, che inviano le varie codeword in forma matriciale, inviando gli errori colonna per colonna, così "spalmando" l'attacco in singoli frammenti. Questa tecnica è nota con il nome di *interleaving* (o *tecnica della matrice invertita*). In questa viene calcolato un bit di parità per ognuna delle n colonne e tutti i bit dati come k righe, inserendo nell'ultima riga n parity bit e grazie proprio ai bit di parità ci accorgiamo dei possibili flip. Si misura che ogni decimo di secondo si ha un errore all'interno di RAM o tecnologie simili (unità di misura, il quadrilione, per esempio un doppio errore ogni 8 quadrilioni di secondi). Importante come il design di un sistema sia tenere conto degli errori minimi (livello 1), tralasciando le potenze più alte se possibile. Le RAM di nuova generazione sono più lente in quanto contemporaneamente correggono anche gli errori rispetto alle precedenti, con circa una penalità dell'11% rispetto alle precedenti; tali funzionalità devono essere supportate dalla scheda madre. Se supportasse questo tipo di codice, quasi certamente, usa anche SECDED (Single Error Correcting/Double Error Detecting); essi vengono usati in vari ambiti, tipo le schede SD. Essi correggono errori di potenza 1 e 2 (nel caso quest'ultimo di SECDED, come intuibile dal nome).

Reed-Solomon

Il *codice di correzione Reed-Solomon* (anche chiamato RS) è basato su aritmetica polinomiale, che agisce su Y parole usandone X, quindi *RS(X,Y)* in grado di correggere $\frac{(X-Y-1)}{2}$ errori. Esso viene per esempio usato nei DVD o nei CD. Questo codice considera come unità di base blocchi di bytes al posto dei bit, sopportando per esempio nel caso dei CD errori di burst lunghi 4000 b, il tutto in maniera autocorrettiva; RS viene usato anche in ADSL, Blu-Ray, ecc. Con questo codice possiamo essere in grado di prevenire le cancellazioni e anche di recuperare i dati persi, con un'efficacia del 200%. Il codice RS corregge fino al doppio degli errori, anche contemporaneamente *erasures* (cancellature) ed errori quali burst oppure bit flipped.

Cercando di ridurre l'error rate a zero, diminuisce anche il data rate in maniera esponenziale. Shannon ha dimostrato che la curva dei codici migliori riesce ancora ad avere un data rate positivo, usando il termine di limite di Shannon (oltrepassando ogni limite presente agli altissimi livelli della banda, mantenendo più del 50 per cento della banda). Il data rate massimo è di fatto pari all'entropia del canale, dato un certo tasso di errore X. Ogni 10 letture dell'hard disk c'è un errore (quindi un tasso dello 0,1). Shannon, dando questa formula, afferma che grazie all'entropia ne bastano due di dischi per poter mantenere tutti i dati. A questo scopo, con varie singole combinazioni, sono utilizzati i dischi RAID, per evitare gli errori nei dischi. I codici utilizzati per esempio sono quelli *LDPC - Low Density Parity Check*, che utilizzano sempre matrici di parità ed ogni bit di output è formato da una frazione dei bit di input. Essi sono codici molto potenti e questa potenza viene pagata in tempo di calcolo (np-completo, in tempo polinomiale, esecuzione potenzialmente lunghissima). La stessa cosa vale all'interno dei programmi di zippaggio/compressione; pur di dare un risultato, basta limitarsi a ciò che si è già calcolato, tagliando i dati che non si riescono a recuperare.

CRC ed esempio

Parliamo ora del codice *CRC, Cyclic Redundancy Check,* basato sull'aritmetica polinomiale in base 2 (definito GF(2)[], polinomi in base 2). Le operazioni possibili in GF(2)[] sono ad esempio addizioni e sottrazioni, dove la sottrazione è uguale all'addizione; semplice somma di polinomi ma solo avendo 0 e 1 come standard di trasmissione dati. Si sceglie quindi il polinomio generatore *G(x)* (che deve avere come primo ed ultimo bit 1), avendo un messaggio *M(x)* e calcolandosi il resto *R(x)*, facendo quindi l'encoding. Se il resto finale è diverso da 0, vuol dire che ci sono degli errori.

Per fare il decoding si utilizza uno shift a destra, dividendo quindi per *x^r*. Attenzione che se moltiplichiamo di una certa potenza x^r, stiamo facendo lo shift a sx di r posizioni. Una volta fatto lo shift, si aggiunge un certo numero di bit a 0, affinché *G(x)* contenga m+r bit e corrisponda al polinomio *x' M(x)*. Si divide poi *G(x)* per *x' M(x)* e poi si sottrae il resto; questo è il checksum pronto alla trasmissione, definito come *T(x)*. I singoli errori sono del tipo *E(x)=x^i*, mentre per doppi errori o errori burst, il polinomio di errore diventa *E(x) = x^i + x^j*, dato che possiamo scegliere il polinomio generatore, basta che G(x) non sia multiplo di x per poter funzionare. Sotto un esempio applicativo:

Esempio Si calcoli il CRC di 10011101 usando $G(x) = x^4 + x + 1$
- $M(x) = x^7 + x^4 + x^3 + x^2 + 1$
- $P(x) = M(x) \cdot x^{deg(G(x))} = x^{11} + x^8 + x^7 + x^6 + x^4$
- Divido $P(x)$ per $G(x)$ e calcolo il resto $R(x)$

$x^{11}+x^8+x^7+x^6+x^4$ | x^4+x+1
$x^{11}+x^8+x^7$ | x^7+x^2+1
x^6+x^4
$x^6+x^3+x^2$
$x^4+x^3+x^2$
x^4+x+1
$x^3+x^2+x+1=R(x)$

- Eseguo la XOR tra $P(x)$ e $R(x)$

100111010000
000000001111
100111011111

Il risultato è il dato da trasmettere.

Per fornire una protezione ben sufficiente si utilizza G alla potenza esponenziale che riesce ad essere protetto da ogni possibile errore. Alcuni esempi dell'uso di questo codice sono CRC-16, usato nella trasmissione USB, CRC-16CCITT che si usa nel Bluetooth e CRC-32, usato nei modem o negli zip. Essendo infine i codici CRC autocorrettivi, abbiamo anche la possibilità di generare degli errori volontariamente, contando sul fatto che poi vengano corretti; ad esempio i codici QR code BBC, dove viene inserito come "errore" il logo BBC, risultando comunque in un messaggio dati valido e comprensibile.

Stop and wait, timeout, piggybacking, sliding windows, Go Back N, Selective repeat

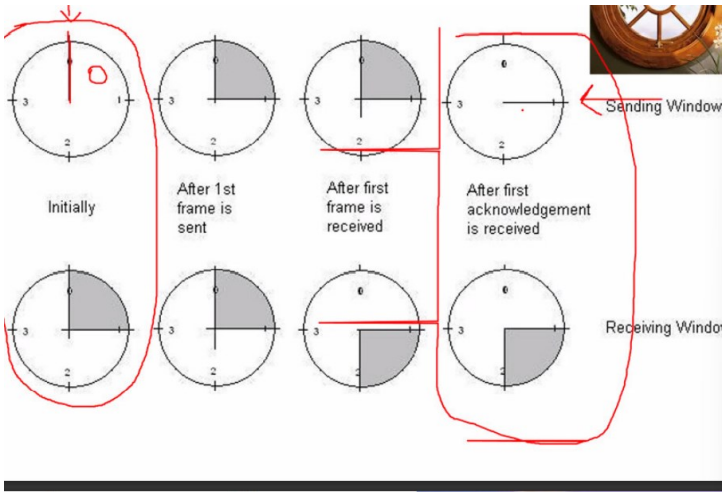
All'interno della trasmissione di rete, è importante tenere sotto controllo il flusso della trasmissione compresi quanti dati vengono effettivamente trasmessi; l'approccio è chiaramente limitativo, magari trasmettendo meno dati, ma facendo in modo il ricevente sia in grado davvero di gestire ciò che gli sta arrivando. Si introduce un protocollo ***stop and wait***, aspettando che il ricevente mandi il messaggio di conferma (ACK), dopo aver mandato un frame di dati. L'error detection viene eseguito e viene mandato una conferma una volta ricevuto il messaggio. Può essere usato sia in canali half-duplex che in canali full-duplex, essendo una comunicazione non contemporanea. Potrebbe anche succedere che, a causa anche dei codici di correzione utilizzati, un determinato pacchetto venga inviato all'interno di una rete e si rimanga indefinitamente in attesa (roundtrip delay elevato, che può portare oltre a questo anche alla ricezione dello stesso frame da parte del destinatario).

Viene introdotto il concetto di ***timeout*** per cui se, entro il tempo limite, non si riceve un messaggio di conferma, il pacchetto viene rimandato. Nella trasmissione i singoli dati vengono inviati sempre due volte e, per accorgerci delle ripetizioni di pacchetto, ognuno viene numerato e il ricevente sa se ha già ricevuto o meno un particolare frame. Il numero che infatti viene utilizzato nella trasmissione deve sempre essere abbastanza grande/piccolo compatibilmente con ciò che si deve mandare; basta anche un solo bit, distinguendo semplicemente tra pacchetto precedente e pacchetto successivo. Nel caso il pacchetto vada perso, esiste una classe di protocolli chiamata *PAR (Positive Acknowledgement with Retransmission)*. In questi protocolli la trasmissione può essere in un senso (half-duplex) o in tutti e due i versi (full-duplex).

