

## **INTRODUZIONE.**

### **Rete.**

#### **Infrastruttura .**

**Host** – sistema terminale che si interconnette alla rete attraverso dei **Collegamenti (link)** come ad esempio Cavo coassiale/fibra ottica/onde elettromagnetiche.

**Router** – tra i vari collegamenti vi sono dei dispositivi di interconnessione che creano i nodi di Reti

**Frequenza di Trasmissione** connessa all'ampiezza di banda = ovvero quanti bit/s posso trasferire.

**Obiettivo** – della connessione è di trasferire informazioni e comunicare a distanza

**Protocollo** – definisce il formato, l'ordine e le regole dei messaggi scambiati tra due o più entità in comunicazione (es. TCP, IP, http etc.), si tratta di dispositivi sia hardware che software

### **Struttura.**

**Confini** – parte esterna ci sono le reti domestiche o aziendali fatte da sistemi terminali e applicazioni. Essi comunicano con due tipi di architetture:

**Client/Server** – L'host client richiede e riceve servizio da programma server in esecuzione su un altro terminale con struttura gerarchizzata (Browser, Web)

**Peer to Peer** – nodi non gerarchizzati in maniera fissa ma chi è client può diventare server e viceversa (Skype, Bit Torrent)

**Intermezzo** – Dispositivi fisici che permettono la comunicazione (Collegamenti)

**Centro** – i router interconnessi che creano poi internet etc.

**Reti di Accesso** – si indica la parte di rete destinata al collegamento fra la sede dei singoli utenti finali (terminali) e fino alla parte centrale (provider). Si distinguono per risorse:

- ☐ **Dedicate** – dedicate a una singola comunicazione
- ☐ **Condivise** – dedicate a molteplici comunicazioni e sono di tre tipologie:

**Reti di accesso residenziale** – (punto a punto) due tipologie:

**Modem Dial- Up** : si richiede la connessione al modem e realizza la connessione tra computer usando la banda fornita a bassa frequenza. Non è possibile navigare e telefonare allo stesso momento.

**DSL** - linea dedicata per la connessione = connessione permanente

**Rete di Distribuzione Telefonica** – Parte dal **permutatore** (centrale) passa per la rete primaria (~ 1 km) per arrivare all'**armadio**, poi si dirige attraverso la rete secondaria (~ 200 m) per arrivare al **distributore**, da cui attraverso il raccordo (~ 50 m) arriva alle case.

**Reti di accesso aziendale** – (locali) chiamate LAN collegano i sistemi terminali di aziende e università ad una LOCAL AREA NETWORK ovvero ad un router. A seconda dell'estensione geografica della rete si potrebbe arrivare ad ottenere una WAN (WIDE AREA NETWORK ) che collega reti con estensione geografica maggiore.

☐

### **Architetture ibride rame-fibra (FTTx)**

**FTTE** = Fiber to the Exchange

**FTTCab** = L'ONU è situato nell'armadio telefonico nelle strade, Fiber to the Cabinet

**FTTCurb** = L'ONU è situato in prossimità degli edifici dove si trovano gli utenti, Fiber to the Curb

**FTTB** = Nelle aree con edifici a sviluppo verticale, Fiber to the Building

**FTTH** = Anche nel caso di case individuali, Fiber to the Home (Onu presente nell'edificio)

**ONU(Optical Network Unit)**: converte il segnale ottico trasmesso attraverso la fibra in segnale elettrico.

### Mezzi Trasmissivi.

**Mezzo fisico** – fisicamente ciò che sta tra trasmittente e ricevente

**Mezzo guidato** – il mezzo fisico tramite il quale il segnale si propaga (fibra ottica, filo di rame, cavo coassiale etc.) □

**Mezzi a onda libera** – i segnali si propagano nell'atmosfera o spazio

### **ESEMPL.**

**Twisted Pair** – due fili di rame attorcigliati che trasferiscono il segnale

**Cavo coassiale** – due conduttori in rame concentrici, bidirezionale

**Fibra ottica** – Mezzo sottile e flessibile che conduce impulsi di luce, alta frequenza trasmissiva e basso tasso di errore, immune all'interferenza elettromagnetica.

**Canali Radio** – in un canale radio non tutte le frequenze uguali e a seconda della frequenza la trasmissione più veloce o più lenta ovvero più forte ad attraversare le cose o più lenta. Microonde, Satellitare, Wifi etc.

### Modi di trasferimento dati.

**Commutazione di Circuito** – (Circuit Switching) è un circuito dedicato per l'intera sessione che garantisce la prestazione della comunicazione ma non permette più connessioni allo stesso tempo e necessita dell'impostazione iniziale della chiamata. Per sorvolare il problema della mancata condivisione si può suddividere le risorse di rete(banda) in pezzi e ogni pezzo viene allocato ad un collegamento. Ovviamente più la banda è piccola o più sono molti i collegamenti più perdo qualità della connessione. Questa divisione si può fare in due modi :

**Divisione per FREQUENZA** – TimeDivisionMultiplex TDM – a ogni collegamento affido una frequenza

- **Divisione per TEMPO** - FrequencyDivisionMultiplex FDM - a ogni collegamento affido uno slot di tempo

**Commutazione di Pacchetto** – (Packet Switching) il flusso dei dati viene suddiviso in pacchetti di bit . Non c'è una linea dedicata e le risorse vengono utilizzate a seconda della necessità (Moltiplicazione Statistica). Utilizzo delle risorse più dinamico ma potrebbe incorrere in congestione di pacchetti = ritardi. Questo trasferimento inoltre utilizza il metodo **Store and Forward**: il commutatore deve ricevere l'intero pacchetto prima di poterlo trasmettere sul collegamento in uscita. Ottima per dati a "burst", ma sono necessari protocolli per il trasferimento affidabile dei dati e per il controllo della congestione.

**Moltiplicazione Statistica**: non vi è una sequenza fissa di chi utilizza risorse (condivisione di risorse esu richiesta), l'ordine è casuale a seconda della necessità. = maggiore utilizzo efficiente della banda. Il nodo quindi memorizza i pacchetti (Buffer) ma se lo spazio non basta il pacchetto potrebbe essere perso. = **BEST EFFORT**

**Ritardi**: i pacchetti si accodano nel buffer dei router, se il tasso di arrivo è maggiore della capacità di inoltro del collegamento si crea una coda e si creano dei ritardi:

- **Ritardo di elaborazione** - ritardo nel controllo di errori e nell'instradamento
- **Ritardo di accodamento** – ritardo di trasmissione (può essere solo stimato), livello di congestione  
 $Ritardo = La / R$  dove  $a$  = tasso medio di arrivo pacchetti , tanto più si avvicina ad 1 tanto più il ritardo di attesa
- **Ritardo di trasmissione** – dipende dalla velocità della linea

**Ritardo** =  $L/R$  dove  $R$  = velocità/frequenza di trasmissione in bps e  $L$  = lunghezza del pacchetto in bit

□ **Ritardo di propagazione** – dipende dal tempo dei pacchetti

**Ritardo** =  $d/s$  dove  $d$  = lunghezza collegamento fisico,  $s$  = velocità di propagazione del collegamento □

$$d_{\text{link}} = d_{\text{elab}} + d_{\text{queue}} + d_{\text{trasm}} + d_{\text{prop}} \quad \text{Ritardo di link}$$

$d_{\text{elab}}$  = ritardo di elaborazione (**processing delay**): In genere pochi microsecondi, o anche meno

$d_{\text{queue}}$  = ritardo di accodamento (**queuing delay**): Dipende dalla congestione

$d_{\text{trasm}}$  = ritardo di trasmissione (**transmission delay**): Significativo sui collegamenti a bassa velocità

$d_{\text{prop}}$  = ritardo di propagazione (**propagation delay**): Da pochi microsecondi a centinaia di millisecondi

### Perdita di pacchetti

Una coda ha capacità finita: Quando il pacchetto trova la coda piena, viene scartato (e quindi va perso)

Un pacchetto perso può essere ritrasmesso dal nodo precedente, dal sistema terminale che lo ha generato, o non essere ritrasmesso affatto

### Altri valori :

**Throughput** – Frequenza alla quale i bit sono trasferiti tra mittente e ricevente in bit/s . Esso può essere calcolato in un preciso istante (**Istantaneo**) e in un periodo (**Medio**). Ma se il mio collegamento è fatto di diversi collegamenti con frequenze diverse alla fine quello con portata minore determinerà la portata complessiva (**collo di bottiglia anche detto Bottleneck**) .

### Internet.

Rete delle reti con struttura gerarchica e standardizzata :

Esistono provider ISP di tre livelli diversi: Gli ISP di livello 1 sono direttamente connessi a ciascuno degli altri ISP di livello 1, gli ISP di livello 2: ISP più piccoli (nazionali o distrettuali) si possono connettere solo ad alcuni ISP di livello 1 e possibilmente ad altri ISP di livello 2, gli ISP di livello 3 o ISP locali sono gli ISP finali a cui si collega l'utente.

**Internet Engineering Task Force** – coloro che sviluppano gli standard e li pubblicano poi in

**RFC = Request for Comment** – documento di testo con regole con tempo di vita, se viene commentato ed eseguito allora approvato.

**Medium access control (MAC)** regola l'accesso ai mezzi condivisi.

**Indirizzi** identificano il punto di accesso alla rete (interfaccia)

### ARCHITETTURA A STRATI.

La rete è un sistema complesso e per organizzare il sistema si utilizza una stratificazione gerarchica delle funzionalità. In questa maniera ogni processo della comunicazione risulta indipendente dagli altri, lo si può aggiornare o testare individualmente. I vari strati vengono poi collegati dai protocolli che effettuano delle chiamate ai servizi offerti dallo strato inferiore.

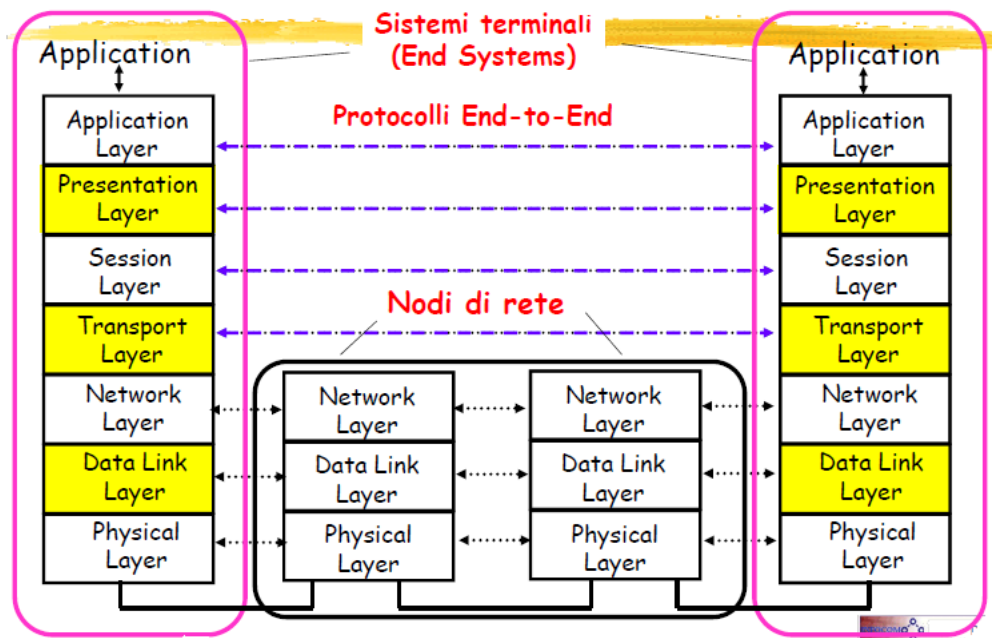
Il modello di riferimento per quest'architettura è :

**OSI = Open Systems Interconnection** : che definisce un modello di rete a 7 strati e i protocolli di ogni strato per permettere l'interconnessione tra sistemi che utilizzano anche architetture diverse.

Oggi questo sistema è stato superato dal TCP/IP che utilizza però gli stessi concetti.

Due terminali comunicano attraversando i loro 7 strati di funzioni. Alcune funzioni avvengono solo (end to end) nei terminali meno onerose ma meno immediate, altre vengono ripetute anche nei

nodi (sezione per sezione) che vengono inseriti per motivi di distanze e topologia della rete.



### **STRATO FISICO (BIT)**

Strato che effettua operazioni sezione per sezione, ma viene implementato in maniera diversa a seconda del mezzo trasmissivo.

**Scopo** – trasferire bit informativi sui mezzi trasmissivi

**Caratteristiche** – Meccaniche (tipo cavo) , Elettriche / ottiche ( potenza segnali, tensione..) , Funzionali (procedure per attivare o disattivare link fisici)

### **STRATO DI COLLEGAMENTO (FRAME)**

**Scopo** – realizzare trasferimento affidabile

**Operazioni** :

- ☐ - Riceve bit dallo strato fisico e li incapsula in unità Frame
- ☐ - Controlla che i bit ricevuti son corretti ed eventuale loro correzione
- ☐ - Controlla il flusso, regolarizza la velocità alla quale i frame viaggiano
- ☐ - Controlla la condivisione delle risorse tramite il MAC

### **STRATO DI RETE (PACCHETTI)**

**Scopo** – trasferire pacchetti attraverso una serie di reti/link (**Internetworking**)

**Operazioni**:

- ☐ **Indirizzamento di rete** ( MultiCast molti , BroadCast tutti, Unique Terminal uno)
- ☐ Definisce procedure **instradamento (routing)** per determinare nei nodi i cammini di rete  
Definisce le procedure di **rilancio (forwarding)** dei pacchetti nei nodi
- ☐ **Controllo congestione** – evita sovraccarico sulla rete  
Definisce le procedure di **setup, gestione e teardown** delle connessioni di rete

### **STRATO DI TRASPORTO (SEGMENTI)**

**Scopo** – Trasferire i dati end-to-end dal processo attivo in un host al processo dell'host remoto

**Operazioni**:

- ☐ Gestisce port numbers (indirizzi interni ai sistemi terminali)
- ☐ Segmenta i pacchetti nell'invio e li riassembla all'arrivo

### **STRATO DI APPLICAZIONE (MESSAGGIO)**

**Scopo** – fornire servizi richiesti dalle applicazioni (DNS, web access, file transfer, email...)

Sono incorporati in esso due funzioni:

- **Presentation Layer** – interpretare significato dei dati ( cifratura, compressione, convenzioni specifiche della macchina)
- **Session Layer** – sincronizzare e controllare il dialogo + recupero dati

### **INTERAZIONE STRATI**

Lo strato è quindi un'entità che implementa delle funzioni e che comunica attraverso le unità dati **protocol data unit (PDU)** che rappresenta la SDU + informazioni di controllo (PCI) per eseguire le funzioni dello strato seguendo delle regole condivise dai sistemi ai quali è interconnesso.

Le entità che eseguono le funzioni di uno strato all'interno di sistemi comunicanti sono dette:

**peer processes.**

La cooperazione tra entità dello stesso strato è regolata dal protocollo di strato n (**layer-n protocol**)

Se esiste uno strato in un terminale deve esistere anche nel terminale con cui comunica.

I dati che sono ricevuti da uno strato dallo strato superiore sono denominate **Service Data Unit (SDU)**

Le SDU sono incapsulate nelle PDU nelle quali sono anche aggiunte le informazioni di controllo per l'esecuzione delle funzioni di strato.

La comunicazione tra strati è virtuale ed indiretta, infatti lo strato n+1 invoca il servizio fornito dallo strato n e gli passa le informazioni che gli servono. Questo meccanismo avviene tramite l'interfaccia.

### **Segmentation & Reassembly**

Uno strato può imporre un limite massimo alla dimensione del blocco dati che può essere trasferito, se le n-SDU superano questo limite non possono essere trasferite in un'unica n-PDU

**Lato emittente:** la SDU è segmentata in PDU multiple

**Lato ricevente:** la SDU è riassemblata a partire dalla sequenza di PDU ricevute

Alla ricezione dei bit nel lato ricevente ogni strato toglie la porzione di informazione relativa al suo strato e lo rimanda in su = Multiplazione - Demultiplazione

### **Headers & Trailers**

**PCI (Protocol control information).** Ogni protocollo usa **un'intestazione (header)** per ogni strato e un **suffisso (trailer)** sul controllo dei bit che contengono le informazioni di controllo necessarie per l'esecuzione delle funzioni di strato: Indirizzi, numeri di sequenza, flag, codici di controllo d'errore...

