

# Appunti FCI

## Storia di internet

### Storia

Leonard Kleinrock è stato il primo a spedire un pacchetto alla UCLA. Lo scopo di internet era di creare una rete dove era possibile distribuire tra vari computer le risorse. Il primo esperimento è stato di fare un login da un computer della UCLA a uno della SRI, ma è fallito: dei tre caratteri (LOG) sono stati inviati solo i primi 2 (LO). La prima rete è stata ARPAnet.

Negli anni '70 nasce il primo protocollo di internet (NCP) e il primo programma di e-mail. La prima rete wireless è stata la ALOHAnet. Nel '76 nasce Ethernet. Nel '82 nasce la base dell'internet moderno: SMTP, TCP/IP, DNS. Nascono nuove reti nazionali. Le prime applicazioni furono telnet (remote login), e-mail, FTP.

Negli anni '90 ARPAnet viene dismessa. Tim Berners-Lee inventa il Web moderno al CERN. Nasce il primo browser (Mosaic, diventato Netscape e poi Mozilla) e l'internet viene commercializzato e diventa com'è oggi.

### Commento di Kleinrock sulla rete

Il problema principale della rete è che è stata concepita per essere utilizzata da piattaforme fidate. Con l'adozione di massa, la sicurezza è diventata un problema e aggiungerla a questo punto è difficile.

L'internet ha, però, ha un grande potenziale in quanto permette lo sviluppo di applicazioni che non sono state concepite al momento della sua creazione: si pensi ai contenuti multimediali, appstore e tutti i servizi moderni.

Il prossimo passo di internet è diventare trasparente, "come la corrente elettrica". Ciò si potrà raggiungere con l'avvento dell'IoT.

## Internet oggi

Il traffico aumenta ogni anno. La maggior parte del traffico odierno è multimediale. Il numero di utenti è altissimo. Viene utilizzato principalmente IPv4. IPv6 è meno diffuso e non sarà scopo del corso. La maggior parte della capacità della rete è fornita da cavi sottomarini (le dorsali).

La rete oggi è molto densa: per andare da un punto all'altro basta attraversare solo pochi nodi. In totale i nodi sono più di un miliardo.

## Programma del corso

Ha un approccio TOP-DOWN: si parte dalle architetture e si arriva ai metodi di trasmissione dei bit:

1. Introduzione e architetture
2. Sistemi di comunicazione
3. Modelli funzionali
4. Livello fisico
5. Protocolli applicativi
6. Il livello di trasporto (UDP, TCP)
7. Networking (IP, DHCP, DNS, ICMP)
8. Inoltro e instradamento in internet (Routing)
9. Reti locali e livello di linea (Reti locali, bridge/switch, Ethernet, WIFI)

## Laboratorio

1. Sniffer di rete
2. Ping, traceroute, Dig, strumenti browser (chrome)
3. Protocolli applicativi
4. Programmi socket (python)
5. Configurazione e simulazione di rete (PT)
6. Attività sperimentali su reti wireless

## Cos'è internet

1. Architettura fisica fatta di componenti
2. Un'architettura di rete
3. Servizio di comunicazione che ci permette di trasferire informazione tramite un protocollo di comunicazione

## Componenti fisici

- Sono i terminali di rete (host)
- Canali di comunicazioni (link) (fibre ottiche e simili)
- Router o nodi di rete
- Altri nodi

**Gli host** Gli host sono i nostri pc, dei server o delle macchine virtuali. Anche telefoni, smartwatch e IoT sono host. Gli host sono in grado di ricevere o mandare informazioni per le loro applicazioni

**I link** Possono essere di natura fisica diversa: fibre ottiche, doppini telefonici, antenne radio e cavi coassiali. La differenza principale tra i vari link è la tecnologia usata per trasmettere i dati e la velocità di trasmissione dei suddetti (rate, misurato in Kb/s ecc...)

**Nodi di rete** I router sono unità che smistano informazione suddivisa in pacchetti: sequenze finite di bit.

Oltre ai router esistono altri tipi di nodi che svolgo compiti di collegamento:

- i switch
- gli access point wifi

## Architettura fisica

**Accesso a internet** Per accesso si intende la tecnologia che consentono alla nostra rete locale di collegarsi al resto di internet. Prima, però, si passa per la rete dell'ISP.

Per accedere alla rete dell'ISP ci sono molte tecnologie diverse:

- Dialup (modem analogico)
  - Fino a 56KB/s
  - Accesso diretto tramite circuito telefonico
  - Trasmissione segnale in banda fonica
- ADSL (asymmetric digital subscriber line)
  - Sempre basata sul doppino telefonico, ma non passa per la rete telefonica permettendo velocità più alte. Poiché condivide l'accesso (doppino) con il telefono, esiste un meccanismo di splitting tra il segnale del modem e quello telefonico (nella centrale telefonica)
  - L'unico limite in velocità è il mezzo fisico stesso
- Fibra ottica
  - Sta sostituendo il doppino
  - Ci sono diversi tipi di collegamento:
    - \* Fino a casa
    - \* fino a edificio: il pezzo fino all'utilizzatore è in doppino
    - \* fino quartiere: il pezzo fino all'utilizzatore è in doppino
- Rete cellulare (reti radio)
  - Dipende dal mezzo
  - Sono sistemi molto più complessi

**Reti di reti** Internet è un puzzle di reti interconnesse tramite diverse tecnologie. La tecnologia di internet più essere usata per interconnettere reti di tipo diverso:

- le diverse porzioni di rete sono composte da tecnologie diverse e possono essere anche "sottoreti" (reti locali che gestiscono autonomamente i propri nodi e link)
- I link sono di vario tipo

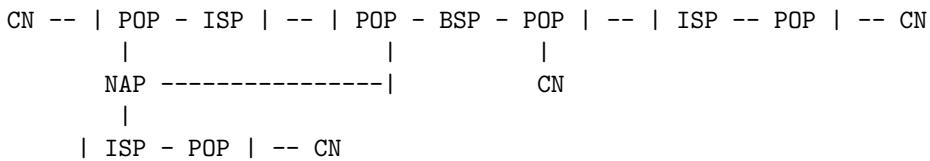
La rete mondiale è composta da tanti reti degli ISP indipendenti che si accordano per collegarle insieme.

**Eterogeneità** Una classifica di reti è in base alla dimensione:

- LAN: Local Area Network. Impiegate in aree limitate
- MAN: Metropolitan Area Network. Interconnettono LAN e hanno estensioni di decine di chilometri
- WAN: Wide Area Network. Ampiezza a piacere, anche globale.

Di solito le LAN sono delle CN: reti terminali che permettono l'accesso ai consumatori. Le CN si collegano attraverso un POP (Point of Presence) alla rete dell'ISP.

Gli ISP si connettono agli altri grazie ad altri ISP che offrono collegamenti a lunga distanza (Backbone Service Provider) tramite un POP. Gli ISP si collegano tra di loro anche tramite le NAP (Neuralt Access Point). I NAP prendono il nome di Internet Exchange Point.



### Servizi di comunicazione e Protocolli

- Infrastruttura di comunicazione: Consente la creazione di applicazioni distribuite
- Servizio di comunicazione: Le modalità con cui sono trasferite le informazioni con delle determinate regole
- Protocollo di comunicazione: Regole con le quali l'informazione viene costruita e trasferita

**Servizio di comunicazione** E' un servizio di trasporto delle informazioni tra processi. Il servizio prende una parte di memoria e la trasferisce in un'altra parte di memoria nel processo remoto (lato server). Le regole con cui si scrive e si leggono le informazioni fanno parte del servizio di comunicazione. I tipi di servizi sono diversi adatti a diversi tipi di messaggi con diversi gradi di affidabilità.

I meccanismi vengono chiamati Socket.

**Protocolli di comunicazione** I protocolli permettono di impacchettare l'informazione in modo che il server capisca la nostra richiesta e la esegua correttamente. Esempio:

- Il client "saluta" il server, server risponde facendo capire che è pronto
- Il client richiede un risorsa, il server gliela invia