Esiste la tecnica del ***piggybacking***, che invia il messaggio di conferma non da solo ma mandandolo con il primo frame dati da noi inviato. Si ha quindi un overhead minimo rispetto a prima. La trasmissione in questo senso deve essere bilanciata, perché magari si rimane in attesa per troppo tempo del piggyback di uno o dell'altro, aumentando quindi lo spreco di banda. Se prendiamo l'esempio della comunicazione satellitare, l'attesa genererebbe troppo delay sul margine di trasmissione (si calcola per esempio un 96% del tempo in stop. Quando il prodotto *bandwidth * round-trip-delay* è grande, stiamo sottoutilizzando il canale. Se il canale ha una capacità C (bit/s), con taglia del frame S (size, sempre in bit) e tempo di round-trip (R) (quindi tempo di ricezione di risposta), l'utilizzo della linea con protocolli con ack viene calcolato con la formula:

S / (S+CR); *Attenzione che se S < CR abbiamo un'efficienza minore del 50%*

Mandiamo dati aspettando per ognuno gli ACK (acknowledgement/ricevute di ritorno), gestendo il tutto in parallelo come una pipeline, infatti la tecnica viene chiamata *pipelining* (quindi uno attivo e gli altri in attesa round-robin, proprio come avviene nelle pipelines). La tecnica delle ***sliding windows*** si basa su questo principio, preoccupandosi quando c'è un numero di frame a rischio maggiore rispetto a quelli n trasmessi e tipicamente viene scelta una potenza di 2 per non sprecare bit. La taglia può variare sia da sender che receiver. Le finestre sono sia in entrata che in uscita; ciò è visibile anche sotto (nell'es. n=4, size=1).



Ogni finestra grigia corrisponde ad un certo numero "*n*" di frame trasmessi aspettando per questo uno ack (solitamente n è una potenza di 2). Nella fase di invio del frame da parte del mittente, resta nella finestra finché non viene ricevuto l'ACK corrispondente; a seguito dell'arrivo di questo, viene poi aggiornata. Se invece si riceve il frame, si controlla che il numero del pacchetto corrisponda a quello atteso e se corrisponde invia l'ACK. È possibile inviare più frame in contemporanea e alcuni protocolli sfruttano il principio delle sliding windows, come il *Go Back N*, in cui la finestra del ricevente ha ampiezza 1 e quella di chi trasmette ha ampiezza N e, se il destinatario trova un frame corrotto, scarta tutti gli N successivi, aspettando che scada il timeout. Essa viene usata in caso di alto prodotto *bandwidth * roundtrip-delay*. Altra tecnica in cui anche il ricevente ha un buffer per poter contenere più pacchetti in contemporanea è quella del *selective repeat*, in cui a seguito di pacchetto corrotto, il destinatario risollecita la trasmissione usando un *NAK (Not Acknowledgement)*; fino a quando non arriva nuovamente il pacchetto giusto da parte della sorgente, il destinatario ospita tutti i pacchetti nel suo buffer, aspettando che, selettivamente, arrivi il pacchetto interessato, sulla base di un *NAK timer.* Sotto si nota l'efficienza di NAK e selective repeat:

Quando si inizia la trasmissione, solitamente, la taglia richiesta del buffer è l'ampiezza massima della finestra. Attenzione ad usare troppo parallelismo, in quanto potrebbe risultare confusionario perché non è garantita la sequenzialità del canale e risulta essere svantaggioso in termini di allocazione di risorse (come nel caso poco sopra descritto di selective repeat). Potremmo trovarci nella situazione in cui i pacchetti sono troppi rispetto alla trasmissione esistente sulla sliding window, perché non ci sono stati degli ACK; conseguenza di questo è l'invio e la ricezione dei singoli pacchetti in un ordine ben diverso rispetto a quello

previsto o anche la possibile sovrapposizione tra i singoli pacchetti, generando ambiguità. In questo caso conviene che l'apertura del buffer sia al massimo metà della grandezza della sliding window.

HDLC

Diamo ora alcuni esempi di protocolli reali. Il primo protocollo utilizzato è il frame *HDLC (High Level Data Link Control)*, utilizzato ad esempio nei Bancomat. La tecnica usata anche da HDLC è il bit stuffing. Nei pacchetti HDLC si hanno tre campi, in particolare la correzione nella parte *Checksum* (con *CRC*), una parte *Data* che contiene l'informazione (*payload*) e la parte *Address* che serve come componente per l'indirizzamento.

Ci possono essere 3 tipi di frame in una trasmissione, quindi *Information* (utile per le conversazioni multiple e usa alcuni byte per poter fare framing, come ad esempio *Seq* per determinare la sequenza di flusso, oppure *P/F (Poll/Final)*, indicanti inizio/fine trasmissione), *Supervisory* (determina se ci siano stati o meno ACK/NAK) e *Unnumbered* (usato nel caso di perdita di frame e per gestione della connessione). La sliding windows deve avere grandezza massima di 3 bit di dati in parallelo inviati in un determinato momento (8 spicchi, in potenza 2^3); un'altra variante di controllo di flusso si fa con una sliding windows grande 128 spicchi, usata ad esempio nelle comunicazioni satellitari.

Alcuni esempi di byte *Supervisory* sono ad esempio il tipo 0: ACK, usato quando il flusso dati è sbilanciato e non si può fare piggybacking, il tipo 1: REJECT, che rappresenta un NAK generalizzato, il quale segnala i frame che saranno ritrasmessi successivamente a quello già inviato (partendo quindi da un elemento next), tipo 2: RECEIVE NOT READY, dove la trasmissione viene bloccata da altri eventi esterni che congestionano il traffico di rete, tipo 3: SELECTIVE REJECT (classico NAK). Diamo quindi anche esempi di frame *Unnumbered*, ad esempio DISC (DISConnect), che segnala che la macchina sta uscendo dalla rete, SNRM (Set Normal Response Mode), che segnala l'entrata di un dispositivo in una rete (con priorità inferiore), SABM (stessa cosa di prima, ma crea connessione bilanciata), oppure il FRMR (Frame Reject), indicando che è arrivato un frame con una sequenza di controllo non corretta o sconosciuta.

PPP, LCP, NCP, MTU, ATM

Il protocollo *PPP (Point-to-Point Protocol)* definisce il pacchetto dei dati e come le connessioni attivano test e negoziazione. Usa lo stesso principio di HDLC e usa byte stuffing invece che bit stuffing. Il PPP possiede il campo *Protocol* (variabile nelle dimensioni a seconda delle esigenze trasmissive), che specifica il protocollo utilizzato ed effettua la negoziazione come un vero e proprio livello fisico (*Link Layer Control Protocol (LCP)*) oppure come livello di rete (*Network Layer Control Protocol (NCP)*, campo *Payload* di dati e variabile nelle dimensioni, campo *Checksum*, che calcola gli errori con *CRC* ma non facendo error recovery. Come si vede anche sotto, vi sono poi le flag, realizzate con il principio di byte stuffing. Una panoramica riassuntiva dei campi è la seguente:

Nome	Numero di bytes	Descrizione
Flag	1	indica l'inizio o la fine del frame
Address	1	indirizzo broadcast
Control	1	byte di controllo
Protocol	2	indica il protocollo del campo data
Data	variabile (da 0 a 1500)	campo di dati
FCS	2 (o 4)	somma di correzione

Per la LCP sono previsti quindi 11 campi, in particolare 4 di configurazione, 2 di terminazione, 2 di rifiuto, 2 di echo ed 1 di test. Sono previsti alcuni comandi, *configure-request*, configurando la linea e i suoi parametri, *configure-ack*, che segnala con un ACK di aver accettato la configurazione, *configure-nack*, segnalante il disaccordo con la configurazione proposta, *configure-reject*, che rigetta semplicemente la

configurazione proposta. Con un'opzione, si possono togliere del tutto i campi Address e Protocol perché sono fissi, risparmiando quindi 2 byte a frame. I comandi di fine codice e rifiuto in LCP sono similari a prima, chiamati *code-reject* e *protocol-reject*, inviati quando non si capisce il protocollo usato oppure perché non si capisce la richiesta. Altri tipi di comandi sono gli echo, quindi *echo-request* ed *echo-reply* che testano la qualità di comunicazione. Prima dei comandi echo, vi è il comando di *test/discard request* che non fa nulla di pratico, ma serve a testare la qualità della linea prima della trasmissione o per trovare dei loop. La dimensione massima del payload chiamata *MTU (Maximum Transmission Unit)*, fissato per ogni protocollo ed è riconfigurabile verso il basso. Si cerca di scegliere dei pacchetti più grandi, dando meno overhead e avendo quindi più banda (quindi MTU grande); per converso scegliendo MTU piccolo si ha una diminuzione della dimensione dei pacchetti, aumento di overhead e meno banda (cio è utile quando il canale ha molti errori).

Vi sono quindi due varianti di *PPPoE* e *PPPoA* (*PPP over Ethernet* oppure *PPP over ATM*), con pacchetti incapsulati dentro altri pacchetti Ethernet o ATM appunto. I flussi dati variano per complessità, passando da computer a router (flusso 1), router/modem (flusso 2), modem/provider (flusso 3), flusso 4 DSLAM ("Digital Subscriber Line Access Multiplexer", multiplex per più dispositivi) e rete Internet (flusso 5).

Approfondiamo quindi il concetto di *ATM (Asynchronous Transfer Mode)*, che usa TDM per dividere i dati in celle di ampiezza fissa e realizza controllo di flusso con sliding windows (16 spicchi solitamente), error detection (con CRC-8) ed indirizzamento, che funziona con una gerarchia fatta di cammini (paths) e canali (channels). Altre etichette sono le VPI e VCI, identificatori per Path e Channel, il tutto realizzato con una logica di tipo connection-oriented. I multiplexing possono essere incapsulati in livelli di link logico di controllo, chiamati *LLC* (Logical Link Control, protocolli multipli per canale) o *VC–Mux* (Virtual Connection Multiplexing, un solo protocollo per canale).