### **Service Access Point**

Interazione può avvenire in due modi:

**Servizio con Connessione (Connection Oriented):** due entità stabiliscono la connessione e le regole per il trasferimento e ad ogni trasferimento si attende la conferma di ricezione ed infine dichiarano terminata la connessione.

Strutturazione in tre fasi temporali:

- Negoziiazione dei parametri di trasferimento
- Indirizzamento con identificatori di connessione
- Legame logico tra i segmenti informativi scambiati

**Servizio senza Connessione:** non si stabilisce la connessione iniziale, c'è una sola fase temporale e non c'è conferma di ricezione. (**Connectionless**)

## **STRATO FISICO**

### **Trasmissione Digitale.**

I messaggi che oggi si trasmettono sono digitali (audio, video). Esistono due modi per trasmetterli:

**Informazione a blocchi** – il messaggio è strutturato in unità indipendenti (blocchi) con un numero di bit per blocco fisso.

**Informazione Stream** – trasmissione continua di bit (es. streaming).

Esse poi possono avere:

**Bit Rate** = frequenza di bit/s che riesco a trasmettere

**Costant Bit Rate(CBR)** – flussi a bit rate costante (la rete deve fornire un canale con banda minima uguale al bit rate)

**Variable Bit Rate(VBR)** – flussi con bit rate variabile nel tempo (la rete deve supportare la variabilità)

### **Delay di trasferimento di un messaggio**

L = Numero di bit in un messaggio, si riduce mediante tecniche di compressione

R = Velocità del sistema di trasmissione (bit/s), si aumenta mediante adeguate tecniche di trasmissione

$T_{prop}$  = tempo di propagazione lungo il mezzo trasmissivo

d = lunghezza del collegamento, si riduce riducendo la lunghezza del collegamento

c = velocità di propagazione sul mezzo trasmissivo (3x10<sup>8</sup> m/s nel vuoto, 2x10<sup>8</sup> m/s nei mezzi guidati)

**Ritardo totale** = Ritardo di TRASFERIMENTO minimo:  $T_{prop} + L/R = d/c + L/R$

### **Compressione**

Si riducono i bit necessari a rappresentare l'informazione, riducendo ridondanza.

**Senza perdita (Lossless)**: l'informazione originale è ricostruita esattamente: zip, GIF, fax

**Con perdita (lossy)**: l'informazione decompressa non è identica all'originale: JPEG

**Rapporto di compressione** =  $R_c = B_{orig} / B_{compr}$  (#bit originali / #bit compressi),  $R > 1$  per non perdere info, più R grande più compressione efficiente.

### **Digitalizzazione dell'analogico.**

Un segnale analogico varia nel tempo quindi ci saranno momenti in cui userò più bit e altri meno.

**Campionamento**: Divido il segnale in livelli. Ogni tot tempo prendo un campione (**sampling**) e lo approssimo al livello più vicino. A seconda del livello sul quale cade gli assegno un certo numero di bit. Il numero di bit dipende dal numero di livelli in cui divido il segnale. (3 bit = 2<sup>3</sup> 8 livelli, 8 bit = 2<sup>8</sup> 256 livelli)

Il bit rate **BR** in questo caso diventerà = #bit/campione \* #campioni/secondo e da esso dipenderà la qualità del segnale.

### **Teorema del campionamento**

La larghezza di Banda  $W_s$ (Hz) indica quanto velocemente il segnale varia nel tempo. Più essa è grande più dovro prendere campioni frequentemente per avere buona qualità.

Si giunge così al fatto che la frequenza di campionamento minima x preservare le informazioni è:

$$F_c = 2 \times W_s \text{ e da questo ottengo che } T_c = 1/F_c \text{ è il tempo di ogni quanto campione.}$$

Sebbene ci sia un minimo se campiono più frequentemente del dovuto non ho un guadagno.

### Valori per trasmissione Stream

**Ritardo (Delay):** Per ogni servizio occorre individuare il vincolo sul ritardo massimo di attraversamento della rete

**Variabilità del ritardo (Jitter):** Per ogni servizio occorre individuare il vincolo sulla variabilità massima consentita del ritardo di attraversamento della rete

**Perdita di informazioni (Loss):** Per ogni servizio occorre individuare il vincolo sulla percentuale massima di bit persi (per errori o congestione) sul totale dei bit trasmessi

**Ritmo di Picco** =  $R_p$

**Ritmo Medio** =  $R_m = a R_p$  (dove  $a$  è coefficiente di attività, vale 1 se sempre informazioni, minore di 1 se vi sono pause)

**Grado di Intermittenza**  $G = R_p/R_m$

### Sistema di Trasmissione

**Trasmittitore:** converte flusso informativo prodotto da una sorgente in un **segnale** adatto alla trasmissione

**Canale di Comunicazione:** Cavi, Fibra, Radio

**Ricevitore:** converte il segnale ricevuto in forma utilizzabile

**Possibili alterazioni :** Attenuazione del segnale / Distorsione segnale / Rumore additivo / Interferenza con altri segnali

### Posso trasmettere

**Segnale Analogico** in cui il ricevente deve ricostruire tutti i dettagli, maggiore possibilità di errore, limite di distanza.

Nel caso di trasmissioni analogiche a lunga distanza devo far uso di **ripetitori**.

Ogni ripetitore ha lo scopo di **rigenerare** il segnale in uscita in modo che sia quanto più possibile simile a quello ricevuto in ingresso

La rigenerazione è non ideale: le distorsioni non sono completamente eliminabili, il rumore e le interferenze sono solo parzialmente rimosse, per questo la qualità del segnale diminuisce al crescere del numero di ripetitori

**Segnale Digitale** in cui il ricevente deve ricostruire solo i livelli discreti del segnale, probabilità di errore piccola, possibili comunicazioni a lunga distanza.

Nel caso di trasmissioni analogiche a lunga distanza devo far uso di **rigeneratori**.

Un rigeneratore ricostruisce la sequenza iniziale di bit e la ritrasmette sulla tratta successiva

Il segnale rigenerato è in pratica identico a quello originale

### Trasmissione ad impulsi

**Obiettivo:** Rendere massimo il rate di trasmissione degli impulsi in un canale, ovvero rendere  $T$  il più piccolo possibile facendo attenzione che gli impulsi brevi e ravvicinati potrebbero sovrapporsi (**interferenza intersimbolica**).

Esiste un minimo di frequenza affinché questo non accada:

**Frequenza di Nyquist:**  $F = 2 \times W_c$  dove  $W_c$  è la larghezza della banda del canale (**bandwidth**)

Se non c'è interferenza intersimbolica e non c'è rumore il massimo rate di trasmissione è  $R_{max} = 2W_c$

### Larghezza di banda di un canale trasmissivo

Se il segnale di ingresso ad un canale è una sinusoide di frequenza  $f$  allora l'uscita sarà una sinusoide della stessa frequenza  $f$  attenuata di un fattore  $A(f)$  che dipende dalla frequenza  $f$

**Segnale trasmesso:**  $X(t) = a \cos(2\pi ft)$

**Segnale ricevuto:**  $Y(t) = A(f) a \cos(2\pi ft)$



$A(f) \approx 1 \Rightarrow$  il segnale transita inalterato;  $A(f) \approx 0 \Rightarrow$  il segnale è bloccato

### Trasmissione multilivello (PAM)

Raggruppa i bit in parole di dimensione  $N = \log_2 M$

**M: numero di livelli; N: numero di bit trasmessi in un unico impulso;**

Assegna ad ogni parola di N bit un livello tra gli M disponibili:

I livelli adiacenti corrispondono a parole di codice che differiscono per un solo bit (Codifica di Gray)

Un errore tra due livelli adiacenti comporta un errore su un solo bit

Se il segnale può assumere  $M = 2^m$  livelli, si ha un Bit Rate uguale a:

$$m \text{ bit/impulso} \times 2W_c \text{ impulsi/sec} = 2mW_c \text{ bit/s}$$

Il bit rate può essere aumentato incrementando il numero di livelli, tuttavia il segnale include il rumore additivo che limita il numero di livelli che possono essere usati.

### Rumore

Una misura del rumore consiste nel rapporto segnale-rumore (*signal-to-noise ratio*)

$SNR = \text{Potenza media segnale} / \text{potenza media} = 10 \log_{10} (SNR)$

Se l'SNR è elevato il segnale ha un rumore trascurabile, se è basso l'errore è evidente.

### Limite di Shannon alla capacità di un canale

Dato un canale con banda W e rumore Gaussiano e fissato un valore di SRN, il massimo rate di trasmissione raggiungibile per cui è ottenibile un BER arbitrariamente piccolo è dato da:

$C_{\max} = W_c \log_2 (1 + SNR) \text{ bit/s}$  che è una funzione della larghezza di banda e del rapporto segnale rumore

Gli errori nella rivelazione del segnale ricevuto appaiono quando la separazione tra i livelli del segnale è comparabile con il livello di rumore

Il **Bit Error Rate (BER)** aumenta quando diminuisce l'SNR = #bit giusti / #bit complessivi

Il rumore pone un limite al numero di livelli che possono essere utilizzati nella trasmissione di impulsi e quindi un limite al bit rate in trasmissione

Se il bit rate di trasmissione R è inferiore a  $C_{\max}$  ( $R < C_{\max}$ ) è possibile ottenere un BER arbitrariamente piccolo.

Se  $R > C_{\max}$ , non è possibile ridurre il BER a valori arbitrariamente piccoli

La capacità  $C_{\max}$  può essere utilizzata come una misura di riferimento per stabilire quanto un sistema di trasmissione reale è vicino alle migliori prestazioni possibili

### **POTENZA DI UN SEGNALE FORMULE**

#### **Sviluppo in serie di Fourier per un segnale periodico**

#### **Teorema di Parseval**

**Trasformata di Fourier:** Il generico segnale è costituito da una composizione di sinusoidi che rappresentano le frequenze. Maggiori sono le frequenze maggiore il segnale si avvicina all'originale.

#### Digitalizzazione di segnali analogici

**Campionamento:** estrazione di campioni del segnale  $x(t)$  uniformemente spaziatati nel tempo

**Quantizzazione:** codifica di ogni campione con una stringa di bit (con precisione finita)

**Compressione:** applicazione di metodi di riduzione del bit rate

### Larghezza di banda di un segnale

Un segnale reale  $x(t)$  si dice limitato in banda  $[-W, W]$  se la sua trasformata di Fourier  $X(f)$  è nulla per  $f \notin [-W, W]$  dove W è la Larghezza di banda del segnale  $x(t)$



### **Teorema del campionamento**

Un segnale limitato in banda  $W_s$  può essere perfettamente ricostruito a partire dalla sequenza dei suoi campioni se la frequenza di campionamento:

$$F_c = 1/T > 2W_s \quad (\text{Frequenza di Nyquist})$$

### **Canali di comunicazione**

Unione dei mezzi trasmissivi e dei dispositivi (elettronici o ottici) che sono attraversati dal segnale lungo il percorso tra sorgente e destinazione: Equalizzatori, amplificatori, ecc.

Spesso si usa il termine **filtro** per indicare gli effetti del canale sul segnale che lo attraversa

**Canale**: set di sinusoidi che passano inalterate nel canale

**Segnale**: set di ampiezze che sono lo spettro del segnale. La parte più alta della banda canale costituisce la banda segnale.

**Canale con banda infinita**: canale sul quale passano tutte le frequenze ma non è ideale perché gli impulsi (funzione che trasporta i bit) si allargano e la coda di un segnale finisce sull'inizio di un altro ottenendo un'interferenza intersimbolica.

### **Filtro passa basso ideale**

Tutte le frequenze  $f < W_c$  non subiscono attenuazione e sono ritardate di  $\tau$  secondi, invece le frequenze  $f > W_c$  vengono bloccate

### **Filtro passa basso reale**

Le frequenze sono attenuate in modo diverso e subiscono ritardi diversi

### **Canale passabanda**

Alcuni canali di comunicazione si comportano come un filtro passa-banda quindi bloccano le basse e le alte frequenze

La larghezza di banda è l'ampiezza dell'intervallo di frequenze per cui il segnale in uscita ha una potenza non trascurabile

### **Distorsione**

Il canale introduce sul segnale in ingresso  $x(t)$  due effetti:

Se la risposta in frequenza non è "piatta", le componenti di frequenza del segnale d'uscita  $y(t)$  avranno ampiezza diversa rispetto a quelle del segnale d'ingresso  $x(t)$

Se la risposta in fase non è "piatta", le componenti di frequenza del segnale d'ingresso  $x(t)$  subiranno ritardi diversi

### **Risposta impulsiva di un sistema lineare**

La risposta impulsiva  $h(t)$  di un sistema lineare e permanente (filtro) è definita come l'uscita  $y(t)$  del sistema quando all'ingresso è applicato l'impulso unitario  $x(t) = \delta(t)$

## **FORMULE + CONVOLUZIONE**

### **Risposta impulsiva di un filtro ideale**

Per canali ideali passa basso di larghezza di banda  $W_c$ , la risposta impulsiva è rappresentata dalla funzione impulso di Nyquist  $h(t) = s(t - \tau)$ , dove  $T = 1/2 W_c$ , e  $s(t)$  vale zero in  $t = kT$ ,  $k = \pm 1, \pm 2, \dots$

### Caratterizzazione del rumore

Il rumore termico è inevitabile.

Il rumore può essere caratterizzato mediante la densità di probabilità dell'ampiezza dei campioni

La distribuzione del rumore è Gaussiana

### Modulazione (Trasportare bit attraverso un segnale)

**Canale Passa Basso – Passa Banda** : Ammettiamo di avere un segnale ad una certa frequenza, ma vogliamo spostarlo e centrarlo attorno ad una frequenza che chiamiamo Frequenza Portante.

Quest'operazione si chiama **Modulazione in Banda** ed il canale che passa per frequenze tutte centrate attorno ad  $F_c$  viene chiamata non più Canale Passa Basso ma Canale Passa Banda.

I dispositivi che effettuano la modulazione vengono chiamati Modulatori Numerici (es Modem) ed essi moltiplicano la frequenza base per la sinusoidale  $\cos(2 \pi f_c t)$

Quest'operazione è necessaria perché ogni canale ha una frequenza portante differente, e devo poter trasmettere il mio segnale a canali diversi con  $F_c$  diversi.

Quest'operazione diminuisce il limite della velocità della banda che invece di essere  $2W_c$  diventa  $W_c$ .

### Modulazione di Ampiezza(ASK)

Mappa ogni bit informativo nell'ampiezza di una sinusoidale a frequenza  $f_c$  :

- “1” trasmissione del segnale sinusoidale
- “0” nessun segnale

Il demodulatore individua i periodi in cui è presente il segnale e i periodi in cui il segnale è assente

### Modulazione di Frequenza(FSK)

Mappa ogni bit informativo nella frequenza di un segnale sinusoidale

- “1” trasmissione di un segnale di frequenza  $f_c + \delta$
- “0” trasmissione di un segnale di  $f_c - \delta$

Un demodulatore individua la potenza intorno alle frequenze  $f_c + \delta$  oppure  $f_c - \delta$

### Modulazione di Fase (PSK)

Mappa ogni bit informativo nella fase di un segnale sinusoidale:

- “1” trasmissione del segnale  $A \cos(2 \pi f_t)$   $\rightarrow$  fase 0
- “0” trasmissione del segnale  $A \cos(2 \pi f_t + \pi)$   $\rightarrow$  fase  $\pi$

Poniamo  $A_k$  sia il nostro segnale:

Un segnale  $\cos(2 \pi f_c t)$  viene **modulato** moltiplicandolo per  $A_k$  per  $t$  secondi (durata di un simbolo):

$$Y_i(t) = A_k \cos(2 \pi f_c t)$$

Il segnale ricevuto viene **demodulato** moltiplicandolo per  $2\cos(2 \pi f_c t)$  per  $t$  secondi e successivamente filtrandolo con un filtro passa-basso

$$Y(t) = A_k [1 + \cos(4 \pi f_c t)]$$

### QAM (Quadrature Amplitude Modulation)

Si tratta di una particolare modulazione che permette di trasmettere due segnali diversi (due impulsi) =

Un segnale  $A_k$  modulato in Fase come abbiamo visto, ed il secondo  $B_k$  modulato in quadratura.