### Esempio: Posta elettronica

- Quando ci colleghiamo al server si presenta
- Il client risponde indicando il suo nome
- Il server riconosce il client ed è pronto per gestire le email
- Il client specifica mittente e mail
- Il server risponde indicando che è tutto in regola
- Il client indica il ricevitore
- Il server risponde indicando che è tutto in regola
- Il client chiede di poter inserire il messaggio
- Il server concede il permesso specificando il destinatario
- Il client inserisce il messaggio
- Il server accetta il messaggio e lo spedisce
- Il client chiude la comunicazione

```
S: 220 hamburger.edu
C: HELO cerpes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Bla bla bla
C: Bla bla Bla
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Nota: assomiglia molto al linguaggio umano. I protocolli sono convenzioni.

## Modelli

- Modello client/server: Il client fanno richieste, i server rispondono I client chiedono un servizio e il server lo fornisce.
  - E' una comunicazione asimmetrica
- P2P: Tutti comunicano senza distinzione di ruoli.
  - I tipi di messaggi sono di diverso tipi

## Come funziona la rete

### Commutazione di circuito

Funzionamento della vecchie rete telefonica: viene allocata una risorsa che connette fisicamente i due terminali per l'intera comunicazione.

Come funziona:

- Tramite una richiesta il client chiede alla rete di comunicare con un altro nodo
- Viene creato un canale di comunicazione con l'altro terminale e viene aperto
- Alla fine della chiamata viene rilasciato il circuito per uso da altre telefonate

Le risorse di rete sono suddivise in “pezzi”. Ciascun pezzo viene allocato a vari collegamenti. Le risorse rimangono inattive se non utilizzate.

La banda è quindi divisa in pezzi: divisione di frequenza o di tempo.

**Nodo: Comutatore di circuito** Il commutatore connette un canale in ingresso con uno in uscita.

### Commutazione di pacchetto

I collegamenti non sono suddivisi come nella commutazione di circuito, ma bensì si forma un rete di diversi router collegati tra loro.

Ciascun router ha al suo interno una tabella di instradamento che gli permette di scegliere il prossimo nodo in base alla destinazione dell'informazione.

**I pacchetti** L'informazione viene divisa in pacchetti. Il pacchetto viaggia in rete in modo autonomo (simile al servizio postale). Il pacchetto è identificato da un header che specifica la sua destinazione (simile alla busta di una lettera).

Pacchetto:

| header | Pacchetto ... |

Tutti i pacchetti condividono le risorse di rete e utilizza il canale trasmissivo solo per il tempo necessario alla trasmissione.

**I nodi di rete: packet switch/router** I pacchetti vengono smistati in 3 passaggi

- I pacchetti arrivano in modo asincrono da diversi ingressi. Man mano che arrivano vengono memorizzati in una coda di entrata dal router per essere processati.
- Il router guarda la tabella e decide dove trasferire il pacchetto.
- In uscita possono esserci dei conflitti temporali tra pacchetti: due pacchetti si avviano verso l'uscita contemporaneamente. Si forma, quindi, una coda di uscita per ogni canale.

Il router deve prima ricevere tutto il pacchetto prima di poter elaborarlo e spedirlo in uscita (store and forward).

L'ordine con cui i pacchetti vengono inviati rispecchia quello di arrivo (se viene usato un meccanismo FIFO e un'elaborazione semplice). Ciò viene chiamato multiplazione statistica.

## Confronto tra pacchetti e circuiti

Esempio:

- Collegamento di 1MB/s di N utenti. Ciascun utente genera 100 KB/s quando è attivo (attivo 10% del tempo).
  - Commutazione a circuito: possono essere serviti massimo 10 utenti ( $1 \text{ MB/s} / 100 \text{ KB/s} = 10$ )
  - Commutazione di pacchetto: Con 35 utenti, la probabilità di averne più di 10 è inferiore allo 0,0004.
- Collegamento di 2048 MB/s di 25 utenti. Ciascun utente chiede risorse di 50 KB ogni 62.5s in media.
  - Commutazione a circuito: 1 canale a 64 KB/s per utente, ritardo di trasferimento pagina di 6.25s ( $400 \text{ KB} / 64 \text{ KB/s} = 6,25\text{s}$ )
  - Commutazione a pacchetto: Ritardo di trasferimento medio di 0.22s (teoria delle code)

La commutazione di circuito permette:

- di servire molti più utenti con pochi errori, ma dobbiamo accettare la probabilità che qualcosa potrebbe andare storto (troppi pacchetti che non vengono smistati in tempo).
- di servire gli utenti più in fretta.

Internet è costruito sulla commutazione a pacchetti. Ciò, però, causa il problema delle code: ritardo e perdita di pacchetti. Esistono dei protocolli per gestire i pacchetti e la perdita dei suddetti.

### Altri tipi di commutazione di pacchetto

**Datagram** Utilizzata da internet. In ciascun pacchetto viene indicata solo l'indirizzo di destinazione. Ciascun pacchetto viene gestito indipendentemente dagli altri, anche se fanno parte della stessa informazione.

Il meccanismo datagram è stato il primo ed è il più semplice e più utilizzato, specialmente nella rete periferica.

**Circuito virtuale** I nodi identificano i pacchetti di un flusso informativo sulla base di un identificativo di circuito virtuale (CVI o label). Nella tabella di instradamento ci sono solo i label dei vari circuiti virtuali.

I circuiti virtuali vanno creati in fase di setup. I pacchetti seguono, quindi, lo stesso percorso in quanto identificati dallo stesso label.

Il circuito virtuale permette di ridurre notevolmente la dimensione della tabella di instradamento del router a patto che il numero di circuiti non sia troppo elevato.

Nel cuore della rete viene usato questo approccio in quanto più efficiente quando le tabelle di instradamento diventano troppo lunghe.

## Velocità di trasmissione (rate, R)

La velocità con il quale l'informazione viene trasmessa sulla linea. Si misura in bit/s (bps). Esistono multipli (basati sulle potenze di 10): kbps, Mbps...

Sono usati anche i byte/s (Bps = 8bps). Esistono anche di questi i multipli (kBps, MBps).

Noi useremo la variante basta sui bit.

### Cosa significa

La velocità di trasmissione di un bit è uguale all'inverso del suo tempo di trasmissione (ricavabile dalla semplice formula della velocità).

## Tempo di trasmissione (T)

Il tempo tra la trasmissione del primo e quella dell'ultimo bit. Legata alla velocità di trasmissione:  $T = L/R$ , dove L è il numero di bit.

## Ritardo di propagazione (latency)

Il tempo  $\tau$  affinché un impulso trasmesso da un trasmettitore raggiunga un altro. Essa dipende dalla distanza (D) e dalla velocità di propagazione v (prossima a quella della luce):  $\tau = D/v$ .

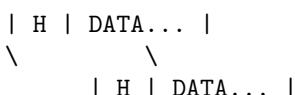
## Tempo di attraversamento del canale

Tempo fra la trasmissione del primo bit e la ricezione dell'ultimo:  $T_{tot} = T + \tau$

## Store and forward e cut-through

Nello store-and-forward, il pacchetto deve essere prima ricevuto nella sua interezza ( $T_{tot} = T + \tau$ ).

La tecnica cut-through viene usata per reti più piccole e permette di ridurre i tempi salvando solo l'header del pacchetto, processare quello e poi fare un passthrough dell'area dati:



L'architettura cut-through presenta problemi se la velocità del canale in uscita è più lenta di quello in entrata o viceversa.

## Architettura semplificata di un nodo

Architettura general-purpose di un nodo:

- CPU: elabora le informazioni
- Memoria
- Bus: permette lo scambio di informazioni tra CPU, periferiche e memoria
- NIC (Network interface card): sono porte d'ingresso e/o di uscita.

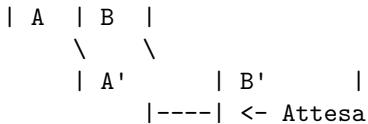
Esistono architetture specifiche con hardware dedicato. Queste architetture sono basate su delle tabelle (match table) che ci permettono di effettuare in hardware il processo. Una switch matrix permette di collegare l'input con la coda di output.

## Il tempo di elaborazione

Tra l'istante di ricezione e quello di invio avviene il processo dell'informazione (lookup e instradamento). Nei sistemi fatti bene, questo tempo è approssimabile a 0. Esistono però, casi in cui questo non è trascurabile. Noi lo assumeremo sempre trascurabile.

## Ritardo di accodamento

Se la linea di uscita è occupata, allora occorre aspettare in coda.



Nel calcolare l'intervallo complessivo di trasmissione bisogna contare il fatto che la ricezione di un pacchetto possa avvenire in contemporanea alla trasmissione di quello precedente. Fai attenzione però: interfacce indipendenti hanno code distinte e lavorano in parallelo.

**Multiplazione statistica e packet-loss** Il ritardo di accodamento dipende dalla multiplazione statistica dovuta all'arrivo asincrono dei pacchetti alle code di uscita. Tutto ciò possono essere caratterizzato da grandezze statistiche.

Quando il rate di arrivo è maggiore al rate di uscita, la coda di uscita verrà saturata e i pacchetti che arrivano dopo la saturazione vengono persi. I pacchetti persi possono essere ritrasmessi a seconda del livello/protocollo che gestisce l'evento di perdita.

Il ritardo di accodamento medio è basato sui risultati della teoria delle code. Dati:

- R la velocità del link
- L la lunghezza del pacchetto
- $\lambda$  la frequenza di arrivo dei pacchetti
- $\mu$  la frequenza di trasmissione dei pacchetti pari a  $\mu = R/L$

L'intensità del traffico è:

$$\frac{L\lambda}{R}$$

Se:

- l'intensità di traffico tende a 0, il ritardo è piccolo
- se l'intensità di traffico tende a 1, il ritardo tende a  $\infty$

Si può dimostrare che in alcune condizioni (code infinite) si ha che il ritardo medio di accodamento è:

$$T_a = \frac{1}{\mu - \lambda} - \frac{1}{\mu}$$

## **Il servizio di comunicazione**

Dati due entità remote, quando parliamo di servizio di comunicazione intendiamo il fornitore del servizio di trasporto dell'informazione.

Il colloquio tra due entità non corrisponde con il servizio di comunicazione: il colloquio è il contenuto, il servizio di comunicazione è ciò che sposta il contenuto.

IL servizio di comunicazione gestisce lo scambio di informazione fra due entità. E' in generale un servizio di trasporto di unità informative.

### **Regole di accesso (primitive di servizio)**

Il servizio di comunicazione può essere descritto mediante delle "chiamate a servizio" dette primitive di servizio. Esse servono a descrivere, a richiederlo e a ricevere informazioni su di esso.

Le primitive sono caratterizzate da parametri tra cui:

- informazione da trasferire
- indicazione del destinatario
- caratteristiche del servizio richiesto
- ...

Le primitive di servizio dell'interfaccia socket sono discusse a laboratorio.

### **Caratteristiche**

#### **Modalità a connessione**

- Instaurazione della connessione
- Trasferimento
- Rilascio della connessione

Richiede una fase preliminare (instaurazione della connessione).

#### **Modalità senza connessione**

- Si trasferisce direttamente l'informazione

## **Protocollo di comunicazione**

Un protocollo è l'insieme delle regole che gestiscono il colloquio tra entità:

- formato dei messaggi
- informazioni di servizio
- algoritmi di trasferimento
- ...

Quando due entità colloquiano tra di loro sono entità di pari livello.

### **Cosa viene scambiato**

L'oggetto dello scambio di informazioni sono le PDU (Packet Data Units). Le PDU possono contenere sia tutte le informazioni di servizio (header) e i dati veri e propri (data).

I dati di solito non sono legati alle regole del protocollo. I dati quindi vengono da un livello superiore.

## **Livelli di comunicazione**

Due entità che colloquiano tra di loro possono offrire questo servizio di comunicazione a entità terze. Queste entità terze sono dette di livello superiore.

Proprio perché i dati nel PDU possono essere forniti da un'entità di livello superiore essi sono detti che provengono da un livello superiore.

Le entità di un livello, quindi, comunicano con le loro pari per offrire un servizio di comunicazione alle entità di livello superiore. Nel fare ciò usano il servizio di comunicazione offerto da altre entità di livello inferiore. Ciò quindi permette di creare una gerarchia di livelli tra le vari entità.

Il servizio di comunicazione offerto dal livello superiore è più ricco e complesso grazie alle funzioni implementate dal livello inferiore.

I sistemi di comunicazione, quindi, vengono articolati a livelli: dal primo che offre solo il trasporto di bit ad un livello dove sono definiti servizi complessi. Internet è diviso principalmente in 5 livelli:

- L'applicazione crea un PDU di livello 5 che viene passato al servizio di comunicazione (livello 4);
- Il livello 4 vede questa PDU come un'unica entità alla quale aggiunge il suo header e lo manda al livello 3;
- Il livello 3 incapsula con il suo header i dati ricevuti;
- Il livello 2 incapsula con il suo header i dati ricevuti;
- Il livello 1 incapsula con il suo header i dati ricevuti e lo spedisce lungo la linea
- L'informazione arriva al primo livello del ricevente. Usa le informazioni del header di primo livello e le scarta passando i dati al livello 2;
- Il livello 2 usa le informazioni del header e le scarta passando i dati al livello 3;
- Il livello 3 usa le informazioni del header e le scarta passando i dati al livello 4;
- Il livello 4 usa le informazioni del header e le scarta passando i dati al livello 5;
- L'informazione è arrivata al livello 5 del ricevente e viene processata.

Si può vedere che ogni livello gestisce lo scambio di informazioni con un suo pari.

### Perché i livelli

- Facile identificazione dei servizi
- Facile gestione ed update: trasparenza verso i livelli superiori

La divisione in livelli può essere pericolosa?

### Modello a livelli di internet (TCP/IP)

Livello	Nome	PDU
5	Applicativo	Messaggi
4	Trasporto	Segmenti
3	Rete	Pacchetti
2	Link	Trame (frames)
1	Fisico	Bit

### Funzioni dei vari livelli

Molteplici sono le funzioni che possono essere svolte da un livello.

**Multiplazione** Permette a più livelli superiori di condividere lo stesso servizio di comunicazione. Nel momento in cui io trasferisco più informazioni di utilizzatori diversi, alla ricezione devo essere in grado di risepararle dalla parte del ricevitore.

Esempio: 1 singolo indirizzo di rete, la multiplazione è data dal numero di porta (80, 25 ecc...)

**Controllo d'errore** E' possibile garantire l'affidabilità della comunicazione anche in presenza di errori sul canale.

Esempio: il TCP, quando invia un pacchetto, mette nell'header una sezione per il controllo dell'errore. Il ricevitore userà questa sezione per verificare il pacchetto. Se tutto è andato bene, il ricevitore manda un segnale di tutto ok (ACK). Se il segnale non arriva al trasmittente, viene ritrasmesso il pacchetto.

**Instradamento (routing)** Il livello superiore passa al livello con instradamento il parametro di indirizzo. L'indirizzo viene aggiunto nell'header in modo che possa essere instradato.

I pacchetti possono anche essere in entrata: l'entità instradante legge l'indirizzo e lo confronta con una tabella di instradamento e decide se rimandarlo attraverso una porta d'uscita (forward).

La tabella di routing possono essere scritte:

- a mano (Human defined Networking). Le rotte scritte a mano vengono dette rotte statiche;
- protocolli di instradamento distribuiti. I nodi della rete scambiano tra di loro dei messaggi di servizio che permettono ai router di “imparare” da sé le rotte;
- software defined networking. Viene automatizzato con software il HDN;

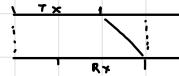
Nota: il pacchetto in arrivo non sale mai più in alto nei livelli salvo che non sia giunto a destinazione.

Nota: i nodi di rete spesso non salgono più in alto del livello di instradamento. Un esempio di ciò sono i Router IP. La funzione di rete, però, non per forza è implementata al livello 3:

- instradamento al livello 2: LAN switch;
- instradamento al livello 5: Proxy.

## ESERCIZI

Un sistema binomistico della velocità di 100 Kbps prende una lunghezza di 500 Km. Si calcoli il tempo che intercorre fra la tr. del primo bit e l'arrivo dell'ultimo di un pacch. di 200 b, assumendo ritardo di prop. di 5 ps/Km.



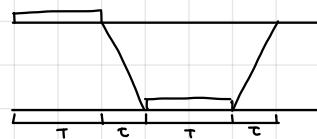
$$T = \frac{l}{v} = \frac{500 \text{ Km}}{2 \cdot 10^8 \text{ Km/s}} = 2,5 \cdot 10^{-3} \text{ s} = 2,5 \text{ ms} = 5 \mu\text{s}/\text{Km} \cdot 500 \text{ Km}$$

$$V = \frac{2}{3} C = \frac{2}{3} \cdot 3 \cdot 10^8 = 2 \cdot 10^8 \text{ m/s} = 2 \cdot 10^5 \text{ Km/s}$$

$$T = \frac{L}{R} = \frac{2000}{10} = 20 \text{ ms}$$

$$T_{\text{tot}} = 20 \text{ ms} + 2,5 \text{ ms} = 22,5 \text{ ms}$$

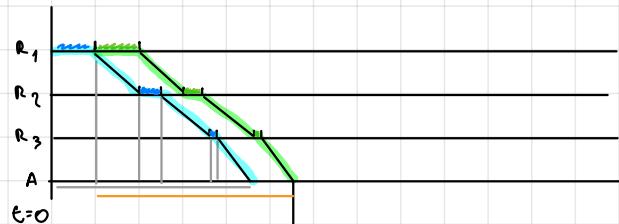
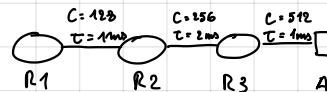
Un pacchetto di 1000b viene inviato dal nudo a velocità di 100 Kbps su 100 Km. Il pacchetto viene ricevuto e ripetuto. In entrambi i percorsi la velocità è uguale. Calcolare il tempo tot, assumendo  $v = 2 \cdot 10^5 \text{ Km/s}$ . Si ripeta con rete 10 Gbps.



$$T_{\text{tot}} = 2(T + \tau) = 2 \left( \frac{10 \text{ Kb}}{100 \text{ Kbps}} + \frac{100 \text{ Km}}{2 \cdot 10^5 \text{ Km/s}} \right) = 2(0,1 \text{ s} + 0,5 \cdot 10^{-3} \text{ s}) = 2(100 \text{ ms} + 0,5 \text{ ms}) = 201 \text{ ms}$$

$$T_{\text{tot}}^1 = 2(T + \tau) = 2 \left( \frac{1 \cdot 10^{-3} \text{ Gb}}{10 \text{ Gbps}} + 0,5 \cdot 10^{-3} \text{ s} \right) = 2(0,001 \text{ ms} + 0,5 \text{ ms}) = 1,002 \text{ ms}$$

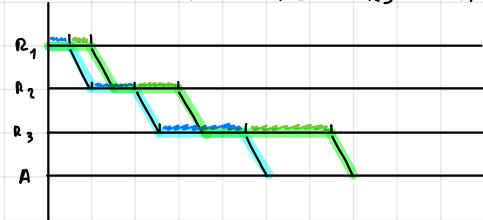
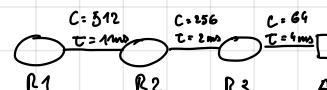
Si consideri la rete. In  $t=0$  la coda d'uscita di  $R_1$  ha 2 pacchetti diretti ad  $A$ . Assumendo  $L = 512 \text{ b}$ , si indichi per ciascun pacchetto l'intervallo in cui viene ricevuto a destinazione.



$$T_{\text{tot}}^1 = \frac{L}{C_1} + \tau_1 + \frac{L}{C_2} + \tau_2 + \frac{L}{C_3} + \tau_3 = \frac{512}{128 \cdot 10^3} + 1 \text{ ms} + \frac{512}{256 \cdot 10^3} + 2 \text{ ms} + \frac{512}{512 \cdot 10^3} + 1 \text{ ms} = 4 + 1 + 2 + 2 + 1 + 1 = 11 \text{ ms}$$

$$T_{\text{tot}}^2 = \tau_1 + T_{\text{tot}}^1 = 4 + 11 = 15 \text{ ms}$$

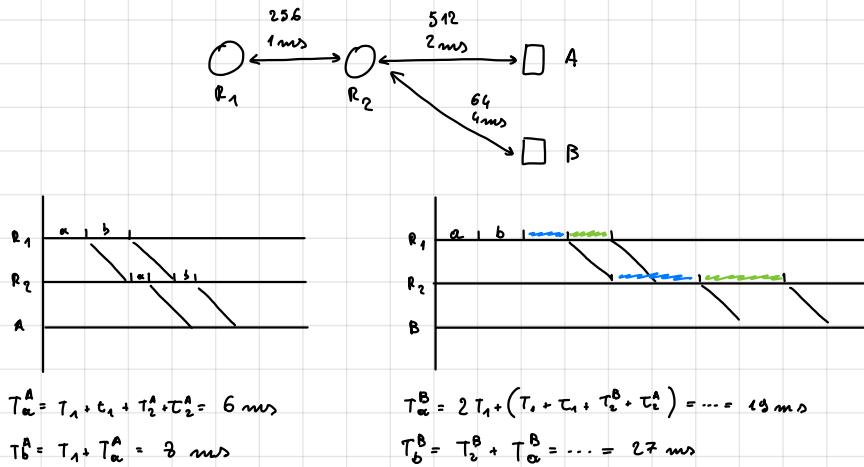
Si consideri la rete. In  $t=0$  la coda d'uscita di  $R_1$  ha 2 pacchetti diretti ad  $A$ . Assumendo  $L = 512 \text{ b}$ , si indichi per ciascun pacchetto l'intervallo in cui viene ricevuto a destinazione.



$$T_{\text{tot}}^1 = \tau_1 + \tau_2 + \tau_3 + \tau_4 = 18 \text{ ms}$$

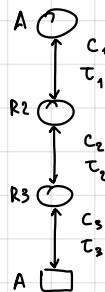
$$T_{\text{tot}}^2 = T_{\text{tot}}^1 + \tau_3 = \dots = 26 \text{ ms}$$

Si consideri la rete. Al tempo  $t=0$  la coda di  $R_1$  ha 4 pacchetti diretti a A, A, B, B. Assumendo  $L=512 \text{ Kbps}$ , si indichi per ciascun pacchetto l'intervallo in cui viene ricevuto.