Nel caso della trasmissione in ADSL, questa inizia con un frame *PPPoE Active Discovery Initation*, dando il proprio indirizzo fisico (MAC). Ogni servizio in quel momento disponibile dà la propria disponibilità tramite un comando *PADO (PPPoE Active Discovery Offer)*, in cui comunica di voler connettersi ed iniziare la trasmissione. Vi sarà quindi un *PADR (PPPoE Active Discovery Request)*, segnalando quindi il servizio scelto e *PADS (PPPoE Active Discovery Session-confirmation)*, che dà un'ACK di avvenuta configurazione. Alla fine la trasmissione ha un frame *PADT (PPPoE Active Discovery Termination)*, che semplicemente termina la connessione e, di conseguenza, tutta la trasmissione.

Contention systems, carrier/no carrier sense, ALOHA, Slotted Aloha, CSMA e Vari tipi

Nei sistemi di comunicazione multipla, c'è un unico canale condiviso con dei protocolli di comunicazione multipla in cui possono creare contenziosi (*contention systems*), non essendo più point-to-point (quindi traffico indipendente, canale singole per le N stazioni e collisioni osservabili). All'interno dei protocolli ovviamente devono essere stabilite delle regole, assumendo alcune informazioni, ad esempio lo *Station Model ("modello a stazione"),* in cui le entità che hanno iniziato la trasmissione non fanno altro finché non è stato trasmesso il frame che stanno inviando in quel momento. Altra regola utile è l'utilizzo di un canale singolo (single channel, disponibile per tutti) e si possono incontrare delle collision (collisioni), quando due frame si sovrappongono, rendendoli inutilizzabili. Altre assunzioni che vengono fatte sono sul carrier (mezzo di trasporto), vedendo se il canale è già in uso prima di una trasmissione (*carrier sense*) o entità cieche (*no carrier sense*) e conoscendo lo stato del canale prima di utilizzarlo (rilevamento della portante). Vengono inoltre fatte assunzioni sulla temporizzazione delle stazioni, quindi se a *tempo continuo* (senza frammentazione) oppure *diviso ad intervalli*.

Una rivoluzione nelle comunicazioni fu la rete *ALOHA*, sistema a contesa ideato all'interno delle Hawaii per trasmettere i segnali in broadcast in maniera casuale, mandando i frame in maniera asincrona da un punto

di vista temporale. Se il frame viene trasmesso correttamente, verrà trasmesso il frame successivo. Se la trasmissione non riesce, la sorgente invierà di nuovo lo stesso frame. ALOHA funziona bene con sistemi di trasmissione wireless o collegamenti a due vie half-duplex. Quando invece la rete diventa più complessa, come una Ethernet con più origini e destinazioni che utilizza un percorso dati comune, si verificano problemi a causa della collisione dei frame di dati. Quando il volume di comunicazione aumenta, il problema della collisione peggiora. Ciò può ridurre l'efficienza di una rete poiché i frame in collisione causeranno la perdita di dati in entrambi i frame (capita ad esempio in caso di banale sovrapposizione).

Viene utilizzata la distribuzione di Poisson con la formula $\Pr[k] = \frac{G^k e^{-G}}{k!}$, prendendo la media delle trasmissioni nell'unità di tempo e determina la probabilità che ci siano *k* trasmissioni nonostante possibili eventi caotici. Con Poisson, se con N frame di media, si ha N > 1, la frequenza dei frame è più elevata rispetto a quella gestibile dal canale, altrimenti se compreso tra 0 ed 1, il carico è ragionevole. Come detto nel protocollo ALOHA, la trasmissione può fallire in ogni momento per effetto delle collisioni, essendo le trasmissioni casuali e si ha una possibilità di collisione nel tempo *t* più il tempo di trasmissione del file (ad esempio *t0*). Si verifica quindi quale sia la probabilità che ci sia una sola trasmissione in un tempo doppio dell'unità. Il numero medio di frame è $2G$ e la probabilità che nessuno trasmetta è Ge^{-2G} (la probabilità di invio di 0 frame è e^{-G}).

Si attua in particolare un meccanismo di *back*-off, per cui la ritrasmissione avviene dopo un ritardo selezionato compreso tra T (tempo di trasmissione del frame) e K (che dipende dal numero di collisioni già avvenute). Il problema principale è il fatto che tutto è in maniera randomica, pertanto non si può sapere se un certo frame è giò stato trasmesso; per esempio capita quindi che un certo frame venga ritrasmesso e collide con altri intervalli di trasmissione e frame conseguenti. Il vantaggio di ALOHA è di essere particolarmente scalabile (scalabilità infinita) come sistema, mettendo moltissimi dispositivi allo stesso momento, *sperando al massimo di utilizzare il canale al 18%* (ciò dipende dal numero di entità della rete ma come detto dalla natura randomica del sistema). Un'altra cosa che viene fatta è fissare randomicamente la lunghezza di un frame, non utile perché il sistema diventa asimmetrico ed occorre fissare la lunghezza dei dati trasmessi.

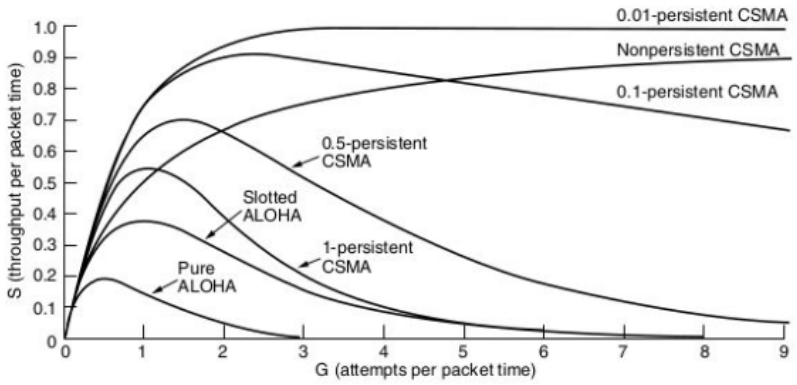
Viene quindi adottata questa idea in una trasmissione fissa (*Slotted ALOHA*), tramite una stazione principale che manda un apposito segnale di sincronizzazione e i singoli utenti devono sincronizzarsi all'inizio di ogni intervallo. Il periodo critico è dimezzato e l'efficienza risulta essere Ge^{-G}. Si possono avere problemi *solamente quando due frame vengono inviati contemporaneamente nello stesso slot*, situazione quindi non comune. La migliore situazione ottenibile in Slotted ALOHA prevede il 37% di intervalli vuoti e il 26% di collisioni. In particolare, la probabilità che una trasmissione richieda esattamente k tentativi è:

$P(k) = e^{-G}(1 - e^{-G})^{k-1}$. Ultima nota, né ALOHA né la versione Slotted sono dei protocolli *carrier sense*, come visto, perché lo stato del canale non è conosciuto a priori. Sotto, un confronto grafico per efficacia:

Un esempio di protocolli carrier sense è *CSMA (Carrier Sense Multiple Access)* in cui, prima di trasmettere, si controlla che non ci sia già una trasmissione in corso. Dato che possono succedere delle collisioni, si aspetta un tempo casuale per trasmettere e poi si riprova. Nel primo caso si ha una trasmissione chiamata *1-persistent CSMA*, in cui una stazione ha probabilità 1 di trasmissione quando trova il canale libero. Qui le collisioni continuano a sussistere, a causa del ritardo di propagazione. Questo protocollo, pur ascoltando il canale per sapere se è già occupato, ottiene un data rate superiore al 50% ma i problemi principali sono dovuti alla mancata considerazione del tempo di propagazione del segnale (quindi una stazione non ha modo di sapere subito se il canale è occupato e passa del tempo prima che ne venga a conoscenza) e il fatto che, se un canale diventa libero, si ha la possibilità che due stazioni trasmettano in contemporanea.

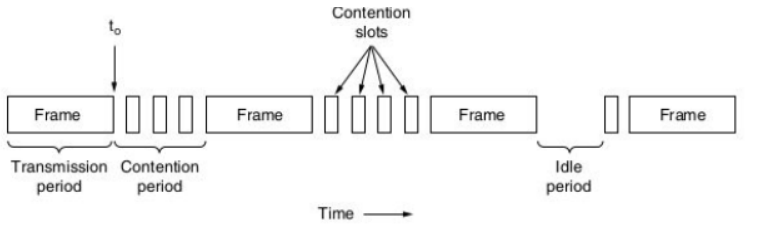
Si ha anche *p-persistent CSMA*, che si applica ai canali divisi in intervalli temporali e ogni stazione controlla il canale; se libero, trasmette subito con probabilità "p" altrimenti rimanda fino all'intervallo successivo con

probabilità $q = 1 - p$. Il processo si ripete fino a quando il frame non è stato trasmesso oppure un'altra stazione non inizia a trasmettere; in questo caso, la stazione sfortunata si comporta come se ci fosse stata una collisione (quindi attendo un intervallo di tempo casuale e poi ricomincia). Si può avere anche *nonpersistent CSMA*, che raggiunge performance vicine al 90%. Anche qui si attende un tempo casuale per trasmettere e di conseguenza si ha un miglior utilizzo del canale ma con maggior ritardo, per il fatto che se il canale è occupato, la stazione non esegue un controllo continuo per trasmettere subito il proprio frame e attende un intervallo casuale, allungando i ritardi di trasmissione. Tutte queste trasmissioni si rivelano inefficaci in caso di collisione. Qui a confronto i vari CSMA:



CSMA-CD/CA, Limited contention protocols, wireless, stazione esposta e nascosta

Un miglioramento rispetto ai precedenti si ha con *CSMA-CD (Collision Detection)*, che consente ad ogni stazione di annullare la propria trasmissione in caso di collisione, risparmiando tempo e banda. Questo avviene per il fatto che, vedendo il tipo di segnale trasmesso, capisce se si tratta di un segnale errato o meno. Occorre ragionare quanto tempo allocare ad una certa zona per decidere la sorte della trasmissione. Il tempo da allocare sarà due volte il tempo di trasmissione tra le stazioni più distanti (solitamente due stazioni, questo è il tempo di *roundtrip*). Tipicamente quindi il periodo di contention ha come durata il tempo massimo di roundtrip e poi usando tecniche di protocollo multiaccesso, in maniera tale da aggiudicarsi il canale. La rete viene separata in 3 stati qui, quindi fase *transmission, contention ed idle.*

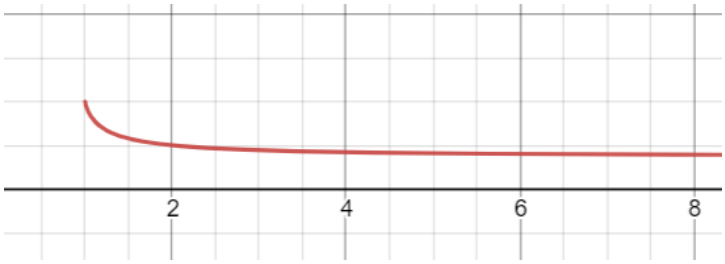


Per poter evitare le collisioni, si introducono dei protocolli *collision free*, nel qual caso *CSMA/CA (Collision Avoidance)*, facendo un uso intelligente del contention period, dividendolo in intervalli uguali e quindi riservando il "mini-slot" ad un processo in maniera riservata (*reservation protocol*). I frame in questa trasmissione sono in broadcast e quindi, sapendo chi ha trasmesso in maniera ordinata (dando precedenze e quant'altro), grazie alla reservation risulta essere efficiente.

