Reti e Sicurezza

Esempi di ret

Internet
Ethernet
ADSL
Computer + Perferiche

Classificazione reti

Broadcast
Multicast
Point-to-Point

Tipi di trasmissione

Esistono due tipi di trasmissione, le trasmissioni Wired e le trasmissioni Wireless
Comincieremo dalle trasmissioni Wired analizzando i tipi di cavi che consentono la comunicazione

Wired

Coppia Annodata (Twisted Pair)
- UTP: Unshielded Twisted Pair

Il twist serve a limitare le interferenze reciproche
- UTP3 e UTP5
- Bandwidth UTP3: 250Mhz
- Bandwidth UTP5: 600Mhz

Cavo coassiale
Ha una schermatura decisamente migliore dei cavi UTP
Viene utilizzato per le reti MAN (Metropolitan Area Network)
- Bandwidth: 1Ghz

Fibra Ottica
Un raggio di luce dentro a una fibra con angoli differenti (luce intrappolata da riflessione totale)
Può essere:
- A fibra singola
- A tre fibre

Connessioni tra fibre

Connettore: Si tende a perdere circa tra il 10% e il 20% del segnale
Allineatore meccanico: 10% di perdita
Fusione: Si perde "solo" l'1% del segnale

La fibra trasporta luce, ma di quale tipo?
Esistono vari tipi di luce: "normale" (lampade a incandescenza), laser, led.

Ogni tipo di luce ha le sue specifiche proprietà

Luce:	Led	Laser
Banda	Low	High
Distanza	Low	High
Durata sorgente	High	Low
Temperatura	Low	High
Costo	Low	High

Fibra ottica vs filo di rame
La fibra è molto più costosa, non è flessibile e più laboriosa da unire ma offre molta più banda e regge in modo migliore la perdita di segnale inoltre è più piccola e costa di meno ed è dielettrica, quindi non interessata dalle interferenze elettriche.

Wireless

Anche qui esistono vari modi di trasmettere i dati, la differenza sta nel decidere quale frequenze usare.

Lo spettro elettromagnetico
Nello spettro elettromagnetico vengono definite tutte le frequenze possibili, tra cui Radio, Microonde, onde infrarosse ecc...

Politiche dello spettro elettromagnetico
Ovviamente c'è bisogno di qualcosa che regoli l'assegnazione delle frequenze
I metodi principali sono:

- Beauty contest: La frequenza viene assegnata a chi la utilizzerebbe meglio
- Lotteria/Asta: La frequenza viene assegnata a chi fa l'offerta migliore
- Banda ISM (Industrial, Scientific, Medical), è considerata zona libera
- La banda ISM viene utilizzata per esempio dal Bluetooth, alcune reti 802.11, telefoni cordless, forni a microonde, telecomandi ecc...

La trasmissione Radio

LF - Low Frequency (30KHz - 300KHz)
MF - Medium Frequency (300Khz - 3 MHz)
HF - High Frequency (3MHz - 30MHz)
VHF - Very High Frequency (30MHz - 300MHz)
UHF - Ultra High Frequency (300MHz - 3GHz)

Le trasmissioni radio sono omnidirezionali

-> vantaggioso, non servono particolari allineamenti tra trasmettitore e ricevente

Onde radio a bassa frequenza
Essendo a bassa frequenza passano gli ostacoli può facilmente, ma si disperdono più in fretta
essendo a basso quantitativo energetico
Le radio AM utilizzano frequenze MF

Onde radio ad alta frequenza

A differenza delle prime le onde radio ad alta frequenza hanno più difficoltà nel passare gli ostacoli e possono essere assorbite dalla pioggia

MF vs HF
Le onde MF tendono a seguire meglio la curvatura terrestre mentre le onde HF tendono a rimbalzare sulla ionosfera, per questo vengono utilizzate nelle comunicazioni militari in quanto essendo direzionabili sono più difficili da intercettare

RDS - Radio Data System

I dati RDS sono digitali e vengono trasmessi su frequenze ancora più alte, 57KHz (la terza armonica del segnale pilota stereo (19KHz))
Il data rate è di circa 1,12Kbps
Esiste già da tempo la radio digitale
HD Radio (Hybrid Digital) in America
DAB (Digital Audio Broadcasting) e DAB+ in Europa

Perchè la radio digitale non ha preso piede?

Tre fattori principali
- I costi: il costo di upgrade ma questa c'era anche per la tv analogica.
- Il vero problema è che non si può fare un upgrade.
Nella tv l'apparato ricevente è esterno e quindi si può mettere un ricevitore in mezzo, ma non si può fare nella radio normale.
Nelle autoradio si potrebbe, ma il problema è l'accessibilità della soluzione.

Secondo fattore: Le distorsioni

Nella televisione, l'apparato ricevente non solo è esterno ma è lontano dalle interferenze domestiche, (forni a microonde, televisori, linee elettriche,
luci a led...).
Nell'autoradio ci sono problemi aggiunti creati dall'effetto Doppler.

Terzo fattore: La qualità
Per risparmiare banda i canali audio sono compressi (mpeg2)
-> problema: qui non c'è video come media primario che cattura i nostri sensi, ma audio, quindi rispetto all'analogico la qualità peggiora in maniera percettibile.

La radio stereo
Nella stereofonia ci sono due canali che devono essere trasmessi
Il problema della stereofonia è che essendo state sviluppate molto tempo dopo l'invenzione della radio, dovevano essere compatibili.
La banda mono va dai 30Hz ai 15KHz
Quindi si è reso necessario introdurre un segnale pilota (19KHz) che avverti l'apparecchio della presenza del segnale stereo
I due canali S e D vengono uniti in nuovo segnale che è la media dei due canali mono

Microonde

Sopra i 100Mhz viaggiano quasi in linea retta e quindi possono essere meglio focalizzate e viaggiare a distanze più elevate, però trasmettitore e ricevente devono essere allineati.

La rete a microonde è stata per anni uno dei migliori metodi utilizzati per le telecomunicazioni a grandi distanze

MCI = Microwave Communications Inc.

Come per le onde radio ad alta frequenza le microonde non passano bene gli edifici e sono soggette a interferenze atmosferiche

Visto il relativo affollamento ci si spinge a frequenze sempre più alte che però sono sempre più soggette alle interferenze del mondo

Trasmissioni millimetriche e ad infrarossi

- Utilizzate nei telecomandi
- Direzionali
- Non passano gli oggetti solidi
- Economiche
- Sicure

Trasmissioni luminose

Sono state in uso per molto tempo (già al tempo dei romani)
Sono state reinterpretate molto pià tardi, nel 1880 con l'invenzione del fotofono

Con l'avvento del laser sono tornate in auge.

- Fascio di luce con poca dispersione, arrivo focalizzato a buona distanza
- Soggetta a problemi atmosferici

Comunicazioni satellitari

Tipi di satelliti

Satelliti geostazionari (GEO) > 35k Km
Satelliti medio-orbitali (MEO) 5k - 15k Km
Satelliti basso-orbitali (LEO) < 5k Km

Più basso è il satellite più satelliti servono per coprire la superficie di interesse, ma il tempo di latenza è inferiore (1-7 ms)
Ovviamente più è alto il satellite più la latenza aumenta, e aumenta anche la potenza richiesta per trasmettere (la potenza va circa al quadrato dell'altitudine)

Attualmente sono in orbita circa 14000 satelliti

Bande principali

	D	V
P	0,1Ghz	1Ghz
L	2GHZ	4Ghz
C	4Ghz	8Ghz
Ku	12Ghz	18Ghz
Ka	26,5Ghz	40Ghz

Satelliti MEO

I primi satelliti dell'umanità sono MEO
1957 - Sputnik (55cm di diametro)

Dal lancio dello Sputnik l'uso dei satelliti ha avuto un boom produttivo

GPS

Il GPS usa satelliti MEO ed è costituito da circa 30 satelliti
Inizialmente in esclusiva al dipartimento della difesa statunitense, Ronald Reagan decide di donarlo al mondo dopo l'incidente del KAL007, un'aereo delle linee aeree coreane che il 1/09/1983 sconfina nello spazio areo russo e viene abbattuto dai caccia.

GLONNASS (24+4 satelliti) - Controparte russa del sistema GPS

A-GPS (GPS Assistito)
- Barometri
- Informazione sull'altitudine per regolare l'angolo di ricezione

NAVSAT
- Sistema di navigazione satellitare per le navi in uso dal '64
- Derivato dallo Sputnik

Satelliti GEO

Per molti aspetti i più appetibili
Stanno in orbita sopra l'equatore
Esiste un limite massimo di 180 satelliti oltre il quale si verrebbe a creare troppa interferenza

- Satelliti spia
- Meteo
- Televisione via satellite
- Televisione analogica e digitale on demand
- Sky Television (1989)

Satelliti LEO

IRIDIUM - Sistema di 77 satelliti, (anche sa dal progetto originale sono stati ridotti a 66)
Copre tutta la terra e spesso sono scambiati per ufo

Formano 6 collane da 11 satelliti, un satellite genera 48 celle telefoniche.

Comunicazione Intra-Satellitare

- 1 novembre 1998 viene lanciato in orbita il sistema IRIDIUM

- 13 agosto 1999, il sistema dichiara bancarotta
  -> Costi troppo elevati rispetto al GSM, mercato troppo ampio e telefono troppo ingombrante
- 2001, IRIDIUM viene riattivato

Costo telefonata

3-14 $ al minuto da fisso a IRIDIUM
1,50 $ al minuto da IRIDIUM a fisso
0,99 $ al minuto da IRIDIUM a IRIDIUM

Inoltre il sistema fa parte del Tsunami Warning System

Utilizza una banda compresa tra i 2 e i 4Khz, richiede tecniche di compressione avanzate

GLOBAL STAR

Sistema diverso da IRIDIUM
- Utilizza una serie di ripetitori bent-pipe
- Utilizza meno satelliti rispetto a IRIDIUM (52) e sono meno costosi
- Non copre tutta la terra
- Non necessita di un telefono apposito
- I satelliti hanno una durata di 7,5 anni contro i 23 di IRIDIUM
- I primi satelliti sono già stati rottamati
- 2000, lancio dell'ultimo satellite
- 2002, viene dichiarata bancarotta
- 2004, viene riattivato

I rottami spaziali

Più di 44 milioni di detriti nella fascia GEO di grandezza compresa tra 0,1 - 1 cm
Due milioni di detriti di grandezza compresa tra 1-10cm
34mila > 10 cm
Siamo intrappolati dai detriti spaziali
Occorre rottamare "ecologicamente" i satelliti
- Si potrebbe deviare l'orbita del satellite e farlo entrare nell'atmosfera, ma è costoso e pericoloso
- La soluzione migliore è di allontanarlo nelle cosiddetta orbita cimitero
  -> Richiede meno carburante ed è meno rischioso che qualcosa vada storto

IASDCC - International Agency Space Debris Coordinations Committe
- Composta da tutte le principali agenzie spaziali del mondo
Si occupa di tracciare tutti i rottami spaziali in grado di causare problemi
Il problema è che questi oggetti sono tantissimi e le misure non sono accurate

Esempio:
Abbiamo un satellite in orbita, quanti avvisi di possibile collisione riceviamo?
- 10 avvisi a settimana
Cambiare rotta è impossibile in quanto altererebbe il sistema
Quindi si spera che non accada niente

KOSMOS-2251
Satellite di comunicazione russo della rete Strela-2M
- Lanciato nel 1993
- Dismesso nel 1995

Una compagnia riceve nei suoi 400 avvisi settimanali, un avvertimento di bassissima probabilità di collisione (circa 586 metri di distanza)
Il KOSMOS si schianta contro il satellite ad una velocità di 42 120 Km/h
Il satellite pienamente operativo viene distrutto
- Il satellite faceva parte del sistema IRIDIUM 33

Storicamente 30 anni fa la comunicazione satellitare sembrava il futuro
Motivo? Nessun progresso nella comunicazione terrestre, perchè i grandi monopoli avevano bloccato tutto.
1984, gli USA introducono il Revised Telecomunication Act, frammentando le grandi aziende telefoniche incentivando la concorrenza.

Ora però è la fibra a vincere sui satelliti
Il satellite è utile per reti strategiche/militari e/o impervie/poco popolate
Le reti broadcasting utilizzano un segnale unidirezionale

Satelliti di tipo MOLNIVA
Satelliti di tipo TUNDRA

Le basi della comunicazione
- Fourier Analysis
- Segnali a banda limitata
- Data rate

Problema fondamentale: bisogna definire qual è la misura di informazione/trasmissione

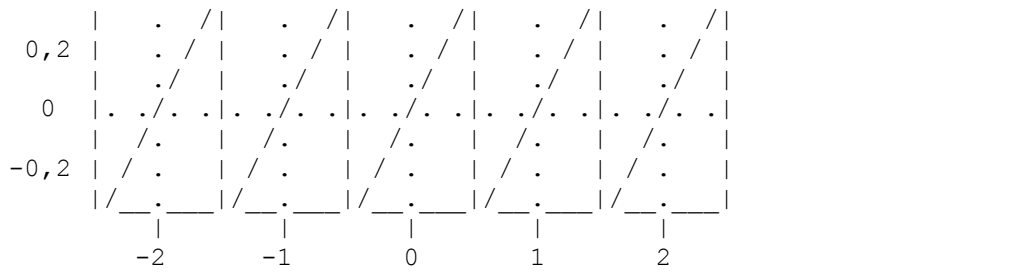
Misura fisica: la banda | Misura informativa = (Data rate: bit rate, bps, Baud rate, Bauds)
La banda: L'insieme di frequenze che trasmettiamo sul canale, si misura in Hertz
Il data rate: Bit rate, e quanti bit trasmettiamo al secondo
Baud Rate: quanti simboli trasmettiamo al secondo

Baud rate vs bit rate
Esempio: Posso trasmettere un impulso usando frequenze diverse
Il mio alfabeto è composto da 4 simboli
Ognuno di questi simboli porta due bit di informazione
Log2(v) numero di informazione esprimibile in bit con un dato alfabeto V
Bit rate = baud rate * log2(v)

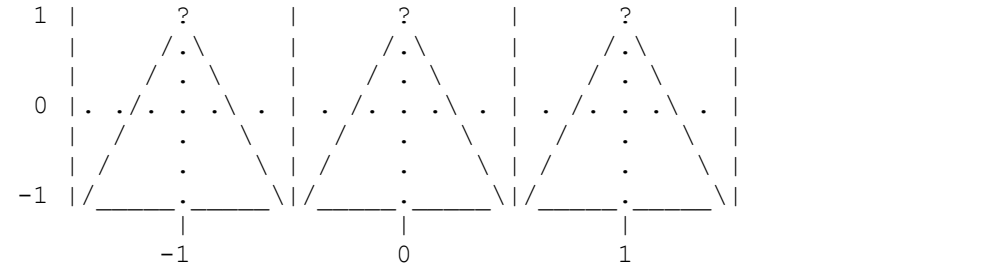
La quantità di informazione dipende dal materiale e dal segnale
Più frequenze riusciamo a trasmettere più alto sarà il data rate
Per trasmettere più frequenze abbiamo bisogno di più potenza
La potenza richiesta è sempre il quadrato della frequenza
Si fissa sempre un certo limite di banda

Trasformata di Fourier

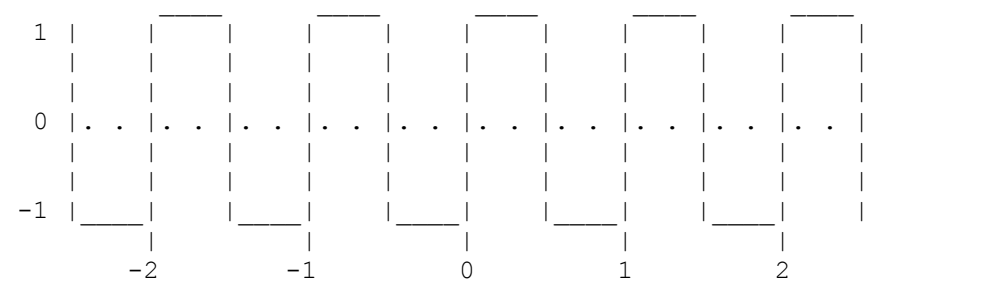
Onda a dente di sega



Onda triangolare



Onda quadra



Ogni impulso trasmesso in un mezzo che non sia il vuoto subisce una attenuazione in potenza
Attenuazione calcolata in decibel
(Log10(Potenza trasmessa/Potenza ricevuta))*10
L'attenuazione dipende dalla frequenza, quindi una forma d'onda in generale subisce attenuazioni diverse a seconda delle sue componenti nella trasformata di Fourier

La prima grande rete: Il telegrafo

Cyrus Field

Nel 1854 fu uno dei fondatori della New York, Newfoundland and London Telegraph Company, una società costituita per realizzare la posa di cavi sottomarini al di sotto dell'Oceano Atlantico.
Due anni dopo contribuì a organizzare l'Atlantic Telegraph Company, una società britannica avente gli stessi obiettivi.