Si calcoli al forma parametrica il tempo necessario a trasmettere da A a B:

$$T_{\text{TOT}} = \frac{h+D}{C_1} + \tau_1 + \frac{h+D}{C_2} + \tau_2 + \frac{h+D}{C_3} + \tau_3$$



Si assume di dividere il pacchetto in due frammenti e che  $\tau_2 = 2\tau_1 \Rightarrow \frac{L}{C_2} = 2 \frac{L}{C_1}$ :

$$d = \frac{D}{2}$$

$$T = \frac{h+d}{C_1} + \tau_1 + 2 \frac{h+d}{C_2} + \tau_2 + \frac{h+d}{C_3} + \tau_3$$

Qual è il numero di frammenti che minimizza il ritardo?

$$T = \frac{h+\frac{Dn}{2}}{C_1} + \tau_1 + n \frac{h+\frac{Dn}{2}}{C_2} + \tau_2 + \frac{h+\frac{Dn}{2}}{C_3} + \tau_3 = \left( \frac{h}{C_1} + \tau_1 + \frac{D}{C_2} + \tau_2 + \frac{h}{C_3} + \tau_3 \right) + \frac{D}{2C_1} + \frac{nh}{2C_2} + \frac{D}{2C_3}$$

$$T' = \frac{h}{C_2} - \frac{D}{n^2 C_1} - \frac{D}{n^2 C_3} \dots n^* = \sqrt{\frac{C_2}{h} \left( \frac{D}{C_1} + \frac{D}{C_3} \right)}$$

Nota

Se l'header si può trascurare, si può frammentare fino a trasmettere 1 bit e trascurare il tempo di trasmissione:

$$T = \frac{D}{C_2} + \tau_1 + \tau_2 + \tau_3$$

Il bottleneck del link più lento è evidente. Questa è una buona approssimazione di un file transfer dove le dimensioni del file rendono quelle dei pacchetti trascurabili.

## 2. LIVELLO FISICO

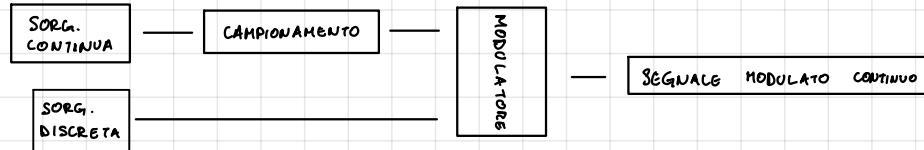
### 2.1 SEGNALE E FREQUENZE

Esistono 2 tipi di segnali:

1. Segnali logici: reverenti naturalmente numeriche
2. Segnali fisici: associati a grandezze fisiche

Le definizioni di segnali analogici e digitali è troppo riduttiva e non verrà utilizzata.

Lo schema di trasmissione è



Il segnale può essere rappresentato in due modi:

- $s(t)$ : legato al tempo (dominio del tempo)
- $S(f)$ : legato alle frequenze (dominio delle frequenze)

Si passa tra le due con l'analisi di Fourier: essa ci permette di studiare scomponendolo in sinusoidi delle armoniche.

Studiamo l'onda quadra. È possibile scomporla in una serie discreta di sinusoidi della serie di Fourier. L'onda quadra possiede armoniche discrete.

Un segnale  $s(t)$  è equivalente alla somma di tutte le sue armoniche. Esso è rappresentato nel dominio delle frequenze dalle frequenze delle sue armoniche. Le frequenze delle armoniche sono lo spettro di  $s(t)$ .

L'analisi di Fourier è utilizzabile a tutti i tipi di segnali. Se  $s(t)$  non è periodica,  $S(f)$  è una funzione continua. Se, invece, è periodica,  $S(f)$  sarà discinta.

La banda del segnale è l'intervallo di frequenze o sinusoidi in cui il segnale non è nullo.

Un segnale si dice a banda larga se variano molto nel tempo e a banda stretta se variano poco.

### 2.2 CONVERSIONE ANALOGICO-DIGITALE

Per trasmettere un segnale continuo bisogna convertirlo in digitale perché al mondo dei computer è digitale.

Per ottenere tale trasformazione un segnale viene campionato ossia vengono presi dei "campioni" a intervalli regolari. Passare dal segnale campionato all'originale è possibile solo se la banda è finita e se la freq. di campionamento è almeno  $f \geq 2B$  (TEOREMA DI NYQUIST). Campioni più frequenti di  $2B$  è ecceso. La banda rappresenta il contenuto informativo.

Per ricongenerare dal campionato viene passato in un filtro low-pass con frequenza di taglio  $2B$ .

Le ampiezze dei campioni sono ancora continue. È necessario, quindi, quantizzare le ampiezze: si divide il range in una rete di gradini.

Nel fare ciò compriremo un errore  $E$ . Il numero di livelli utilizzabili dipende dal numero di bit che vogliamo utilizzare nella rappresentazione:  $N_b = 2^b$

La quantizzazione è un'operazione irreversibile. È importante, quindi, scegliere adeguatamente il numero di livelli.

Il flusso binario non è altro che:  $F = f_c \cdot b$  → numero bit  
freq. camp.

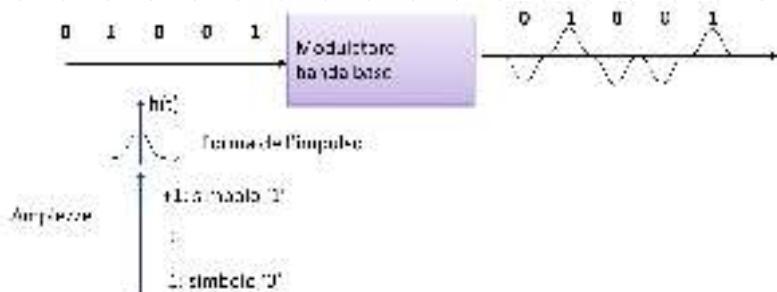
• •

### 2.3 MODULAZIONE E TRASMISSIONE

La trasmissione di un segnale digitale richiede di creare un segnale adatto a essere trasportato da un mezzo trasmissivo. La sequenza digitale viene usata per MODULARE una grandezza fisica di una dato segnale.

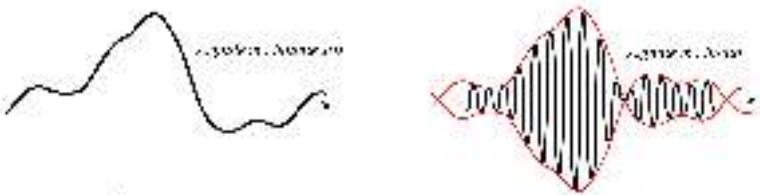
Esistono due tipi di modulazione:

- 1) **BANDA BASE:** i segnali usati hanno uno spettro continuo rispetto all'origine (PAM)



- 2) **BANDA TRASLATA:** i segnali hanno uno spettro basato su intervalli non contigui all'origine

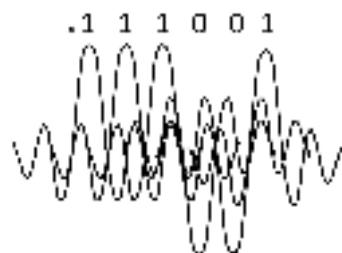
Per un segnale modulante si usa un'onda elettromagnetica sinusoidale a frequenza  $f_p$  (detta portante) per trasportare il segnale.



Se per la modulazione PAM usiamo la funzione  $P(t)$ , otteniamo un'ultra risultato:

$$P(t) = A \frac{\sin(\pi \frac{t}{T})}{\frac{\pi t}{T}} \rightarrow \text{Sintesi di Fourier}$$

Usciamo questo funzione perché è molto compatta nel dominio delle frequenze: un rettangolo con lato  $T$  rimbalzo rispetto a  $y$ . Un segnale modulato avrà forma:



La banda occupata dal segnale sarà, quindi, pari al più bit rate:  $R \approx 2B$ . L'efficienza spettrale  $\eta$  non è altro che il rapporto tra bit rate e banda. Per il segnale che consideriamo abbiamo:

$$\eta = \frac{R}{B} = 2 \text{ bps/Hz}$$

Se usiamo la banda traslata,  $\eta$  sarà:  $\eta = \frac{T}{T} = 1 \text{ bps/Hz}$

La portante dei segnali è un'onda elettromagnetica a opportuna frequenza. Essa si propaga nell'atmosfera o guidata in varie... Le onde elettromagnetiche sono classificate in base alla frequenza.

Le trasmissioni radio avvengono su frequenze regolate a livello internazionale per evitare interferenze. Im più il mezzo trasmissivo ci può vincolare nell'uso di alcune date frequenze. Per questo motivo, è prevalente l'uso della banda traslata in modo da tramutare il segnale in banda portante senza alterarlo.

Esistono diversi modi di modulare lo portante:

- ASK: si modifica l'ampiezza
- FSK: si modifica la frequenza tra più frequenze note
- PSK: si modifica la fase
- QAM: si modifica sia ampiezza che fase.

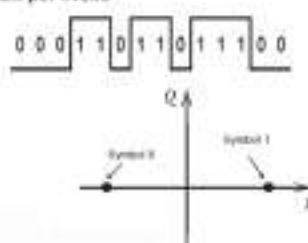


Per migliorare l'efficienza, si può modulare su più livelli:

- il flusso è diviso in gruppi di  $\log_2 N$  ( $N$  livelli) bit
- Per ogni livello di ampiezza (chiamato simbolo) corrispondono  $n = \log_2 N$  bit

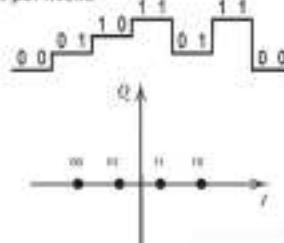
ASK o PAM binario

2 livelli di ampiezza  
1 bit per livello

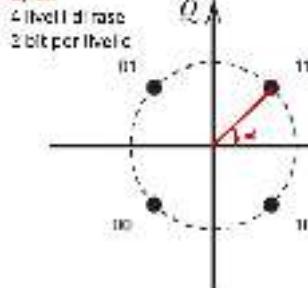


4-ASK o 4PAM

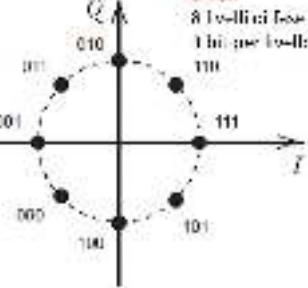
4 livelli di ampiezza  
2 bit per livello



QPSK

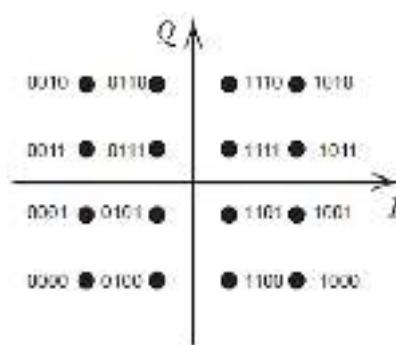


8-PSK



16QAM

16 livelli di fase e ampiezza  
(diametri anche SIMBOLI)  
4 bit per simbolo



L'efficienza spettrale aumenta di circa  $n$  volte rispetto a quella binaria. Definendo  $R_s$  il rate in simboli al secondo (band) e con  $R_b$  il rate in bps si ha:

$$R_b = n R_s$$

$$\eta_s = n \eta_b$$

Queste equazioni valgono sia per banda base che per banda traslata e passante.

Si definisce canale trasmissivo l'insieme di:

- trasmittore
- rumore: modifica la trasmissione  $s_t(f)$  con  $H(f)$ :  $s_r(f) = s_t(f) \cdot H(f)$
- ricevitore.

Esso è caratterizzato da una velocità di trasmissione (capacità/rate)  $R$  e da un ritardo di propagazione  $\tau$ .

Il rumore trasmissivo introduce principalmente:

- attenuazione della potenza in funzione della distanza e della frequenza:

$$A = \frac{P_{out}}{P_{in}} \quad A_{dB} = 10 \log_{10} \left( \frac{P_{out}}{P_{in}} \right)$$

- dispersione, ovvero un ritardo differente per diverse componenti in freq. del segnale

Questa dispersione ha una frequenza in cui lavora meglio i filtri **BANDA PASSANTE DEL CANALE**. La banda passante si riduce all'aumentare della distanza. Affinché il segnale ricevuto contenga, la banda di trasmissione deve essere più ampia di quella ricevuta.

In ricezione si può sommare un rumore casuale causato da: temperatura, interferenza ecc.... Le alterazioni causate dal rumore è più ampia del livello commettibile un errore. Il più livello di rumore la probabilità di errore dipende dalla distanza tra i livelli della energia d'impulso. Tale energia dipende dalla potenza del segnale ricevuto. L'attenuazione riduce l'energia tanto più aumenta la distanza percorsa dal segnale. Si può ridurre la probabilità di errore attraverso dei codici di parità e poi ritrasmettere il pacchetto.

Esiste un limite massimo alla velocità di un canale (Teorema di Shannon):

$$C = B \log_2 \left( 1 + \frac{S}{N} \right)$$

con: C capacità

B banda

S potenza segnale

N potenza rumore

Il Teorema di Shannon costituisce il limite inviolabile che non ci permette di aumentare o piccare il rate di un link: ad esempio l'aumento di livelli comporta la riduzione dell'energia di impulso e l'aumento della probabilità di errore; per superare il rumore bisogna aumentare la potenza.

### 3 LIVELLO APPLICATIVO

Per comunicare tra processi remoti bisogna avere un metodo di indirizzamento e un protocollo di scambio di dati. Lo scambio avviene grazie all'utensile di servizio **Access Point** (servizio di comunicazione).

Per identificare l'applicazione usiamo:

- 1) un indirizzo dell'host (IP)
- 2) un numero di porta che identifica il SAP (porta)

Un socket è una porta di comunicazione identificata da un indirizzo e una porta.

Si possono usare diversi servizi offerti dal S.O.:

- 1) TCP: sistema affidabile in quanto garantisce la ricezione corretta e ordinata dei pacchetti. Non garantisce velocità
- 2) UDP: sistema non affidabile in quanto non garantisce nulla se non velocità.

Le applicazioni possono essere strutturate in modo:

- client-server
- P2P: le applicazioni sono sia client che server
- client

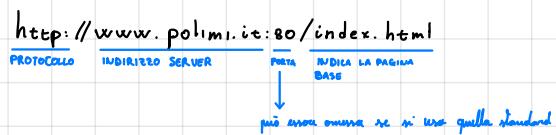
Nell'architettura client-server, il server è sempre disponibile (IP fisso, attivo 24/7). Il client, invece, non ha vincoli.

Nella P2P pura non c'è nessun server fisso e i domini comunicano direttamente in modo intermittente. Essi possono anche comunicare indirizzando. Una parte fisso di solito c'è, ma riguarda il repository di informazioni su come comunicare (es. BitTorrent e i torrent).

• •

### 3.1 HTTP (RFC 1995, 2616)

Le pagine web sono fatte da oggetti. Un oggetto è un file HTML, PHP ecc.. Tipicamente si ha un oggetto base (HTML / PHP) che chiama gli altri oggetti. Tutti gli oggetti hanno un URL. Esempio:



Lo comunicazione HTTP è client-server.

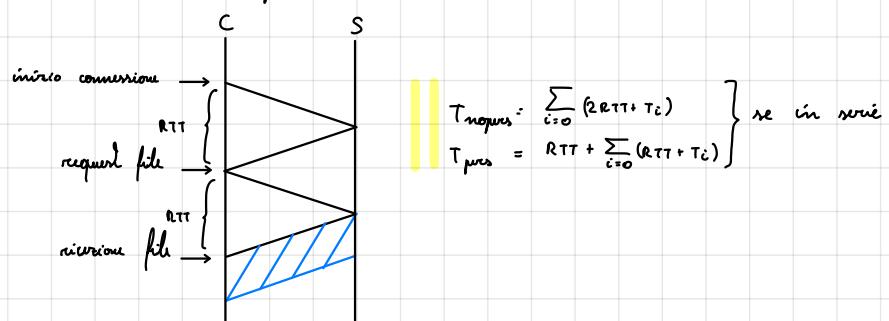
- il client fa richieste HTTP di oggetti
- il server fornisce le risorse richieste con risposte HTTP

Non viene mantenuta memoria sul server (STATELESS).

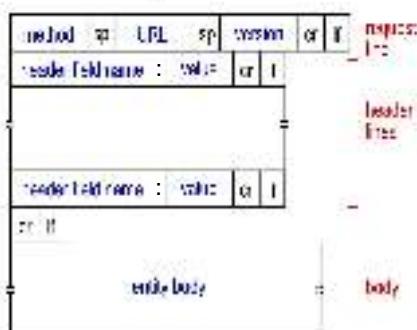
Lo HTTP usa TCP per trasportare le informazioni. Ci sono due modalità di comunicazione:

- non persistente: viene creata una connessione TCP per ogni oggetto
- persistente: viene creata una connessione TCP per l'oggetto base e viene mantenuta quella. Le richieste possono essere:
  - senza pipelining: richieste in serie
  - con pipelining: richieste in parallelo (default in HTTP 1.1)

Nella stima del tempo di trasferimento in HTTP vengono trascurate le richieste in quanto molto piccole. Il tempo per mandare una richiesta e ricevere la risposta si chiama RTT (roundtrip-time)



Le richieste sono codificate in ASCII e suddivise così:

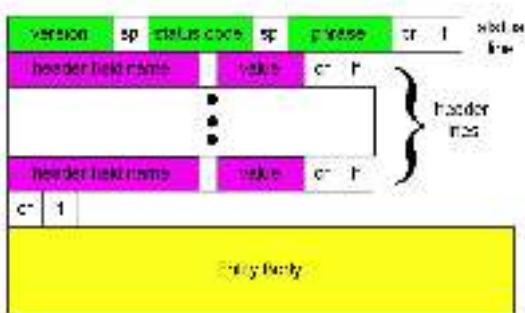


GET	Richiesta di download di un documento. Si specifica il nome del file o il percorso relativo al documento richiesto.
HEAD	Richiesta di download di un documento ma senza riceverne il contenuto. Si utilizza questo metodo quando si vuole verificare se il documento esiste.
POST	Richiesta di upload di un documento. Si specifica il nome del file da inviare.
PUT	Cambia il contenuto di un documento. Si utilizza questo metodo per modificare un documento esistente.
DELETE	Silenziosa cancellazione dello URL.

Header name : Header value

Cache-control	Informazione sulla cache
Accept	Formati accettati
Accept-language	Lingua accettata
Authorization	Mostra i permessi del client
If-modified-since	Invia il doc. solo se modificato
User-agent	Tipo di user agent

Ciò che le risposte sono codificate in ASCII:



Codes:

- 1xx: informazione
- 2xx: successo
- 3xx: redirigenza
- 4xx: errori client (richiesta errata)
- 5xx: errori server

Le richieste GET possono essere condizionate in base a quanto un oggetto è stato modificato. Se la risorsa è stata modificata verrà inviata, senno' risponderà "NOT MODIFIED" (codice 304).

GET ...

If-modified-since::<date>

HTTP/v · 304· Not Modified ...

OGGETTO NON MODIFICATO

GET ...

If-modified-since::<date>

HTTP/v · 200· OK

<DATA>

OGGETTO MODIFICATO

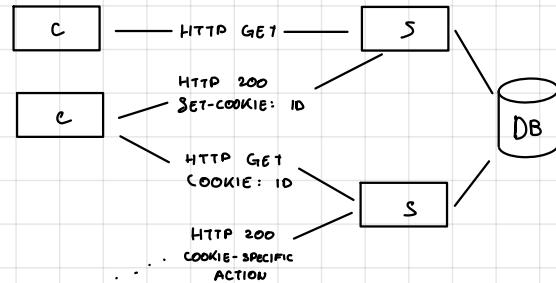
### 3.1 HTTP (RFC 1345, 2616)

...

#### 3.1.1 i COOKIES

I cookies sono uno strumento per mantenere uno stato. I cookie sono header che vengono salvati sull'host. Un database di cookie viene mantenuto dal server.

Quando il client visita per la prima volta il sito esso viene identificato da un id unico.



#### 3.1.2 PROXY HTTP

I proxy sono dei middle-man che hanno compito di rispondere alle richieste senza coinvolgere il server. Il client, quindi, parla con il proxy. Il server vedrà solo richieste da parte del proxy.

Il proxy può anche cacheare delle risorse in modo da rendere più accessibili ad altri utenti del proxy.

I proxy sono degli application gateway, ovvero interratori a livello applicativo.

#### 3.1.2 HTTP 2.0

L'obiettivo è ridurre i loading time dei siti e alcuni problemi del HTTP 1.1:

- lavora in binario (trasporta frame)
- usa il pipelining (comunicazione in parallelo)
- gli header vengono compresi:
  - codifica di Huffman
  - indexing
  - codifica differenziale
- si usa il server push: il server manda risposte senza richiesta
- usa TLS
- offre flow control. (tramite stream)

#### 3.1.3 HTTPS

HTTP non offre garanzie su integrità, confidencialità e autenticazione tra server e client.

Le parti di sicurezza possono essere aggiunte a livello di trasporto da SSL e TLS.

Le connessioni SSL/TLS si dividono in:

- HANDSHAKE: le due parti si identificano e si mettono d'accordo su crittografia e si scambiano le chiavi
- TRANSFER: ogni PDU è cifrata con la crittografia decisa
- CLOSE: viene chiusa la connessione e le chiavi vengono eliminate.

La parte di handshaking è la più critica.

- viene scambiato un certificato generato da una CA. Esistono diverse CA riconosciute valide nei browser delle nostre certificazioni. Un certificato contiene:
  - chiave pubblica dell'entità certificata;
  - informazioni sul server;
  - firma digitale della CA;
- viene generata una chiave simmetrica per la cifratura delle PDV; (PUB → ENCRYPT; PRIV → DECRYPT)
- le chiavi simmetriche vengono inviate su una connessione con cifratura simmetrica (vengono utilizzate la chiave pubblica nel cert.)

### 3.2 POSTA ELETTRONICA (SMTP, POP3, IMAP)

La posta elettronica è composta da client (User agent) e da mail server. Ogni mail server contiene le varie caselle di posta degli utenti e una coda di messaggi da inviare.

I server di posta comunicano in SMTP e si comportano sia da server, quando riceve posta da un User Agent, che da client, quando deve inviare un messaggio ad altri server.

Per leggere la posta vengono usati i protocolli POP3 e IMAP

#### 3.2.1 SMTP

È client-server codificato ASCII. Usa il TCP sulla porta 25. Sia header che corpo sono codificati in ASCII, quindi anche i messaggi devono essere codicati in ASCII.

Il colloquio avviene con comandi tipo HELO, MAIL FROM, RCPT TO, DATA e QUIT. Non vi è nessun tipo di autenticazione nel protocollo. I meccanismi di autenticazione sono stati aggiunti tramite una connessione a un server d'entrata (verifica mittente) e uno d'uscita (verifica ricevitore). La sicurezza è molto leggera.

I dati della mail sono strutturati in HEADER e BODY (ASCII). Alcuni header sono 'Subject:', 'To:' e 'From:'.

Per aggirare la restrizione ASCII viene usata l'estensione MIME che permette di trasformare mail multimediali, con ogni parte identificata da un tipo. I dati binari vengono convertiti in ASCII grazie alla codifica in base64.

#### 3.2.2 ACCESSO MAILBOX

Vengono usati principalmente POP3 e IMAP. Il POP3 è più vecchio e più semplice.

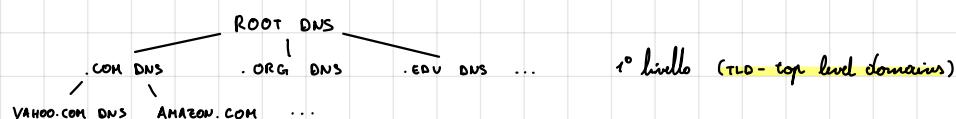
La comunicazione POP3 avviene di solito sulla porta 110 e dunque in:

- autenticazione: il client si identifica coi comandi 'USER' e 'PASS'. Il server risponde +OK o -ERR.
- transazione: il client lavora sulla sua casella con comandi tipo 'list', 'retr', 'dele' e 'quit'.

### 3.3 DNS

Gli indirizzi IP sono numeri. I numeri sono difficili da ricordare. Il DNS è un'applicazione che traduce nomi simbolici in indirizzi IP. Il DNS ha un database distribuito che contiene le varie associazioni e usa una comunicazione UDP per scambiare info. Offre anche altri servizi come ad esempio load distribution.

Il DNS è strutturato in gerarchie:



I vari name servers (NS) sono di due tipi:

- LOCAL: forniti dall'ISP collegati direttamente con l'host.
- AUTHORITATIVE: ns responsabile di un particolare hostnames.  
Ogni host ha configurato l'indirizzo del LNS.

La risoluzione può avvenire in due modi: ITERATIVO e RICORSIVO.

Esempio iterativo:

1. il client dns dell'host comunica con LNS
2. LNS contatta il root ns.
3. il Root ns segnala al LNS il TLD responsabile
4. il LNS contatta il TLD e gli chiede il sito
5. il TLD segnala l'autoritativo ns del sito
6. il NS autoritativo comunica l'IP del sito.

Esempio ricorsivo:

1. il client dns dell'host comunica con LNS
2. LNS contatta il root ns.
3. il Root ns contatta il TLD responsabile
4. il TLD contatta il NS autoritativo.
5. il NS autoritativo risponde l'ip
6. 7. 8. l'informazione torna su

Un NS, una volta risolto l'ip di un dominio su cui non ho autorità può memorizzarla per rendere successivi lookups più veloci. Ogni cache ha un TTL (Time to live) che decide per quanto tempo la cache è valida.

Le informazioni memorizzate nel DNS (resource record - RR) hanno questo formato:

NAME, VALUE, TYPE, TTL

Il tipo può essere:

- 'A': NAME è il nome di un host e VALUE è l'IP.
- 'NS': NAME è un domain e VALUE è il nome di un NS che può risolvere le info di ns.
- 'CNAME': NAME è un alias per un host il cui vero nome è VALUE
- 'MX': NAME è dominio di mail o alias e VALUE è il nome del mail server.

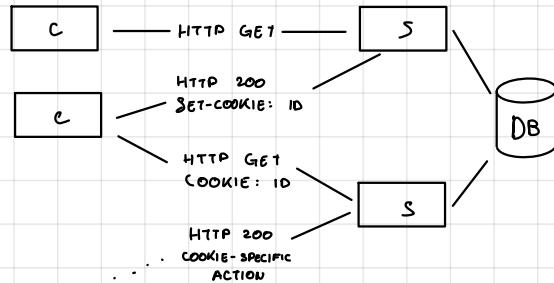
### 3.1 HTTP (RFC 1345, 2616)

...

#### 3.1.1 i COOKIES

I cookies sono uno strumento per mantenere uno stato. I cookie sono header che vengono salvate nell'host. Un database di cookie viene mantenuto dal server.