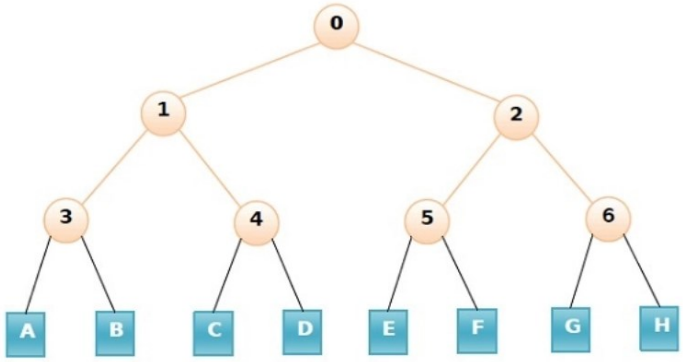
Con tante stazioni questa situazione diventa un problema (a pieno carico, se la taglia del frame è *d*, l'efficienza sarà $\frac{d}{(d+1)}$; inoltre, funziona bene a carico basso di richieste, peggio a carico alto. Il comportamento di CSMA/CA fissato il numero di stazioni ha formula $Np(1-p)^{N-1}$. Una buona soluzione combina sia CSMA/CA che CSMA/CD, gestendo bene sia pochi che molti utenti sulla banda. Ci possono essere due meccanismi di scelta per la trasmissione su un canale, per esempio tramite la scelta accurata del

frame di trasmissione e le stazioni che vogliono trasmettere lo fanno partendo dal bit più significativo (*binary countdown*).

La probabilità che la trasmissione funzioni bene in caso di contesa nella trasmissione, quindi con un *protocollo simmetrico* (caso ALOHA slotted) è $\left(\frac{N-1}{N}\right)^{N-1}$; essa decresce rapidamente e in maniera asintotica converge ad $\frac{1}{e}$. Si stabilizza così (sempre caso ALOHA slotted):



La competizione non sarà determinata a priori, ma magari dinamicamente diminuisce la competizione (*limited-contention protocols*, di cui segue subito esempio). Qui non si ha per forza simmetria, perché bisogna limitare la contesa privilegiando alcune stazioni. Per poter realizzare una suddivisione equa della trasmissione, si utilizza un *adaptive tree protocol*, dividendo il tutto in gruppi e sottogruppi, tagliando esponenzialmente la banda passibile per l'utilizzo della rete e tra competizioni di parti.

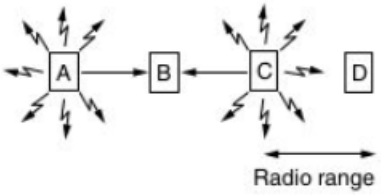


La scala quindi è in modo logaritmico (essendo un albero chiaramente) e, scendendo verso le foglie, la competizione diminuisce, dato che poi si seleziona una sottoparte dell'albero. Analizzando il traffico recente, possiamo renderci conto di quante stazioni stanno cercando di trasmettere in un certo momento; probabilisticamente, questo modello è realistico, perché considera che il carico di trasmissione possa variare ma non essere pesantemente sbilanciato.

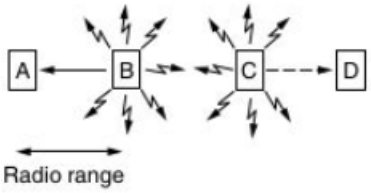
Sapendo quante sono le stazioni attive, inutile sprecare tempo cercando dalla cima dell'albero, ma piuttosto cercare da un sottoalbero qualsiasi (per esempio, volendo trasmettere al 90%; piuttosto che iniziare dal 100% e poi scendere 75%, 50%, ecc. , voglio andare direttamente ad un sottopezzo X). Ad una profondità P, i nodi sotto sono all'incirca 2^P. Se le stazioni attive sono equamente distribuite, vi saranno circa $\frac{A}{2^P}$ stazioni attive in un sottoalbero; l'ottimalità si ha con quando si ha $\frac{A}{2^P} = 1$ e quindi P = $\log_2$A. Ad esempio, avendo circa 8 stazioni attive, conviene cominciare a cercare a profondità 3. In particolare, quindi, se un nodo superiore ha collisione, si procede verso le foglie cercando di far trasmettere i figli, preservando almeno una quantità di possibile data rate. Qualora si verifichi una collisione, si cerca ricorsivamente un canale libero e si applica il procedimento di ricerca descritto matematicamente qui sopra.

Pensando ad esempio al caso *wireless*, vi sono ulteriori difficoltà dovute al cambiamento della topologia di rete che, essendo mobile (come dispositivo), *non ha una topologia fissa*. Non vi è più quindi un singolo

canale, ma le singole zone variano a seconda dei dispositivi che in quel momento stanno comunicando tra di loro. Ecco quindi che il controllo diventa locale, non più globale. Vediamo due esempi di trasmissione:



A che trasmette a B, che vi è collegato per mezzo della bolla trasmissiva. D è al di fuori della bolla, pertanto non sente, ma decide che vuole trasmettere, per esempio a C (collegato a B e D). B sarà quindi sovrapposto tra la bolla di A e la bolla di C. Ora si hanno dei problemi dovuti alla *stazione nascosta* (*hidden station problem*), perché non riesce a rievare potenziali concorrenti nel mezzo a causa dell'eccessiva distanza.



Dualmente, consideriamo il caso di B che trasmette e C riceve, non trasmettendo perché già B sta trasmettendo. Ora il problema è che C avrebbe potuto trasmettere ma non lo ha fatto (*exposed station problem, problema della stazione esposta*), dovuto anche qui ad una cattiva ricezione che porta ad oscurare dei membri nella trasmissione. *MACA (Multiple Access with Collision Avoidance)* è stato uno dei primi protocolli per le reti WLAN (Wireless LAN) e si basa sull'idea di condivisione del canale affinché anche le altre stazioni siano a conoscenza del suo stato. Esso utilizza alcuni frame, ad esempio *RTS (Request to Send)* tra un host e un altro, richiedendo uno specifico frame e, una volta calcolata la lunghezza viene inviato nel segnale "via libera", chiamato *CTS (Clear to Send)*; grazie a questo semplice sistema, anche con bolle di trasmissione intersecate o al di fuori della stessa bolla di azione, non ci sono problemi, perché le stazioni vicine rimangono in silenzio e attendono un tempo casuale per ritrasmettere (modalità non-persistent). Esiste anche la variante *MACAW (MACA Wireless)*, versione migliorata che introduce un ACK dopo ogni frame inviato correttamente e CSMA per controllare se le stazioni vicino stanno trasmettendo frame RTS ad una stessa destinazione.

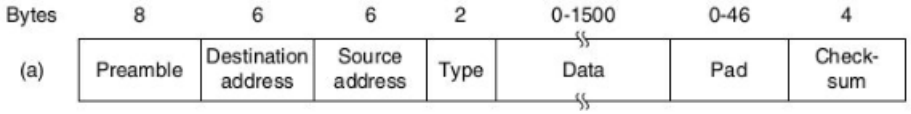
Ethernet, Manchester encoding

Parliamo ora di *Ethernet (IEEE 802.3)*, la cui invenzione si deve a Metcalfe e deriva da ARPANET, oltre che essere stata ispirata direttamente anche da ALOHA. Essa è una forma di trasmissione semplice e considera tassi di trasmissione da 3 a 10 Mbps (versione *classica*) oppure potenziata utilizzando hub o switch e raggiungendo velocità massime fino a 10000 Mbps (versione *switched/commutata).* Il creatore, Metcalfe, discute anche il *principio di qualità della rete*, determinandone il valore in base al numero degli utenti (solitamente il quadrato del loro numero e questa viene chiamata legge di Metcalfe). Il design di questo tipo di rete (nello strato fisico) usa un dispositivo chiamato *transceiver* (che incorpora carrier detection e collision detection) ad ogni giunzione del pezzo di rete.

Distinguiamo i vari cavi usati in questo tipo di trasmissione, per esempio *10Base5 – thick Ethernet*, 500 metri e 100 stazioni. Versioni successive a questa sono la *10Base2 – thin Ethernet*, con cavo coassiale fine e con 200m. massimi di lunghezza, 30 stazioni massime, *10BaseT*, 100 m. massimi di lunghezza, 1024 stazioni massime e usa cavo twisted pair telefonici e anche la *10Base-F* che, come suggerisce il nome, usa la Fibra ottica, permettendone dei segmenti stesi fino a 2 Km. Per codificare i bit 0 e 1, la normale codifica porta a problemi di sincronizzazione. Metcalfe decide quindi di implementare una nuova codifica non usata fino ad

allora, cioè *Manchester encoding*, che differisce rispetto alla codifica classica in quanto usa due forme d'onda alternate in un senso e nell'altro, dividendo il periodo di bit in 2 intervalli uguali, transizioni dall'alto al basso (caso 1) e viceversa (caso 0). Di fatto mixa il segnale di clock con il segnale dati, facendo uno XOR tra i due e, così facendo, il clock compie una transizione per ogni bit. Così facendo si elimina la necessità di un clock per il controllo dati sulla trasmissione, ma si dimezza la banda.