Si trasmette la somma delle componenti in fase ed in quadratura:  $Y_i(t) + Y_q(t)$

$$Y_q(t) = B_k \sin(2 \pi f_c t).$$

Questo è possibile perché il seno ed il coseno sono ortogonali tra loro e quindi possono viaggiare insieme sommandoli.

L'operazione di traslazione alla frequenza portante mi fa perdere metà della velocità della banda traslata e da  $W_c$  il limite diventa  $W_c/2$ , ma inviando poi due impulsi nella stessa trasmissione recupero la perdita.

Se quindi su una componente mando  $v$  bit su quella componente avrò  $2^v$  livelli, ma anche sull'altra componente avrò lo stesso e quindi alla fine manderò  $2v$  bit alla volta.

### Costellazione dei Segnali

Insieme dei punti che può assumere un segnale che è sempre uguale a tutte le possibili combinazioni tra i livelli del segnale A e B.

Se su A voglio mandare  $v=2$  bit per impulso. Significa che su A avrò  $2^2$  livelli = 4. Anche su B avrò quindi 4 livelli e potrò mandare 2 bit per impulso. Quindi i possibili punti saranno  $4 \times 4 = 16$  QAM

Ad ogni modo sebbene con quest'operazione si possa arrivare ad un numero maggiore di bit per impulso il limite dei livelli è sempre dato dal Rumore che si aggiunge. Più livelli ho, meno saranno distanziati, più la possibilità di errore con la somma del Rumore aumenta.

### STRATO DI COLLEGAMENTO

In una rete esistono diversi nodi (Host, Router) collegati da dei link che sono appunto i collegamenti. Le PDU dello strato di collegamento sono chiamate **Frame**

I frame possono essere gestiti da collegamenti diversi (cablati, wireless, LAN) e quindi da **protocolli** diversi che erogano servizi diversi. Per identificare origine e destinatario vengono utilizzati indirizzi "MAC"

### Servizi

Nella modalità **full-duplex** gli estremi di un collegamento possono trasmettere contemporaneamente

Nella modalità **half-duplex** la trasmissione nei due versi è alternata

**Framing**: incapsulamento dei pacchetti dello strato superiore in Frame ai quali si aggiunge:

**Flag**: una sequenza di bit fissa all'inizio e alla fine di ogni frame cosicché il ricevente capisca il termine di ogni frame quest'operazione si chiama funzione di **Delimitazione**. Il tipo di sequenza di bit dipende dal tipo di protocollo.

- ☐ **Bit Stuffing** - potrebbe accadere di avere una sequenza di bit proprio uguale alla mia flag, per evitare aggiungo uno zero dopo ogni seq di cinque 1.
- ☐ **Bit destuffing** - in ricezione ogni sequenza di cinque 1 se il bit successivo è un 1 è finito il frame, se invece è 0 è un bit di stuffing e lo si può togliere.
- ☐ **Byte stuffing (si usa una sequenza di «control escape» 01111101)**: - se nella sequenza di bit, c'è un pezzo identico al flag, allora lo ripeto due volte.
- ☐ **Byte destuffing** - se incontro due byte vicini uguali ne tolgo uno.

### Implementazione

Si tratta di un livello che è implementato sia nel software che nell'hardware.

A livello pratico si tratta di una scheda di rete che implementa le funzioni dello strato fisico e di collegamento e si chiama **Network Interface Card (NIC)**.

### Rivelazione e Correzione errori

Il nodo ricevente rivela l'errore e possibilmente lo corregge, non sempre sono dovuti allo strato fisico.

Possono essere di diversi tipi (ricevo + bit o meno del dovuto)

Controlla gli accessi al MAC (se accesso condiviso da più terminali)

Controllo flusso: si accerta di non sovraccaricare il ricevente

Consegna affidabile: se richiesta dall'applicazione, se ci sono errori ritrasmette il messaggio

**Due approcci possibili:**

**Error detection & retransmission (ARQ)** – ricevente rileva l'errore, chiede di ritrasmettere l'informazione.

**Forward Error Correction(FEC)** – il ricevente rileva l'errore e lo corregge senza chiedere la ripetizione

Entrambe si basano comunque sull'aggiunta di extra-informazione

### Rilevamento Errori

**Codice di Parità:** I dati trasmessi si organizzano in modo da mandare blocchi di codice(*codeword*), alla ricezione se il blocco non è una codeword è considerato un errore.

Per fare questo è necessario aggiungere dei bit di controllo che vengono chiamati **Codici di controllo di Parità** posso effettuare:

### Controllo di Parità Singola:

Prendo la sequenza dei bit da trasmettere e conto quanti 1 ci sono, se ci sono un numero pari di 1 aggiungo uno 0 alla fine, invece se ci sono un numero dispari di 1 aggiungo un 1 alla fine

Il ricevente conta gli 1 : se sono pari è corretto, se sono dispari vuol dire che qualcosa è andato storto.

**Due problemi:** una volta rivelato l'errore non posso correggerlo, se avviene un numero pari di errori non mi accorgo dell'errore.

### Controllo di Parità Bidimensionale

Si struttura la sequenza dei bit informativi in colonne, per ogni colonna si aggiunge un bit di parità e si aggiunge poi una colonna di parità

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |

La colonna finale è formata dai bit di parità di ogni riga

La riga finale è formata dai bit di controllo di ogni colonna

**Problemi:** se si verificano due errori sulla stessa colonna non posso rilevarli, quando gli errori sono maggiori di 4 non sempre è possibile rilevarli

### Altri codici di rivelazione d'errore

I codici a parità singola hanno scarse prestazioni: Elevata probabilità di non rivelare errori

I codici bi-dimensionali hanno overhead elevato: Richiedono un numero elevato di bit di controllo

**Internet Checksums:** implementato nello strato di Trasporto, si aggiunge bit di controllo nell'header che controlli i bit di parità (controllo sul controllo)

Si considera la stringa di bit da proteggere composta da L parole di 16 bit, il checksum che aggiungo è anch'esso una stringa da 16 bit.

Prendo ogni parola della stringa e la converto in intero da binario e sommo tutti i vari interi e faccio:

$$-x \text{ modulo } (2^{\text{\#bit}} - 1)$$

Se lo si vuole fare in binario = sommo tutti i vari numeri in binario con complemento a 1 (quindi se ho un bit di riporto lo sommo al risultato). Dopodiché ottengo il checksum come complemento a 1 del risultato della somma (ovvero inverte tutti i bit)

Il ricevente somma tutte le sequenze e alla fine somma anche il checksum = se il risultato è composto da tutti 1 il pacchetto è valido

Non è comunque un metodo sicuro al 100% = se la somma ha qualche zero sono sicuro ci sia stato un errore, ma se la somma è tutti 1, non posso esser certa che non ci sia stato errore.

Più è robusta la codeword più facilmente potrà correggere gli errori

### Codici polinomiali a ridondanza ciclica (CRC)

Le cifre della stringa da proteggere son trattate come coefficienti (1 o 0) di un polinomio  $P(x)$ . Il polinomio avrà grado uguale alla lunghezza della stringa -1.

$$P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \dots + a_1x^1 + a_0$$

In particolare, l'i-esimo bit ( $a_i$ ) della stringa è il coefficiente del termine  $x^{i-1}$  di  $P(x)$

Le entità emittente e ricevente utilizzano un polinomio comune  $G(x)$ , detto **polinomio generatore**

Coefficienti del grado massimo e grado nullo devono essere = 1

I coefficienti di  $G(x)$  sono binari, come quelli di  $P(x)$ , supponiamo che questo polinomio sia di grado  $z$

**Emittente** : prende il polinomio  $P(x)$  lo moltiplica per  $x^z$  e lo divide per  $G(x)$ , il risultato è:

$$\frac{x^z P(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

**Addizione e sottrazione** sono operazioni identiche: equivalgono ad un XOR sui bit degli operandi

La **moltiplicazione** per  $x^z$  = significa aggiungere  $z$  zeri alla stringa originaria, il che equivale ad uno shift verso sinistra di  $z$  posizioni

**Ricevente**: prende il polinomio  $T(x) = x^z P(x) + R(x)$  (parola di codice) e divide tutto per  $G(x)$  ottenendo il  $Q(x)$  se il quoziente è senza resto allora non ci sono errori, altrimenti ci sono.

□

Divisione: esempio:

□  $P(x) = x^3 + x^2$   $G(x) = x^3 + x + 1$   $x^z = x^3$

□  $P(x) + x^3 = x^6 + x^5$

□  $P(x) + x^3 / G(x) =$  prendo tanti polinomi quanti il divisore.

Divido il primo per il primo = risultato lo scrivo sotto, moltiplico il risultato per ogni polinomio del divisore e lo metto sotto alla colonna giusta, cambiato di segno, faccio somma algebrica e faccio scendere un altro monomio. Seguo così finché mi rimane uno che è il resto.

**L'emittente** calcola quindi il  $T(x)$  e manda al ricevente il  $P(x)$  di  $k$  bit e il  $R(x)$  di  $z$  bit. Il ricevente riassembla il  $T(x)$  e lo divide per  $G(x)$  e controlla se c'è il resto o no.

**Il ricevente** ottiene una sequenza di bit di  $T(x)$  e di  $G(x)$ . La divisione in binario vien fatta mettendo divisore sotto al dividendo a partire dalla prima cifra. E faccio operazione di xor. A ogni giro rimetto il divisore sotto, shiftato di uno a dx.

### Rilevazione dell'errore

Rappresentiamo con un altro polinomio  $E(x)$  il polinomio dell'errore. Il polinomio  $T(x)$  errato sarà la somma di  $T(x) + E(x)$ , sempre divisibile per  $G(x)$ . E la storia è uguale: se il resto è nullo non ci sono errori, altrimenti ci sono errori.

Ogni bit "1" in  $E(x)$  corrisponde ad un bit che è stato invertito e quindi a un errore *isolato*

Un errore a *burst* di lunghezza  $n$  è caratterizzato in  $E(x)$  da un "1" iniziale, una mescolanza di "0" e "1", e un "1" finale per un complesso di  $n$  coefficienti binari

$$\text{Resto} \left[ \frac{T(x) + E(x)}{G(x)} \right] = \text{Resto} \left[ \frac{E(x)}{G(x)} \right]$$

L'unico problema incorre quando  $E(x) = G(x)$  perché ottengo zero anche se c'è l'errore.

**Errori non rilevabili con CRC:** tutti gli errori pari a  $G(x)$  o multipli di  $G(x)$

### Codici CRC: polinomi generatori

Sono standard i seguenti polinomi generatori:

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

### Forward Error Correction (FEC)

Date due stringhe binarie di ugual lunghezza,  $X$  e  $Y$  e posto  $W(A)$  = numero di bit 1 della stringa  $A$ , si definisce **distanza di Hamming** tra  $X$  e  $Y$  la quantità:

$$HD(X, Y) = W(X \text{ xor } Y)$$

Un codice con parole di  $n$  bit può rappresentare simboli di  $m$  bit e la capacità di correzione è funzione della ridondanza  $r = n - m$ ; il valore minimo della HD tra tutte le coppie di parole di codice è la HD del codice; Un codice con  $HD = 2d + 1$  può correggere fino a  $d$  errori binari e può rivelarne fino a  $2d$

### Protocolli di accesso multiplo (MAC)

Due tipi di Rete : **Punto-Punto (PPP)**, **BroadCast** (cavo o canale condiviso) nel quale comunicano centinaia o migliaia di nodi. Per evitare collisioni esistono protocolli di accesso multiplo che fissano le modalità di trasmissione su canali condivisi. Il protocollo viene implementato in tutti i nodi (decentralizzato) senza controllo centrale. Ne esistono due tipi:

**Protocolli a suddivisione del canale:** suddivide canale in parti (x tempo, frequenza) e per ogni nodo allocata una parte (es. telefonia, satelliti)

**A suddivisione di tempo (TDMA):** Si divide il canale x intervalli di tempo, ogni slot allocato a nodo anche se il nodo non ha nulla da trasmettere.

**A suddivisione di frequenza (FDMA):** Si divide il canale x frequenze, ogni nodo una banda di frequenza, anche se il nodo non ha nulla da trasmettere.

Vantaggio è che se il canale è libero il nodo trasmette al massimo  $R$ , se avviene collisione il protocollo deve saperla gestire. (*aloha*, *ethernet*, *CSMA*).

### Protocolli ad accesso dinamico

**Protocolli ad accesso casuale (random access):** Canali non divisi, se avviene collisione i nodi ritrasmettono il messaggio. La rete come se usasse un bus e tutti trasmettono quando vogliono.

**Protocolli ad accesso controllato:** ogni nodo ha un turno, più o meno lungo a seconda della necessità. Il turno è di colui che possiede il token, quindi la rete è come se fosse un

anello in cui ci si passa il token.(wireless, LAN)

### Random Access

Rete a bus: Una stazione trasmette quando è pronta; Possibili collisioni, strategie di ritrasmissione

### Protocolli ad accesso casuale

Quando un nodo deve inviare un pacchetto trasmette sempre alla massima velocità del canale, cioè  $R$  bit/s e non c'è nessun coordinamento a priori tra i nodi

Se due o più nodi trasmettono “contemporaneamente” si ha una “collisione”

Un protocollo ad accesso casuale definisce come rilevare un'eventuale collisione

Esempi di protocolli ad accesso casuale sono: **ALOHA**; **slotted ALOHA**; **CSMA**, **CSMA/CD**, **CSMA/CA**

### Prodotto Banda Ritardo

$$PBR = Rd(\text{bit})$$

$R$  (bit/s): banda del canale;  $d$  (sec): ritardo di propagazione end-to-end

È il numero di bit che si trovano contemporaneamente sul canale. Ritardo prop = distanza km / velocità prop colleg.

### Calcolo dell'efficienza

Dato  $R$ : bit rate del canale (bit/s) e  $L$  lunghezza di una frame (bit)

$$\text{Efficienza} = \rho_{\max} = \frac{L}{L + 2t_{\text{prop}}R} = \frac{1}{1 + 2t_{\text{prop}}R/L} = \frac{1}{1 + 2a}$$

$$\text{Throughput Massimo} = R_{\text{eff}} = \frac{L}{L/R + 2t_{\text{prop}}} = \frac{1}{1 + 2a} R \text{ bit/s}$$

$$\text{Prodotto banda ritardo normalizzato} \quad a = \frac{t_{\text{prop}}}{L/R}$$

← Ritardo di Propagazione  
← Tempo di trasmissione di una frame

### Protocollo ALOHA

Ad accesso casuale, quindi il nodo trasmette quando ha una frame pronta, se c'è collisione la frame viene persa, il nodo aspetta l'ACK entro un tempo timeout, se scade, il nodo calcola il tempo di ritrasmissione (**backoff time**) e allo scadere di esso ritrasmette la frame.

#### Definizioni:

- ☐  $X$  = tempo di trasmissione frame (costante)
- ☐  $S$  = throughput (numero medio di frame trasmette con successo nell'intervallo di  $X$  sec)
- ☐  $G$  = load ( numero medio di tentativi di trasmissione in un intervallo  $X$  sec)
- ☐  $P_{\text{succ}}$  = probabilità di trasmissione della trama con successo
- ☐  $S = G * P_{\text{succ}} = G e^{-2G}$  (**Throughput**)
- ☐ **Intervallo di vulnerabilità** =  $2X$

### Slotted Aloha

I pacchetti hanno stessa dimensione e il tempo diviso in slot uguali al tempo di trasmissione.