Quando il client visita per la prima volta il sito esso viene identificato da un id unico.



#### 3.1.2 PROXY HTTP

I proxy sono dei middle-man che hanno compito di rispondere alle richieste senza coinvolgere il server. Il client, quindi, parla con il proxy. Il server vedrà solo richieste da parte del proxy.

Il proxy può anche cacheare delle risorse in modo da rendere più accessibili ad altri utenti del proxy.

I proxy sono degli application gateway, ovvero interratori a livello applicativo.

#### 3.1.2 HTTP 2.0

L'obiettivo è ridurre i loading time dei siti e alcuni problemi del HTTP 1.1:

- lavora in binario (trasporta frame)
- usa il pipelining (comunicazione in parallelo)
- gli header vengono compresi:
  - codifica di Huffman
  - indexing
  - codifica differenziale
- si usa il server push: il server manda risposte senza richiesta
- usa TLS
- offre flow control. (tramite stream)

#### 3.1.3 HTTPS

HTTP non offre garanzie su integrità, confidencialità e autenticazione tra server e client.

Le parti di sicurezza possono essere aggiunte a livello di trasporto da SSL e TLS.

Le connessioni SSL/TLS si dividono in:

- HANDSHAKE: le due parti si identificano e si mettono d'accordo su crittografia e si scambiano le chiavi
- TRANSFER: ogni PDU è cifrata con la crittografia decisa
- CLOSE: viene chiusa la connessione e le chiavi vengono eliminate.

La parte di handshaking è la più critica.

- viene scambiato un certificato generato da una CA. Esistono diverse CA riconosciute valide nei browser delle nostre certificazioni. Un certificato contiene:
  - chiave pubblica dell'entità certificata;
  - informazioni sul server;
  - firma digitale della CA;
- viene generata una chiave simmetrica per la cifratura delle PDV; (PUB → ENCRYPT; PRIV → DECRYPT)
- le chiavi simmetriche vengono inviate su una connessione con cifratura simmetrica (vengono utilizzate la chiave pubblica nel cert.)

### 3.2 POSTA ELETTRONICA (SMTP, POP3, IMAP)

La posta elettronica è composta da client (User agent) e da mail server. Ogni mail server contiene le varie caselle di posta degli utenti e una coda di messaggi da inviare.

I server di posta comunicano in SMTP e si comportano sia da server, quando riceve posta da un User Agent, che da client, quando deve inviare un messaggio ad altri server.

Per leggere la posta vengono usati i protocolli POP3 e IMAP

#### 3.2.1 SMTP

È client-server codificato ASCII. Usa il TCP sulla porta 25. Sia header che corpo sono codificati in ASCII, quindi anche i messaggi devono essere codicati in ASCII.

Il colloquio avviene con comandi tipo HELO, MAIL FROM, RCPT TO, DATA e QUIT. Non vi è nessun tipo di autenticazione nel protocollo. I meccanismi di autenticazione sono stati aggiunti tramite una connessione a un server d'entrata (verifica mittente) e uno d'uscita (verifica ricevitore). La sicurezza è molto leggera.

I dati della mail sono strutturati in HEADER e BODY (ASCII). Alcuni header sono 'Subject:', 'To:' e 'From:'.

Per aggirare la restrizione ASCII viene usata l'estensione MIME che permette di trasformare mail multimediali, con ogni parte identificata da un tipo. I dati binari vengono convertiti in ASCII grazie alla codifica in base64.

#### 3.2.2 ACCESSO MAILBOX

Vengono usati principalmente POP3 e IMAP. Il POP3 è più vecchio e più semplice.

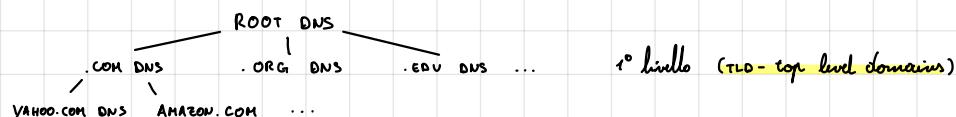
La comunicazione POP3 avviene di solito sulla porta 110 e dunque in:

- autenticazione: il client si identifica coi comandi 'USER' e 'PASS'. Il server risponde +OK o -ERR.
- transazione: il client lavora sulla sua casella con comandi tipo 'list', 'retr', 'dele' e 'quit'.

### 3.3 DNS

Gli indirizzi IP sono numeri. I numeri sono difficili da ricordare. Il DNS è un'applicazione che traduce nomi simbolici in indirizzi IP. Il DNS ha un database distribuito che contiene le varie associazioni e usa una comunicazione UDP per scambiare info. Offre anche altri servizi come ad esempio load distribution.

Il DNS è strutturato in gerarchie:



I vari name servers (NS) sono di due tipi:

- LOCAL: forniti dall'ISP collegati direttamente con l'host.
- AUTHORITATIVE: ns responsabile di un particolare hostnames.  
Ogni host ha configurato l'indirizzo del LNS.

La risoluzione può avvenire in due modi: ITERATIVO e RICORSIVO.

Esempio iterativo:

1. il client dns dell'host comunica con LNS
2. LNS contatta il root ns.
3. il Root ns segnala al LNS il TLD responsabile
4. il LNS contatta il TLD e gli chiede il sito
5. il TLD segnala l'autoritativo ns del sito
6. il NS autoritativo comunica l'IP del sito.

Esempio ricorsivo:

1. il client dns dell'host comunica con LNS
2. LNS contatta il root ns.
3. il Root ns contatta il TLD responsabile
4. il TLD contatta il NS autoritativo.
5. il NS autoritativo risponde l'ip
6. 7. 8. l'informazione torna su

Un NS, una volta risolto l'ip di un dominio su cui non ho autorità può memorizzarla per rendere successivi lookups più veloci. Ogni cache ha un TTL (Time to live) che decide per quanto tempo la cache è valida.

Le informazioni memorizzate nel DNS (resource record - RR) hanno questo formato:

NAME, VALUE, TYPE, TTL

Il tipo può essere:

- 'A': NAME è il nome di un host e VALUE è l'IP.
- 'NS': NAME è un domain e VALUE è il nome di un NS che può risolvere le info di ns.
- 'CNAME': NAME è un alias per un host il cui vero nome è VALUE
- 'MX': NAME è dominio di mail o alias e VALUE è il nome del mail server.

• • •

### 3.4 DNS

I messaggi DNS sono codificati in binario e contiene header, domande e/o risposte. Domande e risposte possono essere mescolate tra loro. Ogni risposta contiene dentro la corrispondente domanda (rende "affidabile" l'UDP).

- header: identificazione coppia client/server
- flag: richiesta/risposta, autorizzazione/altro, iterative/recursiva
- numero di riferimento al numero di campo nello stesso successivo
- question: nome richiesto e tipo (ad es. A o MX)
- answers: resource records compresi tutti in risposta
- authority: canaliere altri record forniti da altri server
- additional: informazioni aggiuntive, ad es. i record con IP ADDR per il MX fornito in risposta

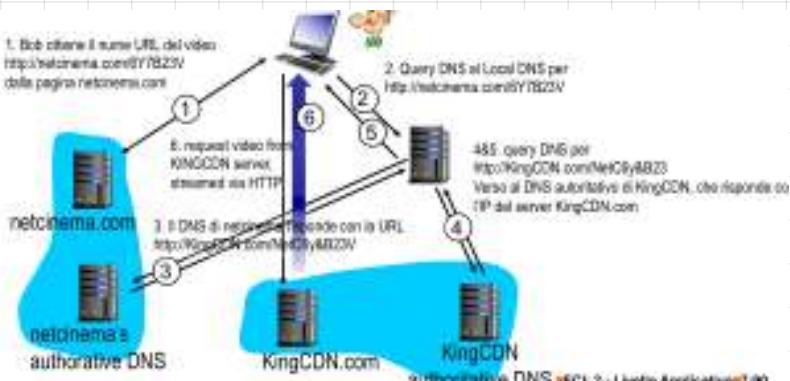


Per registrare un dominio bisogna registrare il dominio presso un DNS registrar. Poi bisogna fornire al DNS registrar nome simbolico e IP del DNS che ha autorità sul dominio. Il DNS registrar poi carica i record nel TLD corrispondente.

I-Like-Networking, dns.I-Like-Networking.com, NS  
dns1.I-Like-Networking.com, 212.212.212.1, A

Le DNS registrar sono le uniche entità con i permessi di modificare il TLD.

Le CDN servono a creare una cache di contenuti distribuita geograficamente in modo da velocizzare il download da parte dei client.



### ESERCIZI

$$L = 200 \text{ Kb} \quad C = 100 \text{ Kb/s} \quad \tau = 10 \text{ ms} \quad n\_obj = 11$$

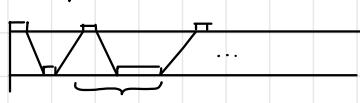
$$l = 100 \text{ bit}$$

1) Connessioni TCP parallele:



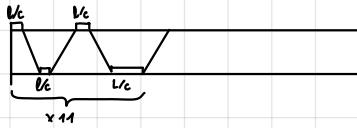
$$\left( 3 \frac{l}{C} + 3\tau \right) + \frac{L}{C} + \tau + \left[ \left( 3 \frac{l}{n} + \tau \right) + \frac{L}{n} + \tau \right]$$

2) Connessioni TCP parallele in serie



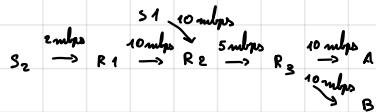
$$\left( 3 \frac{l}{C} + 3\tau \right) + \frac{L}{C} + \tau + 10 \left( \frac{l}{C} + \frac{L}{C} + 2\tau \right)$$

## 2) Connessioni TCP in serie



$$TCP + DDC. BASE + OBS$$

$$m \left( 3 \frac{L}{C} + 4 \tau + \frac{L}{C} \right)$$



$$1) \quad S_2 \rightarrow B: 1 \text{ stream} \quad R_a = \frac{5}{3} \text{ mbps}$$

$$S_1 \rightarrow A: 2 \text{ stream} \quad R_b = \frac{5}{3} \text{ mbps}$$

$$3) \quad S_2 \rightarrow B: 5 \text{ stream} \quad R_a = \frac{25}{3} \text{ mbps}$$

$$S_1 \rightarrow A: 1 \text{ stream} \quad R_b = 5 \cdot 2 \text{ mbps}$$

$$2) \quad S_2 \rightarrow B: 2 \text{ stream} \quad R_a = \frac{2}{2} \text{ Mbps}$$

$$S_1 \rightarrow A: 1 \text{ stream} \quad R_b = 5 \cdot 2 \text{ Mbps}$$

$$4) \quad S_2 \rightarrow B: 5 \text{ stream} \quad R_a = \frac{5}{20} \text{ mbps}$$

$$S_1 \rightarrow A: 15 \text{ stream} \quad R_b = \frac{5}{20} \text{ mbps}$$

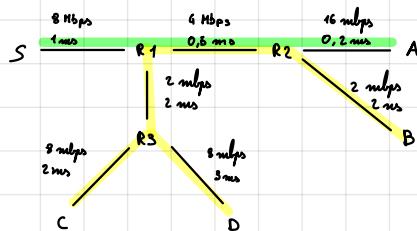
Algoritmo: per ogni collegamento dividilo  $\frac{C_i}{n_i}$  dove  $n_i$  è il n di flussi che attraversano il collegamento; prendo il minimo e calcolo la capacità residua dei collegamenti attraversati dai flussi  $n_i$ ; ripeto usando come  $C_i$  le capacità residue.

• • •

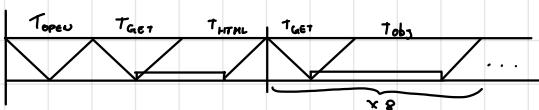
### ESERCIZI

$$A \rightarrow S \quad \text{HTTP} \quad l = 100 \text{ B}, \quad L = 1 \text{ MB} \quad n = 8$$

$$D \rightarrow B, \quad C \rightarrow B \quad \text{FTP}$$



1) parallelismo:



$$RTT = 2(\tau_1 + \tau_2 + \tau_3)$$

$$T_{OPEN} = T_{GET} = RTT$$

$$T_{HTML} = \frac{l}{R_{HTML}} = \frac{100 \cdot 8}{8 \cdot 10^6} = 0,1 \text{ ms}$$

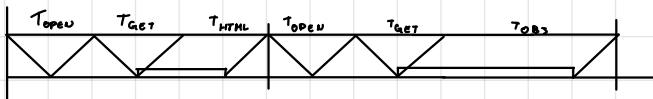
$$T_{OBJ} = \frac{L}{R_{OBJ}} = \frac{8 \cdot 10^6}{2 \cdot 10^6} = 4 \text{ s}$$

$$R_{HTML} = R_{OBJ} = 2 \text{ mbps}$$

1 mbps vs.  $\frac{4}{3}$  mbps  $\rightarrow$  cap. res. di 2 mbps per S-A

$$T_A = T_{OPEN} + T_{GET} + T_{HTML} + 2(T_{GET} + T_{OBJ}) = 82,03 \text{ s}$$

2) non parallelismo in parallelo



$$RTT = 2(\tau_1 + \tau_2 + \tau_3)$$

$$T_{OPEN} = T_{GET} = RTT$$

$$R_{HTML} = 2 \text{ mbps}, \quad R_{OBJ} = \frac{4}{10} \text{ mbps}$$