Struttura del frame Ethernet



I frame Ethernet hanno un *preambolo* di 8 byte, con 6 byte di *indirizzo di destinazione*, 2 per *indirizzo sorgente* e 2 di tipo e dati (da 0 a 15 byte). Il primo bit segnala la comunicazione multicast nello spazio del campo indirizzi, che ricordiamo essere particolarmente grande (6 byte). Il secondo bit distingue indirizzi locali da quelli globali, avendo come spazio ben 46 bit, che infatti rappresentano i *MAC Address* (identificativo nella rete globale della rete locale, dove MAC è Media Access Control). Negli indirizzi MAC i primi 3 byte identificano il loro produttore (*Organizationally Unique Identifier, OUI*), i secondi 3 lo spazio di indirizzi.

Molte aziende (colossi della tecnologia mondiale, ma anche compagnie/aziende che non si occupano propriamente di informatica) detengono un loro OUI, chiaramente acquistato perché proprietario. Ciò significa che, con molta probabilità, le attività dell'azienda vertono alla comunicazione di singoli dispositivi all'interno della rete. Continuando la specifica e la strutturazione del frame di Ethernet (chiamato anche DIX frame) seguono una checksum di 4 Byte che usa CRC-32, la quale fa solo error detection, e il campo *Pad*, con 46 byte massimi. Esso è importante nella correttezza dei dati trasmessi in quanto rende efficace la collision detection, assicurando che la lunghezza minima del frame trasmesso sia almeno il tempo di roundtrip e desincronizza i pacchetti singolarmente. Questo aiuta soprattutto in caso di pacchetti errati o pacchetti particolarmente grandi, per evitare attese quindi o che altri trasmettano in modo improprio.

Binary exponential backoff, hubs, switch, spanning tree, router, flooding e distance vector routing

Il problema principale di Ethernet è che la sua efficienza è inversamente proporzionale al prodotto *banda*lunghezza* (aumentando la banda quindi diminuisce l'efficienza) ed è bene limitare la lunghezza massima della rete nel caso di aumento della banda. Per evitare si debba aspettare molto, si aspetta in maniera esponenziale rispetto ai bit mandati e si ritrasmette quando la trasmissione non vada a buon fine (*binary exponential backoff*). Esiste anche la versione *truncated*, che si ferma dopo 10 collisioni.

Il problema nelle reti è unire le singole reti più grandi e farle interagire tra di loro. Infatti quello che li fa interagire sono una serie di dispositivi noti, come *ripetitori/hubs* (che amplificano il segnale e propagano il segnale lungo le varie porte di rete), *bridge/switch* (che agiscono a livello logico, dato che in una rete, si ha una fase preventiva di *learning*, in cui ad esempio lo switch impara la configurazione di rete e ne gestisce il traffico; ciò avviene anche all'inverso, *backward learning*, in cui si costruiscono delle hash tables all'interno di ogni macchina associate, quindi i dati dei dispositivi collegati ad ognuna). Dentro gli switch vi è anche un timeout di fading per ovviare ai cambiamenti nella topologia di rete, dopo il quale le hash tables vengono cancellate e quindi le informazioni precedenti.

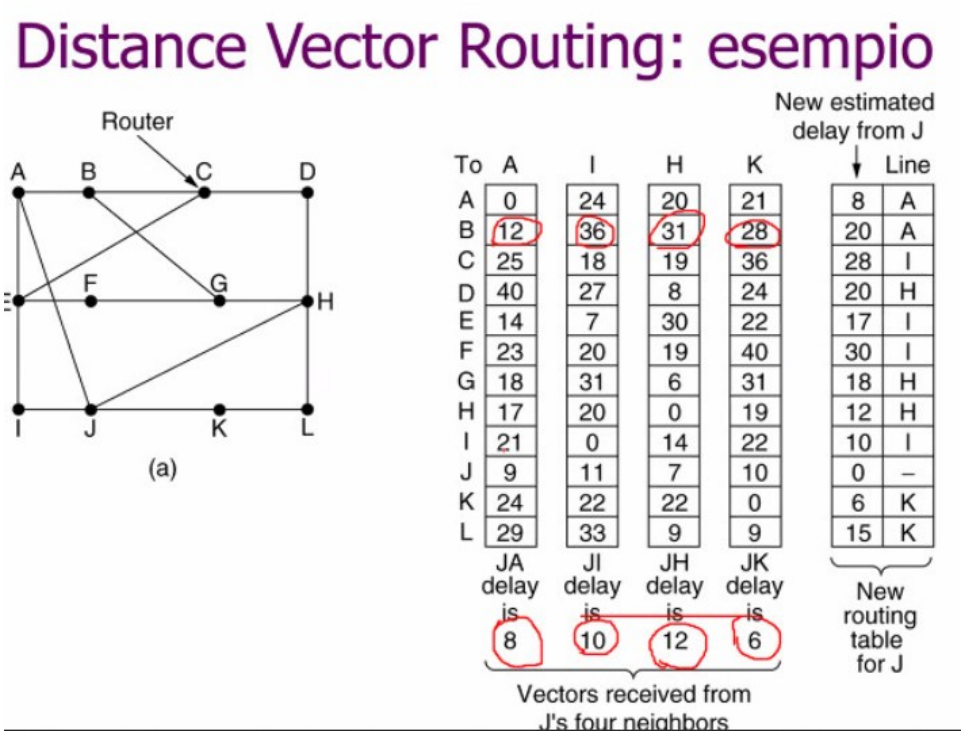
Per poter capire se vi sono loop all'interno di una rete, gli switch realizzano anche il *calcolo dello spanning tree*, mettendo in comunicazione tutti gli switch e, una volta eletto un nodo root, vengono costruiti percorsi con i nodi più brevi tra root e gli altri, stavolta approfittando dei bridge. Una volta imparata tutta la

configurazione, si usano tutti i dati raccolti tramite broadcast da tutti i dispositivi collegati alla rete, clonando la singola configurazione e propagando pacchetti ad altre reti finché non individua il giusto destinatario di un certo pacchetto.

Sempre importanti anche i *router*, che portano a destinazione i pacchetti sulle reti scegliendo il percorso migliore dato sulla base della distanza tra le stazioni sulla base del numero di "hops" (misura delle stazioni incontrate durante il cammino). Un metodo di routing potente è il *flooding*, che ritrasmette ogni pacchetto a tutte le linee. Naturalmente sarebbe un problema se mal gestito, infatti la rete sarebbe sommersa di pacchetti inutili; abbiamo quindi due tecniche per gestirlo, quindi l'*hop counting*, associando un massimo numero di hop ad ogni pacchetto, oltre i quali il pacchetto muore o il *tracking*, che tiene traccia dei pacchetti già arrivati adottando uno speciale contatore che numera i pacchetti già ricevuti, scartando i pacchetti con contatore doppio. Dato che sceglie tutti i percorsi, il flooding *sceglie sempre la via migliore* ed è quindi il *sistema più robusto*. Questo è utile in tutti i casi in cui il carico di rete non è molto alto o in cui è critico che un certo messaggio arrivi il prima possibile (esempio pratico d'uso: ambito militare).

I router utilizzano degli algoritmi per realizzare l'instradamento/routing, ad esempio il primo utilizzato fu il *distance vector routing*, usato sin da ARPANET (noto anche come Bellman-Ford, nome molto più comune infatti). Ogni router possiede una tabella con tutte le informazioni sulla rete, compresa la connessione e i dettagli di velocità, capendo anche qual è la via migliore per raggiungere un certo dispositivo e ciascuno chiede ai propri vicini le loro tabelle, usandole per costruirsi una propria tabella selezionando i percorsi migliori per ogni destinazione.

Un esempio d'uso:



To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is	JI delay is	JH delay is	JK delay is		
	8	10	12	6		

Con questa tecnica, i pezzi più lenti funzionano meglio e sono in grado di recepire facilmente informazioni; tuttavia, il problema si ha con il *count-to-infinity*, quindi il fatto che la topologia tenda a configurarsi in maniera sbilanciata, creando esponenzialmente cammini irrealizzabili da un punto di vista di trasmissione dati passo/passo, in cui i router considerano rotte con costo sempre più alto per raggiungere i vicini fino potenzialmente ad infinito; ciò può essere dovuto ad un sovraccarico o ad una mancata buona organizzazione nella disposizione degli host e dell'infrastruttura.

Link state routing, QOS, Bucket (leaky, token), load shedding, IP e Basi

Parliamo invece di *link state routing*, in cui ogni nodo costruisce, sotto forma di grafo, una mappa di connettività della rete. Qui ciascun router scopre vicini e relativi indirizzi di rete, ne misura la distanza, costruisce poi un pacchetto con tutte le informazioni che viene inviato in broadcast agli altri; questo fondamentalmente è un flooding con un refresh che, per quanto sprechi più banda, garantisce che ogni nodo sia in grado di costruirsi una mappa completa della rete, calcolandosi i percorsi migliori. In ogni momento viene elaborato il percorso più breve, proprio come un algoritmo di Dijkstra (shortest path); è tuttavia un carico elaborativo non da poco mantenere questa informazione in ogni nodo, in particolare per reti di grandi dimensioni. Essendo la rete grande, si può decidere di fare in modo che ogni router conosca solo quelli della propria regione (*routing gerarchico*), oppure capendo se il pacchetto ricevuto fa parte della linea utilizzata solitamente e quindi, ipoteticamente, il percorso migliore fino a quel momento (*reverse path forwarding*). Altro parametro interessante è la *QOS (Quality of Service)*, con una serie di parametri che descrivono la rete e la sua qualità, principalmente 4.