All'inizio degli slot sincronizzati i nodi trasmettono pacchetti, se collidono rilevano la collisione prima della fine dello slot e ritrasmettono frame negli slot dopo con probabilità  $p$ .

Ogni nodo può arrivare alla massima velocità di trasmissione e non c'è controllo centralizzato, ma una certa frazione degli slot presenterà collisioni e di conseguenza andrà “sprecata”, mentre un'alta frazione degli slot rimane vuota, quindi inattiva.



***Intervallo di vulnerabilità = X***

### **L'efficienza dello Slotted Aloha**

Definita come la frazione di slot in cui avviene una trasmissione utile in presenza di un elevato numero di nodi attivi, che hanno sempre un elevato numero pacchetti da spedire.

Supponiamo N nodi con pacchetti da spedire, ognuno trasmette i pacchetti in uno slot con probabilità p

La probabilità di successo di un dato nodo =  $p(1-p)^{N-1}$

La probabilità che un nodo arbitrario abbia successo =  $Np(1-p)^{N-1}$

Per ottenere la massima efficienza con N nodi attivi, bisogna trovare il valore  $p^*$  che massimizza

$Np(1-p)^{N-1} \rightarrow p^*=1/N$

Per un elevato numero di nodi, ricaviamo che nel caso migliore, solo il 36% degli slot sono utilizzati in modo utile

### **Protocollo CSMA (Carrier Sensing Multiple Access)**

I nodi prima di trasmettere ascoltano il canale, se è libero trasmettono, se è occupato aspettano prossimo intervallo di tempo.

□ **Intervallo di vulnerabilità** :  $2t_{prop}$

Quando il nodo rivela il canale occupato si applicano degli algoritmi di persistenza:

**1-persistent CSMA**: appena il canale si libera il nodo inizia la trasmissione, basso ritardo e bassa efficienza

**Non persistent CSMA**: il nodo applica un **backoff** e ricomincia da capo con il CSMA, alto ritardo e alta efficienza

**P-persistent CSMA**: il nodo aspetta che il canale si liberi, con probabilità p trasmette, con probabilità 1-p attende un mini-slot e ricomincia il CSMA

### **CSMA/CD (with collision detection)**

Il nodo ascolta sia prima di trasmettere e mentre trasmette, così se rivela una collisione invia un segnale di disturbo (**jam**) e interrompe la trasmissione e tutti i nodi coinvolti rischedulano dopo un tempo di backoff. Se si arriva all'n-esima collisione consecutiva stabilisce un valore k (tra 0 e  $2^{n-1}$ ) e la scheda di rete aspetta un tempo pari a K volte 512 bit. Così non spreca i tempi di rilevazione di collisioni e la collisione viene rivelata al max ad un tempo  $t=2t_{prop}$

### **Confronto Protocolli**

Per piccoli valori di prodotto banda ritardo, CSMA-CD ha il miglior throughput, per grandi valori di A Aloha e Slotted Aloha hanno migliori prestazioni.

### **Protocolli ad accesso controllato**

**Polling**: c'è un nodo master che gestisce i turni degli altri, elimina le collisioni e gli slot vuoti ma se si rompe il nodo master tutto il canale risulta inattivo.

**Token-Passing**: la rete è ad anello e ci si passa un messaggio di controllo (token) per chi ha il turno, quindi decentralizzato ma il guasto di un nodo può mettere inattivo tutto il canale.

I nodi in stato ready aspettano il token, quando lo ricevono e cominciano a trasmettere cambiano ultimo bit del flag per dire che è busy, alla fine della trasmissione rimette il bit per dire che è free.

### **Metodi di reinserimento del token**

**Ring Latency**: numero di bit che possono essere trasmessi sul ring simultaneamente

**MultiToken operation:** appena viene trasmesso l'ultimo bit del frame il nodo trasmette il free token  
**Throughput** =  $1/(1+a/M)$  (M=nodi e a= tempo x bit di circolare sul ring/tempo trasmissione frame)  
**Single Token operation:** t- Il Free token è inserito dopo che l'ultimo bit del busy token è ritornato al nodo origine. **Throughput** =  $1/(a/M + \max(1,a))$   
**Single Frame operation** – Il Free token è inserito dopo che il nodo emittente ha ricevuto l'ultimo bit della sua frame. **Throughput** =  $1/(1+a(1+1/M))$

□

### **Controllo Errore – Controllo di Flusso**

**Hop-by-Hop** = operazione effettuata tratta per tratta (Strato di Collegamento). Il controllo vien fatto a ogni hop e la reazione è immediata, nel caso di errore può essere ritrasmesso. Affidabile.

**End-to-End** = operazione effettuata da estremo a estremo (Strato di Trasposto).

Il controllo avviene tra sorgente e destinazione e i segmenti possono essere persi o subire ritardi.

Il controllo è più complesso e meno affidabile.

### **Automatic Repeat Request (ARO)**

Servizio offerto dagli strati sottostanti per assicurare che una sequenza di PDU sia consegnata in ordine e senza errori o duplicazioni in presenza di un servizio offerto dagli strati sottostanti.

### **Stop-and-Wait**

L'entità **emittente(Transmitter)** che si trova inizialmente nello stato Ready, in attesa che uno strato superiore gli richieda di inviare un pacchetto. All'arrivo del pacchetto trasmette un frame (composto da Pacchetto+Frame di controllo = Header+CRC) dove nell'Header c'è il numero di sequenza del frame Slast. Dopodiché passa allo stato Wait e attiva un temporizzatore nell'attesa di un ACK (riscontro positivo) da parte del ricevente.

In questo stato la ricezione delle richieste dallo strato superiore vengono bloccate

Se scade il timeout il frame viene ritrasmesso. Se ricevo l'ACK = se il numero di sequenza  $R_{next} = Slast+1$  è corretto e torna nello stato ready, se il numero di sequenza invece è sbagliato l'ACK viene ignorato.

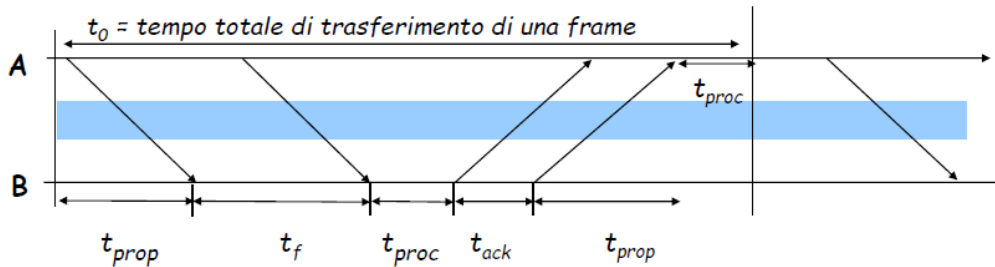
L'entità **ricevente(Receiver)** a sua volta sta sempre nello stato Ready in attesa di ricezione di una nuova frame. Quando arriva la frame effettua il controllo CRC.

Se non ci sono errori e il numero di sequenza  $Slast = R_{next}$  è corretto, allora la frame viene accettata, aggiorna il valore di  $R_{next}$ , emette l'ACK con  $R_{next}$  (numero di sequenza che si aspetta di ricevere), consegna il pacchetto allo strato superiore.

Se non ci sono errori e il numero di sequenza sbagliato: la frame viene scartata e viene mandato un ACK con  $R_{next}$  uguale a prima per richiedere l'ultimo frame

Se ci sono errori : la frame viene scartata.

## Tempo di trasferimento



$$\begin{aligned}
 t_0 &= 2t_{prop} + 2t_{proc} + t_f + t_{ack} \\
 &= 2t_{prop} + 2t_{proc} + \frac{n_f}{R} + \frac{n_a}{R}
 \end{aligned}$$

Annotations:   
 -  $t_f$ : Lunghezza di una frame   
 -  $t_{ack}$ : Lunghezza di un ACK   
 -  $R$ : Bit rate canale

## Rate di Efficienza di trasferimento

$$R_{eff}^0 = \frac{\text{numero di bit informativi consegnati a destinazione}}{\text{tempo totale necessario per la consegna dei bit informativi}} = \frac{n_f - n_o}{t_0}$$

Annotation:  $n_o$ : bit di overhead

## Efficienza di trasmissione

$$\eta_0 = \frac{R_{eff}}{R} = \frac{n_f - n_o}{t_0} = \frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}}$$

Annotations:   
 -  $1 - \frac{n_o}{n_f}$ : Effetto dell'overhead di una frame   
 -  $1 + \frac{n_a}{n_f}$ : Effetto di un ACK   
 -  $\frac{2(t_{prop} + t_{proc})R}{n_f}$ : Effetto del prodotto Banda-Ritardo

Essa decresce all'aumentare della lunghezza del mezzo perché dipende molto dal prodotto banda-ritardo

Se poi vogliamo tener conto della probabilità di errori si moltiplica all'efficienza  $(1 - p_f)$  che è la probabilità che un frame arrivi senza errori (che dipende dal mezzo trasmissivo).

Minore è  $p_f$  e maggiore sarà l'efficienza.

**Problema**: Anche quando i frame vengono mandati correttamente si spreca tempo nell'attesa degli ACK

## Efficienza su un canale con errori

Sia  $1 - P_f$  = probabilità che una frame arrivi senza errori

$1 / (1 - P_f)$  = numero medio di trasmissioni necessarie per avere una trasmissione corretta di una frame

$T_0 / (1 - P_f)$  = tempo medio di trasferimento di una frame

### Go-back N

Il **Transmitter** rimane attivo perché manda una finestra di frame (il numero di frame mandabili è limitato dal  $W_s$ ) e numera i frame con  $m$  bit.

Se riceve ACK delle frame emesse prima che termini quella finestra, allora la finestra viene aggiornata con quella successiva e continua la trasmissione

Se la finestra termina e non riceve ancora ACK : si mette in attesa degli ACK

Se non riceve ACK allo scadere di timeout: ritrasmette la finestra richiesta in  $R_{next}$

Il **Receiver** se tutto va bene rimanda con l'ACK il numero di frame che si aspetta di ricevere in  $R_{next}$  al termine della finestra. Altrimenti se c'è un problema rimanda con ACK il numero del frame ricevuto male e continua a scartare gli altri fintanto che il Transmitter non ricomincia dal frame errato.

**Efficienza:** Nei casi di elevato Prodotto Banda Ritardo è meglio di S&W ma nei casi di elevato BER l'efficienza diminuisce

**Problema:** anche quando certe frame consegnate correttamente, rimanda dal punto in cui ACK negativo, ritrasmettendo quelle corrette

### Sliding window

Il Transmitter attende gli ACK (con numero di sequenza  $S \geq S_{last}$ ) e quando arriva un ACK, con numero di sequenza  $S$ , viene posto  $S_{last} = S$

L'estremo superiore della finestra sarà quindi  $S_{last} + W_s - 1$

**Il massimo valore della finestra** =  $W_s = M - 1 = 2^m - 1$

**PiggyBacking:** quando due entità mandano e ricevono e quindi nel mandare il messaggio caricano sulla schiena anche gli ACK di ciò che hanno ricevuto.

**Tempo Timeout( $T_{out}$  componenti ):** Il timeout è il tempo che ci vuole per arrivare da punto a punto ed è uguale alla somma di  $= 2t_{prop} + t_{proc} + t_{trasmissione\ frame} + t_{trasmissione\ ack}$ .

$W_s$  deve quindi essere abbastanza grande per mantenere il canale occupato per tutto il  $T_{out}$ .

□

### Selective Repeat

Go-Back-N è inefficiente poiché, in caso di ritrasmissione, viene rimesso un numero elevato di frame, anche se ricevute correttamente dal receiver

Selective Repeat ritrasmette solo le frame che sono state perse

**Transmitter:** All'esaurimento del Timeout il Transmitter rimanda solo la frame corrispondente al NAK più vecchio.

**Receiver:** gestisce una finestra in ricezione con sequenza di numeri che possono essere accettati.

Quando NAK comincia a mandare ACK sempre con  $R_{next}$  relativo al NAK.

Ma in realtà dentro di se aggiorna un altro contatore così quando viene rimandato quello il NAK riprende dal punto lasciato. Quelle che arrivano fuori sequenza vengono bufferizzate.

**Efficienza:** meglio di GBN e S&W ma comunque l'efficienza diminuisce all'aumentare del BER.

Il massimo valore permesso è:  $W_s + W_r = 2^m$

### Flow Control

Il ricevitore dispone di un buffer limitato per memorizzare le frame entranti

Nel buffer di ricezione si possono verificare fenomeni di overflow a causa della differenza tra il rate di arrivo delle frame e il rate con cui il ricevitore elabora le frame oppure picchi nell'arrivo delle frame

Ha lo scopo di prevenire gli overflow del buffer di ricezione regolando il tasso di emissione delle frame da parte del Transmitter

### **XON / XOFF (Backpressure)**

Si deve attivare il segnale di Off in modo da evitare la perdita di pacchetti, lo spazio disponibile nel buffer deve essere almeno uguale a  $2T_{prop} R$  bit

### **Window Flow Control**

Finestra scorrevole di ampiezza  $W_s$  uguale al buffer disponibile in cui il Transmitter non può in nessun caso emettere più di  $W_s$  frame

Gli ACK possono essere interpretati come permessi a trasmettere e possono regolare il rate di trasmissione

**Problemi:** Scelta della dimensione della finestra, Interazione tra rate di trasmissione e ritrasmissioni, TCP separa error & flow control

### **PPP(point-to-point protocol)**

Sono presenti un mittente e un destinatario dove non è necessaria la funzione di controllo di accesso al mezzo (MAC) e non occorre un'indirizzamento MAC esplicito

### **Funzioni del PPP**

**Framing dei pacchetti:** Incapsula un pacchetto a livello di rete all'interno del pacchetto PPP a livello di link

**Trasparenza:** Non deve porre alcuna restrizione ai dati che sono contenuti nel pacchetto a livello di rete

**Rilevazione degli errori (ma non la correzione)**

**Disponibilità della connessione:** Il protocollo deve rilevare la presenza di eventuali guasti a livello di link e segnalare l'errore al livello di rete

**Negoziazione degli indirizzi di rete:** Deve fornire un meccanismo alle entità di strato di rete per ottenere o configurare gli indirizzi di rete

**NON Controlla il flusso né la sequenza:** Non deve necessariamente trasferire le frame al ricevente mantenendo lo stesso ordine

### **Formato dei pacchetti dati PPP**

**Flag:** ogni frame inizia e termina con un byte con valore 01111110

**Address:** unico valore (11111111)

**Control:** unico valore; ulteriori valori potrebbero essere stabiliti in futuro

**Protocol:** indica al PPP del ricevente qual è il protocollo del livello superiore cui appartengono i dati incapsulati

**Information:** incapsula la PDU (es. pacchetto IP) trasmesso da un protocollo del livello superiore sul collegamento PPP

**Checksum:** utilizzato per rilevare gli errori nei bit contenuti in un pacchetto; utilizza un codice a ridondanza ciclica a due o a quattro byte

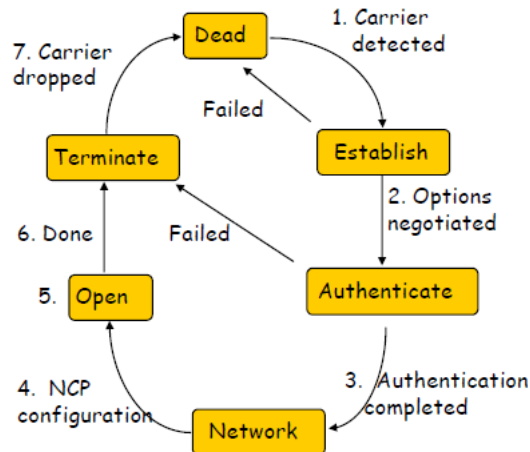
### **Delimitazione (Byte stuffing)**

**Requisito di trasparenza:** Nel campo informazioni deve essere possibile inserire una stringa <01111110>

**Transmitter:** Aggiunge un byte <01111101> prima di ogni byte di dati <01111110> o <01111101>

**Receiver:** Se rivela due byte <01111101> consecutivi si scarta il primo e continua la ricezione dei dati  
Se rivela una sequenza <01111101> <01111110> si scarta il primo byte e continua la ricezione dei dati  
Se rivela un singolo byte <01111110> si tratta di un flag

## Collegamento PC-ISP: fasi del PPP



1. Il PC si connette all'ISP via modem
2. Il PC e l'ISP scambiano pacchetti LCP per negoziare i parametri del protocollo PPP
3. Controllo delle identità
4. Scambio di pacchetti NCP per configurare lo strato di rete (es. IP address assignment)
5. Emissione e ricezione di pacchetti IP send/receive IP packets
6. Il protocollo NCP è usato per abbattere lo strato di rete (rilascio degli IP address); Il protocollo LCP abbatte la connessione di data link layer
7. Il Modem si disconnette

### PPP Authentication

**Password Authentication Protocol:** La parte “*Initiator*” deve inviare la coppia [userID & password] mentre la parte “*Authenticator*” replica indicando il successo o il fallimento dell'autenticazione. Dopo alcuni tentativi falliti, il collegamento viene chiuso. Se la trasmissione non è cifrata, la coppia [userID & password] può essere intercettata.