Dopo essersi assicurato il finanziamento in Inghilterra e il sostegno da parte del governo americano e britannico, l'Atlantic Telegraph Company iniziò la posa del primo cavo telegrafico transatlantico, utilizzando un altopiano sottomarino poco profondo che correva tra l'Irlanda e Terranova.
Il cavo è stato ufficialmente inaugurato il 16 agosto 1858, quando la regina Vittoria inviò al presidente James Buchanan un messaggio in codice Morse. Anche se l'esultanza alla prodezza era diffusa, il cavo stesso fu di breve durata: si è rotto tre settimane dopo.

Nel 1866, Field posò un nuovo cavo transatlantico, più durevole del precedente, utilizzando la Brunel SS Great Eastern. La Great Eastern era, al momento, la più grande nave oceanica al mondo.
Il suo nuovo cavo, era in grado di avviare una comunicazione quasi istantanea attraverso l'Atlantico.
Al suo ritorno a Terranova, prese il cavo che aveva tentato di porre l'anno precedente e lo utilizzò come cavo di backup.
Nel 1871, organizzò la posa del cavo fra San Francisco e le isole Hawaii.

1902, per la prima volta il telegrafo fa il giro del mondo
1956, viene installato il TAT-1, che va a sostituire il cavo di Field
- Supporto fino a 36 canali contemporaneamente
- Il 36° canale viene utilizzato per portare 22 linee del telegrafo

1988, viene installato TAT-8
- Primo cavo in fibra ottica

SHR - Self Healing Ring

La seconda grande rete: Il sistema telefonico

Sviluppato inizialmente come un overlay sopra alla grande rete telegrafica
1876, ogni linea era realizzata point to point fra coppie di telefoni
1878, vengono creati i primi switch inserter (centralini)
Con la crescita della rete telefonica vengono creati centralini di secondo livello in supporto ai primi
Con la crescita della rete si è arrivati fino a centralini di 5° livello

PSTN - Public Switched Telephone Network
Il PSTN è gerarchico, più ci si allontana dalla fonte del segnale più si riduce la qualità del cavo - Fibra
- Cavo coassiale
- Cavi UTP

Switching
Ricordiamo la struttura gerarchica della rete telefonica PSTN
Abbiamo detto che ci sono gli switching center

Tre tecniche di switching
1) Circuit switching
2) Message switching
3) Packet switching

1) Si crea un collegamento fisico tra i due che comunicano
   Per creare un collegamento fisico serve del tempo
   - Delay iniziale
   - Collegamento fisico dedicato
2) Message switching

Invece di creare il cammino e poi iniziare la trasmissione, lanciamo direttamente il messaggio

Quando arriviamo ad uno switch, aspettiamo che ci dicano dove andare, si chiama anche storage- and-forward
Problemi di storage, messaggi troppo "grossi" causano problemi alla rete

3) Packet switching

Con il packet switching si divide il messaggio in tanti sottomessaggi di lunghezza massima prefissata

xDSL - x Digital Subscriber Line

E' una famiglia di tecnologie che fornisce trasmissione digitale di dati attraverso l'ultimo miglio della rete telefonica fissa, ovvero su doppino telefonico dalla prima centrale di commutazione fino all'utente finale e viceversa.

Deve esserci multiplexing
Splitter, separa le frequenze
Il multiplexing in frequenza si chiama FDM - Frequency Division Multiplexing
In generale si allocano vari slot di frequenze per i vari canali e poi si fa l'opportuno encoding/decoding

Il Modem

Questo dispositivo permette la MOdulazione e la DEModulazione dei segnali contenenti informazione; dal nome di queste due funzioni principali il dispositivo prende il nome di MODEM. In altre parole, sequenze di bit vengono ricodificate come segnali elettrici.

Modem Analogico/Standard

Il primo tipo di modem era molto comune fino alla fine degli anni '90, che ha segnato la diffusione di internet. Era una scheda di espansione interna o un'unità esterna che connessa al computer e alla linea telefonica consentiva di accedere ad internet attraverso la linea telefonica base occupandola costantemente a bassa velocità di connessione.

Modem ISDN

A differenza dei modem precedenti non usano frequenze sul doppino telefonico ma veri e propri segnali digitali discreti.
Sono utilizzabili solo se in possesso della linea telefonica specifica per questo tipo di connessione.
Hanno le medesime funzioni di quelli analogici con la differenza che sono più veloci e sfruttano una linea dati digitale, con una velocità tra i 64 kbit/s e 128 kbit/s, occupando entrambi i canali ISDN.

ADSL - Asymmetric DSL

- Asimmetrica perchè analogamente allo standard per il modem telefonico v.90 e v.92 da più spazio al download e minore all'upload

0-4KHz banda dedicata alla voce
25,875-138KHz banda dedicata all'upload
138-1104KHz banda dedicata al download
il modo preciso in cui si usa l'FDM per gli standard ADSL è il cosiddetto Discrete MultiTone (DMT)
Si spezza la banda in canali minori

256 canali da 4312.5 Hz ciascuno
1 per la voce
5 vuoti
32 upstream
Il resto downstream

Canali indipendenti
Indipendenti significa che ogni canale viene trattato come una connessione telefonica a se stante
Si usa una specie di v.34 e come nel caso di una singola connessione modem, c'è controllo costante sulla qualità della trasmissione
Ogni canale può essere rallentato/accelerato indipendentemente
I canali vuoti sono utilizzati per evitare interferenze tra voce e dati

Standard ADSL
ADSL Lite
1.5Mbit/s downstream, 0.5Mbit/s upstream
ADSL
8Mbit/s downstream, 1Mbit/s upstream
ADSL2
12Mbit/s d, 1Mbit/s u
ADSL2(Annex J):
12Mbit/s d, 3.5Mbit/s u
ADSL2+
24Mbit/s d, 1Mbit/s u
ADSL2+:(Annex M)
28Mbit/s d, 3,5Mbit/s u
le ADSL2+ usano una banda doppia, cioè 2.2MHz invece di 1.1MHz
- hanno bisogno di una linea particolarmente buona varianti all-digital, in cui si guadagnano 256Kbps in upstream rinunciando alla parte voce

FDM si usa anche per il telefono e internet

Multiplexing nei telefoni

Una soluzione possibile è appunto l'FDM
12 canali voce (4KHz) vanno da 80-160KHz
Esempio: Supergroup(5x)/Mastergroup(10x)
Un mastergroup di terzo livello può reggere 600 conversazioni contemporaneamente mentre con altri standard si arriva a 230k canali voce
Nel caso della fibra ottica la FDM si chiama WDM (Wavelength Division Multiplexing)
Gli encoder ed i decoder dovendo trattare la luce possono essere costruiti a tecnologia completamente passiva e quindi danno componenti molto più affidabili e duraturi

Dove si arriva usando fibra WDM
1990 - 8 canali da 2.5Gbps
19998 - 40 canali da 2.5Gbps
2001 - 96 canali da 10Gbps
2007 - 124 canali da 50Gbps
esempio:
2001 96x10=960Gbps si possono trasmettere 160 film al secondo
2007 - 124x96 = 6200Gbps 1033 film al secondo

Non c'è solo l'FDM
Un altro multiplex è quello temporale (TDM)

24 canali voce(1.544 Mbps di dati)
Viene usata nella infrastruttura di internet
Viene usata nei cd audio
WAV, PCM Lineare, AIFF

La telefonia mobile: 0G, 1G, 2G, 3G, 4G

Premessa di marketing
Uno dei pochi ambiti in cui l'UE e l'Italia se la contendono con gli USA
- L'Europa dopo i primi tentativi si rende conto che serve uno standard europeo
- Gli USA per eccesso di liberismo e libera concorrenza hanno lasciato che ci fossero standard molteplici creando così reti incompatibili

Principio della conoscenza tariffaria

La telefonia mobile costa di più, ma finora non è stato un problema perchè i numeri dei cellulari sono diversi dai fissi, quindi quando si effettua una chiamata si conosce in anticipo il costo
Mentre negli USA non c'è distinzione e quindi si è deciso di far pagare il surplus al possessore del cellulare
-> rallentamento del mercato
Inoltre esiste una grande competizione per quanto riguarda la telefonia fissa, e ciò fa sì che i prezzi siano relativamente bassi
In Europa invece (specialmente in Italia) accade il contrario e quindi i mercati alternativi (mobile) sono fioriti molto più velocemente

La telefonia si distingue in varie generazioni
0G-1G - reti analogiche
2G-4G - reti digitali

Tutta la telefonia mobile si basa su un problema fondamentale: la suddivisione del territorio
Come gestire l'infrastruttura fissa per permettere il mobile?
- Si usa la switching center

0G (1950) Deriva dalle trasmissioni radio, noti come PPT - Push To Talk
- Un solo canale per ricevere e trasmettere
-> Half Duplex
Il PTT è stato reintrodotto in alcuni cellulari di ultima generazione (es. moto talk)
Essenzialmente il cellulare può funzionare come un walkie-talkie

1960, il sistema IMTS - Improved Mobile Telephone System
- passa a due frequenze
- aumenta il livello di privacy, in quanto non è più possibile sentire le comunicazioni altrui
- vengono utilizzati super trasmettitori ad altissima frequenza
- per evitare interferenze le celle coprono centinaia di chilometri
- 23 canali nella banda 100-450 MHz
-> troppo limitativo
- Ancora in uso in certe zone remote (Canada)

1G (1982)
I Bell Labs introducono l'AMPS - Advanced Mobile Phone System
- Conosciuto in Europa come TACS - Total Access Communication System
- Ora le celle sono molto più piccole (10-20Km)

- La capacità di gestione è aumentata e diminuisce la potenza richiesta per la trasmissione
- Problema delle interferenze tra le celle
- Si usano frequenze diverse tra celle vicine

Quante frequenze usiamo?
Più frequenze usiamo per separare le celle meno banda avremo a disposizione

Viene usato il teorema dei 4 colori
Il teorema afferma che data una superficie piana divisa in regioni connesse, come ad esempio una carta geografica politica (o in celle come nel nostro caso), sono sufficienti quattro colori per colorare ogni regione facendo in modo che regioni adiacenti non abbiano lo stesso colore.

Quindi basta usare 4 frequenze diverse

Ma in pratica come sistemiamo le celle?
Supponiamo di avere un buon controllo del territorio
Come interagisce la rete wireless con la strada?
In città si sfasano le celle cercando di usare una matrice esagonale che permette di ridurre le interferenze.
In questo caso quanti colori servono? Dipende!

Quando il sistema va in overload si suddividono le celle principali in microcelle
Il numero di colori cambia in base alle infrastrutture e alla densità d'uso e delle necessità di espansione

Tipicamente si usano dai 3 ai 7 colori

Un cellulare è sempre connesso ad ina sola cella finchè non si sposta.
ma quando si sposta cosa accade?

Handoff

L'Handoff classico 1G
Quando il segnale è troppo debole lo switching center office chiede alle celle vicine quanta potenza ricevono dal telefono
Nell' Hard Handoff la vecchia stazione molla il cellulare che si aggancia a quella nuova
- C'è del lag dovuto al cambiamento (circa 0,3 s) e in casi rari la linea cade
Nel Soft Handoff, la nuova cella acquisisce il cellulare prima che la cella vecchia lo molli
- Il problema è che il cellulare deve essere in grado di supportare due frequenze diverse contemporaneamente

Caratteristiche principali del 1G
- Ogni cella gestisce molti utenti, quindi occorre un multiplex (FDM)
- Ogni cella è in grado di gestire 832 canali Full Duplex (due canali simplex)
- Alcuni canali sono riservati per il controllo e la gestione, mentre altri possono essere usati per lo smistamento delle frequenze tra celle

Ogni cellulare ha un numero seriale di 32 bit e un numero telefonico di 10 cifre (34 bit)

Ogni 15 minuti circa il cellulare manda un segnale in broadcast contenente i suoi dati per registrarsi alla cella più vicina

Quando si chiama si usa il canale apposito (condiviso!) per attivare la richiesta
In ricezione invece c'è un canale apposito di paging che i cellulari controllano per sapere se ci sono chiamate in arrivo

2G

Anche nel 2G non si è arrivato ad uno standard mondiale

D-AMPS - standard statunitense (retrocompatibile con AMPS)
PDC - standard giapponese, funziona con la stessa tecnologia di D-AMPS

D-AMPS usa tutte le frequenze di AMPS (850MHz)
- Trasmette solo in digitale
- Usa frequenze aggiuntive per aumentare la banda (1850-1990MHz)
- In questa banda la lunghezza d'onda diminuisce quindi bastano antenne piccole
- Trasmettendo in digitale si usa la compressione della voce
- Dai classici 56kbps si arriva a 8kbps (si arriva anche a 4kbps)
- Sei volte gli utenti standard di AMPS

Esempio di compressione: Delta Modulation
Le comunicazioni compresse vengono gestite in multiplexing usando un classico TDM
La qualità del suono è peggiorata da 1G a 2G

Gestione dell'handoff
In AMPS è lo switching center a gestire tutto, ma potenziale collo di bottiglia
In D-AMPS l'onere viene lasciato ai singoli cellulari
Il cellulare è in "patnership" con la cella, periodicamente tiene sotto controllo la potenza del segnale
Quando il segnale è basso, è il cellulare a "protestare" con la base, quindi la cella lo molla
A quel punto il cellulare cerca la cella con il segnale più potente
MAHO - Mobile Assited HandOff)
Il carico sul cellulare è minimo perchè si sfruttano i tempi morti dovuti al multiplexing temporale per misurare la potenza del segnale

GSM - Global System for Mobile Communications, nata in Europa ora utilizzata in tutto il mondo

Simile a D-AMPS in quanto usa l'FDM temporale
Differenza: I canali GSM sono molto più ampi di quelli AMPS (200KHz vs 30KHz)
Tengono più utenti ed hanno un data rate per utente molto più ampio
- 124 canali, ognuno con 8 slot TDFM
- Ogni canale GSM gestisce un data rate di 270Kbps, che per 8 utenti fanno 33,854 kbps a testa
- Meno overhead
  -> 24,7 kbps, e dopo la correzione degli errori 13 kbps

Struttura delle celle GSM
Quattro tipi di celle: Macro, Micro, Pico e Umbrella

- Macro: 35 Km di raggio, sopraelevate rispetto agli edifici
- Micro: Cella più piccola posta sopra i tetti degli edifici
- Pico: Molto piccola, usata per aree densamente popolate (utilizzate sopratutto indoor)
- Umbrella: usata per corpire i buchi tra le celle principali

SIM - Subscriber identity Module
- Varie taglie, dai 4kb fino ai 512kb
- Contiene varie informazioni, di cui le due più importanti sono la IMSI e la Ki

IMSI - International Mobile Subscriber identity (Identificazione della SIM)
Ki - chiave di autenticazione (autenticazione crittografica a chiave condivisa)

Collegamento crypto GSM
- Il cellulare manda L'IMSi della SIM all'operatore
- L'operatore genera un numero casuale e lo manda al cellulare
- Il cellulare firma il numero con la Ki e lo rimanda all'operatore
- L'operatore nel suo DB ha registrato l'IMSI e la Ki associata.
- L'operatore defirma con la ki il numero casuale e controlla che il numero sia lo stesso

CDMA - Code Division Multiple Access
Mentre D-AMPS e GSM sono abbastanza simili, CDMA funziona in modo diverso, infatti non usa ne FDM ne TDM
CDMA è stato introdotto solo recentemente per via di un problema che non lo rendeva fattibile, infatti CDMA "assume che ogni cellulare parli il suo linguaggio con lo stesso volume di tutti gli altri"

E' un protocollo di accesso multiplo a canale condiviso, tecnicamente CDMA moltiplica l'informazione binaria generata da una sorgente per un'opportuna parola codice detta chip; la sequenza in uscita sarà successivamente modulata e infine trasmessa sul canale.
In ricezione, il segnale ricevuto sarà costituito dalla somma vettoriale (cioè compresa di modulo e fase) di tutti i segnali trasmessi dalle singole sorgenti. Se i chip delle sorgenti sono ortogonali tra loro, l'estrazione dell'informazione associata a ciascuna sorgente potrà essere fatta in maniera del tutto complementare.
Il risultato di quest'operazione permette di ottenere un segnale che è dato dalla somma di un segnale di ampiezza dominante, che è il segnale proveniente dalla sorgente, e di un segnale ad ampiezza minore, costituito dal rumore e dai segnali delle altre sorgenti.