$$T_{HTML} = 0,1 \text{ ms}$$

$$T_{OBJ} = \dots = 20 \text{ s}$$

$$T = 2T_{OPEN} + 2T_{GET} + T_{HTML} + T_{OBJ} =$$

1 mbps vs.  $\frac{4}{10}$  mbps  $\rightarrow$  tutti ritardati a 0,1 mbps.

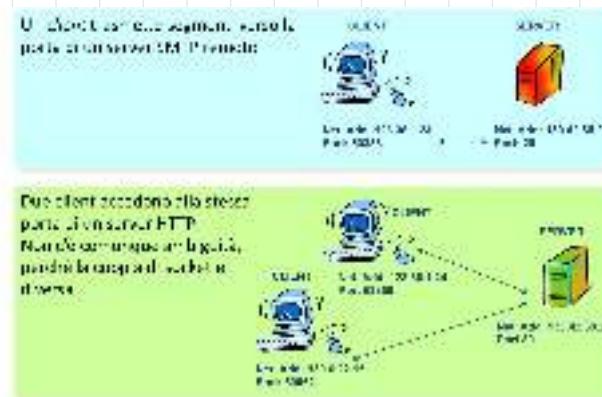
## 4. TRASPORTO

Il livello di trasporto ha il compito di individuare un collegamento logico tra le applicazioni esistenti su host remoti. Il trasporto rende trasportante il trasporto dei messaggi delle applicazioni. Il livello di trasporto è presente solo negli host e consente di creare un collegamento logico tra processi applicativi. Il livello di trasporto esegue multiplexing quindi implementa indirettamente tramite le porte.

Le porte sono salvate nelle PDU e sono valori da 16 bit. Il range è suddiviso in:

- numeri statici [0; 1023]: numeri riservati per servizi di ampio utilizzo. Utilizzati principalmente dai server per ascolto;
- numeri registrati [1024; 49151]: numeri registrati da IANA per loro servizi (tipicamente proprietari). Utilizzati dai server per ascolto;
- numeri dinamici [49151; 65535]: numeri usati dal client per la comunicazione;

L'identificazione fornita dal livello di trasporto è il socket: coppia IP-porta che identifica univocamente un applicativo.



I protocolli di trasporto sono implementati a livello S.O. A ogni socket, il S.O. crea una coda di entrata e una di uscita. Questa è la cosiddetta funzionalità di buffering.

Il livello di rete non è un livello affidabile. Il livello di trasporto, quindi, deve ricorrere a garantisce alla applicazioni affidabilità. Esistono due tipi di trasporto:

- TCP (TRANSPORT CONTROL PROTOCOL): orientato alla comunicazione + AFFIDABILE, - SPEED
- UDP (USER DATAGRAM PROTOCOL): senza comunicazione - SPEED, + AFFIDABILE

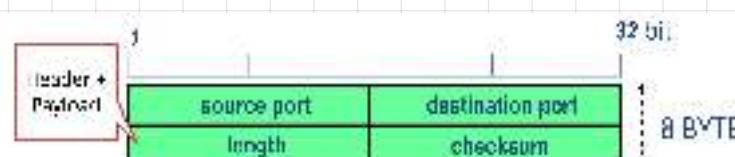
### 4.1 UDP

Le uniche cose che aggiunge al livello rete sono:

- multiplexione, demultiplexione
- controllo (severa corrispondenza) nell'header

Non esiste nemmeno controllo di flusso e di errori e non garantisce la consegna.

La PDU UDP ha formato header (8 byte) + payload



Il checksum viene calcolato da trasm. e salvato. Il ricevitore poi esegue lo stesso calcolo e, se i checksum coincidono, la PDU è valida. L'algoritmo di calcolo del checksum è molto semplice:

#### lato trasmettitore

- l'insieme di bit è diviso in blocchi da 16 bit
- il campo Checksum è inizializzato a 0
- Tutti i blocchi vengono sommati in aritmetica complemento a uno
- Il risultato è complementato ad inizio nel campo di checksum da seguire.

#### lato ricevitore

- L'insieme di bit è diviso ancora in blocchi da 16 bit
- Tutti i blocchi vengono sommati in aritmetica complemento a uno
- Il risultato è complementato
  - Se sono tutti 0 il pacchetto è accettato
  - Altrimenti è scartato

Nel calcolo del checksum vengono usati uno pseudo header e il payload.

## 4.2 TRASPORTO AFFIDABILE

### 4.2.1 PROTOCOLLI DI RITRASMISSIONE

Per operare, un protocollo di ritrasmisione necessita:

- messaggio di ACK/NACK per segnalare la corretta (o no) ricezione
- canale di servizio per trasmettere sull'altro messaggi. (anche ACK/NACK possono subire errori)

} COMUNICAZIONE BIDIREZIONALE

Per inviare un pacchetto, quindi, lo si manda e si aspetta l'ACK. Se arriva NACK o si va in timeout, il pacchetto viene ripetuto.

Se va avanti così finché il pacchetto non è stato trasmesso completamente.

Il controllo dell'errore viene eseguito ai livelli 2 e 4:

- nel 2 si controlla che i bit trasmessi arrivino correttamente (oggi non si controlla più a livello 2)
- nel 4 si controlla sia l'integrità che la ricezione, l'ordine di arrivo e la non duplicazione.

#### 4.2.1.1 STOP-AND-WAIT

Il primo protocollo di ritrasmisione è lo Stop and Wait. È il più semplice in quanto manica:

- ACK
- timer di timeout
- numerazione di pacchetti (SN) e ACK (RN)

Il funzionamento prevede la ritrasmisione se l'ACK non arriva entro il timeout. Il tempo di trasmissione dell'ACK è pari a:  $2 \tau + T_{ACK}$ . Il protocollo funziona solo se  $T_{ACK} \geq 2 \tau + T_{ACK}$ .

La parure della numerazione permette di passare alle applicazioni i pacchetti nell'ordine giusto e senza duplicazioni.  
L'efficienza del protocollo è:

$$\eta = \frac{T}{T + 2\tau + T_{ACK}} < 1$$

Si può notare che il ritardo di trasmissione influenza molto sull'efficienza. Inoltre non è adatto a sistemi con un alto ritmo di trasmissione. Lo stop-and-wait si puota molto bene alla comunicazione half-duplex (comunicazione radio walkie-talkie).

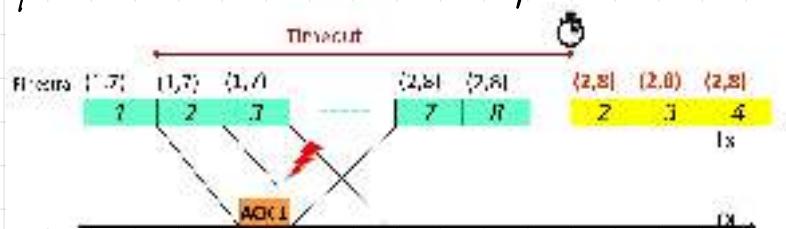
Usi dello stop-and-wait: WiFi.

#### 4.2.1.2 GO-BACK-N

Il protocollo Go-back-N è una generalizzazione dello stop-and-wait. Esso prevede la trasmissione fino a  $N$  pacchetti senza ricevere l'ACK. Questi  $N$  pacchetti sono detti finestra di trasmissione. Man mano che gli ACK vengono ricevuti, la finestra viene spostata in avanti (sliding window).

Se non ci sono errori il Go-back-N riesce ad offrire un efficienza pari a  $\eta=1$  (100%).

Nel caso venga riscontrato un errore, la finestra viene spostata indietro fino al primo pacchetto con errore e si ritrasmette. L'errore viene definito da un timeout come nello stop-and-wait.



I pacchetti fuori ordine vengono scartati. Lo stop-and-wait, quindi, è un Go-back-N con  $N=1$ .

L'ACK può anche essere collettivo: se per qualche motivo gli ACK prima di ACK  $N$  sono stati persi, l'ACK  $N$  può essere interpretato come corollata ricezione di tutti i pacchetti fino a  $N$ , rimediando alla perdita di ACK (sempre supponendo che il timeout non sia già scaduto).

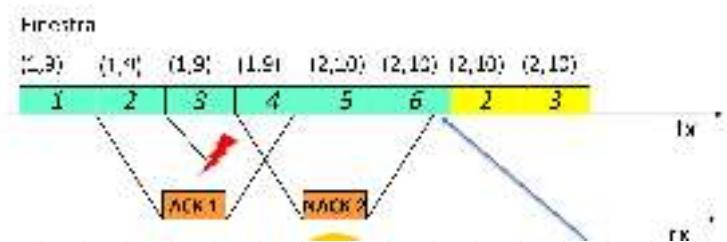
#### 4.2.1.1 GO-BACK-N

NOTA: per connessione nell'ACK si inserisce il numero del prossimo pacchetto da ricevere.

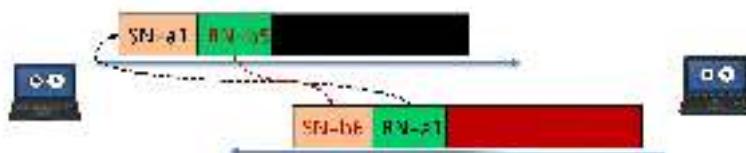
La dimensione ottimale della finestra è pari al numero di pacchetti inviati in un RTT:  $\lceil \frac{T + T_{ACK} + EC}{T} \rceil$ . Il dimensionamento della finestra è, quindi, un problema in quanto RTT è variabile. Ci sono 2 rimedi:

- finestra molto grande: non ideale in caso di errore
- uso di un qualche tipo di NACK
- stimare RTT e adattare N e timeout.

Il NACK viene utilizzato per ottimizzare gli errori: viene mandato se i pacchetti ricevuti sono fuori sequenza, evitando di far scadrà il timeout.



Con il GO-BACK-N è possibile usare la comunicazione full-duplex lungo un canale bidirezionale. Per farlo funzionare, si usa il cosiddetto piggy-backing degli ACK: vengono inseriti negli header dei pacchetti da entrambi le parti.



SN: numero di pacchetto

RN (ACK): numero del prossimo pacchetto che si vuole ricevere.

Pochi SN e RN devono essere sincronizzati, c'è bisogno di uno scambio iniziale per iniziare il protocollo. Questo è il motivo per cui i protocolli che usano controllo di errori sono orientati a connessione.

I pacchetti vengono contati a partire da un numero casuale in modo da rendere più facile l'identificazione delle connessioni.

• • •

**ESERCIZIO**

$$d = 3300 \text{ Km}$$

$$C = 2 \text{ Mb/s}$$

$$L = 1500 \text{ B}$$

ACK loss.

a) T per  $F=600 \text{ KB} + \text{efficienza } (s-a-w)$

b) T per  $F=600 \text{ KB} + \text{efficienza } (G-B-N, N=30)$

c) valore minimo della finestra.

d) T nel caso che n=31 sia errato ( $T_0=1s, N=30$ )

e) T nel caso che n=[31;40] siano variati ( $T_0=1s, N=30$ )

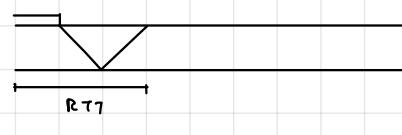
$$t = \frac{d}{v} = \dots = 195 \text{ ms}$$

$$T = \frac{L}{C} = \dots = 6 \text{ ms/s}$$

$$RTT = T + 2t = 396 \text{ ms}$$

$$n = \frac{600 \cdot 10^3 \text{ B}}{1500 \text{ B}} = 400$$

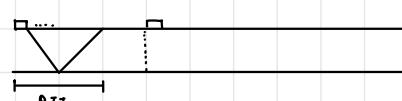
a)



$$T_{TOT} = n \cdot RTT = \dots = 158,4 \text{ s}$$

$$\eta = \frac{T}{RTT} \approx 0,015$$

b)



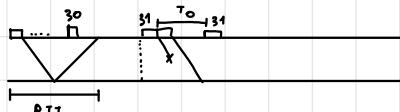
?  $N \cdot T < RTT \rightarrow \text{si}$

$$m = \lceil \frac{n}{N} \rceil = \lceil 13,33 \rceil = 14$$

$$T_{TOT} = (m-1)RTT + 10T + 2t + T_0 = 5,208 \text{ s}$$

c)  $N \geq \left\lceil \frac{RTT}{T} \right\rceil = w_{min} = 66 \rightarrow T_{TOT} = 400T + 2t = 2,79 \text{ s}$

d)



$$T_{TOT} = (m-1)RTT + 10T + 2t + T_0 = \dots$$

e) uguale a d)

#### 4.2.2 CONTROLLO DI FLUSSO

Lo scopo del controllo di flusso è quello di non sovraccaricare il buffer di ricezione ed evitare la perdita di pacchetti. Nei protocolli tipo GBN la velocità di trasmissione dipende dalla dimensione  $N$  della finestra. Nel flow control permane lo stesso concetto.

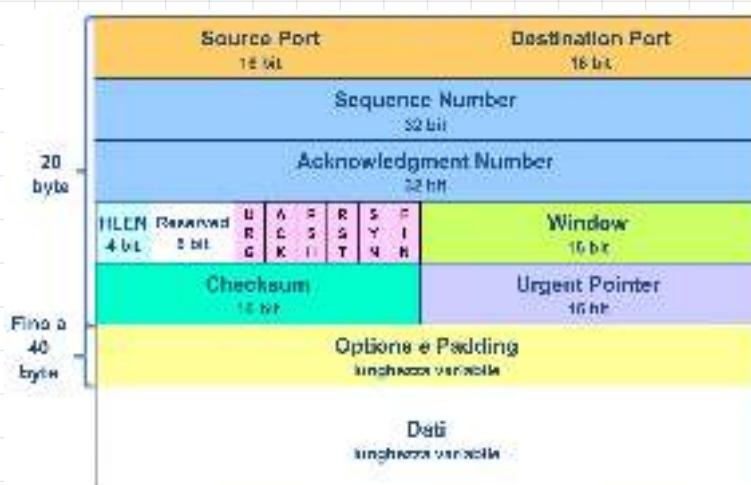
La sliding window viene spostata solo a ritorno inviato, ovvero quando il pacchetto viene preso dalla coda dal livello superiore. Se il livello superiore è troppo lento, il timeout potrebbe scadere, causando inefficienza.

Per risolvere i problemi della sliding window si usa il campo 'w' dell'header: il ricevitore lo usa nei ritorni per indicare quanti pacchetti può inviare prima della saturazione. Non è necessario dire la metà in  $w$ , in modo da mandare un messaggio o altri motivi.

#### 4.2.3 TCP

Il TCP è un protocollo a connessione, quindi esiste hand-shake, e full-duplex. Il TCP trasmette stream di byte che divide in segmenti per il trasporto. La lunghezza dei segmenti viene decisa dal protocollo per ragioni di performance.