### **Challenge-Handshake Authentication Protocol (CHAP):**

Initiator & authenticator condividono una chiave segreta. L'Authenticator emette una “sfida” (un numero random), dopo di che l'Initiator e l'Authenticator ne calcolano la versione cifrata utilizzando la chiave segreta condivisa. L'Initiator trasmette la versione cifrata della sfida verso l'Authenticator mentre l'Authenticator confronta la risposta con la propria versione.

### High-Level Data Link Control (HDLC)

**Normal Response Mode:** Linee multidrop con polling

**Asynchronous Balanced Mode:** Link full-duplex, point-to-point

### Formato delle Frame

**Flag:** funzioni di delimitazione (01111110)

**Address:** identifica la stazione secondaria (1 ottetto), nella modalità ABM, una station può agire come primario o secondario quindi il valore del campo può cambiare

**Information:** dati d'utente (lunghezza variabile)

**Frame Check Sequence:** CRC 16 o 32-bit

Il campo di controllo è formato da: **Information Frame** che contiene un numero di sequenza N(S),

**Supervisory Frame** che implementano le funzioni di error control (ACK, NAK) e flow control e

**Unnumbered Frame**

**Indirizzamento**(Strato Collegamento).

**Indirizzo Ip:** Indirizzo a livello di rete associato a un dispositivo per una sessione (può variare)

**Indirizzo MAC:** Unico a ogni dispositivo indipendentemente dalla rete a cui è collegato.

Ogni indirizzo è associato ad una scheda di rete e gli indirizzi vengono gestiti dalla IEEE che vende alle società che vogliono costruire schede di rete blocchi di indirizzi.

**Indirizzo BroadCast :** FF-FF-FF-FF-FF-FF

**Tabella ARP:** Ogni nodo IP nella LAN ha una tabella ARP che contiene la corrispondenza tra indirizzi IP e MAC. E' una tabella plug-and-play che si costituisce automaticamente senza bisogno di essere configurata

□

**Protocollo ARP nella stessa sottorete**

Il nodo A vuole inviare a B ma non ha l'indirizzo nella tabella, dunque invia un messaggio broadcast con l'indirizzo IP di B richiedendo l'indirizzo MAC. Solo B risponde ad A con il suo indirizzo MAC che A scrive nella sua tabella.

**Protocollo ARP in rete esterna**

Il nodo A deve comunque conoscere l'indirizzo IP di B e le due reti sono collegate da un router che ha due tabelle ARP una per ciascuna LAN. A crea il pacchetto con origine A e destinazione B, usa la sua tabella per trovare l'indirizzo MAC del router e gli invia il pacchetto. Il router guarda la destinazione, cerca l'indirizzo MAC di B e invia il frame a B

**Ethernet**

Oggi le reti hanno una struttura a stella al centro delle quali si trova un dispositivo chiamato **switch** o **hub**. Ciascun nodo esegue il protocollo Ethernet separatamente dagli altri.

**Valore principale:**Slot time

**Struttura**

La scheda di rete incapsula i pacchetti IP in una frame Ethernet formata da:

**Preambolo:** 7 byte che serve per attivare le schede dei riceventi e sincronizzare i clock con quelli dell'emittente

**Start Delimiter:** 1 byte che indica l'inizio della frame (10101011)

**Source e Destination Address:** 6 byte ciascuno , sono gli indirizzi MAC di ricevente e destinatario, quando i riceventi vedono in destination se stessi copiano il frame nel buffer di ricezione mentre gli altri vengono ignorati.

**Lenght:** 2 byte indica la lunghezza del campo informativo che può essere al massimo 1500 bytes, se poi invece calcolo la lunghezza massima del frame senza preambolo e SD allora sono 1518

**PAD:** assicura la lunghezza minima del frame = 64 byte

**CRC:** 4 byte per il controllo

**Funzionamento**

Si tratta del funzionamento del protocollo **CSMA/CD**, il protocollo rimane standard e anche la struttura dei frame, ciò che si differenzia sono le velocità e i mezzi trasmissivi

La scheda di rete prepara una frame Ethernet e se il canale è inattivo, inizia la trasmissione.

Se invece il canale risulta occupato, resta in attesa fino a quando non rileva più il segnale

Durante la trasmissione verifica la presenza di eventuali segnali provenienti da altri terminali, se non ne rileva considera il pacchetto spedito

Se invece rileva segnali da altri adattatori (evento di collisione), interrompe immediatamente la trasmissione del pacchetto e invia un segnale di disturbo (jam)



La scheda di rete calcola l'intervallo di backoff e se si è arrivati all'n-esima collisione consecutiva, stabilisce un valore K tra  $\{0,1,2,...,2n-1\}$  di attesa prima di riprendere

**Segnale di disturbo (jam):** avvisa della collisione tutti gli altri adattatori che sono in fase trasmissiva, ha una lunghezza di 48 bit

**Intervallo di backoff:** adatta il tempo di attesa al numero di nodi coinvolti nella collisione

### Hub

Ripetitore che opera allo strato fisico che rigenera il segnale analogico e lo ritrasmette su tutte le interfacce, decodifica e ri-codifica il codice di linea, rileva collisioni e le inoltra su tutte le porte, isola segmenti di rete se si verificano 30 collisioni consecutive e permette di aumentare le dimensioni di una LAN.

**Dominio di collisione:** Sezione di rete in cui qualsiasi coppia di stazioni che trasmettono contemporaneamente generano una collisione.

### Switch

Dispositivo che opera a livello di collegamento che esamina l'indirizzo MAC di destinazione e, se possibile, lo invia all'interfaccia corrispondente alla sua destinazione, collega diversi domini di collisione (reti collegate ad un hub) in cui le reti non sanno che c'è un switch tra di loro.

Gli switch non hanno bisogno di essere configurati, apprendono autonomamente la topologia di rete e le regole di instradamento delle frame poichè viene collegato direttamente a ogni host

Permette trasmissioni simultanee tra 4 reti

### Tabella di commutazione (Switch Table)

Ogni switch ha una tabella di commutazione che comprende l'indirizzo MAC di un nodo, l'interfaccia collegata al nodo e il timestamp.

**Autoapprendimento:** Ogni volta che un nodo gli manda frame, lui prende i dati del mittente e li registra.

Ovviamente c'è un timeout che forza periodicamente l'apprendimento di ogni indirizzo di nuovo.

Le informazioni che non vengono rinfrescate da molto vengono cancellate dopo un tempo massimo.

Se trova la destinazione nella tabella = **Selective Send**, se invece non la trova allora la inoltra a tutti = **Flood**

### **Switch e router a confronto**

Entrambi sono di tipo store-and-forward.

Il **Router** sono attivi a livello di rete e mantengono le tabelle di routing e implementano algoritmi di instradamento, mentre gli **Switch** sono attivi a livello di collegamento e mantengono le tabelle di commutazione e implementano il filtraggio e algoritmi di autoapprendimento.

### Protocollo Spanning Tree (STP)

Regola il processo di forwarding in presenza di loop nella rete.

Da una rete a maglia crea un albero e lo tiene in memoria.

L'algoritmo viene avviato ogni volta che si rileva un cambiamento nella topologia della rete.

**Operazioni:** Inizialmente si determina lo switch radice (**Root Bridge**) : all'inizio tutti si credono root bridge e mandano trame e messaggi(in particolare **BPDU(bridge protocol data unit)**, in cui mettono nel campo root id il loro indirizzo MAC. Quando ricevono una trama con l'id diverso dal loro lo confrontano. Se l'id è minore del proprio lo definiscono root, altrimenti continuano a credersi root. Quindi alla fine il root sarà quello con l'id minore.

Determina poi per tutti gli altri switch la **Root Port**, la porta che li porta al Root Bridge: una

volta eletta la root bridge ogni switch elegge la porta a costo minimo per raggiungere il root bridge. Seleziona la **designated port**, la porta per ogni lan per ricevere e trasmettere frame : una volta che gli switch hanno scelto le loro porte con i costi per raggiungere il root bridge, le lan sceglieranno lo switch che raggiunge il root a costo minore. Tutte le altre porte di lan e switch vengono disabilitate

Le **BPDU** (Bridge Protocol Data Unit) contengono:

**Root id:** identificativo del root bridge

**Switch id:** identificativo del bridge emittente

**Root path cost:** costo totale per raggiungere root bridge

**Flag:** con il Topology Change Notification = 1 trasmesso se cambia qualcosa nella topologia della rete

### **STP: cambiamenti di topologia**

Cambiamenti della topologia vengono notificati al Root Bridge attraverso **Topology Change**

#### **Notification BPDU**

Il Root Bridge invia Conf. BPDU con flag TC = 1 verso tutti gli altri bridge

I bridge reagiscono al cambiamento della topologia impostando il timer ageing-time al valore forward delay (trasportato nelle Conf. BPDU ... raccomandato 15 s)

### **Wireless LAN**

Le operazioni di **Collision Detection** sono difficili perché una stazione non è in grado di ascoltare le proprie trasmissioni e quindi rivelare eventuali collisioni

### **Tipologie di rete**

**Infrastructure network:** Le comunicazioni avvengono esclusivamente tra i terminali e l'Access Point (AP) e non direttamente tra i terminali

**Basic Service Set (BSS):** Terminali e AP all'interno della stessa area di copertura

**Extended Service Set (ESS):** Diverse BSS connesse tra loro

**Ad-hoc network:** Nessun AP, ogni terminale comunica direttamente con gli altri

### **Strato MAC**

Sono definiti due modalità di accesso:

**Distributed Coordination Function (DCF)** basato su un protocollo CSMA/Collision Avoidance (CSMA/CA):

**Point Coordination Function (PCF):** Metodo di polling di tipo contention-free polling adatto a servizi con requisiti stringenti di ritardo

Il DCF offre un servizio di trasferimento asincrono, mentre il PCF offre sia un servizio asincrono sia un servizio time-bounded

### **Funzionamento base CSMA/CA**

Se il mezzo è libero il terminale attende un intervallo di tempo denominato DIFS dopodiché trasmette la frame, se invece il mezzo è occupato il terminale effettua il backoff della trasmissione

### **Hidden Terminal Problem**

Pensiamo in riferimento tre nodi A,B,C in cui i nodi A e C non possono ascoltarsi e B è tra di loro

A emette una frame verso B, C non può ascoltare la trasmissione di A

C vuole emettere una frame verso B, C "ascolta" a "free" medium (CS commette un errore)

Si verifica una collisione in B

A non può rivelare la collisione (il meccanismo CD non funziona)

### **Meccanismo RTS/CTS**

Meccanismo di “prenotazione” del canale (opzionale): Permette al sender di “prenotare” il canale invece di competere per il suo utilizzo attraverso un accesso casuale

Il “sender”, prima di emettere una frame, trasmette verso il receiver (o l’AP) un pacchetto, di lunghezza molto breve, denominato “**Request-To-Send (RTS)**”:

Il pacchetto RTS è trasmesso usando il meccanismo CSMA, può subire collisioni con altri pacchetti RTS, le collisioni sono poco probabili perché il pacchetto RTS è breve

Il “receiver” (o l’AP) quando riceve l’RTS emette un pacchetto “**clear-to-send (CTS)**” che viene ricevuto da tutti i nodi

Il sender quando riceve il pacchetto CTS può trasmettere la frame mentre gli altri terminali posticipano le proprie trasmissioni

All’interno dei pacchetti RTS e CTS è indicato l’intervallo di tempo in cui il canale sarà occupato per la trasmissione della frame

**Net Allocation Vector (NAV)**: È un temporizzatore che indica l’intervallo di tempo che le altre stazioni devono attendere per effettuare il test del canale e verificare se il canale libero

Ogni nodo alla ricezione dell’RTS inizializza il proprio NAV che specifica l’istante in cui il nodo può tentare nuovamente di accedere al mezzo

### **Formato delle frame**

**Frame control (FC)** (2 byte): Definisce il tipo di frame e contiene alcune informazioni di controllo

**Duration (D)** (2 byte): Nella maggioranza delle frame indica la durata della trasmissione, viene usato dagli altri nodi per definire il NAV

**Addresses (4 x 6 Byte)**: Ci sono 4 campi di indirizzo MAC ognuno di lunghezza 6 byte (48 bit)

Il significato di questi indirizzi dipende dai flag “TDS” e “FDS” contenuti nel campo FC

**Sequence control (SC)** (2 byte): Numero di sequenza della frame usato per la funzione di flow control

**Frame body** (0 – 2312 byte): Contiene le informazioni dell’utente (payload)

**Frame Check Sequence (FCS)** (4 byte): Contiene un CRC-32 per la rivelazione di errore

### **Addressing**

Il meccanismo di indirizzamento specifica quattro casi:

**Caso 1 (00)** : “To DS”= 0 , “From DS”= 0:

La frame non proviene da e non è diretta verso il Distribution System e i nodi sorgente e destinazione sono interni allo stesso Basic Service Set (BSS):

Address 1: Destination Address

Address 2: Source Address

Address 3: Identificatore del BSS a cui appartengono i nodi sorgente e destinazione (indirizzo dell’AP del BSS)

Address 4: non usato

Il riscontro ACK deve essere inviato direttamente al nodo sorgente

**Caso 2 (01)** : “To DS”=0 , “From DS”=1:

La frame proviene dal Distribution System ed è emessa dall’Access Point (AP):

Address 1: Destination Address

Address 2: Identificatore del BSS a cui appartiene l’AP sorgente (AP address)

Address 3: Source address (indirizzo del nodo sorgente che si trova in un altro BSS)

Address 4: non usato

Il riscontro ACK deve essere inviato all’AP

**Caso 3 (10) :** “To DS”=1 , “From DS”=0:

La frame è diretta verso il Distribution System, verso un AP diverso rispetto a quello del BSS a cui appartiene il nodo sorgente:

Address 1: identificatore del BSS a cui appartiene il nodo di destinazione

Address 2: Source Address

Address 3: Destination address (indirizzo del nodo destinazione che si trova in un altro BSS)

Address 4: non usato

Il riscontro ACK deve essere inviato al nodo sorgente

**Caso 4 (11) :** “To DS”=1 , “From DS”=1:

È il caso in cui una frame è emessa da un AP ed è diretta verso un altro AP dell

Address 1: Receiving Address (Indirizzo dell'AP di destinazione)

Address 2: Transmitting Address (Indirizzo dell'AP di origine)

Address 3: Destination Address (indirizzo dell'effettivo nodo di destinazione)

Address 4: Source Address (indirizzo dell'effettivo nodo sorgente)

### ***Roaming***

Un nodo può migrare da una BSS ad un'altra

Procedura di “Re-association”: Un nodo decide che il collegamento verso l'AP non è affidabile, dunque esegue la funzione di “scanning” del mezzo radio per trovare un altro AP

In caso di esito positivo, il nodo emette una “***Re-association Request***” verso il nuovo AP, se risulta positiva il nodo entra a far parte della BSS gestita dal nuovo AP (roaming), altrimenti cerca un ulteriore AP.