Relazione del cellulare con le celle
- La base trasmette ad una potenza fissa, nota al cellulare
- Dalla potenza del segnale che riceve, il cellulare può calcolare quanto è distante dalla base
- Quindi cerca di trasmettere ad una potenza opportuna

Rispetto a D-AMPS e GSM, CDMA è in grado di gestire le celle senza sprecare frequenza, infatti CDMA sfrutta al meglio la caratteristica del traffico e dell'intermittenza
La sua efficienza è dovuta al fatto che ciascun canale utilizza l'intera banda disponibile per tutto il tempo che desidera

Tra 2G e 3G: GPRS - General Packet Radio Service
E' classificato come 2.5G
E' un overlay del 2G che permette la gestione del traffico a pacchetti

GSM è costruito essenzialmente per trattare voce dato che viene riservato un intero canale per la comunicazione
Quindi navigando in internet si sprecava la banda del canale, dato che durante la navigazione sono frequenti i tempi morti in cui non si trasmette niente,
Inoltre la tariffa era a tempo e non a traffico
Quindi il GPRS permette, analogamente al concetto di switch di cui abbiamo parlato, la navigazione a pacchetti con tutti i vantaggi conseguenti

- Non si spreca banda
- Non serve più un canale dedicato
- Si possono usare tariffe a traffico dati
- Alloca dinamicamente i canali Internet e quelli voce a seconda delle richieste del traffico

Tipo di cellulari

classe C: possono connettersi come GSM o come GPRS, l'utente deve settare a mano quale comunicazione usare
Classe B: Si può connettere o come GSM o come GPRS automaticamente
Pseudo classe A: Può usare contemporaneamente GSM e GPRS usando la stessa frequenza (però la rete lo deve supportare)
Classe A: Usa contemporaneamente GSM e GPRS usando due frequenze diverse

EDGE - Enhanced Data rate for GSM Evolution
Classificato come 2.75G
Oltre alla modulazione di frequenza usa la modulazione di fase

- Ne esistono varie versioni
- velocità variabile: da 64kbps a 236Kbps (o EDGE Evolution)

3G

Rispetto al 2G ha più data rate e può supportare più utenti

- Nasce per gestire la banda video
- usa tipicamente bande di frequenza più larghe
- Due standard principali: WCDMA e CDMA2000

WCDMA: Wideband CDMA

- E' conosciuto in Europa come UMTS - Universal Mobile Telecomunication System
- Usa una banda di 5MHz
- Il data rate tipico è di 384 kbps

CDMA2000

- Chiesto dagli USA (a differenza di EU e Giappone)
- Usa una banda di 1,25MHz
- Ha un data rate di 144kbps

HSDPA - High Speed Downlink Packet Access
Classificato come 3.5G
Evoluzione dell'UMTS (usa CDMA e QAM)
Esistono varie versioni di HSDPA supportati dagli operatori e dalle celle
- 1,8; 3,6; 7,2; 14,4 Mbps in Download
- Da 384kbps ai 2 Mbps in Upload

Attualmente in Italia in 3G si viaggia sui 7,2Mbps

HSUPA: High Speed Uplink Packet Access
Classificato 3.75G
Evoluzione dell'HSPDA con download massimo (teorico) a 5,6 Mbps
(la nuova versione arriva a 11,5 Mbps)

Come è possibile che il 3.5G superi il 3.75G?

HSDPA+ - Evolved HSDPA
E' stato sviluppato in contemporanea di HSUPA, ma anche se considerato come 3.5G è riuscito a superare in prestazioni HSUPA
- Velocità massima in download teorica: 28Mbps
- velocità massima in upload teorica: 11Mbps

4G
HSOPA - High Speed OFDM Packet Access
- Conosciuto anche come LTE
- Bande variabili, da 1,25Mhz a 20Mhz
- Download fino a 1,2 Gbps e in Upload fino a 600Mhz

Le prime versioni furono commercializzate nel 2009 in Svezia e Finlandia

Come fa ad andare così veloce?
- Usa varie tecniche combinate tra cui: FDM, TDM e altre tecniche commutate da trasmissioni satellitari.
- Usa più banda

La banda 4G LTE interferisce con la banda del digitale terrestre, quindi adesso occorrono filtri aggiuntivi ad ogni antenna TV

Lo strato Data Link

- Fornisce un'interfaccia allo strato di rete sovrastante (network layer)
- Si occupa degli errori di trasmissione (error control)
- Regola il flusso di dati a seconda delle capacità della rete e del ricevente (flow control)

Esistono vari modi in cui questo servizio può essere forte.

Unacknowledged connectionless

In questo servizio i pacchetti vengono inviati senza aspettare conferma di una eventuale ricevuti(unacknowledged)e tantomeno senza stabilire una connessione dedicata (connectionless).
Utile ad esempio per voce/streaming media, o quando il canale è molto affidabile.

Acknowledeged connectionless

Altro servizio analogo, è come il precedente solo che in questo caso ho la conferma della ricevuta dei pacchetti.

Acknowledged connection-oriented

Una connessione dedicata con ricevuta di ritorno (servizio di lusso)

La codifica: Framing

L'approccio classico è di prendere dei pacchetti dati dallo strato superiore e codificarli in appositi frames

[packet] -> [Header|Payload Field|Trailer]

Payload (contenuto del pacchetto)

Uno dei problemi della trasmissione dei frame è accorgersi dove inizia e dove finisce un frame.
Si potrebbe usare la sincronizzazione degli orologi, ma è impraticabile per vari motivi

Metodo del Character count

Semplicemente mettiamo nell'header l'informazione del numero di caratteri che costituiscono il corpo dati.
Simile a quello che fanno certi linguaggi di programmazione internamente.

ES:

```
[5]|0|1|2|3|4|[5]|5|6|7|8|9|[8]|0|1|2|3|4|5|6|7|
   |---------|---|---------|---|---------------|
   | frame 1 |   | frame 2 |   |    frame 3    |
   |5 charac.|   |5 charac.|   |  8 character  |
```

Problemi

```
                !->errore
[5]|0|1|2|3|4|[7]|5|6|7|8|9|[5]|0|1|2|3|4|
   |---------|---|---------|---|---------|
   | frame 1 |   | frame 2 |   | frame 3 |
   |         |   | (wrong) |   |         |
```

Il character count è stato uno dei primi metodi (derivati dai primi linguaggi di programmazione, ma l'ambiente delle reti dove ci possono essere errori molto più frequentemente che non su un desktop ha imposto di cambiare metodo.

Metodo dei Flag Bytes

Il problema del Character count è che perdiamo tutti i dati in caso di errore di sincronizzazione
->usiamo un byte speciale per segnalare (flag byte) l'inizio e la fine di ogni frame

Byte stuffing
Usato per fare l'escaping, in modo da trasmettere qualunque messaggio

[FLAG | Header | Payload Field | Trailer | FLAG]

[A][FLAG][B] ---> [A][ESC][FLAG][B]
[A][ESC][B] ----> [A][ESC][ESC][B]

Un problema analogo con i linguaggi di programmazione si è avuto quando si è passati dai caratteri ASCII ai caratteri più globali (UNICODE)
Usare bytes o in ogni caso grandezze fisse prima o poi non và più bene

Il bit stuffing
E' analogo al byte stuffing, stavolta a livello di bit

Ad esempio, si prende come "flag" 01111110
Escaping: Se lo stream di bits ha cinque 1 di fila allora dopo il quinto bit viene inserito uno 0

a) 0110111111111111111111110010
b) 01101111101111101111101111110010
| | |
bits stuffing

Se tecniche come quelle che abbiamo visto ci permettono di sapere come identificare gli errori in un frame nella rete, vogliamo sapere come comportarci in caso di errori.

ERROR CONTROL

L' error detection ci permette di controllare se ci sono stati errori durante la trasmissione; nel caso richiediamo una ritrasmissione.
Error detection è utile quando il canale è affidabile o gli errori non sono critici, ma spesso serve anche error correction, cioè cercare di correggere gli errori.

Error Correction (1)

Nelle reti quando il canale non è affidabile, ritrasmettere troppi pacchetti diventerebbe oneroso, facendo direttamente correzioni degli errori sui pacchetti si evita la ritrasmissione.

Error control: distanze
Prima di procedere ci occorre una misura del "danno" che un errore può fare; gli errori meno gravi sono quelli che danneggiano un solo bit nel messaggio o che ne danneggiano 2 e via dicendo.

La distanza

Diciamo allora che due messaggi di lunghezza uguale sono distanti 1 se sono diversi solo per 1 bit, distanti 2 se sono diversi per 2 bit

Si chiama distanza di Hamming

Tecniche di Error Control

Sono più o meno potenti a seconda di quanta distanza di Hamming riescono a sopportare nei messaggi (ovvero la gravità massima dell'errore che riescono a sopportare)

Error Detection

Dovremmo avere informazione extra rispetto ai dati che ci permetta stavolta di individuare l'errore
Avremo quindi un encoding (aggiungiamo una protezione ai dati) e poi una fase di decoding dove ci liberiamo della protezione dei dati.
L'error detection quindi sarà nella fase di decoding, dove il ricevente controlla se ci sono stati errori.

Modo semplice? Basta ad esempio verificare se il messaggio arrivato non può essere stato creato con l'encoding (Messaggio illegale)

Condizione sufficiente ma non necessaria.

Potrebbero esserci talmente tanti errori da trasformare un messaggio "legale" in un altro messaggio "legale", quindi, quand'è che siamo sicuri di trovare l'errore?

Dobbiamo limitare la potenza dell'errore!

Dato un certo modo di proteggerci dagli errori (encoding/decoding) vorremmo sapere: qual è il massimo errore che riusciamo a trovare?
->(la distanza minima tra i messaggi legali) - 1

Una semplice tecnica è quella del bit di parità (parity bit)
Ogni tot bit (n), inseriamo uno 0 o un 1 a seconda che la somma degli n bit precedenti sia pari o dispari
n = 2 : 01 -> 01(1) 10-> 10(1)

Quanto è potente il parity bit "n"?
Facile vedere qualunque sia l'n: Presi due messaggi diversi, i loro messaggi codificati sono a minima distanza 2

Il fattore n

Possiamo avere un errore (un bit "flippato") ogni n+1 bits -> possiamo identificare un error rate di 1/(n+1)
Il codice migliore di controllo si ha con n=1: in quel caso, possiamo identificare un error rate del 50%

Quanto stiamo sprecando per controllo dell'errore? 1/(n+1)

Esempio con n = 3 -> 1/4 del datarate sprecato
Ad ogni modo il parity bit ha potenza 1: significa che non appena c'è un errore di potenza due il codice fallisce.

I Repetitions Code

Nel repetition code ogni bit viene ripetuto N volte

R(3) 110 -> 111111000
010 -> 000111000

E' facile vedere che per ogni due messaggi diversi, i loro messaggi codificati con Rn sono distanti almeno N
Quindi con Rn possiamo fare error detection fino a potenza N-1
-> possiamo alzare la potenza quanto vogliamo.

Error Correction (2)

Significa che dato un valore sbagliato, possiamo farlo tornare al valore corretto

Repetition N ci dà una affidabilità a K errori tale che K sia meno della metà di N

Ad esempio abbiamo che con R3 il K massimo è (3-1)/2=1
-> R3 oltre a fare error detection a potenza, è un codice di error-correction
Però la banda diventa 1/N

Error detection viene usato un diversi ambiti, come il codice IMEI del telefono, o il codice ISBN dei libri, e nelle carte di credito
Altri esempi: codici a barre (UPC - Universal Product Code)
(somma delle cifre dispari) x 3 + (somma delle cifre pari) == Multiplo di 10

ISBN13 - C'è un nuovo tipo di codici ISBN dal 2007: ISB13 (13 cifre invece di 10)
Usa lo stesso codice di error detection usato dai codici a barre

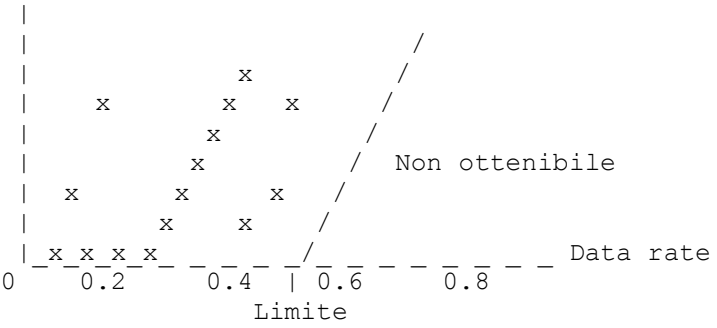
Torniamo ai codici error detection

Abbiamo visto i codici a ripetizione, ma sprecano molta banda

Limite Error rate vs Data rate

Man mano che vogliamo ridurre l'error rate a zero, anche il data rate diventa zero
Conosciuto come No Free Lunch Principle

Shannon ha dimostrato che invece, anche se sembra impossibile, la curva dei codici riesce ad avere un data rate positivo, anche se si vuole un tasso degli
errori sempre più piccolo.



Il teorema di Shannon

Hr(x) = xlog(1/x) + 1(1-x)*log(1/1-x)

Conseguenza

Dischi rigidi "scassoni" con un tasso di errore del 10%
Però vogliamo un tasso di errore decente ad esempio un errore ogni 10^15 letture
Usando al meglio le nostre tecnologie, dovremmo usare 60 dischi

Shannon però ci dice che in realtà c'è un codice per cui da un error rate di 0,1 ad uno di 10^15 bastano (usando la formula dell'entropia) 2 dischi

Quali codici si avvicinano

LDPC - Low Density Parity Check
- Usato nella TV digitale e nel WIMAX
- Potenza: molto vicino al limite di Shannon
- Il decoding è NP-completo

I codici di Hamming

Sono una famiglia di codici famosissima, fanno parte dei cosiddetti codici lineari perchè le loro operazioni si possono esprimere tramite combinazioni lineari

Il codice Hamming(7,4)

Un codice Hamming(X,Y) significa che codifica Y bits di dati usandone X in totale, quindi con X-Y bit extra.
I bit extra si dividono equamente la parità dei bit dati
L'encoding si può scrivere linearmente usando una matrice, che si chiama matrice generatrice

```
  |1 0 0 0 |0 1 1|
G=|0 1 0 0 |1 0 1|
  |0 0 1 0 |1 1 0|
  |0 0 0 1 |1 1 1|
  ---------|------
      |       |
  Bit dati   Bit di
            protezione
```

Che proprietà hanno questi codici?
Fanno parte della famiglia più grande, come detto, dei codici lineari (sottospazi lineari di uno spazio vettoriale su campi finiti)
Un codice Hamming (X,Y) e che ha distanza minima di Hamming Z viene anche detto (X,Y,Z)

Diciamo che il peso (weight) di un messaggio binario è il numero di 1 che ha.