Questi sono:

- affidabilità di trasmissione (*reliability*)
- banda (*bandwidth*)
- ritardo di trasmissione (*delay*)
- variazione nel tempo di arrivo dei pacchetti (*jitter*)

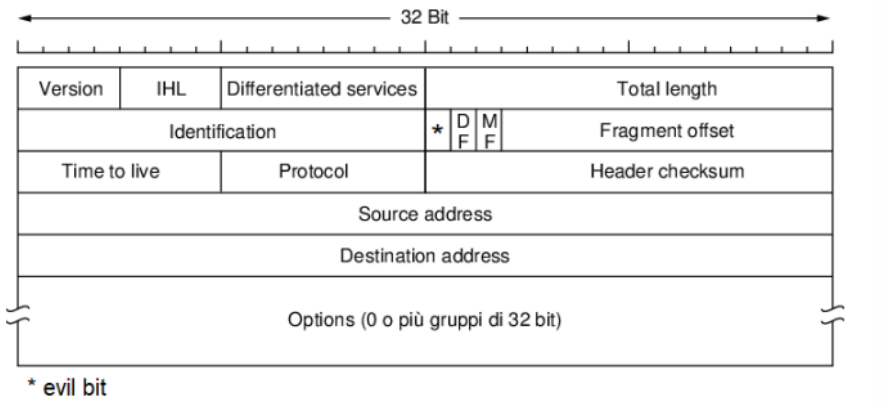
I parametri cambiano e servono a seconda del tipo di servizio (video, audio, e-mail, trasferimento file, ecc.). Per le singole misure, si adottano varie soluzioni (ad esempio per la reliability esistono già varie soluzioni da noi viste come error control oppure error correction). Le altre misure derivano da altre cause, che devono essere risolte se si vuole mantenere una buona QoS. Il problema più grosso è la *congestione*, quindi quando la capacità della linea si satura. A quel punto la performance di rete potenzialmente crolla. Aggiungere capacità ad una rete può portare ad una diminuzione della qualità (paradosso di Braess).

Una tecnica usata sono i *choke packets*, che segnala quando si verifica una congestione; quando un host riceve questo pacchetto, dimezza il suo data rate. Se la rete è grande, una richiesta di choke può metterci troppo tempo per sistemare le cose; la variante usata infatti è il *choke hop by hop*, che non agisce solo su chi sta inviando dati ma agisce anche mentre passa da un punto ad un altro della rete. Se dopo un certo tempo dalla ricezione degli ultimi choke packets non se ne hanno altri, si usa il *fading* per tornare ai parametri di velocità normale.

Per limitare il traffico si usano dei filtri che sono chiamati *buckets*, attivati nei punti critici della rete e funzionano come limitatori. Il bucket più semplice è il *leaky bucket* (“secchio che perde”), tipo di filtro che garantisce un data rate massimo costante (evitando i burst possibili nella rete che possono sovraccaricarla); tuttavia, in presenza di traffico sostenuto, converrebbe aumentare un po’ il data rate. I pacchetti in arrivo possono uscire solo se usano un token disponibile (*token buckets*); questi non sono altro che contatori dal punto di vista software/hardware e vengono generati ad ogni ciclo di clock e, in caso di scarso traffico, vengono accumulati per meglio gestire i burst. Citiamo altre tecniche che possono essere usate, quindi segnalare la congestione in piggyback nei pacchetti mandati, oppure il *load shedding*, decidendo di scartare i pacchetti più nuovi (*milk*) o quelli più vecchi (*wine*) migliorando quindi il buffering. Si può anche adattare il traffico in base ai dati trasmessi e usando QoS come parametro di bontà della linea (*traffic shaping*) oppure riconoscere preventivamente quando i router iniziano a scartare pacchetti in modo anomalo (*random early detection/RED*).

Ora si parla finalmente di *IP (Internet Protocol)*, (versione IP-V4) originariamente con il protocollo NCP (Network Control Protocol), poi inserito in TCP (Trasmission Control, da cui TCP-IP). Internet è quindi una grande rete che si forma sull’eterogeneità di parametri di trasmissione. All’interno di IP è presente un

header, di grandezza fissa a 20 bytes e una a lunghezza variabile. L'altra è la lunghezza dell'header e una checksum dell'header (*header checksum*), la lunghezza del datagramma (il campo *DF, Don't Fragment*, indica la scelta di non frammentare), sapendo se vi è stata frammentazione dei dati in più datagrammi (*More Fragment, MF*). Ogni pacchetto ha un certo tempo di vita, misurato per certo dal *Time to Live (TTL)*, quindi numero di hop massimi. Anche qui si ha un apposito campo protocollo (*Protocol*), controllo dell'errore (checksum, realizzato con somme a complemento ad uno), nonché il campo opzioni (*Options*), eventualmente utilizzato (registrazione percorso/opzioni sicurezza, ecc.). In ultimo, chiaramente, si hanno indirizzo sorgente e destinazione (*source* e *destination address*). Riassumendo:



In ultimo un commento; essendo che l'idea di IP è stata concordata a partire dal Dipartimento della Difesa statunitense (DOD) per scopi militari non ci devono essere overload e non si tracciano errori intermedi, perché le perdite possono essere ignorate e l'IP gestisce male la congestione del traffico di per sé. Il protocollo quindi si occupa principalmente di indirizzare e instradare i pacchetti tra sottoreti, assegnando a ciascun terminale di rete il proprio indirizzo univoco (indirizzo IP), definendo le modalità di individuazione del percorso di rete (capendo quindi a chi appartiene un certo indirizzo), non facendo controlli sull'affidabilità di rete o di trasmissione.

CIDR, NAT, ICMP, DHCP

Il più grosso problema di IP è fondamentalmente l'*estendibilità* (lunghezza massima 60 bytes e TTL massimo 255); il protocollo è mal disegnato e non gerarchico se non ben implementato. In ogni caso, il router può intervenire ed aumentare il TTL. Gli indirizzi sono formati da 32 bit ed è un grosso problema, perché lo spazio degli indirizzi è fisso e non estendibile. Questi sono assegnati da un'apposita entità centralizzata, dandone uno ad uno per ogni dispositivo. Se dovessimo mantenere traccia di ognuno di questi, si dovrebbero avere tabelle molto grandi.

Gli indirizzi IP sono suddivisi a blocchi per questo motivo e, per realizzare ciò, si ha una suddivisione in classi (*classful addressing*), in particolare *classe A* (8 bit rete +24 bit host), *classe B* (16+16) e *classe C* (24+8) e le taglie delle reti ne conseguono, quindi 1/2/3 bytes. Il fatto è che non essendoci una dimensione intermedia, tante aziende hanno utilizzato la classe B sprecando molti indirizzi rispetto a quelli effettivamente usati; questo comporta un limite evidente in merito all'effettiva progettazione della classe di indirizzi, puramente tecnico e distante dall'effettiva realtà. Non potendo tornare indietro alla scelta di indirizzi possibili, si usa come una possibile soluzione *CIDR (Classless InterDomain Routing)*, introducendo blocchi di lunghezza variabile. La gestione però diventa complicata in questo modo, usando le cosiddette *entry aggregate*; se dispongono di un prefisso comune posso "raggrupparle" tutte all'interno di uno stesso raggruppamento. Quindi si adotta la regola del longest matching, quindi l'entrata con il prefisso di rete più lungo ha priorità.

L' idea che salva la situazione è *NAT (Network Addresss Translation)*, traducendo e cambiando la serie di indirizzi rendendo tutto invisibile all'esterno e mostrando all'esterno un singolo indirizzo IP ottenuto dal

proprio provider internet ISP, risolvendo quindi il problema dell'esaurimento degli indirizzi. Alcuni indirizzi però sono riservati per le reti interne, non come reti normali (esempi: 10.0.0.0, 172.16.0.0, 192.168.0.0). Il problema sorge nel riconoscere i pacchetti in entrata, in quanto posseggono un IP pubblico e, tramite il payload dati, si cerca di ricostruire l'indirizzo IP privato; decisamente poco efficiente, ma utile per come è stato mal progettato IP. Per gestire eventi inaspettati in Internet si usa *ICMP (Internet Control Message Protocol)*, già incapsulato dentro IP. Questo contiene una serie di parametri che controllano qualitativamente l'andamento della rete, ad esempio destinazione irraggiungibile, redirezioni di pacchetti, gestione dei choke packets, comandi di echo, ecc.

Nello spazio di rete si ha come astrazione IP che identifica i dispositivi; nello strato fisico invece sia ha un identificativo specifico data link per ogni dispositivo (*l'indirizzo MAC*, univoco di ogni macchina e dentro la scheda di rete di ciascuna). Per far corrispondere spazio network e data-link bisogna quindi farli coesistere e collaborare. Qui interviene il *protocollo di configurazione dinamica DHCP*, che converte indirizzi MAC con indirizzi IP. Esso invia come primo pacchetto *DHCP Discovery*, suggerendo la propria configurazione e ricevendo poi un ACK dopo l'avvenuta configurazione. DHCP mantiene in ogni caso dinamicamente le informazioni dentro ogni macchina rinnovandole con il meccanismo del *leasing*, secondo il quale sia gestore che macchina interessata sanno quando scade l'informazione. Esso collabora con il protocollo *ARP (Address Resolution Protocol)* che gestisce le singole corrispondenze, mantenendo una tabella locale tra indirizzi IP e MAC. All'interno delle richieste e pacchetti inviati come ARP, per piggyback, si inserisce anche la propria configurazione, in maniera tale da dimezzare le richieste in una rete. Una conseguenza di questa implementazione è il fatto che una macchina, quando entra in una rete, invii un pacchetto di *who* per la rete dopo aver inviato *DHCP Request*, chiedendo chi possiede il proprio indirizzo IP e poi settando un collegamento con i dispositivi che lo possiedono e capisce possibili indirizzi IP duplicati.