## **STRATO DI RETE**

### **Funzioni**

***Forwarding:*** inoltro di pacchetti da un interfaccia di ingresso di un nodo verso l'opportuna interfaccia di uscita (Funzione attuativa)

***Routing:*** instradamento per determinare il percorso dei pacchetti dall'origine alla destinazione , lo fa attraverso protocolli e algoritmi specifici(Funzione Decisionale Servizi)

***Senza connessione:*** i pacchetti vengono inviati senza un preventivo accordo, ogni pacchetto è indipendente dagli altri, operazione di instradamento su ogni pacchetto singolo, stateless

***Con connessione:*** instaurata una connessione di rete prima dell'invio dei pacchetti, durante l'instaurazione si decide il cammino dei pacchetti, i nodi mantengono info riguardo allo stato delle connessioni (statefull)

### **Tipologia Reti**

***Switching Circuit:*** Rete con connessione garantisce affidabilità e qualità ma perde tempo nell'istaurazione della connessione (RETI A DATAGRAMMA).

***Packet Switching:*** Rete senza connessione , più semplice e veloce ma non ho affidabilità, qualità e non conosco il percorso dei pacchetti. (RETI A CIRCUITO). Ad esempio il Router.

***Virtual Circuit:*** Una via di mezzo tra le prime due. Non stabilisco le risorse allocate per i pacchetti ma stabilisco la strada che i pacchetti faranno e tento di utilizzare quella strada ogni qual volta essa è libera. Utilizza un protocollo di segnalazione: esso emette messaggi di segnalazione sul path definito per la connessione e fa due operazioni: o inizializza le tabelle di forwarding dei nodi o determina la funzione di routing di ogni nodo per identificare il nodo successivo.

Per verificare la connessione utilizza un ***VCI (Virtual Circuit Identifier-*** identificatore di connessione) che è un tag così che gli switch possano aggiornare le loro tabelle con la relazione tra il tag di entrata e di uscita.

Questo comporta comunque un ritardo che si va ad addizionare al ritardo di trasferimento = **Connection Setup Delay**, pesante se si devono mandare pochi pacchetti, ok se è minore del tempo di trasferimento dei pacchetti

**Virtual Circuit Forwarding Table:** è una tabella che ha ogni porta di ingresso di ogni nodo (switch) e utilizza il VCI di ogni pacchetto per fare le ricerche nella tabella.

Per ogni VCI è associata una porta di uscita e il valore del VCI nel link di uscita.

Quindi scrive il VCI di uscita nel pacchetto e lo trasmette sulla porta giusta.

### **Router( Esempio di Rete senza Connessione)**

I router utilizzano gli indirizzi di destinazione per effettuare il forwarding. Per l'indirizzamento si utilizzano delle Tabelle di Routing con gli indirizzi di rete. Ovviamente non è che ogni router ha una lista di tutti i possibili indirizzi, ma ha una tabella che per ogni intervallo di indirizzi ha l'interfaccia d'uscita e il nodo successivo, al resto penseranno i nodi successivi.

Il router non conosce tutto il cammino completo del pacchetto, ma per scegliere l'interfaccia d'uscita ne utilizza il Prefisso.

Il router possiede un Database Topologico dove memorizza le informazioni sulla topologia della rete a cui è connesso e lo aggiorna in base alle info che riceve da altri router con messaggi sui protocolli di routing.

Sulla base di questo database l'algoritmo di routing, determina ogni volta il cammino minimo tra il router e le possibili reti di destinazione e aggiorna così il next-hop nella tabella di routing.

### **Architettura del router**

**Input Line Card:** Porte di ingresso che effettuano il processamento dell'header e il routing.

Ricevono a livello fisico i bit, a livello di link processano i frame, e a livello di rete determinano la porta di uscita con la tabella di routing.

**Output Line Card:** Porte di uscita che definiscono lo scheduling per la trasmissione dei pacchetti.

Riceve i pacchetti, li bufferizza se son troppi, e stabilisce in che ordine ritrasmetterli.

**Regola per bufferizzazione:** si dice che per essere efficiente c'è bisogno di:

$$B = RTT * C / \sqrt{N}$$

RTT(tempo medio di andata e ritorno), C (collegamento) N (numero di flussi)

**Controller:** dove vengono bufferizzate le risorse se il tasso di arrivo > tasso di inoltro

**Switching Fabric:** inoltra i pacchetti dalle porte di ingresso a quelle di uscita. Esistono tre tecniche di commutazione (inoltra da input a output):

**Architettura Shared Memory:** la cpu centrale controlla il processo, il pacchetto viene copiato in memoria centrale e poi passato alla porta di uscita

**Architettura bus:** le porte di ingresso sono collegate alle porte di uscita tramite un bus e trasferiscono il pacchetto da sole. Tutte le porte si contendono il bus.

**Architettura crossbar:** tanti bus per collegare ogni porta di ingresso a quella di uscita così non ci sono contese.

### **Operazioni**

Estrae dal pacchetto che entra l'indirizzo di destinazione.

Ricerca nella sua routing table con il "longest prefix matching" il record relativo a quell'indirizzo, se non lo trova lo manda al router di default.

Se non riesce a fare nemmeno quello, manda un messaggio **ICMP** dicendo che è undeliverable.

### **Protocollo IP**

Nella Rete abbiamo dunque protocolli di instradamento ( che servono al controllo e al riempimento delle tabelle di routing), le Tabelle di Inoltro(Routing), i Protocolli ICMP (gestisce gli errori che IP non gestisce, controlla flusso del trasferimento e risolve eventuali situazioni anomale)ed infine il Protocollo IP. Esso opera con modalità di trasferimento senza connessione (no garanzie) ed esegue le seguenti funzioni:

- ☐ Definisce il formato dei pacchetti
- ☐ Definisce lo schema di indirizzamento
- ☐ Definisce le modalità di instradamento dei pacchetti
- ☐ Esegue , se necessario, frammentazione e riassettaggio delle unità di dati.

Ogni Rete fisica ha un valore massimo di lunghezza della propria unità informativa = **MTU (Max Transmission Unit)**. Il valore minimo di MTU è 68 byte. Se un pacchetto IP è più grande della MTU della sottorete allora serve frammentazione, fatta prima del rilancio nella sottorete che poi l'host destinazione ricomporrà.

### **Formato del pacchetto**

**Vers:** 4 bit, versione del protocollo

**HLEN:** 4 bit, lunghezza dell'header, al massimo 60 byte

**Total Length:** 16 bit , lunghezza complessiva del pacchetto(specificata in byte), valore massimo 65536 byte (perchè con 16 bit al max posso rappresentare questo numero)

**Service Type:** 8 bit, specifica parametri di qualità del servizio richiesti dall'utente.

**Precedence:** 3 bit, indica il livello di priorità del pacchetto

**Type of Service(TOS):** 4 bit, indica tipo di servizio richiesto a seconda di quale bit è posto a 1.

(1000-minimize delay, 0100 max throughput, 0010 max reliability, 0001 minimize cost, 0000 normal)

**Identification:** 16 bit, numero identificativo del pacchetto da frammentare

**Flags:** 3 bit, X = posto a zero , DF= 0 permessa frammentazione, 1 frammentazione vietata, MF = 0 ultimo frammento del pacchetto, 1 non ultimo frammento

**Fragment Offset:** 13 bit, posizione del frammento all'interno del pacchetto( espresso in unità di 8 byte), per valutare se il pacchetto è arrivato tutto.

**Time to Live (TTL):** 8bit, numero massimo di router che possono essere attraversati dal pacchetto. Inizializzato dall'host sorgente e decrementa ogni salto, se zero viene scartato.

**Protocol:** 8bit, a quale protocollo di strato superiore va mandato (TCP,UDP,ICMP)

**Header Checksum:** 16 bit protegge l'intestazione del pacchetto, se viene rivelato errore il pacchetto viene scartato.

**Source address(32bit) e Destination Address(32bit)**

**Options:** lunghezza variabile a multipli di 8

**Record Route Option(RRO):** lista vuota di indirizzi IP, ogni router inserisce il suo

- ☐ **TimeStamp Option:** come RRO + l'istante in cui il pacchetto attraversa ogni router
- ☐ **Loose Source Routing Option:** lista dei router che il pacchetto attraversa
- ☐ **Strict Source Route Option:** lista di TUTTI i router che il pacchetto attraversa

**Padding** - rende l'intestazione multipla di 32 bit introducendo zeri

### **Protocollo ICMP ( Internet Control Message Protocol)**

Protocollo che è parte integrante di quello IP, incapsulato nella parte dati del pacchetto IP e lo accompagna.

**Obiettivo:** Permettere al router di inviare messaggi all'host sorgente in caso di anomalie (errori di instradamento, TTL scaduto, congestione eccessiva). Non definisce che operazioni fare in caso di errore, notifica solamente dell'errore.

- ☐ **Type:** 4bit, identifica il particolare tipo di messaggio a seconda dell'errore rilevato
- ☐ **Code:** 4bit, contiene il codice dell'errore ☐

**Redirect Message:** significa che i prossimi pacchetti devono essere trasmessi non più al router che manda il messaggio ma a quello specificato nell' ICMP, quindi si modifica la tabella di instradamento dell'host

***Time exceeded:*** TTL si è esaurito

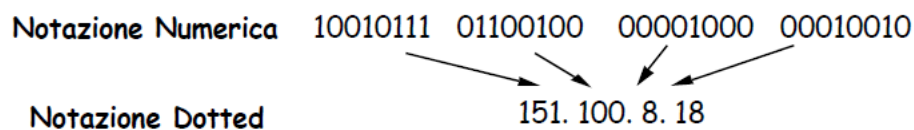
***Destination Unreachable:*** instradamento non completato

***Address mask Request & Address mask Replay:*** utilizzati per determinare la maschera della sottorete a cui è connesso un host.

**TraceRoute:** protocollo utilizzato per capire che strada farà il pacchetto e quindi che nodi attraversa. Ogni volta che il TTL scade infatti il router manda un messaggio all'host sorgente dicendo "Time Exceeded e il nome del router che lo manda".

Di modo che per ogni pacchetto individua il nodo successivo ed il tempo dalla sorgente (RTT) a quel router. Traceroute fa questo procedimento per tre volte, ma i tempi potrebbero anche esser diversi ogni volta.

L'indirizzo IP identifica un'interfaccia di Rete. Esso ha una lunghezza di 32 bit ed è unico nella rete. Noi convertiamo i 4 byte da binario (notazione numerica) a decimale (notazione dotted) e quindi per ognuna delle quattro parti potremo mettere un numero da 0 a 255.



1) Net\_id = identificativo della sottorete 2) Host\_id = identificativo dell'host all'interno della sottorete. Una sottorete è identificata da un prefisso (Net\_Id) che è quella parte dell'indirizzo IP identica per tutte le interfacce che appartengono alla sottorete



### **Schema Indirizzamento Classfull**

In origine gli indirizzi erano divisi in classi, gli indirizzi dovevano essere univoci e così a seconda di quanti host avevi bisogno ti davano un certo intervallo di indirizzi di una certa classe. Le classi erano contraddistinte dai bit iniziali dell'indirizzo e un numero di bit per il net\_id fisso.

**Classi** : A (0 + 7net\_id + 24host\_id) – B(10 + 14net + 16host) + C(110+21netid + 8hostid)

Considerando sempre 2 indirizzi in meno (tutti uni e tutti zeri dedicati al broadcast).

In conclusione gli indirizzi erano divisi in due livelli gerarchici: il net della sottorete e l'id dell'host.

### **Convenzioni Indirizzi Speciali:**

- ☐ Indirizzo Host in fase di boot – tutti zeri
- ☐ Host nella rete locale – tutti zeri (netid) e poi hostid
- ☐ BroadCast su rete locale – tutti uni
- ☐ Broadcast sulla rete netid - \_netid e tutti 1

Si è cominciato ad avere un problema quando gli utenti avevano un numero di indirizzi IP enorme ma non li usavano, ed essendo univoci non si potevano sprecare.

Nasce così il Subnetting in cui gli utenti prendevano un gruppo di indirizzi e li dividevano in due creando un terzo livello gerarchico = Subnet un livello di sottorete. Per contraddistinguersi si usavano un numero di bit dell'hostid, perdendo alcuni bit si perdevano anche un numero di indirizzi ma era necessario per la distinzione = Il subnet ID è identificato da una maschera che prende il nome di **Subnet MASK** – una parola di 32bit in cui i bit uguali a 1 identificano i bit del net\_id e del subnet\_id mentre quelli uguali a 0 identificano i bit dell'host\_id. Nell'indirizzo IP semplice la mask 'segnata da/n dove n sono i primi n bit settati a 1. Esistono due tipi di subnet:

- ☐ **Subnetting Statico** in cui tutte le subnet hanno la stessa maschera e una lunghezza fissa.

Subnet mask - (255.255.255.192)

- ☐ **Subnetting a lunghezza variabile:** permette di gestire diverse sottoreti di dimensione diversa.

Per trovare dall'indirizzo di una sottorete la sua subnet si converte l'indirizzo della sottorete in binario, si scrive la sua subnet = tanti zeri quanti bit riservati alla subnet e poi si fa l'operazione AND.

Il risultato convertito in decimale è l'indirizzo subnet.

### **Routing in reti IP**

Sia gli host che i router hanno tabelle di routing. Se l'host deve inviare ad un host nella sottorete, invia direttamente tramite l'interfaccia di rete con il suo indirizzo MAC. Se invece invia fuori dalla rete, invia al default router (al suo indirizzo MAC) e ci penserà lui.

Il Router esamina l'indirizzo IP di destinazione, se connesso a se lo manda direttamente tramite interfaccia di rete, altrimenti accede alla routing table per verificare il next-hop per quel pacchetto.

Tutto questo avviene con la tecnica del **Longest Prefix Matching**, ovvero il router sceglie la direzione corrispondente al prefisso di lunghezza maggiore.

### **Routing Table**

Contiene gli IP di destinazione + next hop associato + porta di uscita + info statistiche

### **Schema Indirizzamento ClassLess (CIDR, Classless Inter Domain Routing)**

Nasce dal problema che gli indirizzi erano in via di esaurimento e più aumentavano, più le tabelle di routing crescevano in dimensione. Il CIDR rappresentava una soluzione short-term al problema.

La classe C del classfull era quella meno utilizzata perché proponeva solo 256 indirizzi.

Con classless si aggregano blocchi di indirizzi di classe C (fino a 64 blocchi) = **Supernetting** e la rete a cui sono associati questi blocchi di indirizzi sarà unificata da un unico prefisso con lunghezza arbitraria.

Oggi i primi 7 bit sono di prefisso e sono pianificati geograficamente. Ad esempio da 194 a 195 è Europa. CIDR rallenta la crescita delle routing table perché ogni rete è rappresentata da un

prefisso e una maschera e una rete con molti blocchi nella routing table avrà una entry sola.

### **DHCP**

Un host deve essere configurato con IP address, Subnet mask, Default router e Server DNS  
DHCP (***Dynamic Host Configuration Protocol***) autocompila i suddetti campi.