Teorema del peso minimo

Se il peso minimo dei vettori della matrice generatore X per Y è d, allora ogni coppia di messaggi codificati dista almeno d.
Il codice corrispondente è un codice error detecting di potenza d-1 e error correcting di potenza (d-1)/2

Quanto abbiamo guadagnato: l'altro codice a potenza 1 aveva un data rate di 1/3 = 0.33
Qui invece il data rate è di 4/7 = 0.57

I codici lineari sono molto belli perchè l'error correction si fa in un modo molto più veloce (usando la cosiddettà sindrome)

Hamming superiori - Hamming ibridi (8,4)

Hamming (8,4), molto semplice si aggiunge un altro parity bit
La distanza minima è aumentata a 4, quindi error-detection 3 ed error-correcting 1
Il data rate però è sceso, il data rate è del 50%

Hamming superiori - Hamming (11,7)

Distanza minima 5 -> (11,7,5)
Quindi corregge fino a 2 errori
Data rate 7/11 = 0.637

Bisogna notare che i codici di Hamming sono una generalizzazione dei codici che abbiamo vista prima (parity bit, repetition)

Problemi pratici con gli errori
Se non sappiamo nulla degli errori non c'è nulla che possiamo fare
Se invece conosciamo delle proprietà degli errori, cioè se gli errori hanno una struttura allora le cose cambiano.

I burst

Nella pratica, in molti casi gli errori non sono del tutto casuali, ma occorrono in burst ("esplosioni" ravvicinate)
Questo crea problemi mostruosi ad ogni codice error-correcting
Potremmo tenerne conto e anzi sfruttare la loro struttura

Nel nostro computer, anche nel suo piccolo, avviene la comunicazione dei dati, dal processore alla ram al disco rigido etc etc

Esempio:

Consideriamo ad esempio la RAM, ci serve una unità di misura sufficientemente grande, il quadrilione (10^15)

Dati del nostro computer
RAM che va a 1600MHz
1Gb di RAM
Supponiamo, la probabilità di un errore nella RAM sia bassissima: uno ogni cento quadrilioni
Ogni quanto in media ci sarà un errore? Un errore ogni 0.08 secondi!

E un doppio errore?
Senza tenere conto dei burst, un doppio errore circa ogni 8 quadrilioni di secondi
Scelta di design: se vogliamo migliorare il nostro sistema, dovremmo tenere ben conto degli errori di potenza 1, e possiamo tralasciare quelli di potenza 2

E per i dischi rigidi?
La situazione non cambia molto: l'access rate è di certo minore della RAM, ma non di molti ordini di grandezza, il che significa che ci saranno errori in termini di anni invece che di secondi

RAID

RAID 0 - Overlay, striping dei dati
RAID 1 - Repetition 2
RAID 2 - Overlay a livello di bit con Hamming Code
RAID 3 - Overlay con parity bit
RAID 4 - Parity bit
RAID 5 - Parity bit distribuito

La RAM usa un codice di error correction, variante di quello di Hamming, di tipo (72,64)

Se vi siete mai chiesti perchè la RAM di nuova generazione non andavano più veloci delle vecchie:
hanno una penalità dell'11% (il data rate è l'89%)

SECDED - Single Error Correction, Double Error Detection

Attenzione che la scheda madre deve supportare la funzionalità
Anche se nella scheda madre è scritto che "supporta le ECC" significa che potete usarle, ma non necessariamente che ci sia error correction

Esempio paradossale: le schede grafiche di fascia alta possono non supportare SECDED

Altro esempio:
Le schedine di memoria usano ECC con Hamming

Le missioni Voyager (1977)
Funzionano ancora, sono "solo" a 14 bilioni di chilometri

In tutte le trasmissioni spaziali anche le più recenti si fa uso di codici di correzione

Il codice Reed-Solomon

E' basato su aritmetica polinomiale
Un codice RS(X,Y) al solito codifica Y parole usandone X, può correggere (X-Y)/2 errori!
Si usa anche in tantissimi altri ambiti, come i CD dove si usa il codice a correzione di Reed-Solomon, in una sua applicazione a due canali.
Nei DVD
Nelle ADSL
Nei Blu Ray
Nel WiMAX
Nella TV Digitale con un RS(204,188)

Il passo oltre Hamming

Uno dei grandi aspetti del codice di Reed-Salomon è che va oltre Hamming; invece che singoli bit come unità di base, può considerare gruppi di bytes.
Permette ad esempio nei lettori CD di supportare errori di burst lunghi fino a 4000 bits.

Le Erasures

Sono errori ma non nel senso di corruzione del dato, ma proprio di cancellazione del dato: il dato non c'è più.
RS(X,Y) riesce a correggere fino a X-Y erasures, quindi il doppio degli errori.
In generale può correggere contemporaneamente errori ed erasures: gli errori contano doppio delle erasures, quindi 2*errori + erasures < X - Y

Reed-Solomon semplificato

CRC - Cyclic Redundancy Check

Un codice simile al funzionamento di Reed-Solomon che fa solo error detection e non correction
Basato sull'architettura polinomiale a 2
Possiamo vedere ogni numero binario come il corrispondente polinomiale, considerando ogni bit come il coefficiente di potenza sempre crescente

1011 -> x^3 + x + 1
11010 -> x^4 + x^3 + x

Come funziona l'aritmetica in GF(2)[]
- L'addizione è come fare lo XOR
- La sottrazione è come l'addizione

Il controllo di parità è dato dal resto della divisione, però usando l'aritmetica di GF(2)[]

Si sceglie un polinomio, il cosiddetto polinomio generatore G(x)
Abbiamo un messaggio M(x)
Potremmo dividere M(x) per G(x), calcolare il resto R(x) e fare l'encoding trasmettendo M(x) seguito da R(x)
Il problema è che se scegliamo un numero G qualunque con G(x) > M(x), e quindi, se il messaggio M(x) è minore di G(x), allora il resto sarà della stessa grandezza di M(x)
Per risolvere questo problema, allora moltiplichiamo M(x) per x^r con r = (grado(G(x))
Quindi moltiplicare per x^r un polinomio, equivale, ovviamente a fare lo shift a sinistra di r posizioni.

Abbiamo trasmesso M(x) seguito da R(x) -> "seguito da" equivale a x^r * M(X) + R(x) ma per le magiche proprietà di GF(2)[] è lo stesso che x^r * M(X) - R(x)
L'error detectiotn semplicemente prendere il numero trasmesso e calcola il resto della divisione per G(x) che per quanto detto prima deve essere zero. Quindi se il resto è zero è tutto ok, altrimenti c'è stato un errore.

Potenza
Qual è la potenza di un tale metodo?
Supponiamo ci sia un errore, sono bits che sono stati invertiti
In altre parole, per l'aritmetica di GF(2)[] è lo stesso che sommare un polinomio di errore, diciamo E(x)

Abbiamo il polinomio trasmesso T(x) più un polinomio di errore e calcoliamo il resto della divisione per G(x)
T(x) + E(x) mod G(x) = (sappiamo che T(x) mod G(x) = 0) -> E(x) mod G(x)
Quindi non riusciamo a trovare l'errore solo quando E(x) mod G(x) da come resto 0

Singoli errori
Abbiamo E(x)= x^i
->basta che G(x) abbia due o più termini

Doppi errori
E(x)= x^i + x^j
= x^j * (x^(i-j)+1)
-> basta che G(x) non sia multiplo di (x^i + x^j)

Ogni errore con un numero dispari di bits
Basta che x+1 sia un fattore di G(x), (deriva dagli zeri dei polinomi)

G(x) = (x+1) * ...
E(x) = x^i + x^j + ...

Dispari

```
E(x) mod G(x) = 0 -> ( E(x) = G(x) * .... )
E(0) = G(0) * ....                        |     E(1) = G(1) * ....
     = (x+1) * .... * ....                |          = (x+1) * .... * ....
     = (0+1) * .... * ....                |          = (1+1) * .... * ....
     = 1 * .... * ....                    |          = 0 * .... * ....
```

I Burst Errors

E(x) = x^i(x^j+...+1) è il burst error di lunghezza j+1
-> basta che G abbia grado > j e termini con + 1, e correggiamo tutti i burst lunghi fino a j+ 1!
In generale mettere "+1" è una buona scelta per un polinomio
Vaste opzioni: possiamo combinare tra loro più polinomi oppure scegliere (per quanto possibile) i polinomi irriducibili.

All'aumentare del grado G, aumenta la potenza: ogni burst error di lunghezza arbitraria più grande del grado di G
-> probabilità: 0.5^grado(G(x))

Cosa vuol dire nella pratica?
Protezione esponenziale col grado di G

Tipi di polinomi

CRC-1
x+1
-> è il codice parity bit 1

CRC-5	CRC-16
x^5 + x^2 + 1	X^16 + x^15 + x^2 + 1

Vengono usati nelle chiavette USB
Bluetooth

CRC-32
x^32 + x^30 + x^26 + x^25 + x^24 + x^18 + x^15 + x^14 + x^12+ x^11...
Usato nei Modem v.4
Formato .zip
Fibra Ottica
CD
Ethernet
Immagini PNG

Come detto questa tecnica "polinomiale" è alla base poi dei codici di error-correction più avanzati come Reed-Solomon.
Dove informalmente, invece di usare GF(2)[] si va a "ordini superiori", ad esempio GF(2^n)...
RS(255,233) su GF(255) è uno dei principali standard NASA
Potenza fino a ordine 16
Data rate 91%

I codici QR

I codici QR possono memorizzare fino a un massimo di 4.296 caratteri alfanumerici, 7.089 caratteri numerici. Nei codici QR è utilizzato il codice Reed-Solomon per la rilevazione e correzione d'errore: nel caso in cui il QR fosse in parte danneggiato, per esempio da macchie o graffi sul

supporto cartaceo, l'applicazione Reed-Solomon permette di ricostruire i dati persi, ripristinando, durante la decodifica, fino al 30% delle informazioni codificate.

Quattro livelli di error correction
L(Low), M(Medium), Q(Quartile), H(High)
->error recovery del 7%, 15%, 25%, 30%

Passiamo ora all'altro lato del Data Link, i flussi.
Servono quindi delle regole:

I protocolli come detto sono i tramiti cui si passano dati da un punto all'altro della rete

Il problema del flow control

In un canale semplice, se "inviamo troppi dati" rischiamo che il ricevente non riesca a gestirli e che quindi vadano persi.
In questo caso l'error control dentro ai dati non serve a nulla, anche se il dato arriva correttemente, è il ricevente che non riesce a gestirlo.

Potremmo disegnare la rete in modo da inviare i dati al giusto data rate
Questo approccio però è ovviamente limitativo perchè le capacità del ricevente/mittente possono essere diverse.
Occorre necessariamente andare su un altro tipo di approccio dove l'unica soluzione possibile è far parlare il ricevente con chi manda i dati.

Costruiamo il nostro primo protocollo di rete

I protocolli stop-and-wait

Sono una famiglia di protocolli in cui si introduce appunto il parlare tra ricevente e mittente.
L'idea è semplice: si manda un blocco dati e poi si aspetta (stop and wait) che il ricevente ci mandi un messaggio di conferma (ACK - Acknowledgement) segnalandoci che possiamo inviarne un altro.

Vantaggi: per implementare un protocollo di questo tipo basta un canale half duplex, visto che non c'è mai comunicazione contemporanea
Svantaggi: abbastanza lento

Mr Murphy

Supponiamo di dover lottare contro la legge di Murphy: cioè che nel canale di comunicazione ci sia qualche errore.

Ovviamente abbiamo protetto il nostro pacchetto, ma ciò non significa che sia immune da errori.
Quindi il protocollo andrebbe modificato in questo modo: il receiver invia il messaggio di conferma solo se il pacchetto supera il controllo dell'errore altrimenti lo ignora.

Mandiamo un pacchetto
Murphy interviene e corrompe il pacchetto
Il receiver se ne accorge e non ci invia il messaggio di conferma
E noi aspettiamo...
E aspettiamo
...

Così non funziona

Introduciamo un timeout: se entro un certo periodo di tempo non riceviamo nessun messaggio di conferma, rimandiamo il pacchetto.

In questo modo, noi mandiamo il pacchetto, Murphy corrompe il pacchetto, il receiver se ne accorge, non ci arriva il messaggio di conferma, noi aspettiamo e quando scade il timeout noi rinviamo il pacchetto.

Però può accadere che Murphy corrompa il messaggio di conferma del receiver, quindi noi rinviamo il pacchetto.
Quindi al ricevente il flusso di dati arriva con dentro ripetuto due volte un pacchetto dati.

Un modo per uscire fuori da questa situazione è aumentare il carico del messaggio (frame) non solo con l'informazione per il controllo degli errori, ma anche l'informazione sul controllo del flusso stesso.
Ad esempio numerando i pacchettti, in modo tale da accorgerci se ci sono ripetizioni.
Morale: Numeriamo ogni pacchetto
In questo caso se accadesse la situazione di prima il ricevente sa riconoscere un eventuale errore del flusso.
Aumentiamo l'informazione del frame con una parte dedicata ai numeri del controllo del flusso.

Quanto grande lo facciamo questo controllo? Grande quanto è grande il flusso
Se il numero è troppo piccolo rischiamo di non poter trasmettere un flusso di dati che sia più grande.

Pensandoci, in realtà il problema può avvenire solo fra pacchetti che sono consecutivi
Quindi non ci serve distinguere tra due frame qualsiasi ma solo fra frame contigui.
-> Ci bastano due simboli: 0 e 1

I frame vengono spediti usando un bit extra per i controllo di flusso: 0 o 1 alternati tra frame pari e dispari.

Siamo andati oltre il semplice stop-and-wait: famiglie di protocolli di questo tipo, dove si ritrasmettono tramite il timeout nel caso il pacchetto vada perso, si chiamano
PAR (Positive Acknolewedgement
ARQ

Quando il canale è full duplex possiamo dunque prendere un protocollo "simplex", metterne due ad ogni lato della linea e trasmettere da ambo i lati.

Un grosso problema è l'overhead: praticamente per ogni pacchetto che arriva, ne circola un altro per la riconferma.

Piggybacking
Invece di essere frettolosi e mandare un messaggio di conferma ogni volta che riceviamo un frame aspettiamo e inviamo il messaggio di conferma non da solo ma in piggyback sulle spalle del primo frame dati che stiamo ritornando

in questo modo stiamo riducendo la banda usata: essenzialmente il messaggio di conferma arriva gratis con un overhead minimo rispetto ai dati che stiamo trasmettendo (una conferma sono pochi bit rispetto a trasmettere un intero frame)

Per sfruttare bene la tecnica del piggybacking bisogna stare attenti a non aspettare troppo per la ritrasmissione (se ad esempio non stiamo trasmettendo dati).

Quindi funziona bene quando la comunicazione è abbastanza equilibrata. Quindi ci vuole un corretto bilanciamento di flussi, oppure un calcolo accorto del timeout di ritrasmissione .

In tutti questi tipi di protocolli, la logica era sempre quella di aspettare la trasmissione prima di aver avuto la conferma del messaggio precedente.
In alcuni contesti questo tipo di protocollo può rivelarsi estremamente inefficacie.