IPV6, Livello di trasporto: TCP/UDP ed handshake

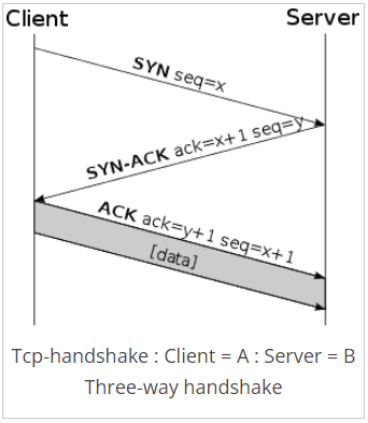
Per ovviare al problema dello spazio degli indirizzi con IPv4 si utilizza *IPv6* che utilizza 16 bytes invece che 4, rimuove il checksum per velocizzare le operazioni di indirizzamento, dando per scontato che i canali di trasmissione odierni siano già abbastanza affidabili. Altre differenze sono il campo *hop limit*, del tutto analogo a *TTL* di IPv4, *flow label*, che marca un gruppo di pacchetti con gli stessi requisiti, *next header*, intestazioni opzionali del pacchetto e restanti altri dati già presenti in IPv4 (*version*, *source/destination IP address, payload length, type of service/QOS*). I singoli blocchi di rete sono degli *Autonomous Systems/AS*, in cui ogni pezzo segue norme di rete definite dal suo gestore e seguono un loro protocollo chiamato *BGP/Border Gateway Protocol* (che significa ad esempio che un pacchetto USA non andrà nella rete della Cina, un pacchetto dati di Facebook non andrà dentro la rete di Google, ecc.). Risulta essere molto costoso, perché per ogni pacchetto memorizza tutto il percorso effettuato ed utilizza molteplici nanosecondi per inviare anche pacchetti di dati molto piccoli.

Parliamo ora del *livello di trasporto*, dove come identificativo si utilizzano le *porte* per ogni processo. Esse sono 65535 massime e per processi noti chiamate *well-known ports* (quindi per protocolli di rete come IP o ARP sono porte apposite, ad esempio porta 80 di TCP usata per HTTP) e le successive 50000 circa (fino alla 49151) sono porte proprietarie, quindi comprate/utilizzate da numerosi gestori di applicazioni, servizi, giochi e similari. All'interno di questo usiamo per esempio il protocollo *UDP-User Datagram Protocol*, che predilige la velocità al controllo; esso è composto da un *header* di 8 byte formati da una porta sorgente/destinazione, lunghezza messaggio, e una parte di dati. Esso è fatto per datagrammi/messaggi veloci ed è connection-less; interagisce con le precedenti in particolare utilizzando le porte, per poter capire destinatari e mittenti nella rete.

Il protocollo *TCP (Trasmission Control Protocol*) è stato progettato per garantire solide prestazioni (reliable) anche in presenza di molti errori e suddivide i dati degli utenti e dei processi in pezzi di dimensioni non superiore a 64 KB. È full-duplex, point-to-point e *connection-oriented*, quindi sia mittente che destinatario

devono creare dei *socket*, i quali solitamente formano una coppia indirizzo/porta permettendo al client/server la comunicazione tramite apposite interfacce applicative (API). Questo ha un header di 20 byte seguito dal payload che è formato da *source/destination port*, *sequence number* di un pacchetto all'interno del flusso dati, *ACK number*, successivo byte previsto nel flusso dati, la *window size* del mittente, la solita *checksum* a complemento ad 1 ed un *urgent pointer*, quindi i dati urgenti. In TCP si usano alcuni flag, come ad esempio *RST*, che indica che ci sono stati problemi nella connessione e serve a fare il reset, *FIN*, che, come visto, termina la connessione, in particolar modo solo uno dei due versi, *URG* e *PSH* che indicano pacchetti ad alta priorità. Per iniziare la connessione vengono usate delle primitive in un processo conosciuto come *three way handshake*, nel qual caso *LISTEN* e successivamente *ACCEPT* per accettare connessioni in ingresso.

Si attua la connessione con *CONNECT* specificando indirizzo e porta di connessione e al momento della ricezione si verifica se uno qualsiasi dei processi abbia eseguito una *LISTEN*; se nessuno l'ha eseguito viene impostato $RST = 1$ (dove *RST* indica connessione non più valida ed è uno dei flag sopra descritti). Se invece la connessione viene accettata, si imposta un corrispondente ACK per indicare l'accettazione. Nel caso in cui invece si vuole terminare la connessione si invia un segmento DR per avviare il rilascio e poi si adopera il flag *FIN*, a cui seguirà uno ACK per confermare la chiusura di una connessione. Essa viene chiusa anche in caso di N timeout senza risposta. In particolare:



Tcp-handshake : Client = A : Server = B
Three-way handshake

1. **A invia un segmento SYN a B** – il *flag SYN* è impostato a 1 e il campo *Sequence number* contiene il valore *x* che specifica l'*Initial Sequence Number* di A;
2. **B invia un segmento SYN/ACK ad A** – i *flag SYN* e *ACK* sono impostati a 1, il campo *Sequence number* contiene il valore *y* che specifica l'*Initial Sequence Number* di B e il campo *Acknowledgment number*contiene il valore *x+1* confermando la ricezione del ISN di A;
3. **A invia un segmento ACK a B** – il *flag ACK* è impostato a 1 e il campo *Acknowledgment number* contiene il valore *y+1* confermando la ricezione del ISN di B.

Livello application: DNS e tipi di attacchi di sicurezza: DOS, MITM

Nel livello *application*, invece, è il livello che si occupa della diretta interazione con l'utente. A questo scopo sono adibiti alcuni protocolli, si veda ad esempio *DNS (Domain Name System)* che converte i nomi immessi dall'utente (tipo Google al posto di www.google.com all'indirizzo IP 8.8.8.8). Esso è un sistema di denominazione gerarchico basato su domini, implementato attraverso un database distribuito tra server in varie parti del mondo, in maniera tale da raggiungere sempre un certo sito. Qui si introduce il concetto di *risoluzione* dell'indirizzo, quindi "tradurre" il nome in un indirizzo IP tramite il *resolver*, che usa il nome come parametro. Esso invia un pacchetto UDP al server locale (gruppo *name server*) che viene restituito se presente, altrimenti si ricerca ricorsivamente una corrispondenza in remoto attraverso tutta la gerarchia detta. Si parte da *root name server*, che ha informazioni sui domini di primo livello (vale a dire i siti che finiscono con *.com*, *.net*), i quali cominciano a ricercare il name server di competenza per quel dominio, fintanto che non si riesce a trovare il server giusto. L'associazione ICANN gestisce tutti questi domini, che possono essere generici (.com, .org) o per nazioni (.it, .de, .fr).

Viene sempre trasmesso un *record DNS*, con una serie di parametri che sono il *nome del dominio* di riferimento, il suo *TTL/Time to Live*, la *classe* di informazioni, il *tipo di record* (quindi se di tipo IPv4 o IPv6) e il *value* del record, che può essere il dominio o una stringa ASCII. Dal punto di vista degli attacchi possibili, DNS può subire attacchi di corruzione dei dati, per cui un attaccante ascolta la trasmissione e risponde alla

richiesta con un record DNS falso (*DNS spoofing*) oppure essere vittima di attacchi *DoS (Denial of Service*), in cui si occupa con una serie di dati inutili una rete per mandarla in sovraccarico, oppure anche gli attacchi DOS distribuiti, parlando di *DDOS (Distributed Denial of Service)*, in cui la rete diventa lenta o non disponibile, sempre per flooding di dati inutili al server (*flood HTTP*), oppure inviando degli indirizzi di rete falsi e sovraccaricando di richieste non evase una rete, sfruttando la lentezza della riconfigurazione ed accettazione dei parametri (*flood SYN/SYN attack*). Esiste una variante di DNS che aggiunge due campi al record, in particolare la chiave pubblica del dominio (*key*) e la firma del server DNS (*SIG*), il cui nome è *DNSsec.*

Questa variante si basa su un'informazione iniziale condivisa, fornita ai root server e conosciuta solo da altri 7 server/host al mondo. Un altro attacco molto noto è il *MTM – Man in the middle*, in cui introducendosi nel canale originale, i pacchetti vengono dirottati verso un altro canale che attraversa l'attaccante e viene sfruttato l'*ARP spoofing/poisoning*, che sfrutta una mancanza del protocollo ARP ovvero il poter rispondere con il proprio indirizzo MAC ad una richiesta di trasmissione anche se l'IP non è quello richiesto, dirottando il traffico nella parte sbagliata. Essi vengono gestiti dalle *CA/Certificate Authorities*, che autenticano le macchine e i siti permettendo di capire chi è sicuro e chi no.

Crittografia, cifrari a sostituzione, OTP

In merito invece alla sicurezza, si parla di *crittografia* per indicare un processo che maschera le informazioni a chiunque non sia il destinatario designato. Kerchoff ne definì i principi fondamentali, in particolare affermando che un sistema dovrebbe essere inviolabile (almeno in teoria), facile da usare e *non dovrebbe richiedere segretezza per la trasmissione, ma solo per la chiave*, che deve essere memorizzabile anche senza essere scritta. Vi è anche l'opposto, chiamato *security by obscurity*, tenendo quindi il tutto al più possibile segreto.