Consente ad un host di ottenere dinamicamente il suo indirizzo IP dal server di rete:

### **Panoramica del DHCP**

L'host invia un messaggio broadcasts "***DHCP discover***": È emesso in modo broadcast da un client per trovare un DHCP server

Il server DHCP invia l'indirizzo con il messaggio "***DHCP offer***" in risposta al DHCP discover

L'host richiede la configurazione con il messaggio "***DHCP request***"

Il server DHCP invia la configurazione richiesta con il messaggio "***DHCP ack***"

Supporta tre meccanismi per la gestione degli indirizzi IP:

***Allocazione automatica:*** DHCP assegna permanentemente un indirizzo IP

***Allocazione dinamica:*** DHCP assegna un indirizzo IP per un intervallo limitato di tempo

***Allocazione manuale:*** L'indirizzo IP è assegnato dall'amministratore di rete

### **Struttura**

***Code:*** Indica una richiesta o una risposta

***HW type:*** Tipo di hardware (es. ethernet, IEEE 802)

***Length:*** Lunghezza del campo client HW address

***Transaction ID:*** Pacchetti di richiesta e di risposta hanno lo stesso numero

***Seconds:*** Indica il tempo trascorso dall'avvio della procedura di boot

***Flag:*** Indica se il pacchetto è unicast o broadcast

***Client IP address:*** È settato dal client, se il client non conosce il proprio indirizzo il suo valore è 0.0.0.0

***Your IP address:*** Indirizzo IP del client assegnato dal server

***Server IP address:*** Indirizzo IP del server

***Client HW address:*** Indirizzo MAC del client

***Options:*** Parametri di configurazione aggiuntivi: router di default, subnet mask, domain name server...

### **Pro**

Semplifica la gestione amministrativa degli indirizzi in rete e l'accesso in rete di utenti in mobilità

Rende possibile l'uso efficiente di un insieme di indirizzi IP dimensionando opportunamente il tempo di lease

### **Contro**

Non garantisce un vero e proprio "plug and play":

Deve essere previsto un server DHCP in rete e gli host devono essere configurati per usare DHCP

DHCP non è sicuro, un utente non autorizzato può accedere alla rete

Problemi di interoperabilità con DNS in caso di riallocazione dinamica degli indirizzi

### **NAT(Network Address Translator)**

Riduce l'utilizzazione dello spazio di indirizzi IP

È utilizzato in una Intranet in cui vengono assegnati un insieme di indirizzi IP pubblici che sono visibili dalle rete esterne

All'interno della Intranet possono essere utilizzati liberamente indirizzi IP privati, anche non unici in rete, appartenenti alla seguenti classi:

Indirizzi di classe A: 10.0.0.0

Indirizzi di classe B: da 172.16.0.0 a 172.31.0.0

Indirizzi di classe C: da 192.68.0.0 a 192.168.255.0

Il dispositivo NAT assegna un indirizzo pubblico ad un host solo nel momento in cui questo deve comunicare con l'esterno eseguendo la traslazione dell'indirizzo privato con un indirizzo pubblico.

Un NAT nasconde i dettagli di una Intranet al mondo esterno infatti è possibile cambiare gli indirizzi delle macchine di una rete privata senza doverlo comunicare all'Internet globale

Dispositivi interni alla rete non esplicitamente indirizzabili e visibili dal mondo esterno (un plus per la sicurezza)

***Quando un router NAT riceve il pacchetto dalla rete locale:***

Genera un nuovo numero di porta d'origine (es. 5001)

Sostituisce l'indirizzo IP di sorgente (privato) con il proprio indirizzo IP (pubblico) sul lato WAN (es. 138.76.29.7)

Sostituisce il numero di porta origine iniziale (es. 3348) con il nuovo numero (5001)

***Quando un router NAT riceve il pacchetto da Internet:***

Legge il numero di porta (es. 5001) ed individua il mapping con l'indirizzo interno

Sostituisce l'indirizzo IP di destinazione con l'indirizzo IP privato dell'host di destinazione

Sostituisce il numero di porta di destinazione (5001) con il numero di porta iniziale (3348)

Il campo numero di porta è lungo 16 bit e il protocollo NAT può supportare più di 60.000 connessioni simultanee con un solo indirizzo IP sul lato WAN

**Instradamento**

Per avviare l'instradamento occorre innanzitutto creare le tabelle di routing, ovvero definire le informazioni che voglio avere riguardo ai link, definire ogni quanto voglio ricevere queste informazioni, e calcolare i cammini migliori per i vari nodi.

Esistono due tipi di instradamento:

- ☐ ***Routing Centralizzato:*** i cammini vengono determinati dal nodo centrale, esso gestisce tutto, ma se si rompe, tutta la rete viene bloccata + difficile notificarlo di cambio topologia della rete
- ☐ ***Routing distribuito:*** i router determinano i cammini con un algoritmo distribuito, e i router si scambiano info utili in caso di cambi di topologia.

***Routing Flat:*** tutti i router sono allo stesso livello (peer)

***Routing gerarchico:*** si divide la rete in domini, sistemi autonomi, alcuni router sono nella backbone, mentre altri comunicano solo con quelli vicini.

***Instradamento Statico:*** cammini configurati manualmente, per reti semplici o per imporre cammini particolari o per definire router di default

***Instradamento Dinamico:*** calcolo automatico dei cammini in base alle info ricevute.

***Indirizzamento e Instradamento non Gerarchici:*** gli indirizzi vicini non hanno relazioni e quindi la dimensione della routing table risulta enorme. □

***Indirizzamento e instradamento gerarchici:*** I prefissi indicano la rete a cui un host è connesso e le reti con lo stesso prefisso sono "vicine"

***Protocolli di instradamento:*** Un sistema autonomo AS è un insieme di router ed host controllato da una singola autorità amministrativa. Un AS particolare è il **CORE AS** (Backbone di internet) il cui router è detto Core Router. Tutti gli altri AS nel mondo devono avere almeno un router connesso ad un core router che viene chiamato **Exterior Gateway**.

Tutti gli altri router interni all'AS vengono invece chiamati **Interior Gateway**.

I router devono quindi sapere se hanno più di un router connesso all'esterno, e sapere ognuno chi raggiunge.

Se il router ha E deve anche sapere quali altri AS può raggiungere tramite gli AS a cui è collegato, se è collegato solo da un gateway allora è detto **Gateway unico**, se invece è collegato da più AS allora è un **Gateway multiplo** e in quel caso il router sceglie quello a cammino minimo.

I protocolli di instradamento all'interno di un AS sono detti **Interior Gateway Protocols** (IGP / intra-AS), mentre le informazioni di instradamento che coinvolgono più di un sistema autonomo sono gestite mediante gli **Exterior Gateway Protocols** (EGP)

□

#### **EGP - svolge tre funzioni:**

Individuazione dei router adiacenti con cui scambiare le informazioni di instradamento

Verifica continua della funzionalità dei router interlocutori

Scambio periodico delle informazioni di instradamento, queste riguardano la sola raggiungibilità delle reti, non la distanza

Quello standard oggi è il **BGP (Border Gateway Protocol)**: non si occupa di scegliere i cammini minimi dentro ai sistemi autonomi, bensì tra i vari AS. Invia messaggi tramite il protocollo TCP di trasporto.

□

#### **Terminologia BGP**

**BGP speaker**: Un router che supporta il protocollo BGP, il quale non necessariamente coincide con un border router

**BGP Neighbors**: Una coppia di BGP speaker che si scambiano informazioni di instradamento inter-AS. Possono essere di due tipi:

**Interni**: se appartengono allo stesso AS

**Esterni**: se appartengono ad AS diversi

**BGP session**: La connessione TCP che supporta il colloquio tra due BGP speaker

**AS number**: Identificatore a 16-bit che identifica univocamente un AS

**AS path**: è la lista di AS che sono attraversati in un cammino

**Politiche di routing**: nel protocollo BGP non sono definite regole fisse per la scelta dei cammini inter-AS, ma le regole sono definite dal gestore di ogni AS:

Un AS multi-homed può rifiutare di operare come AS di transito e farlo solo per alcuni AS

Un AS può scegliere a quale altro AS affidare il traffico di transito

Tra le possibili scelte un BGP speaker sceglie quella da preferire in base alla politica di routing fissata dal gestore, in caso di cammini alternativi, un BGP speaker li mantiene tutti ma ne comunica uno solo agli altri AS

**Traffico**:

**Locale**: Traffico generato o terminato nell'AS

**Transito**: Traffico che non è locale

**AS type**:

**Stub**: Uno stub AS ha una singola connessione inter-AS, trasporta solo traffico locale

**Multihomed**: Ha un insieme di connessioni verso una molteplicità di altri AS, ma non trasporta traffico di transito

**Transit**: Ha un insieme di connessioni verso una molteplicità di altri AS, e trasporta anche traffico di transito

#### **Attributi del percorso e rotte BGP**

Quando un router annuncia un prefisso per una sessione BGP, include anche un certo numero di attributi BGP: Prefisso + attributi = "rotta"

Due dei più importanti attributi sono l'**AS-PATH** che elenca i sistemi autonomi attraverso i quali è passato l'annuncio del prefisso e il **NEXT-HOP**: quando si deve inoltrare un pacchetto tra due sistemi

autonomi, questo potrebbe essere inviato su uno dei vari collegamenti fisici che li connettono direttamente.

Quando un router gateway riceve un annuncio di rotta, utilizza le proprie politiche d'importazione per decidere se accettare o filtrare tale rotta

### **Selezione dei percorsi BGP**

Un router può ricavare più di una rotta verso un determinato prefisso, e deve quindi sceglierne una

#### **Regole di eliminazione:**

Alle rotte viene assegnato come attributo un valore di preferenza locale. Si selezionano quindi le rotte con i più alti valori di preferenza locale

Si seleziona la rotta con valore AS-PATH più breve

Si seleziona quella il cui router di NEXT-HOP è più vicino: instradamento "hot potato"

Se rimane ancora più di una rotta, il router si basa sugli identificatori BGP

### **Messaggi BGP**

I messaggi BGP vengono scambiati attraverso TCP e possono essere di tipo:

**OPEN:** Apre la connessione TCP e autentica il mittente

**UPDATE:** Annuncia il nuovo percorso (o cancella quello vecchio)

**KEEPALIVE:** Mantiene la connessione attiva in mancanza di UPDATE

**NOTIFICATION:** Riporta gli errori del precedente messaggio; usato anche per chiudere il collegamento

**IGP** – quelli più noti sono:

**RIP (Routing Information Protocol):** utilizzato in reti di piccole e medie dimensioni, utilizza l'algoritmo *Distance Vector Protocol*.

Emette due tipi di messaggi:

1) **Request** (per chiedere ai vicini il distance vector)

2) **Response** (per annunciare il suo distance vector).

Questi messaggi vengono scambiati ogni 30 secondi e sono limitati a 25 sottoreti. Ogni messaggio riporta (l'indirizzo di destinazione, subnetmask, next-hop e mask + metriche = n di archi(distanza) e nome del link) . Se un link risulta guasto utilizza la tecnica dell'*inversione avvelenata*.

### **Terminologia OSPF**

**Area:** è un insieme logico di reti e di router (geografico, amministrativo, ...) che ha lo scopo di limitare la dimensione dei database di descrizione della topologia di rete all'interno dei router

All'interno di un'area i router devono avere database identici che descrivono la topologia di rete

**Border Router:** Un Area Border Router trasmette LSA contenenti informazioni sulle reti esterne all'interno dell'area (costo di raggiungimento)

Tutte le reti OSPF devono essere composte da almeno un area, denominata area di backbone

**Intra-Area Router (IAR):** Sono i router che sono situati all'interno di una area OSPF che scambiano LSA con tutti gli altri router dell'area e gestiscono il database relativo alla topologia dell'area

**Area Border Router (ABR):** Sono i router che sono connessi a due o più aree OSPF e gestiscono i database di topologia di tutte le aree a cui sono connessi trasmettendo all'interno di ogni area LSA relativi alle reti presenti in ogni area

**AS Boundary Router (ASBR):** Sono i router che sono situati a bordi del dominio OSPF e scambiano LSA contenenti informazioni di raggiungibilità di reti di altri AS, inviando LSA all'interno del dominio con informazioni sui percorsi esterni

### **Tipi di LSA**

**Link State Advertisements (LSA):** Sono i pacchetti scambiati tra router OSPF per aggiornare i link state database e i percorsi inter-area e inter-AS

**Router link advertisement:** Indicano lo stato dei link uscenti da un router, sono inviati all'interno di una singola area

**Summary link advertisement:** Sono generati dagli ABR e individuano le reti contenute nelle altre aree ed i relativi costi di raggiungimento, sono inviati all'interno di tutte le aree gestite da un ABR

**AS external link advertisement:** sono generati dagli ASBR e indicano i cammini verso le reti esterne al dominio OSPF, sono inviati all'interno di tutte le aree di un dominio OSPF

**Tutti i tipi di LSA hanno lo stesso header:**

**Link State Age:** Indica il tempo (in secondi) di emissione dell'advertisement

**Link State Type:**

- 1: Router link
- 2: Network link
- 3: Summary link: Inter-area, intra-AS route
- 4: Summary link: Route verso l'AS Boundary Router
- 5: AS External link: Route verso reti esterne all'AS

**Link State ID:** Indica il tipo di link a cui si riferisce il messaggio

Tipo 1 e 4: indirizzo IP del Router emittente

Tipo 3 e 5: indirizzo IP della rete a cui si riferisce il messaggio

Tipo 2: indirizzo IP del DR emittente

**Advertising Router:** Indirizzo IP del router che ha emesso il messaggio

Tipo 1 : identico al campo Link State ID

Tipo 2: indirizzo IP del DR

Tipo 3 e 4: indirizzo IP del ABR Tipo 5: indirizzo IP del ASBR

Tipo 5: indirizzo IP del ASBR

**Network Mask:** Maschera della rete a cui si riferisce il pacchetto, l'indicazione della rete è contenuta nell'header

**Metric:** Costo del cammino

**Forwarding Address:** Indirizzo IP a cui deve essere inviato il traffico diretto alla rete indicata

**External Route Tag:** Suffisso ad uso degli ASBR

**OSPF(Open Shortest Path First):** utilizzato in reti backbone, utilizza l'algoritmo **Link State Protocol** e diffonde un messaggio sulla rete con il **flooding** (Esplora tutti i possibili cammini tra origine e destinazione) che si chiama **Link State Advertisement LSA** che viene emesso ogni volta che si verifica un cambiamento sul suo stato.

Se due percorsi hanno lo stesso costo non devo sceglierne uno, posso usare tutti e due a differenza di **RIP**. Gli LSA trasportano con se dei riferimenti temporali di modo che gli altri possano confrontare il riferimento più recente il che risulta più sicuro perché i router sono autenticati, inoltre risolve il problema di reti molto grandi con database grandi adottando un'instradamento gerarchico: divide la rete in aree connesse da un area backbone.

Quindi non tutti hanno bisogno di conoscere la topologia totale, ci si manda pacchetti OSPF che contengono tutte le info necessarie affinché si possa conoscere la parte di rete che mi interessa.

### **IGRP (protocollo proprietario CISCO)**

Il protocollo di instradamento mette in grado ogni router di determinare il modello a grafo della rete.

### Modello a grafo di una rete

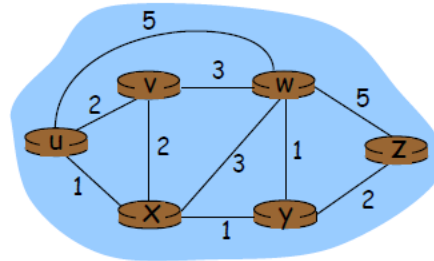
$N$  = insieme di nodi (router) = { u, v, w, x, y, z }

$E$  = insieme di archi (collegamenti) = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

$c$  = insieme dei costi associati ai rami:

$c(x,x') = \text{costo associato ramo } (x,x')$

Grafo Pesato  
 $G = (N, E, c)$



### Algoritmi di instradamento

Determina il cammino a costo minimo tra due nodi della rete (dove il costo di un cammino è definito come la somma di tutti i costi degli archi lungo il cammino).

Negli approcci troviamo due modalità : **Distance Vector Protocol** (dove i nodi adiacenti si scambiano la lista delle distanze (**vettori distanza**) e ognuno determina il next-hop migliore, ogni nodo avvisa solo i suoi vicini quando qualcosa cambia, es. Algoritmo Bellman-Ford), **Link State Protocol** (le informazioni sui costi vengono diffuse in rete (flooding) e ogni router conoscendo l'intera topologia calcola il cammino minimo es. Algoritmo Dijkstra).

□

### Calcolo dei cammini minimi

Se  $D_i$  è la distanza minima dal nodo  $j$  e  $S_j$  e se  $j$  è il nodo adiacente a  $i$  che si trova sul percorso a costo minimo dal nodo  $i$  verso  $S_j$ , si ha:

$$D_i = C_{ij} + D_j$$

### Algoritmo Bellman-Ford

Ogni nodo ha una riga per ogni destinazione, distanza dal nodo  $X$  a se stesso = 0 mentre per tutte le altre distanze viene posta a infinito. A ogni iterazione la destinazione avvisa i suoi vicini sui costi e i vicini aggiornano la tabella.