Esempio

Abbiamo una comunicazione satellitare con un GEO
Tempo di trasmissione: 250 ms
Banda assegnata: 50 kbps
Taglia ogni Frame: 1000 bit

Dopo 20 ms abbiamo inviato il nostro primo frame e poi aspettiamo la ricevuta di ritorno; supponendo che non ci siano errori aspettiamo 250 + 250 = 500 ms
 -> su 520 ms, trasmettiamo per 20 ms
 -> siamo rimasti bloccati per il 96% del tempo.

In generale il problema c'è quando il prodotto (bandwidth) * (round-trip-delay) è grande: i protocolli visti funzionano male, perchè sottoutilizzano il canale.

Se il canale ha capacità C (bit/s) la taglia del frame è S (bit) e il tempo di round trip è R, possiamo calcolare l'utiilizzo della linea nel caso di protocolli con ack: S / (S + C * R)

Se S < CR abbiamo un utilizzo del canale minore del 50%

Soluzione: Sliding Windows

La tecnica delle sliding windows sfrutta l'idea di non preoccuparsi della conferma dei pacchetti fino a quando non trasmettiamo un numero di frame n, (con n > 1)
Tiene conto di quanto siamo stressati: più apriamo la finestra, più siamo rilassati e più pacchetti lasciamo andare senza bisogno di conferma.
La taglia della sliding windows può variare sia per il sender che per il receiver, dando luogo a vari protocolli.

I protocolli "Go Back N"
Si hanno quando la taglia della sliding windows di chi riceve è 1, cioè noi siamo rilassati, mentre il nostro interlocutore è super-apprensivo.
Funziona bene quando non ci sono molti errori ma il prodotto del bandwidth * round-trip-delay è alto

Il nostro rischio è aumentato perchè nel caso peggiore abbiamo n pacchetti inviati, e non sappiamo se sono arrivati, e quindi dobbiamo essere pronti a rinviare quei pacchetti.
 -> dobbiamo avere un buffer di taglia n-frames
 ->Assieme a n timer per l'eventuale ritrasmissione

I selective repeat

Sono quelli in cui anche il nostro interlocutore, finalmente si rilassa un po' e apre la sua finestra.

Funziona bene, ma ha il problema che ora anche il receiver deve allocare un buffer per i pacchetti ricevuti.
La taglia richiesta del buffer è l'ampiezza dell'apertura massima della finestra, non la grandezza complessiva della finestra.

Esempi di protocolli reali

HDLC - High Level Data Link Control

Ha alcune varianti LAP - LAPB

Ideato inizialmente dall'IBM
Si usa ancora per modem/fax, reti vario tipo e per molti altri (bancomat)

Usa dei frames delimitati tramite il bit stuffing che abbiamo visto

La struttura del frame

[01111110 | Address | Control | Data | Checksum | 01111110]

     8   8   8   >0   16   8

la parte Data è il payload, i dati effettivi
La parte Checksum è calcolato usando CRC
La parte Address consente di gestire indirizzi multipli
La parte di Control è quella più interessante. Essenzialmente, ci possono essere tre tipi fi Frame:
- Information
- Supervisory
- Unnumbered

Control del Frame
Il controllo del flusso avviene tramite una sliding window di grandezza massima di tre bit ("8 spicchi")

Il Frame Information

- Seq contiene il numero del controllo di flusso della sliding window
- Next contiene gli ACK(in piggyback)
- P/F sta per Poll/Final. quando il bit indica P si chiede al ricevente di iniziare la trasmissione
  con F si rischiede di terminare la trasmissione

[0 | Seq | P/F | Next]

  1   3   1   3

C'è anche una variante dove il controllo di flusso si fa con una sliding window grande 128 "spicchi", usata per le comunicazioni satellitari.

Il frame Supervisory

Si occupa della supervisione del flusso

Il campo type indica i vari tipi di supervisione: quattro tipi sono possibili
- Type 0: ACK; (in questo protocollo detto Receive Ready), si usa quando il flusso è sbilanciato e non si può fare ACK con piggybacking
- Type 1: Reject; è un NAK generalizzato, segnale che vanno ritrasmessi tutti i frame partire da quello indicato in poi nella sliding window
  Qui il Next indica il primo frame.
- Type 2: Receive Not Ready; questo è qualcosa di concettualmente nuovo: segnala che ci sono problemi di congestione nel receiver, e quindi la trasmissione va bloccata, finchè il receiver non rimanda un ACK.
- Type 3: Selective Reject; questo è il classico NAK.

[1 | 0 | Type | P/F | Next]

Il Frame Unnumbered

Usato per ulteriori comandi di controllo

- DISC: Sta per DISConnect, segnala che la macchina sta uscendo dalla rete in maniera definitiva (quindi diverso dal frame Supervisory di tipo 2
- SNRM: Set Normal Response Mode
  E' il comando duale, segnala che una nuova macchina è entrata nella rete
  Indica un canale asimmetrico, dove il nuovo entrato è meno importante
- SABM: Set Asynchronous Balanced Mode
  E' il comando (introdotto successivamente) per creare una connessione bilanciata, dove chi
  entra ha gli stessi diritti degli altri
- FRMR: FRaMe Reject
  Indica che è arrivato un frame con una sequenza di controllo non corretta/sconosciuta

Infine anche i comandi di controllo unnambered possono essere "toccati" da Murphy
-> c'è bisogno anche qui di SCK
-> comando dedicato, UA (Unnumbered Acknowledgement)

[1 | 1 | Type | P/F | Modifier]

Anche per un semplice protocollo come HDLC abbiamo visto quanta complessità occorre introdurre per semplicemente far funzionare le cose.

Un altro protocollo, che si usa molto di più dell' HDLC, perchè è il protocollo di referenza di Internet per quanto riguarda le connessioni point-to-point

Point to Point
Sono le connessioni internet dedicate punto-a-punto
In internet si usa il protocollo PPP (Point-to-Point Protocol)

Ovviamente dà un metodo di framing per impacchettare i bit

Cosa fa PPP?
Comandi di controllo del flusso per attivare le connessioni, test, negoziazione, chiusura
Questa parte si chiama LCP (Link Control Protocol)

Metodo per negoziare con lo strato superiore dl network layer.
Questa parte si chiama NCP (Network Control Protocol)

Il Frame PPP

E' stato disegnato cercando di essere quanto più simile a HDLC
Ad esempio il delimitatore del frame è lo stesso di HDLC
PPP gestisce frame a lunghezza variabile

[Flag | Address | Control | Protocol | Payload | Checksum | Flag]

PPP vs HDLC
Differenza:
- PPP usa però byte stuffing invece che bit stuffing
- Il campo Address ha lo stesso significato che in HDLC e non si usa: ha sempre il valore costante 11111111.
- Il campo Control non si usa quindi tutti i frame sono di tipo non numerato.
- E Control ha valore fisso 00000011.

Dunque PPP, è un protocollo che non usa le tecniche di numbering e ACKs per creare una comunicazione

Il campo Protocol

In questo campo si specifica il protocollo che PPP sta implementando
Quindi in un certo senso PPP stesso è un meta-protocollo

Il campo payload

Nel campo payload ci sono i dati, il cui significato dipenderà dal campo protocollo specificato in Protocol

Il campo checksum

C'è il checksum del frame, calcolato usando CRC (-> PPP fa error detection, ma non correction)

I protocolli supportati da PPP
Nel campo protocol ci possono essere essenzialmente due tipi di protocolli:
quelli di negoziazione (essenzialmente che restano nel data link layer) e quelli di livello più alto (network layer)

Due famiglie, LCP (che stabilisce la linea) e NCP (che interagisce con i stati superiori)

LCP

Vediamo quali sono i comandi che LCP usa:
Sono 11:
-> 4 di configurazione
-> 2 di terminazione
-> 2 rifiuto
-> 2 di eco
-> 1 di test

Configurazione LCP

Configure-request:
      Sender -> Receiver
            Propone opzioni per la configurazione della linea
Configure-ack:
      Sender <- Receiver
Manca il controllo ACK nel contenitore PPP viene quindi rimpiazzata da una reimplementazione nel sottoprotocollo
Configure-nak:
      Sender <- Receiver
            Il NAK per configure-request
Configure-reject:
      Sender <- Receiver
            Non c'è nulla da fare, (opzioni non negoziabili)

Per evitare sprechi è possibile rimuovere i campi Address e Control

Terminazione LCP

Terminate-request
      Sender-Reciver
Terminate-ack
L'ACK di terminazione-request va bene, comunicazione chiusa

Rifiuto LCP

Code-reject
      Receiver -> Sender
            Non ho capito cosa intendevi, richiesta sconosciuta
Protocl-reject
Non capisco di che protocollo parli, (o c'è stato un errore sulla linea)

Echo LCP

Echo-request
      Sender -> Receiver
            Per favore, rimandami il frame indietro
Echo-reply
      Sender <- Receiver

Serve a controllare/misurare la qualità della linea di comunicazione

Test LCP

Discard-request
Sender -> Receiver
Ignora il frame

Serve a fare un primo test preliminare della linea e trovare eventuali loops.

PPP e le ADSL

Ricordiamo che ci sono vari parametri settabili in PPP
Alcuni ovvi come abbiamo visto, che tolgono byte inutili da ogni frame.
L'altro aspetto è il payload variabile

La taglia dei pacchetti

Significa quindi che dovremmo scegliere una taglia massima per i nostri pacchetti
Quello che in gergo si chiama MTU - Maximum Transmission Unit

Ogni protocollo ha un MTU, fissato dallo standard ed eventualmente riconfigurabile verso il basso

Lo Zen e l'arte dell'MTU ADSL

Intuizione generale sull'MTU

MTU grande -> pacchetti più grandi -> meno overhead -> più banda (va bene se il canale ha pochi errori)
MTU piccolo -> pacchetti più piccoli -> più overhead -> meno banda (va bene se il canale ha molti errori)

PPP si usa in due varianti primarie

Le varianti sono
PPPoE - PPP over Ethernet
PPPoA - PPP over ATM
"over" = incapsulato dentro flussi Ethernet ed ATM

PPP -> PPPoE/PPPoA -> Ethernet/ATM
Ma la situazione è più complessa
Flusso dati 1: dal computer al router
Flusso dati 2: dal router al modem
Flusso dati 3: dal modem al provider

Ma neanche così è finita

Flusso 3: dal modem "al provider"...?
Flusso 3: dal modem al primo DSLAM (Digital Subscriber Line Access Multiplexer)
Flusso 4: dal primo DSLAM fino a qualche punto del provider
Flusso 5: ci si collega alla rete Internet

Nei flussi "lato provider"
ci possono essere due tecnologie, quella ATM e quella Ethernet
Molto spesso il primo tratto è formato da ATM e l'ultimo da Ethernet

ATM

Asynchronous Transfer Mode
Usa TDM, si dividono i dati in flusso di celle di ampiezza fissa.
Analogo di HDLC ma nato per telefonia/bancomat etc e non per internet, gestisce controllo di flusso con sliding windows("16 spicchi"), error detection (CRC-8)

Indirizzamento ATM
Doppia gerarchia, cammmini ("path") e canali ("channel")
Che sono le due etichette VPI e VCI che vediamo nelle configurazioni ADSL
VPI - Virtual Path Identifer
VCI - Virtual Channel Identifer
-> terminologia: Virtual Channel - ATM è connection-oriented

Anche ATM, come PPP, viene concretamente embeddato dentro altri pacchetti.
Giacchè appunto si tratta di canali in multiplex e quindi c'è interazione con altri flussi dati

LLC o VC-MUX
LLC - Logical Link Control: Protocolli multipli per canale
VC-MUX - Virtual Connection Multiplexing: Un solo protocollo per canale

Flusso dati 1: dal computer al router (Es. PPPoEoE PPP -> PPPoE -> Ethernet)
Flusso dati 2: dal router al modem (Es. PPPoEoA Ethernet -> ATM)
Flusso dati 3: dal modem al provider (Es. PPPoA ATM -> LLC/VC-MUX)
Flusso dati 4: dal provider ad un punto interno
Flusso dati 5: da un punto interno ad una rete Ethernet esterna

Se di mezzo c'è il wireless la situazione si fa ancora più complicata

In pratica l'MTU interagisce con tutti gli altri protocolli in tutti i flussi in atto
Quindi occorre stare bene attenti a sapere esattamente cosa si modifica.

caso ideale: linea senza errori
      -> MTU grande
Ma, diminuendo l'MTU sotto certe soglie critiche, la banda migliora.
Deriva tutto dall'interazione con gli altri flussi.
C'è poi da considerare tutta l'interazione con gli strati superiori
In ogni caso ci torneremo quando avremo altre nozioni che ora ci mancano

Alla visione più vicina a PPP, e quella più comune: PPPoE

PPPoE: ciclo iniziale

La connessione ADSL inizia così:
Il nostro computer/modem invia un frame PPPoE (Active Discovery Initiation) col suo indirizzo fisico (MAC)

Ogni servizio ADSL disponibile risponde con un PADO in cui dà il proprio indirizzo e si "offre" per la connessione.
Il nostro computer risponde con un PADR (PPPoE Active Discovery Request) in cui segnala il servizio ADSL che ha scelto.
Il servizio fa l'ACK usando un frame PADS

Competizione e "lock-in"

I protocolli multiaccesso
Finora abbiamo visto il caso point-to-point, in cui c'è uno che parla e uno che ascolta, ed un canale tutto pe loro.
Ovviamente, ci sono molti altri contesti in cui queste assunzioni non sono valide, e ci sono molte entità diverse che vogliono usare lo stesso canale per parlarsi

I contention system
Sono quei sistemi di comunicazione multipla in cui c'è un unico canale condiviso da molti, e si possono creare contenziosi.

Assunzioni

Station Model: sono le entità che trasmettono. Dopo che hanno iniziato la trasmissione di un frame, non fanno altro finchè non è stato trasmesso.

Single Channel: c'è un canale singolo disponibile per tutti
Collision: se due frame si sovrappongono, c'è una collisione e sono inutilizzabili

Sul tempo
O continuo (non c'è un orologio centrale)
oppure slotted (a intervalli)

Su carrier (il mezzo di trasporto)
Carrier Sense (una stazione può vedere se il canale è in uso prima di tentare una trasmissione)
No Carrier Sense

Sfruttamento del caso
Probabilità

La distribuzione di Poisson

Pr[k] = (G^k * e^-G) / k!

Protocollo Aloha

Dobbiamo vedere qual è la probabilità che ci sia una sola trasmissione (pr[1)] in un tempo doppio
    -> la media in quel periodo è 2G
    -> la probabilità (Pr[k] = (G^k * e^-G) / k! è 2G*e-2G

La probabilità che il canale sia usato correttamente in ogni singolo slot di tempo è la meta G * e^-2G

E quindi, a quanto mi conviene settare la velocità di tentativi di accesso al canale (G) per massimizzare le prestazioni?
Il massimo facile da vedere si ottiene con G = 0.5
    -> 1/2e frame/sec -> circa 0.184

18.4% di banda... è poco

Però il 18.4 non dipende da quante entità possono trasmettere, che possono essere tantissimi
Altra variabile possibile: lunghezza del frame random
Ma conviene fissare frame di lunghezza fissa

Motivo intuitivo: perchè avere frame diversi creerebbe rotture di simmetria

Slotted Aloha
Due anni dopo Aloha, Roberts trova un modo per migliorarla
Assume che il tempo sia "slotted" tramite una stazione principale che manda un segnale di sincronizzazione
In tal modo, una stazione deve attendere prima di trasmettere il pacchetto, non c'è trasmissione istantanea.

Come cambiano le prestazioni?
Il periodo "critico" che può generare conflitti stavolta è dimezzato.
Ora la probabilità di successo è Pr[1] = G * e^-G
Il numero massimo di frame al secondo si ha con G=1 e diventa: 1/e
Cioè circa 36.8%, il doppio di Aloha

Abbiamo raddoppiato la velocità alterando le regole del gioco
Si può fare ancora meglio?