Vi sono due tipi di crittografia, *a codice/code*, rimpiazzando ogni parola con altre parole/simboli o *a cifrario/cipher*, trasformando carattere per carattere senza considerare la struttura. Anche qui cambiano i tipi di attacco, distinguendo tra *ciphertext only*, in cui l'attaccante conosce solo il testo cifrato, *known plaintext*, dove si conosce anche il testo in chiaro (plaintext) e *chosen plaintext*, in cui l'attaccante possiede tutte le informazioni utili eccetto la chiave. Il principio contrario a quello di Kerchoff è la *security by obscurity*, quindi non dando o diffondendo informazioni di alcun tipo in modo pubblico; paradossalmente, la diffusione di dati pubblici costringe ad un ripensamento utile della rete perlomeno da un punto di vista progettuale, alzando non di poco l'asticella di ideazione di una rete; nel caso di prima ci vorrebbero almeno 26! – 1 combinazioni di encrypting.

Citiamo quindi i vari tipi di cifrari, per esempio *i cifrari a sostituzione*, ricordando il famoso *cifrario di Cesare (Caesar cipher)*, che sostituisce le lettere k volte nell'alfabeto associando le lettere alle corrispondenti secondo uno schema segreto, in questo caso monoalfabetico. È comunque possibile risalire al messaggio originale esaminando le ripetizioni presenti nel messaggio dato, tramite le *analisi di frequenza/frequency analysis* o anche banalmente "indovinando" una parola o una frase del messaggio. Vi sono anche i *cifrari a trasposizione*, che riordinando le lettere adoperando strutture ausiliarie come una matrice, scrivendo il testo in chiaro per righe e il testo cifrato per colonna.

Per rompere la cifratura si riconosce che il testo è stato trasposto (capendo che ogni lettera rappresenta sé stessa e la frequenza rimane uguale) e si cerca di capire il numero di colonne (anche qui, ripetutamente, si riesce a trovare) poi ricostruendone l'ordine. Altro tipo spesso usato è *OTP One-Time-Pad*, che combina in XOR i bit del messaggio originale con una sequenza pseudocasuale di bit lunga quanto il messaggio, effettuando l'operazione all'inverso per decifrarlo. La chiave viene distrutta una volta usata, da cui il nome, ma richiede che questa sia memorizzata fisicamente nel destinatario, quindi può essere un problema.

P-BOX/S-BOX, ECB, Cifrari di flusso

Consideriamo anche gli *attacchi simmetrici a chiave condivisa*, quindi dove la chiave viene usata sia in cifratura che in decifratura, realizzati sia in hardware che in software, Citiamo quindi *P-BOX*, che permuta i bit facendoli passare per le linee interne, facendo la trasposizione a livello hardware e la *S-BOX*, che sostituisce una sequenza di bit in input con una di bit di output della stessa lunghezza secondo uno schema predefinito e variabile. In pratica i bit vengono impostati in un certo modo già dall'hardware della macchina, poi la P-BOX le cambia e le converte in binario. Un importante standard di sicurezza applicato dagli USA fu *DES (Data Encryption Standard)*, inizialmente pensato con chiavi a 128 poi ridotto a 56 bit. Esso lavorava con blocchi a 64 bit effettuando 19 passaggi per criptare un pacchetto. Essendo poi stato violato, fu introdotto *3DES* per tappare questa falla di sicurezza creata dalla riduzione a 56 bit precedente, facendo 3 volte la fase di encrypting (cifratura) e usando 2 chiavi da 56 bit. Citiamo a questo punto altri due cifrari a blocchi importanti, *AES*, che utilizzava blocchi da 128/256 bit e *Blowfish/Twofish*, che lavoravano su blocchi da 64 con chiavi da 32 (Blowfish) e blocchi da 128 bit con chiavi da 128 a 256 (Twofish).

Discutiamo quindi le *modalità di cifratura*, per esempio la *ECB/Electronic Code Book*, cifrando un testo suddividendolo in blocchi da 64 bit, cifrati tutti con la stessa chiave, ma problematica perché l'attaccante potrebbe spostare in maniera a noi sconosciuta i blocchi e alterarne il contenuto. Abbiamo anche la *cipher block chaining*, che collega i blocchi cifrati in vari modi, resistendo quindi in caso di spostamento e ogni blocco di testo in chiaro è messo in XOR con un vettore di inizializzazione, poi messo in XOR per la fase di decoding.

Altre ancora sono la *stream cipher*, che usa un vettore di inizializzazione per ottenere il primo blocco, poi cifrato per ottenere il secondo e così via, cifrando tutti i blocchi in flusso. Siccome accedere ai dati in maniera casuale richiederebbe di decifrarli tutti, proprio per questo il testo in chiaro (plaintext) viene messo in XOR con un vettore di inizializzazione cifrato, in maniera tale che per accedere ai blocchi successivi basti incrementare il vettore come un contatore (+1 per il blocco successivo, +1 ancora per il blocco dopo, ecc.), raggiungendo così il blocco voluto. Nel caso in corso, si usano altri modi (*cipher modes*) per fare la codifica di encrypting, in particolare considerando che l'IV (Initialization Vector) cambi ogni volta. Prendendo poi questi messaggi in XOR, è possibile combinare i messaggi plaintext (P1 xor P2), ma rimangono vittima di un possibile *keystream reuse attack* (cioè prendendo i singoli messaggi e poi facendo XOR con le chiavi) perché per *frequency analysis* il plain text viene ricostruito.

Algoritmi a chiave pubblica: hash, RSA, WEP

Concludiamo citando gli *algoritmi a chiave pubblica*, facendo arrivare la chiave al destinatario senza essere interrotti da terzi. Qui un noto esempio è *RSA*, che fa scegliere due numeri grandi "p" e "q" ai due host calcolandone il prodotto, usato dal secondo host per codificare la propria chiave. Per la decodifica bisogna avere "p" e "q" separati, l'unico a possederli sarà il mittente. Qui il livello di sicurezza cresce esponenzialmente all'aumentare di x ed è tuttora inviolato. Il problema principale poi in fase di comunicazione in una rete è la segretezza che si deve mantenere, in particolare assicurandosi della veridicità di mittente, sia nel caso di comunicazione simmetrica ed asimmetrica, parlando quindi di firma. Questa firma deve essere autenticata da un ente sicuro, in questo caso le cosiddette *CA/Certificate Authorities*, che firmano e garantiscono in merito alla veridicità di una certa fonte.

Negli algoritmi asimmetrici (*firma digitale asimmetrica con chiave pubblica)* si adopera la chiave privata del destinatario e si usa la chiave pubblica del mittente, il quale decodifica il messaggio. Verificato il mittente, successivamente, il messaggio viene decriptato dal mittente con la sua chiave privata. La cosa è pesante a livello elaborativo, quindi viene calcolata una *message digest/hash*, che comprime il messaggio originale calcolando l'hash, un insieme casuale di caratteri e stringhe molto lungo, difficile da interpretare, modificato anche dopo una piccola modifica al testo originale e sempre univoco. Negli algoritmi simmetrici

(*firma digitale simmetrica con chiave pubblica)* si calcola *l'HMAC* del messaggio (Hashed Message Authentication Code) che corrisponde all'hash del messaggio seguito dalla chiave, solitamente inviato in chiaro e usato in IPSec, versione sicura di IP. Esso viene usato per esempio nelle *VPN (Virtual Private Network)*, in cui si crea un canale sicuro di comunicazione tramite una chiave segreta condivisa, in questo caso data proprio dagli *HMAC* dei dati presenti.

Un altro standard importante è *WEP*, con un chiave da 40 bit, aggiunta ad un IV di 24 bit di lunghezza e processata in modalità stream cipher, che significa processare moltissimi pacchetti in un tasso temporale molto lungo (dato dalla lunghezza della chiave), essendo comunque passibile di furti di dati, come capitò per circa 45 milioni di account di carte di credito. Utilizza un block cipher chiamato RC4 per proteggersi dall'Electronic Book Mode, usando il mode chaining. Risulta però essere suscettibile ad attacchi di tipo dictionary. Alla fine lo standard migliore scelto è stato scelto *WPA*, acronimo di Wi-Fi® Protected Access, è una crittografia dei dati per LAN wireless. Migliora le funzionalità di protezione della WEP utilizzando il protocollo EAP (Extensible Authentication Protocol) per proteggere l'accesso alla rete e un metodo di crittografia per proteggere le trasmissioni di dati. Esso usa TKIP per assicurare la sicurezza, ormai non più molto congeniale nei dispositivi più recenti. Altri standard similari furono *WPA2/3* che utilizzavano una cipher mode basata su AES, chiamata *CCMP*. Tuttavia, nonostante le varie tecniche e processori, esistono sempre modi "brutali" (*caveman attack*), per poter rubare dati, cosa che successe ai cloud di Amazon quando un impiegato sfruttò una bruteforce sulle hash SHA presenti e in 50 minuti riuscì a rubare tutti i dati dai server/cloud presenti.

Bonus - Glossario di Marchiori

La mano del mondo – Le interferenze

Le carte da gioco/carte Matrix: Le frequenze/bande trasmissive

La bacchetta magica: Metodo di correzione degli errori

Il cucchiaio è solo un illusione

Un gusto/multigusto: Usato per descrivere la trasmissione delle singole bande di frequenza, usando una singola modulazione o più di una.

Vedere con il giusto paio di occhiali: Con le conoscenze che abbiamo riusciamo ad intuire un concetto spiegato

Caselle/campo da gioco: Bit e combinazione dei bit per il calcolo/rilevamento degli errori

Boss di livello X: Per spiegare il concetto dei bit e della loro distanza, significa che a seconda della distanza dei bit, ragioneremo ogni distanza-1

Bit di guardia/cittadini/fortino: Bit di protezioni/bit di dati/campo dati

L'intelligenza artificiale dell'aula che spegne le luci e si ricarica ad applausi

L'esame di programmazione 1 sulle schede perforate con il Filè di turno (magari c'era già)

Coperta corta per scoprire un lato del letto: Perdita della banda rispetto al data rate.

Supereroi della rete: Router, bridge, switch, ecc-