Se però qualcosa si rompe e la connessione con la destinazione si interrompe avviene un ciclo di conteggi all'infinito tra i nodi

### Soluzioni al conteggio all'infinito

**Split Horizon:** Un router non trasmette il proprio DV aggiornato verso il router da cui ha ricevuto l'aggiornamento

**Poisoned Reverse:** Un router trasmette il proprio DV aggiornato anche verso il router da cui ha ricevuto l'aggiornamento, ma indicando per la distanza aggiornata al valore  $\infty$

### Algoritmo Dijkstra

Analizza il costo minimo tra se stesso e tutti gli altri, aumentando a ogni iterazione la distanza. Come ricerca operativa viene aggiunto ad ogni ciclo l'insieme dei nodi raggiunti.



## **STRATO DI TRASPORTO**

### **Multiplexing**

A questo livello l'host emittente raccoglie i dati da vari socket (punto di incontro tra processo e canale di comunicazione), li incapsula in pacchetti IP con la sua intestazione e li trasmette all'host destinatario.

### **Demultiplexing**

L'host ricevente riceve i pacchetti IP che trasportano i segmenti.

Per decidere a quale socket mandare l'info ricevuta dipende dal tipo di comunicazione:

**Con connessione:** All'interno dei pacchetti troviamo: indirizzo IP sorgente + indirizzo IP dest + numero porta origine + numero porta dest. Grazie alla combinazione di questi dati l'host decide a quale socket mandare il segmento. Con questo sistema si supportano più connessioni contemporaneamente. (TCP)

**Senza connessione:** All'interno dei pacchetti troviamo : indirizzo ip dest + numero porta destinazione = l'host controlla solo il numero di porta e manda alla socket adatta. Quindi non dipende dall'origine. (UDP)

L'elenco dei numeri di porta è gestito da IANA in tempo reale, alcune delle app più diffuse hanno numeri di porta assegnati (**well-known port numbers**), in generale è il sistema operativo che li assegna all'apertura della connessione.

### **TCP**

E' un protocollo con connessione che interpreta il flusso dati proveniente dallo strato applicativo.

#### **Funzioni:**

- ☐ Indirizzamento di un'applicazione
- ☐ Controllo di sequenza delle unità informative
- ☐ Controllo e recupero di errore
- ☐ Controllo di flusso (regolare quantità dati in trasmissione x fare in modo che l'host ricevente li riceva correttamente)
- ☐ Controllo di congestione (regolare quantità dati in trasmissione x non sovraccaricare la rete)

### **Struttura**

#### **Source port- Dest port**

**Sequence number:** numero d'ordine del primo byte dati nel campo dati

**Acknowledgment Number:** se il bit di ACK in Flag vale 1, contiene numero di sequenza del prossimo byte che il ricevente si aspetta di ricevere.

**HLLEN:** numero di parole da 32 bit contenute nell'intestazione del segmento.

**Flag:** a seconda del bit che imposto ad 1 ho diversi significati:

**URG:** è uguale a uno quando il campo "Urgent Pointer" contiene un valore significativo

**ACK:** è uguale a uno quando il campo "Acknowledgement Number" contiene un valore valido

**PSH:** è uguale a uno quando l'applicazione indica che i dati vengano consegnati all'applicazione ricevente prescindendo dal riempimento dei buffer di ricezione

**RST:** è uguale a uno in caso di richiesta di re-inizializzazione della connessione

**SYN:** è uguale a uno solo nel primo segmento inviato durante la fase di sincronizzazione fra le entità TCP

**FIN:** è uguale a uno quando la sorgente ha esaurito i dati da trasmettere

**Window:** larghezza della finestra che permette di gestire il controllo del flusso

**Checksum:** protegge il segmento ☐

**Urgent pointer:** numero di sequenza dei dati da mandare

**Options - Padding**

### Funzionamento

Handshake a tre vie:

1. Instaurazione della connessione: host A invia segmento SYN all'host B, specificando il numero di sequenza iniziale (x) + gli identificatori dei socket (ip, porta)
2. Host B riceve SYN, alloca i buffer per la ricezione e risponde con un segmento SYN ACK specificando i suoi identificatori di socket e il numero di sequenza iniziale del server(x+1).
3. Host A riceve SYN ACK e risponde con un segmento ACK che contiene già in esso i dati.

### Maximum Segment Size (MSS)

Quando l'entità TCP emittente invia la prima TCP- PDU (SYN) può inserire l'informazione relativa alla massima dimensione del campo dei dati di utente di una TCP-PDU (Maximum Segment Size - MSS)

L'entità ricevente risponde comunicando la propria MSS, nel caso di uno scambio bidirezionale, la dimensione della MSS è scelta in modo indipendente nei due versi e può quindi essere diversa nelle due direzioni

### Chiusura della connessione

1. Host A invia un segmento di controllo FIN al server
2. Host B riceve il segmento FIN e risponde con un ACK
3. Host B chiude la connessione e invia un FIN
4. Host A riceve FIN e risponde con ACK
5. Alla ricezione dell'ACK si avvia un timer al termine del quale si chiude la connessione.

### Controllo di sequenza

Il numero di sequenza è il numero del primo byte del prossimo segmento, esso viene trasmesso insieme all'ACK che viene aggiornato dal ricevente per dire all'emittente da che punto in poi trasmettere ogni volta. (es.  $N^* \text{ seq} = 42$ , lunghezza segmento = 6 – prossimo  $n^* \text{ seq} = 42 + 6 + 1 = 49$ )  
La gestione dei segmenti fuori sequenza non è specificata però in questo standard

### Controllo d'errore

L'entità TCP emittente inserisce una codifica per la rivelazione d'errore nel checksum, e l'entità TCP ricevente la utilizza per farne il controllo. L'entità ricevente può mandare i suoi riscontri ACK con segmenti vuoti appena fa il controllo(*immediata*) o in *piggybacking* se sta trasmettendo anch'essa, tenendoli quindi da parte finché anche lei trasmette qualcosa.

Per evitare però un ritardo troppo lungo attiva un timer che al suo scadere, se non deve mandare nulla, lo manda con il segmento vuoto.(*cumulativa*).

A ogni invio di un segmento, l'emittente attiva un temporizzatore che viene disattivato alla ricezione dell'ACK se questo avviene nel tempo di *retransmission timeout (RTO)*.

Il calcolo del timeout è però dinamico, perché il tempo di trasmissione segmento+ricezione ACK può variare.

Quindi al primo invio il timer è al suo valore massimo, via via che mando i segmenti faccio una media pesata dei tempi e do un valore al RTO aggiungendo sempre un margine di sicurezza dato dalla deviazione che posso avere.

Con questo controllo TCP si da servizio affidabile al servizio inaffidabile IP.

### Round Trip Time (RTT) e timeout

**SampleRTT**: tempo misurato dalla trasmissione di un segmento fino alla ricezione dell'ACK relativo (Ignora le ritrasmissioni), esso varia, quindi occorre una stima "smoothed" di RTT:

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

L'influenza dei vecchi campioni decresce esponenzialmente; Valore tipico:  $\alpha = 0,125$

### Stima della deviazione standard dell'EstimatedRTT

$$\text{DevRTT} = (1 - \beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}| \quad \text{con } \beta = 0,25$$

### Valore Retransmission TimeOut (RTO)

$$\text{RTO} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$

### Exponential RTO Backoff

Determina il valore di RTO associato ad un segmento rimesso:

è consigliabile variare il valore di RTO sui segmenti rimessi perché l'esaurimento dell'RTO è dovuto a congestione in rete, allo stesso modo è consigliabile variare il valore di RTO delle sorgenti che sono coinvolte nella congestione per evitare rimissioni contemporanee

Una sorgente TCP aumenta il valore di RTO per ogni riemissione (exponential backoff process) (normalmente  $q=2$ )

$$\text{RTO}_{i+1} = q \text{RTO}_i$$

### Algoritmo di Karn

L'entità TCP ricevente non distingue se il riscontro si riferisce alla prima emissione del segmento (RTO troppo elevato con perdita di efficienza e inutili ritardi) o alla riemissione del segmento (TRO troppo breve e quindi rimissioni eccessive e nuovi errori di misura).

L'algoritmo di Karn stabilisce di non considerare il RTT dei segmenti rimessi, usare come RTO il valore dato dalla procedura di exponential backoff e ricalcolare il nuovo valore di RTO solo al momento della ricezione di un ACK di un segmento non rimesso

### Fast retransmit

Il periodo di timeout spesso è relativamente lungo perciò l'entità TCP emittente può rivelare precocemente i segmenti perduti tramite l'analisi degli ACK duplicati:

L'entità TCP emittente spesso invia molti segmenti e se un segmento viene smarrito, è probabile che ci saranno molti ACK duplicati

Se l'entità TCP emittente riceve 3 ACK duplicati per lo stesso dato, suppone che il segmento che segue il dato riscontrato sia andato perduto e effettua una ritrasmissione rapida prima che scada il timer

### Controllo di Flusso

Ha lo scopo di limitare il ritmo di emissione dei dati da parte di un host per evitare la saturazione della capacità del buffer di ricezione. Il ricevente comunica la sua disponibilità massima tramite la

Recwindow (finestra di ricezione) che per l'emittente è il numero massimo di dati che può trasmettere senza ricevere un ACK consecutivo e li cumula.

La procedura di controllo di flusso TCP utilizza i seguenti parametri:

**SN (Sequence Number)**: riferito al primo ottetto contenuto nel segmento

**AckN (Acknowledgement Number)**: riferito al prossimo ottetto che l'entità ricevente aspetta di ricevere

**RecWindow (Window)**: esprime il numero massimo di ottetti che l'entità emittente può emettere senza ricevere un riscontro per alcuno di questi

### Throughput di una connessione TCP

Il throughput (TH) di una connessione TCP, nell'ipotesi di overhead nullo e di assenza di ritrasmissioni, è dato da:

$$TH = \begin{cases} 1 & \text{se } W \geq \lceil 2\alpha + 1 \rceil \\ \frac{W}{2\alpha + 1} & \text{se } W < \lceil 2\alpha + 1 \rceil \end{cases}$$

Dove:

C è il ritmo binario della connessione

W è la larghezza della finestra

$\Delta$  è il ritardo di propagazione sulla connessione

$\alpha = C \Delta / 8$  (rapporto tra ritardo di propagazione e tempo di trasmissione di un ottetto)

Se si suppone  $2\alpha \gg 1$ , risulta allora:

$$TH = \begin{cases} 1 & \text{se } W \geq C\Delta / 4 \\ \frac{4W}{C \cdot \Delta} & \text{se } W < C\Delta / 4 \end{cases}$$

In funzione della larghezza della finestra W (in ottetti) e del prodotto banda ritardo C  $\Delta$  (in bit)

### Controllo di congestione

Ha l'obiettivo di non sovraccaricare la rete per evitare di avere ritardi enormi o peggio perdita di pacchetti.

Il metodo adottato da TCP per effettuare quest'operazione è il controllo **punto-punto** ovvero deduce la disponibilità osservando le perdite e i ritardi, l'esaurimento di RTO e imposta quindi una

**Congwindow** (finestra di congestione) che si affianca a quella di ricezione per impostare un limite alla trasmissione di dati.

Questa finestra cambia valore però continuamente e opera con una procedura ciclica in due fasi:

**slow start** = inizio piano (congrwin=1MSS e soglia=congrwin/2), se la rete ed il ricevente ce la fanno continuo ad incrementare esponenzialmente la congrwin fintanto che non rilevo perdite inizio quindi la L'entità emittente determina nel tempo il valore della finestra disponibile (**Available Window - Awdn**):

Awdn = numero di segmenti di lunghezza massima (MSS) che possono essere inviati senza riscontro

Il valore di Awdn non deve superare il minimo tra le larghezze Congwin della finestra di congestione e

RecWindow della finestra di ricezione:  $Awdn \equiv \min \{Congwin, RecWindow\}$

Congwin ed RecWindow sono quantità espresse in numero di segmenti MSS

**RecWindow** è la larghezza comunicata nell'ultimo ACK ricevuto e ottenuta dall'entità TCP emittente dividendo il numero contenuto nel campo Window di questo ACK per il numero di ottetti che compongono una MSS

### Additive-Increase Multiplicative-Decrease

Aumenta il valore di CongWin (sondando la rete) fino a quando non si verifica una perdita

**Incremento additivo**: Aumenta CongWin di 1 MSS a ogni RTT in assenza di eventi di perdita

**Decremento Moltiplicativo**: Riduce a metà CongWin dopo un evento di perdita

### Controllo di congestione TCP

Approssimativamente il rate di emissione dei segmenti è dato da:

$$\text{Frequenza d'invio} = \frac{\text{CongWin}}{\text{RTT}} \text{ byte/sec}$$

Il mittente percepisce la congestione se si esaurisce il timeout o riceve 3 ACK duplicati

Il mittente TCP riduce la frequenza d'invio (CongWin) dopo un evento di perdita

Quando si verifica un evento di perdita si pone:

$$\text{CongWin}(\text{new}) = 1 \text{ MSS}$$

$$\text{Soglia} = \text{CongWin}(\text{old})/2$$

Quando inizia la connessione, la frequenza aumenta in modo esponenziale, fino a quando non si verifica un evento di perdita:

Raddoppia CongWin a ogni RTT, ciò avviene incrementando CongWin per ogni ACK ricevuto

### Congestion avoidance

Se l'aumento che si ha nella fase Slow Start raggiunge e supera il valore di soglia (*Threshold*), e cioè se  $\text{Congwin} \geq \text{Soglia}$ , l'incremento di Congwin diventa lineare al crescere di RTT:

Se  $\text{Congwin} = w$  e se  $w \geq \text{Soglia}$ , dopo l'arrivo di  $w$  riscontri consecutivi, la larghezza Cwnd viene incrementata di 1 MSS in ciascun RTT in cui si registra l'arrivo di un intero gruppo di riscontri dei contenuti della finestra di congestione

Questo incremento lineare continua finché i riscontri arrivano prima dei loro rispettivi RTO

Questo aumento ha un limite superiore corrispondente al raggiungimento di uno stato di saturazione su uno dei collegamenti lungo il percorso o in uno dei nodi attraversati

Nell'ipotesi che  $\text{Congwin} < \text{Recwindow}$ , il limite superiore dell'aumento della Congwin è determinato dal verificarsi di un evento di perdita di un segmento e di un conseguente raggiungimento del relativo RTO

In conclusione, se si trascura la fase di slow start, una entità TCP emittente:

Incrementa Cwnd di 1 Seg.MSS per ogni RTT quando il suo percorso di rete non è congestionato

Diminuisce Cwnd di un fattore 2 per ogni RTT quando il percorso è congestionato

Per questo motivo questa procedura di controllo di congestione è usualmente indicata come algoritmo di incremento additivo e di decremento moltiplicativo (AIMD, Additive-Increase, Multiplicative-Decrease)

### Riassunto

Quando CongWin è sotto la soglia, il mittente è nella fase di slow start; la finestra cresce in modo esponenziale

Quando CongWin è sopra la soglia, il mittente è nella fase di congestion avoidance; la finestra cresce in modo lineare

Quando si verificano tre ACK duplicati, il valore di Soglia viene impostato a  $\text{CongWin}/2$  e CongWin viene impostata al valore di Soglia

Quando scade il timeout, il valore di Soglia viene impostato a  $\text{CongWin}/2$  e CongWin è impostata a 1 MSS

### **Throughput TCP**

Ignoriamo le fasi di slow start; Sia  $W$  la dimensione della finestra quando si verifica una perdita  
Quando la finestra è  $W$ , si ha:

$$TH1 = W / RTT$$

Subito dopo la perdita, la finestra si riduce a  $W/2$ , quindi:

$$TH2 = W/2 * RTT$$

Poiché l'aumento è lineare:

$$T \quad H = (TH1 + TH2)/2 = 0,75 W/RTT$$