Carrier Sense

1 - persistent CSMA
CSMA - Carrier Sense Multiple Access protocol
-> prima di trasmettere, controlla che non ci sia già una trasmissione (carrier sense)
Se c'è una trasmissione, controlla il canale, e non appena si libera, trasmette

Può lo stesso succedere che ci siano collisioni (Murphy...) in questo caso?
Si riaspetta un certo tempo casuale e poi si riprova
Performance: aumenta a più del 50%

Difetto di 1-CSMA
Se molte stazioni stanno aspettando durante una trasmissione, alla fine della trasmissione tutte cercano di trasmettere contemporaneamente provocando una collisione
Se fossero un po' meno "egoiste", il mondo funzionerebbe meglio

p-persistent CSMA

Si ottiene rilassando l'ipotesi che uno debba sempre trasmettere (p=1) quando trova la linea libera, ma invece, trasmette con probabilità p (non deve trasmettere a tutti i costi)
Performance: Migliora, con molte stazioni al diminuire di p

Al limite si può arrivare al 100% dell'efficienza

Nonpersitent CSMA

Usa un metodo alternativo: invece di tenere d'occhio il canale per ricominciare la trasmissione non appena è libero, se trova il canale occupato aspetta un periodo di tempo casuale e poi riprova.
Performance: la curva di performance si comporta differentemente da p-CSMA e poò raggiungere performance vicine al 90&

CSMA/CD - CSMA Collision Detection

Avendo il carrier Sense, possiamo anche essere più furbi quando trasmettiamo: non controlliamo solo quando vogliamo spedir il segnale, ma anche durane tutta la trasmissione, e se ci sono accorgiamo di una collisione, ritrasmettiamo dopo sprecando meno banda
Quindi, rilassiamo l'assunzione station model, possiamo fare anche altro mentre trasmettiamo

In CSMA/CD possiamo separare tre possibili stati della rete trasmissione, contention e idle:
E' nella parte contention, cioè decidere quanto tempo allocare a quella zona in cui si decidono le sorti della trasmissione
Il tempo da allocare sarà due volte il tempo di trasmissione tra le due stazioni più distanti (tempo di roundtrip)
Tipicamente il periodo di contention ha durata il tempo massimo di roundtrip, e si usano più tecniche di protocollo multiaccesso per vincere il round ed essere sicuri che il canale è nostro.

Protocolli collision-free

Anche detti CSMA/CA (Collision avoidance)
In questa classe di protocolli si fa un uso intelligente dei contetion

Basic bit-map
Un modo abbastanza ovvio: prendere il contetion period, e dividerlo in intervalli uguali per ogni stazione
Ogni stazione potrà quindi segnalare nel suo mini slot se vuole trasmettere un frame (fa una prenotazione, quindi è anche un cosiddetto reservation protocol)

Quanto grande sarà il mini-slot?
Batsa un bit per segnalare la nostra intenzione di trasmettere.
      -> Il contention period conterrà un bit per ogni stazione

A pieno carico, se la taglia del frame è d, è ovvio che l'efficienza sarà d/(d+1) (un solo bit sprecato)
Il problema del bit-map è che abbiamo un bit per ogni stazione, quindi con tante stazioni il contention period può diventare molto lungo

Binary count-down
L'idea è invece di riservare un bit separato nel contention protocol a ogni stazione (quindi, prima, per N stazioni ci serviva uno spazio di N bits)

Qual è la codifica più compatta per rappresentare N stazioni?
Ovviamente possiamo usare una rappresentazione binaria, quindi ci servono solo log2(N) bits.

Allora, ogni stazione ha il suo indirizzo in binario
Quando vuole trasmettere, fa un OR boolenao del suo indirizzo alla rovescia

	Bit time			
	0	1	2	3
0010	0	-	-	-
0100	0	-	-	-
1001	1	0	0	-
1010	1	0	1	0
result	1	0	1	0

Efficienza - Prima trasmettevano N bit più il tempo di roundtrip, mentre ora trasmettiamo solo log(N) bit ma per ognuno serve un giro si roundtrip
Questo nuovo metodo quindi è efficiente quando il tempo di roundtrip è piccolo e si può trascurare
Nel qual caso, l'efficienza del protocollo è circa d/(d+log2(N))
Però ciò vale a situazioni ad alto carico cioè con tantissime stazioni che trasmettono contemporaneamente: a basso carico l'efficienza vale come d/(d+1)
A seconda dei casi può essere meglio o peggio uno dei due protocolli.

Le stazioni sono in priorità, quindi si rischia lo stallo di quelle a più bassa priorità
Ovviamente ci sono vari modi per ovviare a questa situazione.
Ad esempio il solito metodo probabilistico in stile Aloha: chi ha trasmesso aspetta un periodo di tempo casuale prima di ritrasmettere
Notare però che questo metodo può portare a basso carico a situazioni di idle

Altro metodo

Manteniamo l'informazione che abbiamo: ogni stazione sa chi ha trasmesso. Se ha trasmesso la stazione k, tutte le stazioni con numero < k possono aumentare la loro priorità di 1, chi ha trasmesso invece va a priorità 0.

Ricapitoliamo
I protocolli CSMA funzionano meglio a carico basso, peggio ad carico alto
    -> troppi conflitti
I protocolli collision free funzionano meglio a carico alto peggio a carico basso
    -> spreco di banda
La cosa migliore ammesso che ci riesca, sarebbe trovare un protocollo che unisca il meglio dei due mondi

Il vero problema di ogni protocollo a multiaccesso è ovviamente quello della competizione
Nel caso CSMA, non abbiamo considerato il numero di stazioni.

Potremmo per cui studiare cosa succede a CSMA quando fissiamo il numero di stazioni
Supponiamo il caso migliore, slotted time, e che ogni stazione trasmetta con probabilità p
Se ci sono N stazioni, qual è la probabilità che una stazione riesca a trasmettere?

$N * p * (1-p)^{(N-1)}$

Avendo un certo numero di stazioni qual è la scelta di p migliore?

$p = 1/N$

Quindi date N stazioni, la miglior probabilità che le cose vadano bene è $((N-1)/N)^{(N-1)}$

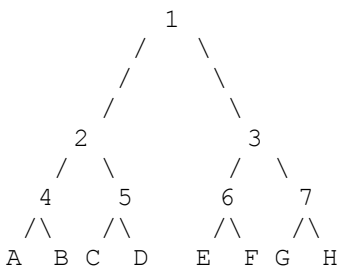
Se la competizione non fosse determinata a priori, ma fosse invece dinamica?

Cioè se avessimo un protocollo che se ci sono tante/troppe stazioni che vogliono trasmettere, diminuisca dinamicamente la competizione?

Limited-Contention protocols

E' una classe speciale, dove si rilassa un po' l'assunzione che tutte le stazioni siano sempre uguali (quindi dal caso simmetrico passiamo al caso asimmetrico)

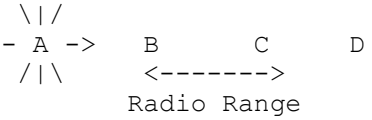
Adaptive Tree Walk Protocol



Quando c'è collisione si diminuisce dinamicamente la competizione, selezionando una sottoparte dell'albero.
Possiamo fare ancora meglio: analizzando il traffico recente, possiamo renderci conto d quante sono le stazioni che cercano di trasmettere in quel lasso di tempo.
Se siamo in un nodo a profondità P, i nodi sotto di lui diminuiscono all'incirca 2^P
Se le stazioni attive (A) sono distribuite equamente, allora ci saranno un circa A/2^P stazioni attive nel sottoalbero.
Vogliamo avere il sottoalbero più piccolo che abbia una stazione attiva
-> A/2^P = 1
-> P = log2(A)
Quindi ad esempio se abbiamo circa 8 stazioni attive conviene cominciare direttamente a profondità 3

Passiamo ora al caso wireless

Il problema è che non c'è più un singolo canale per tutti, ma varie zone spaziali dove alcune stazioni interagiscono, ed altre no.
In altri termini, il controllo da globale diventa locale



A trasmette B
C non sente, quindi conclude che può trasmettere a B
(Hidden Station Problem)

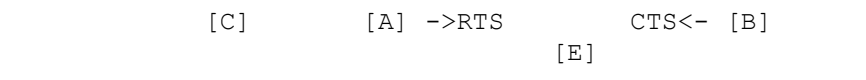
B trasmette ad A

C sente la trasmissione, e conclude che non può trasmettere a D, Quando invece avrebbe potuto.
E' detto il problema (duale) della stazione esposta
(Exposed Station Problem)

MACA - Multiple Access with Collision Avoidance
(Esteso poi a MACAW)

Essenzialmente sfruuta l'idea che chi deve trasmettere renda il suo spazio locale "conosciuto" anche agli altri.
Questa conoscenza locale avviene tramite due comandi:
- RTS (Request to Send) che rende nota la volontà di inviare un messaggio
- CTS (Clear to Send)

Esempio:



A manda il TCS a B
B risponde con il CTS
C sente l'RTS di A, ma non il CTS di B
-> può trasmettere mentre il frame è inviato
E sente sia l'RTS e il CTS
-> non trasmette finchè la trasmissione non viene completata
D non sente l'RTS, ma sente il CTS
-> non trasmette

MACA non è perfetto
Non è collision free ad esempio se B e D trasmettono entrambe a C
-> si usa la tecnica Aloha
Si può aggiungere nei pacchetti la durata della trasmissione per far sapere alle altre stazioni quando potranno trasmettere di nuovo

Murphy può toccare un a parte qualsiasi della comunicazione perfino quella tra A e B
Quindi occorre anche introdurre degli ACK e complicare ulteriormente il protocollo di base

Quelli "attorno" aspettano che A e B parlino e dopo?
Usano sempre Aloha in modalità non persistente

Passiamo ora a vedere qualche esempio pratico di protocollo in uso

Il primo protocollo che vediamo è lo standard IEEE 802.3 cioè Ethernet

Ethernet

Viene in vari tipi, con nome identificativo "XBaseY, dove:
- X è la banda in Bbps
- "Base" indica che è una connessione baseband 8a frequenza unica)
- Y è il tipo di cavo che si usa

Abbiamo già visto come le interferenze del cavo rendano necessario l'uso di ripetitori dopo una certa lunghezza critica
I vari tipi di ethernet differiscono anche, quindi, a seconda della unghezza massima di ogni tratto senza ripetitori.

Vari tipi di cablaggio: a serpente, a pesce...

L'inventore Bob Metcalfe
Studente al MIT
Fan di ARPANET
Scrive un saggio, L'AT&T lo contatta per vedere questo nuovo tipo di rete in azione
Mentre dà la demo, il sistema va in crash
Passa poi a fare il dottorato ad Harvard, dove comincia a sviluppare Ethernet
La sua tesi viene giudicata molto scadente perchè il lavoro viene giudicato carente dal punto di vista teorico
Passa alla XEROX, dove sviluppa la Alto Aloha Network (Alto il pc in uso alla xerox)
Nel 1973 rinomina il sistema in Ethernet
ETHER-net (Etereo)

10Base5

E' la primissima versione
Usava un grosso cavo coassiale
Con tacche ogni 2 metri e mezzo per segnalare dove inserire il cavo
Il 10 significa banda fino a 10 Mbps
Il 5 significa che supporta segmenti fino a 500 metri
Ogni segmento può contenere fino a 100 stazioni

Tipo di connessione

La connessione fisica avviene tramite i vampire taps, il cavo veniva forato con uno spuntone
Notare il design: il transceiver (che si occupa del carrier detection e collision detection) era posto ad ogni giunzione
Con le successive evoluzioni invece, il controller è stato spostato dentro al computer

10Base2

L'evoluzione di 10base5
Fatto con cavo coassiale
Giunzioni a T
```
----][-][----
      |
```
Transceiver nel pc
Economico, più affidabile dei 10Base5
2 -> cavo fino a 200 metri

10Base-T
Usa anche hubs con cavi dedicati per ogni pc
invece del coassiale usa il cavo twisted pair (T)
Poco costoso, ancora più affidabile
      -> E' diventato il tipo dominante
Svantaggi: La massima lunghezza di ogni segmento scende a 100 metri
Vantaggi: Il numero di stazioni per segmento cresce, dai 100 fi 10Base5 ai 30 di 10base2 si arriva ad avere 1024 stazioni!

Ricapitoliamo

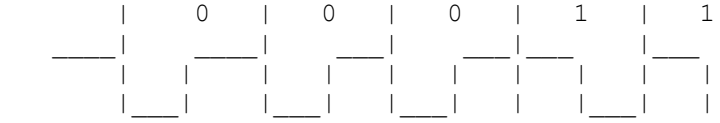
10Base5: 500 metri, 100 stazioni [200 per Km]
10Base2: 200 metri, 30 stazioni [150 per Km]
10Base-T: 100 metri, 1024 stazioni [10240 per Km]

10Base-F
Usa la Fibra ottica
Grande vantaggio: permette segmenti fino a 2 chilometri
Il tapping è molto più difficile, quindi è un buon metodo per connessioni esterne all'edificio

Codifica fisica in Ethernet
Per codificare 0 e 1, non usa 0 volts e X volts: perchè?
Tra le altre cose, per un motivo importante: usare 0 volts per il simbolo 0 porterebbe a seri problemi di sincronizzazione

Esempi:
Inviando 00100000, dobbiamo essere esattamente sincronizzati, altrimenti potremmo conforderci con 0100000 oppure 10000000
Una scelta possibile potrebbe essere quella già vista ad esempio per il CDMA: -X volts per 0 , X volts per 1
Per reti complesse, questa codifica ha lo svantaggio che richiede sempre una buona dose di sincronizzazione

Il Manchester encoding



Svantaggi
il manchester encoding risolve i problemi di sincronizzazione
-> hardware meno costoso
Lo svantaggio è quello di dimezzare la banda
E' usato da Ethernet anche perchè la sua affidabilità ha poi reso possibile incrementare la banda senza far esplodere i costi dell'hardware!

I frames di Ethernet

[Preamble | Destination Addres | Source Addres | Type | Data | Pad | CheckSum]

Un preambolo di 8 bytes formati da 10101010...
L'indirizzo di destinazione (6 bytes)
- Di questi, il primo bit posto a 1 segnala una comunicazione multicast (o a un gruppo)
- Il secondo bit distingue indirizzi locali da quelli globali
- Spazio degli indirizzi globali: 48-2 = 46 bits

C'è il campo type, che specifica il tipo di protocollo o in ogni caso l'uso del frame
Il campo dati: al massimo 1500 bytes
- Frames a lunghezza variabile

Il campo Checksum è il classico CRC-32
- Ethernet fa error detection ma non error correction

Infine il campo Pad
- ha lunghezza variabile (0-46)
- Serve per il controllo di collisione

Pad serve ad assicurare che la lunghezza minima del frame sia almeno il tempo di roundtrip
Per Ethernet a 10 mbps
-> il frame deve essere lungo almeno 500 bits, 512 (64 bytes) per sicurezza

Indirizzi globali e locali

I MAC Address - Media Acces Control
MAC-48 (in futuro EUI-64): gestiti dall'IEEE
I primi 3 bytes. il produttore
- OUI (Organizationally Unique identifier)
I secondi 3: il loro spazio di indirizzi
Durata (circa): fino al 2100...!

Cosa succede quando c'è collision detection

Se c'è collisione, aumentiamo esponenzialmente il tempo di attesa massimo, e facciamo ritrasmettere a caso, finchè non va liscia.
Quello che si chiama binary exponential backoff

Backoff Truncated

In realtà non si rischia così tanto, perchè c'è un limite: dopo 10 raddoppi si mantiene l'intervallo massimo a 1023 slots per altre dieci collisioni e poi si rinuncia
-> binary exponential truncated backoff

Efficienza d Ethernet

Se un frame ci mette in media tempo T per essere trasmesso, l'efficienza media è circa:
T / (T + 2 * roundtrip / alpha)
Dove alpha è la probabilità di trasmissione del canale.

Sostituendo T = lunghezza frame/bandwidth, abbiamo alla fine che l'efficienza è inversamente proporzionale a bandwidth*lunghezza

Quindi notare l'effetto perverso: quanto più aumentiamo la banda, o la lunghezza utile del cavo le prestazioni calano

Bisognerà sacrificare uno dei due aspetti
Visto che alla banda non possiamo rinunciare, essenzialmente serve una rete che abbia lunghezza massima limita
Invece degli hub si usano gli switch!

Fast Ethernet

10mbps erano molti, ma al solito, sono poi diventati pochi col tempo
-> IEEE ha formato due comitati (circa 1992)

Il primo comitato voleva estendere Ethernet mantenedo i suoi problemi
-> standard 802.3u ("Fast Ethernet") 100Mbps
Mentre il secondo comitato voleva creare un'Ethernet migliore, e con supporto di traffico real-time e voce.
-> standard 802.12

L'idea di base era semplice: aumentare il ciclo di clock di un fattore 10

Per le limitazioni che si hanno sulla lunghezza del cavo si sono dovuti introdurre gli switch

Tipi di fast Ethernet
100Base-T4: usa cavi UTP3
      -> Lunghezza fino a 100 metri
100Base-TX: usa cavi UTP5
      -> Lunghezza fino a 100 metri
100Base-FX: Fibra ottica
      -> Lunghezza fino a 2Km

Nota: UTP3 e UTP5 sono uguali?
No, lo standard UTP3 richiede 4 cavi

Ethernet Gigabit

L'evoluzione 10x (802.3z)
Evoluzione notevole: ora funziona in due modi, di cui uno point-to-point che sfrutta la presenza di una switch.
      -> niente carrier sense, molto più veloce

Problemi del Gigabit
Aumentando la velocità, la lunghezza del cavo diminuisce ancora
Soluzione 1: usare del padding ulteriore
      -> l'efficienza generale cala fino del 9%
Soluzione 2. invece di trasmettere un frame solo, si trasmettono blocchi di frame (frame bursting) come se fossero un unico superblocco.
      -> con questa tecnica, arriviamo a 200 metri

L'encoding
E' più sofisticato: per problemi di sincronizzazione, si evitano le sequenze di 0 e 1 troppo lunghe
Encoding: 8B/10B
Ogni 8 bits sono codificati con 10 bits in modo tale che non ci siano mai 4bits identici in una word

Altro piccolo problema: il flusso dati può essere notevole
Se il ricevente (o lo switch) è impegnato per 1 ms si possono accomulare fino a 2000 frames
Ecco spiegato perchè gli switch Gigabit costano relativamente molto di più di quelli normali Ethernet
Ed ecco perchè nel Gigabit ethernet è stato introdotto un nuovo comando di pausa per fermare momentaneamente la trasmissione

Standard Wireless

Il più conosciuto è 802.11
Usa la banda 2.4GHz
Interferenze da forni a microonde, telefoni senza fili, bluetooth e quant'altro

802.11a: fino a 54Mbps
802.11b: fino a 11Mbps
Ma "b" viene prima di "a"
Inoltre il range è 7 volte più grande
Infine 802.11g: va fino a 54Mbps

Trasmissione

Avviene in 5 modi:
- Infrarosso
- FHSS
- OFDM (Orthogonal Frequency Division Multiplexing) è quella usata da 802.11a
  - Usa QAM
- DSSS
- HR-DSSS (High rate Direct Sequence Spread Spectrum)
  - Essenzialmente modulazione di fase
  - E' quella usata da 802.11b
    -> Usa codici di Walsh-Hadamard

Come funzionano gli 802.11
Ci sono sempre i soliti problemi di hidden station e di exposed station
Ci sono due modi operativi
- DCF: Distributed Coordination Function, senza controllo centrale
- PCF: Point Coordination Function, con controllo centrale

DCF e PCF possono coesistere contemporaneamente

DCF

Usa un 802.11 detto CSMA/CA che può funzionare in due modalità diverse
- La prima, è come un classico CSMA/CD
  -> Nonpersistent CSMA (usando il truncated binary exponentioal backoff)
- Nella sua seconda modalità CSMA/CA usa essenzialmente MACAW con le protezioni anti-Murphy, ad esempio la durata della trasmissione messa nei pacchetti

PCF

Qui la stazione base manda un frame "beacon" periodicamente in broadcast, per sincronizzazione e informazione sullo stato della trasmissione
Tutto è dunque centralizzato ed il modo di operazione diventa simile a quello dei telefonini

802.11n (aggiunge il (MIMO - Multiple-input Multiple-output) usa le cosiddette smart antennas)
- -> range praticamente raddoppiato rispetto a 802.11g
- -> si arriva a 70 metri indor e 250 metri outdoor

Bandwith? Dai 54Mbps di 802.11g a 248Mbps fino a 600Mbps
Usa banda più grande (da canali di 20MHz si passa a 40Mhz, restando nella banda 2.4GHz o passando a 5Ghz)
Usa MIMO (fino a 4 antenne)
Codifica QAM (tipicamente QAM-64)
Oltre 802.11n -> 802.11ac
Data rate fino a 860Mbps
In sintesi:
- Più banda per canale (dai 40MHz ai 160MHz)
- Uso della banda ad altra frequenza (5GHz),
- più MIMO (8x)
- Modulazioni più spinte (QAM-256)

Il problema delle reti è farle più grandi
Ma ce ne sono molte e diverse ed anche reti dello stesso tipo hanno limiti

Colla per Reti

Repeater e Hubs

Il ripetitore, come già visto ripete amplificando (in potenza) il segnale

Lo Hub (che può anche essere repeater) è l'equivalente di una connessione fisica, cioè propaga il segnale da una porta a tutte le altre.
Pro: poco complesso, costa poco, affidabile
Contro: non risolve i problemi dei limiti della rete
        Essendo una estensione della stesse rete, non può unire reti diverse (ad esempio una 10Mbps con una a 100Mbps)

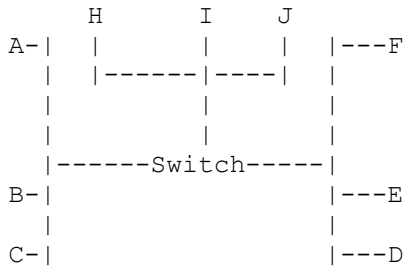
Bridge e Switch

Dispositivi di livello più alto (data link) e quindi interagiscono con la struttura dei frame, cosa che repeaters e Hubs non fanno

L'idea è di unire due o più reti, tenendole per quanto possibili distinte
Crea una rete più grande ma a livello logico (data link) e non fisico
In altre parole, non abbiamo più i limiti di grandezza della rete, ed abbiamo domini di collisione diversi
Questo si ottiene ispezionando i pacchetti, e filtrando il traffico

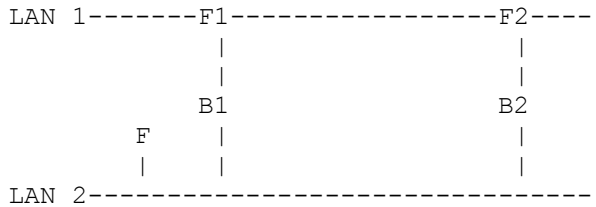
Learning
C'è quindi una fase in cui il Bridge/Switch impara la configurazione di rete, e gestisce il traffico corrispondentemente
Essendo a livello data link, controlla mittente e destinazione dei frame
     -> Tipicamente i MAC Address
Si usa il backward learning: le hash table vengono costruite analizzando il flusso di dati e risposte

Esempio



Timeout
C'è un timeout di fading per ovviare ai cambiamenti nella topologia di rete
Situazione simile
Ai motori di ricerca...pensate a come funziona ad esempio Google

Problema
Possono venire a crearsi dei loop, A può essere collegato a B, che è collegato a C, che a sua volta è collegato ad A



Evitiamo i loop e facciamo solo strutture ad albero
ottima idea in teoria, ma in pratica sarebbe troppo limitato
Soluzione?
Lo fanno gli Switch per noi
Da ogni rete generica con magari dei loops estraiamo una sottostruttura senza senza loops (albero)
-> spamming tree

C'è un protocollo specifico, 802.1D che viene gestito tra bridge/switch per evitare i loops
i bridge/switch si scambiano pacchetti speciali 802.1D (BPDU)
E con questi calcolano le distanze minime tra loro
Anche lo spamming tree avviene in maniera dinamica

Notare come ci voglia tempo per costruire lo spanning tree (e tenerlo aggiornato)
Tipicamente 30 secondi
Nella nuova versione 802.1D si è riusciti a scendere a circa 6 secondi
E ci sono soluzioni ancora più performanti (TRILL)

I routers
Fanno tipicamente parte dello Network layer
Il routing è dunque il processo in cui si decide la via per trasmettere un pacchetto in una rete complessa
Simile ad un grande servizio postale deve scegliere la via migliore
Ad esempio minimizzando la distanza percorsa
Una prima approssimazione molto usata, è contare il numero di hops (balzi), cioè di stazioni incontrate nel cammino

Nel caso statico in cui la rete non cambia, un modo ovvio di operare il routing è precalcolare le distanze minime tra tutti i nodi della rete
L'ambito più generale è invece il caso dinamico, dove la rete può essere può cambiare in ogni istante.

Il flooding
L'"alluvione" è un mezzo potentissimo per fare il routing
In ogni pacchetto viene ritrasmesso a tutte le reti (tranne quella da cui è arrivato)
Mezzo potentissimo sia nel bene e nel male

Cominciamo dai lati negativi: ovviamente la rete col flooding semplice verrebbe sommersa

Occorrono quindi metodi per il "controllo delle acque"
una tecnica è l'hop counting: si associa un numero massimo di hops ad ogni pacchetto dopo i quali il pacchetto muore
Un'altra tecnica alternativa è il tracking: tenere traccia dei pacchetti che sono stati già trasmessi, e non ritrasmetterli

Invece di tenere tutta la lista dei pacchetti, si possono tenere liste separate per ogni router, tenendo un contatore speciale per la lista dei pacchetti 0-N che sono già stati ricevuti
Anche con queste tecniche, il flooding ha l'ovvio svantaggio che gebera una quantità enorme di dati.

Lati positivi
A prima vista il flooding può sembrare una tecnica grossolana
Ma funziona senza alcuna modifica sia per messaggi point-to-point che per messaggi broadcast e multicast
Inoltre il flooding sceglie sempre la via migliore
Ed infine uno dei più grandi vantaggi del flooding: è robustissimo rispetto alle modifiche della rete.
In realtà è facile dimostrare che è il più robusto sistema di routing possibile
utilissimo in quei casi in cui il carico di rete non è molto alto, ma o c'è topologia di rete estremamente variabile,
o è critico che un messaggio svanisca nel nulla.
Ad esempio, ambito militare

Distance Vector Routing

Mentre gli algoritmi di tipo Link State prevedono che ogni router sia informato dei cambiamenti occorsi nell'intera topologia della rete,
i protocolli basati su Distance Vector - come RIP ed EIGRP - sono invece più leggeri: ogni router misura la distanza
(secondo una metrica che può includere vari fattori) che lo separa dai nodi adiacenti ricevendo i dati dai router vicini.
A partire da tali dati, utilizzando l'algoritmo di Bellman-Ford, il router costruisce una tabella che associa ad ogni destinazione conosciuta:

- la stima della distanza che lo separa dalla destinazione
- il primo passo del percorso calcolato

Periodicamente poi il router aggiorna le misure di distanza dai router adiacenti e comunica la propria tabella ai vicini.
Dopo sufficienti scambi di informazioni, ciascun router potrà avere una riga per ogni altro nodo nella rete.

Pro e contro Distance Vector Routing

E' veloce a recepire le "buone notizie"

```
A-----B-----C-----D-----E
      °     °     °     °   Initially
      1     °     °     °   After 1 exchange
      1     2     °     °   After 2 exchanges
      1     2     3     °   After 3 exchanges
      1     2     3     4   After 4 exchanges
```

Quanto questo routing si comporta bene rispetto alle "buone notizie"
tanto male, dualmente, si comporta male con le "cattive notizie"

Il problema del count-to-infinity

```
A-----B-----C-----D-----E
      1     2     3     4   Initially
      3     2     3     4   After 1 exchange
```

3	4	3	4	After 2 exchanges
5	4	5	4	After 3 exchanges
5	6	5	6	After 4 exchanges
7	6	7	6	After 5 exchange
7	8	7	8	After 6 exchanges
	.			
	.			
	.			
∘	∘	∘	∘	

Routing fase 2: il Link State Routing

All'inizio, come nell'altro tipo di routing si costruisce informazione locale
Ogni nodo quindi trova quali sono i suoi vicini

Poi misura quanto lontani sono tramite dei pacchetti ECHO
Quando ha informazione sui suoi vicini ogni nodo costruisce un pacchetto che contiene tutta questa informazione più altra e la manda a tutti gli altri

L'idea è quindi che ogni nodo riceverà i "mattoncini lego" corrispondenti alle informazioni locali di ogni altro nodo
In tal modo potrà ricostruire una mappa completa della rete e quindi calcolare i percorsi migliori

Restano da sistemare varie questioni
Come si fa il broadcast? Si usa il flooding

C'è però una differenza: questo flooding andrà ripetuto ogni tanto per fare il refresh della rete

E Murphy?
Contro Murphy si usano numeri di sequenza e fading in ogni pacchetto link-state

Stiamo sprecando più banda però proprio come nel caso del flooding, quest'uso aggiuntivo della banda (a livello globale) ci permette di essere molto più robusti (a livello locale)

Problemi del routing

Effetto see-saw (effetto altalena)
Nel calcolo della distanza con un vicino, abbiamo visto che si considera il tempo di trasmissione
Ma si considera anche il carico?
Senza considerare il carico, rischiamo l'effetto di congestione

L'effeto see-saw fa avere un effetto oscillatorio molto brutto, che ovviamente porta problemi non da poco

Ricapitolando
Ci sono vari parametri da considerare, che nel complesso definiscono la qualità di una rete: sarebbe utile quindi avere una visione complessiva

Quality of Service

la qualità del servizio (QoS) è una serie di parametri che dettano appunto la qualità del servizio offerto
Nell'ambito delle reti, tipicamente la QoS è data da 4 parametri principali

- Reliability (Affidabilità)
- Bandwidth (Banda)
- Delay (Ritardo di trasmissione)
- Jitter

Jitter
Misura il grado di variazione (deviazione standard) nei tempi di arrivo dei pacchetti

QoS per applicazioni

	R	B	D	J
E-mail	H	L	L	L
File transfer	H	L	L	M
Web access	H	M	L	M
Remote Login	H	M	M	L
Video on demand	L	M	H	H

Soluzioni per il QoS
Riguardo alla Reliabilty abbiamo già visto soluzioni come error detection e error correction
Le altre misure derivano da alcune cause, che devono essere risolte se si vuole mantenere il QoS

Causa: La congestione
Una delle cause principali è la congestione, cioè quando la capacità della linea o di qualche stazione si satura
Gestire la congestione è molto meno ovvio di quello che sembra

Dietrich Braess
Aggiungere capacità ad una rete può portare ad una diminuzione della performance

Congestion Control

Il choke packet
Se un router si accorge che c'è una congestione, può inviare un pacchetto speciale a chi sta inviando dati dicendo di ridurre la trasmissione
Tipicamente dunque, un host ad esempio dimezza il suo data rate non appena riceve un choke rate
Si usa il fading come modo per uscire dal choke
Un altro problema dovuto all'entrata nel choke

A-----B-----C-----D-----E

C'è congestione sulla linea tra A ed E
-> B, C e D inviano dei choke ad A
A riceve il primo choke da B: -50% (0.25)
Riceve il secondo da C: -50% (0.125)
Riceve il terzo da D: -50%
A ha dimezzato la banda fino al 12.5%

Si fa fading alla rovescia, nel senso che non appena riceviamo un choke, per un certo periodo di tempo si ignorano tutti gli altri pacchetti choke che vengono dalla stessa destinazione

Se la rete è grande una richiesta di choke può metterci troppo tempo per sistemare le cose
una soluzione è il choke hop-by-hop

Hop-by-Hop Choke

Il choke hop-by-hop non agisce solo su chi sta inviamo dati, ma agisce anche mentre passa, rallentando i routers

Altra tecnica: Buffering
Ripsetto al QoS, il buffering non risolve la congestione e quindi il delay, ma riduce il jitter
Ovviamente può essere gestito dinamicamente (vedi multimedia sul web)

Buckets
I "buckets" sono essenzialmente una specie di filtri, che servono a garantire buone proprietò QoS

Leaky Buckets

Letteralmente il "secchio che perde" è un tipo di filtro che garantisce un data rate massimo costante (quindi evita i burst che possono sovraccaricare la rete)

Oltre il Leaky Bucket
Il Leaky Bucket ha il vantaggio/svantaggio che il max data rate è sempre costante

Il Token Bucket

Il Token Bucket genera ogni certo intervallo di tempo un token
I pacchetti in arrivo possono uscire solo se "bruciano" un token disponibile
Quindi se il traffico per un certo periodo è lento, ma poi c'è un burst, si riesce a gestirlo meglio consumando i token che si erano accomulati

Load Shedding
Un'altra tecnica è quella derivata dalle reti elettriche, estrema ma molto diffusa
Il Load Shedding fa si che in caso di sovraccarico alcuni pacchetti semplicemente vengano buttati via
Ci sono due modi principali

Wine
- "Old is better than new"
 Ad esempio se ho un remote login, conviene buttare via i pacchetti più nuovi rispetto ai vecchi

Milk
- "New is better than old"
Ad esempio in un file multimediale conviene buttare via un pacchetto vecchio: anche se c'è qualche stato nella trasmissione,
l'importante è che "the show must go on"!

Lo strato di rete di Internet

Dopo aver visto varie tecniche per il routing e il QoS, è arrivato il momento di vedere più nel dettaglio i protocolli di Internet

IP: Internet Protocol

"TCP/IP" o "TCP" / "IP"
1969: ARPANET
Usava il Network Control Protocol (NCP)
      -> inadatto
1974: Transmission Control Protocol (TCP)
1978: Divisione in TCP e IP

L'header di IP

[Version | IHL | Type of service | Void | Total lenght | Identification | Evil Bit |DF | MF | TTL |Protocol| Header Checksum | Source Address | Destination Address]

Ha una parte fissa di 20 bites
La prima parte è la versione di IP usata
La seconda parte è la lunghezza dell'header (IHL - IP Header Lenght)
La terza è il tipo di servizio (campo adatto per la selezione della QoS) anche se spesso ignorato
C'è poi la lunghezza totale del datagramma IP
L'identification serve a identificare se c'è stata frammentazione dei dati in più datagrammi
Il campo MF (More fragment) segnala l'ultimo frammento
Il campo DF (Don't fragment) impone la non-frammentazione dei dati
Infine il cosidetto "Evil Bit"
      - Standardizzato in IETF RfC3514
Il time to live (TTL) è l'età massima del pacchetto
Il campo protocol indica il protocollo di più ato livello che sta usando IP
C'è poi un checksum per l'header, calcolato banalmente tramite somme in complemento a uno
Poi, gli indirizzi IP di chi manda e di chi riceve il datagramma

DoD - Department of Defence of USA

DoD e l'uso militare

ARPANET -> Internet e MILNET
La rete non deve essere sottoposta al pericolo di sovraccarico
      -> niente traccia in IP degli errori intermedi
      -> Niente connessione persistente
Robustezza
In battaglia la perdita di un nodo è prassi comune: le perdite possono essere rendicontate successivamente, l'importante è che la rete sia operativa
      -> la comunicazione IP si riconfigura in caso
Modelli Open e Closed World
Nonostante gli sforzi per rendere TCP/IP robusto ad attacchi, il sistema è stato pensato per l modello closed-world (MILNET)
      -> attacchi esterni
Quando l'attacco è interno (open world model) ci sono molte vulnerabilità ad esempio, IP gestisce male la congestione del traffico

Riflessione: questo spiega alcuni dei problemi di Internet (e i non-problemi di MILNET da quanto si sa)

I limiti di estendibilità

Esempi
"640k ought to be enough for everybody"

Vediamo meglio i limiti intrinsici di IP a livello di estendibilità
IHL è di 4 bit, quindi la lunghezza massima dell'header è di 60 bytes
-> meno i 20 bytes dei campi fissi, fanno 40 bytes per le parti di estensione opzionale
TTL (Time to Live): 255 hops massimi
Può essere un problema se la rete è mal designata e non gerarchica
Comunque il router può intervenire e riaumentare il TTL (o, non diminuirlo) quindi non grosso problema
Gli indirizzi: 32 bits
Questo si è rilevato essere un grandioso problema
Ci deve essere una autorità centralizzata che li assegna
Potremmo assegnare gli indirizzi uno a uno, ma in questo modo, i router dovrebbero mantenere delle tabelle mostruosamente grande

La soluzione è quella di avere anche negli indirizzi IP una struttura gerarchica: invece di assegnare un solo indirizzo alla volta, assegniamo interi blocchi di indirizzi alle varie organizzazioni.

Indirizzi IP inizialmente

Quindi all'inizio di Internet si è pensato di avere vari tipi di blocchi a seconda della grandezza della rete richiesta da un'organizzazione
-> metodo classfull organizzation

```
A [ 0 | Network |           Host            ]  (7 + 24) 128 reti possibili con indirizzi di 24 bits
B [ 10 |   Network     |       Host          ]  (14 + 16) 16384 reti possibili con indirizzi di 16bit
C [ 110 |      Network      |    Host        ]  (21 + 8) 255 reti possibili con indirizzi di 8 bits
D [ 1110 | Multicast Address   |   Host      ]
E [ 1111 | Reserved for future               ]
```

Il problema è che le organizzazioni hanno comprato blocchi più grandi di quelli necessari non sapendo se in futuro potessero tornare utili

Manca una taglia intermedia (ad esempo un migliaio)
-> la maggioranza ha chiesto una classe B sprecando spazio!

Confrontate con Ethernet (3 bytes + 3 bytes)

La soluzione?
CIDR: Classless InterDomain Routing
Classless: si v oltre le vecchie classi di Internet
Ora i blocchi sono di lunghezza variabile

Lunghezza variabile?
Occorre informazione aggiuntiva su quanto grande è il blocco di indirizzi
Questa informazione è contenuta in una maschera (mask) di bits

Questa maschera di 32 bits ha tanti 0 quanto è largo il blocco di indirizzi, mentre gli altri bit sono a 1

Esempio
Chiediamo un blocco di 2048 indirizzi (11bits)
Ci viene assegnato ad esempio come primo indirizzo 194.24.0.0 con maschera composat da 11 bits a 0 e 21 bits a 1
La maschera di solito si scrive mettendo gli 1 come bits più significativi

Tabelle di routing

Il CIDR permette ai router una gestione abbastanza efficiente, usando le cosiddette aggregate entries
Se ci sono varie classi di indirizzi che devono venire indirizzati allo stesso router, possono essere combinate in un'unica entrata se hanno un prefisso comune

Cosa succede se c'è un'altra organizzazione con lo stesso prefisso che invece deve essere mandata ad un altro router?
Si usa la regola del longest matching, cioè l'entrata con il prefisso di rete più lungo ha la priorità

Il vero salvatore dello spazio IP non è stato tanto il CIDR quanto un'altra tecnologia, la tecnologia NAT

NAT - Network Address Translation

L'idea è di simulare una intera sottorete usando un solo indirizzo IP
Internamente la rete funziona con degli indirizzi IP interni, che sono invisibili all'esterno
All'esterno invece, la rete appare come un singolo indirizzo IP!
Ogni pacchetto che esce dalla rete perde quindi il suo indirizzo IP, e viene sostituito dall'unico indirizzo del NAT
Alcuni indirizzi sono riservati per le reti interne al NAT e non possono essere usati come normali indirizzi Internet

- 10.0.0.0 (rete da 16777216 host)
- 172.16.0.0 (rete da 1048576 host)
- 192.168.0.0 (rete da 65536 host)

Il vero punto del NAT è però un altro: l'altro verso dei messaggi
Quando un messaggio è inviato da fuori a un computer della rete interna, come fa il NAT box a mandarlo al computer giusto?
Bisogna aspettare per la risposta

Il resto del Netwotk Layer
Come abbiamo visto, IP è il protocollo di base che serve a trasferire dati nello strato router
Ci sono altri protocolli che servono a far funzionare lo strato network:

- ICMP
- ARP
- DHCP

ICMP - Internet Control Message Protocol
E' il protocollo di controllo di Internet per gestire eventi inaspettati
I mesaggi ICMP vengono impacchettati dentro IP

Esempi:

Messaggi - Descrizione
Destination Unreachable - Packet could not reach destination
Redirect - Teach a router about geography

Supponiamo di dover inviare un pacchetto IP: l'indirizzo del destinatario è nello spazio IP (strato network)
Lo strato data link però tipicamente usa un altro spazio di indirizzi
Ad esempio Ethernet usa uno spazio di indirizzi di 48 bits
Serve un modo per gestire le due corrispondenze tra spazio network e spazio data-link

La soluzione è data dal protocollo DHCP che sta per Dynamic Host Configuration Protocol
Quando un macchina vuole sapere il suo indirizzo IP, manda un pacchetto DHCP di tipo DISCOVERY
Il gestore DHCP della rete risponde con l'indirizzo corrispondente

Il verso opposto da IP a MAC
Una soluzione potrebbe essere, come per DHCP, avere da qualche parte un server o simil-router a cui rivolgersi
ma la situazione è diversa: con gli indirizzi IP abbiamo pieno controllo e possiamo decidere noi, ed incaricare un'autorità centralizzata
nel caso dei MAC Address gli indirizzi sono già assegnati

La soluzione adottata è opposta a quella DHCP: si gestisce la corrispondenza a livello locale
Invece di avere un server centrale che gestisce tutto facciamolo fare indipendentemente a ciascuna macchina
Per comunicare con le altre, manderò messaggi in broadcast

ARP - Address resolution Protocol
Si occupa di gestire queste corrispondenze
Quando vuole inviare un messaggio alla macchina con un certo indirizzo IP X, manda un messaggio ARP chiedendo chi ha l'indirizzo X
Solo la macchina con indirizzo X risponde con un messaggio ARP di ACK

Possiamo anche ottimizzare questo processo, visto che quello che conta sono le tabelle locali ARP
Ad esempio, tipicamente quando un a macchina A comunica con una macchina B, è molto probabile che poi B comunichi con A

Una conseguenza di questa politica è che, tipicamente quando una macchina entra in una rete, invia un pacchetto ARP chiedendo il suo stesso indirizzo IP

Se c'è una nuova informazione su altre assegnazioni IP -> MAC, questa ha priorità sulla vecchia
Ciò permette una gestione dinamica della rete

Oltre IP
Abbiamo già visto la scarsità di indirizzi nello spazio IP
Anche se CIDR e NAT hanno dato fiato +, la situazione alla fine collasserà
Occorre una soluzione definitiva
      -> una nuova versione di IP: IPv6
Gli indirizzi ora sono di 16 bytes invece di 4 bytes

Alcuni volevano meno bytes, altri di più, altri un numero variabile...
Siamo sicuri che con un numero di nuovo fisso gli indirizzi basteranno?
16 bytes corrispondo a 2^128 indirizzi.

IPv6 non ha più il campo checksum
    -> è un protocollo totalmente unreliable
L'idea è che il controllo dell'errore se serve, verrò fatto dai protocolli superiori

Lo strato di trasporto

Sempre più in alto...
Ha una differenza con lo strato network. Siccome è un'astrazione ulteriore, fa multiplexing delle singole risorse network (le macchine, identificate dagli indirizzi IP) tramite le porte (ports)
IP + porta = socket, cioè la divisione in "rete logica", a livello, transport, di una singola entità di rete
Cominciamo quindi con l'applicazione più semplice, UDP, che sta per User Datagram Protocol

Essenzialmente è un IP con le porte aggiunte

Header UDP

[Source Port | Destination | Port UDP | Lenght UDP | Checksum]

UDP è dunque sempre un protocollo connection-less, e si usa in tutte quelle situazioni dove serve inviare messaggi brevi in cui non è necessario
aprire una connessione dedicata.

Esempio d'uso: il DNS - Domain Name Server
Funziona gerarchicamente, usando i resolver dei TLD, quando non c'è informazione disponibile
L'altro protocollo principale di Internet a livello di transport TCP

TCP - Transmission Control Protocol
Mentre in UDP è connection-less, TCP è connection-oriented
Le connessioni TCP sono Full Duplex e Point-to-Point

L'header TCP

[Source Port | Destination Port | Header Lenght | URG | ACK | PSH | RST | SYN | FIN | Window Size | Checksum | Urgent Pointer]
C'è poi un a campo TCP header lenght
C'è poi una serie di flag
 - RST indica che c'è un problema nella connessione e serve fare il reset
 - FIN termina una connessione attiva
    - La terminazione avviene con ACK
    - In realtà termina solo in uno dei due versi
URG e PSH indicano la priorità (URG > PSH)
    - Il numero d'ordine dell'ultimo byte URGente da considerare è messo nel campo urgent pointer
Poi ci sono i campi Sequence Number e Acknowledgement number per la classica lotta contro Murphy
Il campo Window Size gestisce la taglia della sliding window; quindi TCP usa le sliding windows a grandezza variabile (Max 2^16-1)
Di default usa sliding windoes con Go-Back-N

Poi c'è il campo checksum per il controllo dell'errore
Ed infine il flag SYN, che serve ad aprire una connessione

La connessione TCP: 3-way handshake
Si apre una connessione in 3 passaggi
- Io chiedo di aprire una connessione
- Il server risponde con un ACK in piggyback chiedendo di aprire il mio lato
- Noi rispondiamo e cominciamo la connessione

SYN (SEQ = x) ->
SYN(SEQ=y, ACK = x + 1) <-
(SEQ = x + 1, ACK = y + 1) ->

NAT!

Ora che abbiamo visto TCP e UDP possiamo capire come funziona la magia del NAT
Usa le porte
Ricordate che le porte essenzialmente fanno multipexing a livello transport
NAT usa le porte per fare multiplexing a livello network: quindi abusa il concetto di porta e fa si che uno spazio di porte (1-65636) possa essere associato a molti indirizzi IP invece che uno solo
Notare:
Anzitutto che si mescolano i livelli
E, poi, peggio che trasforma Internet in una rete connection-oriented, con tutti gli svantaggi conseguenti: se il NAT box crasha, a differenza di un router, tutte le connessioni saranno perse!

La struttura Internet

La struttura internet è mondiale... quindi di chi è Internet?
Molte parti sono in mano a governi locali, molte altre invece sono in mano private
Come interagiscono queste parti?
Formano Internet, quindi di certo sono integrate!
No
Ogni gestore ha la libertà di decidere la sua politica di integrazione con il resto della rete
i livelli sono bene o male semplicemente reti sempre più grandi, ma c'è un livello particolare
Il livello AS, quello degli autonomous system

Livello AS
ogni gestore definisce la politica di integrazione del suo AS con gli altri
"Gestire la politica" significa essenzialmente definire come funzionano i rapporti tra i vari AS

Routing tra AS
Il routing tra AS viene fatto da un protocollo speciale: BGP
Border Gateway Protocol
Esmpio:
Ogni pacchetto che proviene dal governo USA non deve mai passare per la Cina
ogni pacchetto da un sito interno a Facebook non deve passare mai per Google e viceversa!

Supponiamo di avere pacchetti di 40-64 bytes
A 100Gbit/s quanto tempo ha il router per decidere?
3-5 ns di tempo!
Superiore ai tempi di accesso della DRAM
Si usa SRAM e l'algoritmo di lookup dev'essere velocissimo ("Lulea Scheme" per comprimere)