Il controllo d'errore è in stile GBN. Il TCP numerica i byte, quindi RN sarà il numero del primo byte del segmento e, di conseguenza, RN è il prossimo byte da ricevere. Le dimensioni delle finestre sono espresse in byte.



- **Source port, Destination port:** indirizzi di porta sorgente e porta destinazione di 16 bit
- **Sequence Number:** il numero di sequenza del primo byte nel payload
- **Acknowledge Number:** numero di sequenza del prossimo byte che si intende ricevere (numero valido solo se flag ACK valido)
- **LEN (4 byte words):** contiene la lunghezza complessiva dell'header TCP che DEVE essere un multiplo intero di 32 bit
- **Window:** contiene il valore della finestra di ricezione come comunicato dal ricevitore al trasmettitore
- **Checksum:** il medesimo di UDP, calcolato in maniera uguale
- **Flags:**
  - **URG:** vale 1 se vi sono dati urgenti e quindi il TCP deve passare in modalità urgente; in questo caso urgent pointer punta all'ultimo byte dei dati all'interno del flusso oltre il quale TCP può tornare in modalità normale
  - **ACK:** vale 1 se il pacchetto è un ACK valido; in questo caso l'acknowledge number contiene un numero valido
  - **PSH:** vale 1 quando il trasmettitore intende usare il comando di PUSH; il ricevitore può anche ignorare il comando (dipende dalle implementazioni)
  - **RST:** reset, resetta la connessione senza un tear down esplicito
  - **SYN:** synchronize; usato durante il setup per comunicare i numeri di sequenza iniziale
  - **FIN:** usato per la chiusura esplicita di una connessione
- **Options and Padding:** riempimento (fino a multipli di 32 bit) e campi opzionali, es., durante il setup per comunicare il MSS (il valore di default è 536 byte)

Il TCP offre opzioni. Alcune non fanno nulla, sono lunghe 1B e servono da padding per avere un header con lunghezza multiplo di 32. Le opzioni che vedremo sono MSS e il fattore di scala della finestra.

1

- Definisce la dimensione massima del segmento che verrà usata nella connessione TCP
- La dimensione è decisa dal mittente durante la fase di setup
- Valore di default è 536 byte, il valore massimo 65535 byte

Code (00000010)	Length (00000010)	MSS 16 bit
--------------------	----------------------	---------------

2

- Definisce il fattore di scala della finestra
- Il valore di default è 1
- L'opzione fa sì che venga moltiplicato il valore del campo Window di un fattore pari a 2 elevato al valore contenuto nel campo fattore di scala

Code (00000010)	Length (00000011)	Fattore di scala 8 bit
--------------------	----------------------	---------------------------

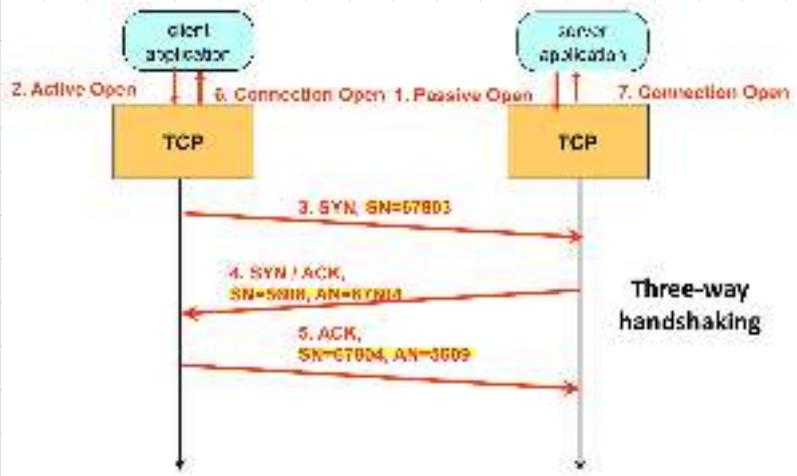
#### 4.2.3.1 SETUP

Prima del 'call setup' il client e quello server si preparano:

1. **PASSIVE OPEN:** il server comunica alla TCP stack che è in ascolto
2. **ACTIVE OPEN:** il client comunica alla TCP stack che è pronto a creare una connessione

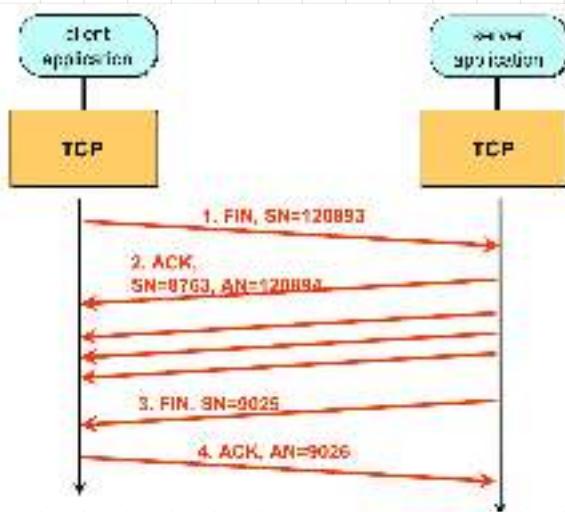
(e eventuali opzioni)

Il client inizia la comunicazione con un messaggio di SYN dove comunica l'inizio della sequenza. Il server risponde con un SYN/ACK che ha come SN = 67804 per riscontro. Il client risponde con un ultimo ACK che ha SN = 67804 e AN la prima sequenza. Le rispettive stack notificano le applicazioni del fatto che la connessione è stata stabilita. Questo è detto 3-way handshake.



#### 4.2.3.2 TEAR-DOWN

Il modo corretto per chiudere la connessione è l'invio di un segmento con flag FIN, seguito da un ACK di conferma dall'altro lato. La connessione rimane aperta dall'altra parte per permettere l'invio degli ultimi segmenti. Quando anche l'altra parte manda un segmento FIN, l'altra risponde con ACK la connessione è del tutto chiusa.



Un altro modo è quello di utilizzare il flag RST dell'header.

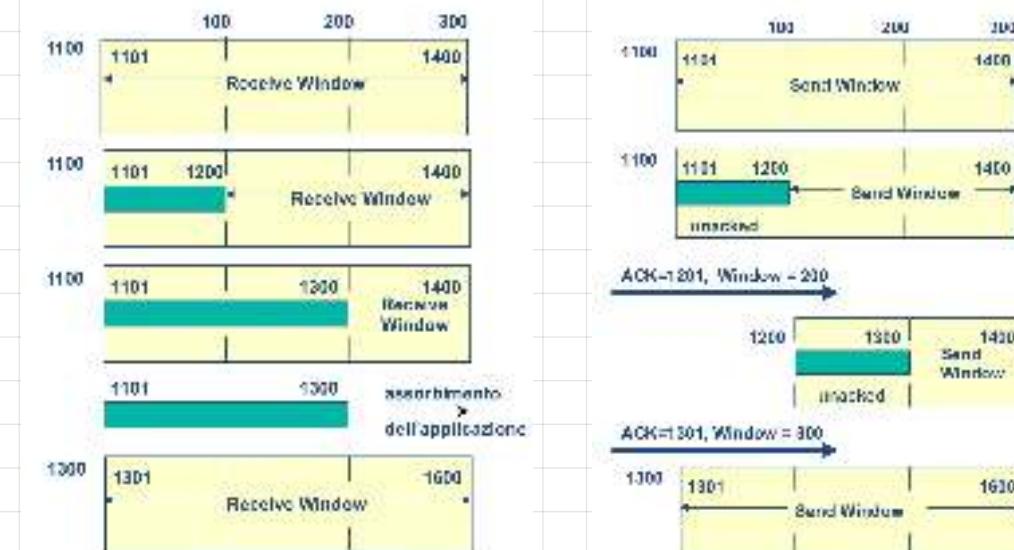
...

#### 4.2.3.3 CONTROLLO FLUSSO

Il TCP prende un buffer in ricezione e in invio. Lo spazio libero nel buffer del ricevitore viene dalla Receive Window (receive). La dimensione della finestra di ricezione è segnalata in ogni segmento.

I byte nella receive window sono numerati e riflettono l'ordine in cui sono dovuti essere ricevuti.

Analogamente la Send Window rappresenta i byte da inviare senza attendere riscontri. La send window si estende dal primo byte non ricevuto fino alla fine della receive window del ricevitore.



Le finestre hanno dei problemi. Uno di questi è la silly window syndrome. Esso è avvenuto al ricevitore ed è causato da un ricevitore nubilo il buffer troppo lentamente e quindi il buffer si riempie. Appena la finestra si riempie di poco il trasmittitore invia pochi byte con molto overhead. La soluzione è l'algoritmo di Clark: il ricevitore segnala finestra nulla finché essa non è allargata grande da ricevere un segmento di dimensione massima.

Se S.W. > avviene anche al trasmittitore guera dati lentamente e, quindi, vengono prodotti pacchetti piccoli. La soluzione è l'algoritmo di Nagle: la prima porzione viene mandata sola, il resto solo se si riempie un segmento di dimensione massima.

I comportamenti sopra possono essere evitati usando il flag PUSH: esso invia i dati anche se non si riempie un segmento massimo.

#### 4.2.3.4 CONTROLLO ERRORE

Il meccanismo è di tipo GBN-TIMEOUT con alcune modifiche:

- i seguenti fuori ordine vengono mantenuti
- quando arrivano i seguenti mancanti la finestra scorre in avanti fino al primo pacchetto non ricevuto fra quelli fuori ordine
- viene mandato un ACK che riscontra collettivamente anche i seguenti fuori ordine.

#### 4.2.3.5 GESTIONE TIMEOUT

Il RTT è molto volatili e, di conseguenza, il timeout va modificato per accomodare. Il TCP usa l'algoritmo di Karn e Jacobson:

- RTT viene definito come il tempo tra trasmissione e ricezione del relativo ACK
- sulla base dei campioni RTT il sender calcola il Smoothed-RTT come:  $SRTT^{(i)} = (1-\alpha)SRTT^{(i-1)} + \alpha RTT^{(i)}$  (o.c.a.c., solito  $\alpha = \frac{1}{2}$ )
- viene anche calcolata una varianza smooth della deviazione standard come:  $SDEV^{(i)} = \frac{3}{4}SDEV^{(i-1)} + \frac{1}{4}\Delta DEV$
- viene calcolato il timeout tramite:  $T_0 = SRTT + 4SDEV$

Il valore iniziale del timeout è 1s. Al seguito di una retransmissione è meglio usare l'algoritmo di Karn

- RTT non viene aggiornato
- Il timeout è moltiplicato per un fattore fisso (tipicamente 2)
- Il timeout cresce finché ad un valore massimo
- Dopo un numero massimo di rtrasmissioni la connessione viene chiusa

#### 4.2.3.6 PERSISTENZA

Se il ricevitore manda finora paci a 0, la trasmissione si ferma finché il ricevitore non manda un ACK con la nuova window size diversa da 0. Per evitare bloccaggi a causa di perdita di questo ACK viene usato un timer di persistenza. Se il timer di persistenza scade viene mandato un messaggio di probe. Se si riceve ACK, si fa riprendere la trasmissione, altrimenti si continua a provare (ad ogni livello del timer) finché non si riceve ACK.

#### 4.2.3.7 CONTROLLO CONGESTIONE

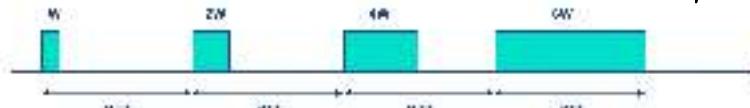
Il controllo congestione è di tipo end-to-end (la rete è, quindi, vulnerabile al sovraccarico in quanto ci si fida dell'host). La congestione viene regolata in base ai dati ricevuti (ACK, RTO...) usando una Congestion Window (cwnd) che viene negoziata opportunamente. Il trasmettitore non può trasmettere più del minimo tra RCVWND e CWND.

Per regolare la finestra di congestione si usano:

- Slow-start ( $CWND \leq SSTHRESH$ )
- Congestion avoidance ( $CWND > SSTHRESH$ )

La connessione parte in slow-start in quanto la finestra di congestione viene inizializzata a 1 MSS con SSTHRESH elevata.

Nello slow-start la cwnd viene incrementata per ogni ACK. Il nome è a caro, poiché i dati altrui che bento.



Il slow-start rende esponenziale l'incremento di cwnd. La velocità di trasmissione è pari a  $R = \frac{CWND}{RTT}$

Un evento di congestione capita quando scade il timeout. Il TCP, in questi casi, ragiona modificando SSTHRESH con questa formula:

$$SSTHRESH = \max\left(2MSS, \frac{\text{FLIGHT-SIZE}}{2}\right)$$

FLIGHT-SIZE byte trasmessi ma non riscontrati  $\cong CWND$

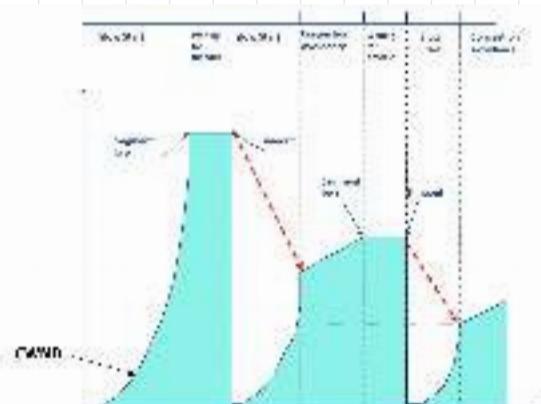
e rimettendo CWND a 1.

La CONGESTION AVOIDANCE prende l'incremento della cwnd di 1/cwnd ad ogni ACK ricevuto. Ciò significa che se la cwnd consente di trasmettere N segmenti, allora la cwnd aumenta di 1 segmento poiché riceverà N ACK.



La CONGESTION AVOIDANCE rende lineare l'incremento della cwnd.

Esempio di controllo della finestra:



• • •

#### 4.2.3.8 DIVISIONE EQUA RISORSE

Si può dimostrare che in condizioni iduali il TCP è in grado di limitare la congestione e dividere in modo equo la capacità del link.

Le condizioni iduali sono alterate da diversi RTT per diversi flussi e da buffer nei nodi minori del percorso.

#### 4.2.3.9 FAST RETRANSMIT/RECOVERY

Il TCP, mantenendo i pacchetti fuori sequenza, è in grado di richiedere un pacchetto mancante per evitare lo scadere del timeout. Il meccanismo viene chiamato degli ACK duplicati: per ogni pacchetto fuori sequenza viene inviato un ACK con AV pari al pacchetto mancante. Gli algoritmi di fast retransmit e recovery riinviano i pacchetti mancanti.

I meccanismi di fast \* non riducono la CWND, in quanto la congestione non è severa. Questo gioco funziona solo se non ci sono perdite multiple nella stessa finestra.

Funzionamento:

- 1) al 2° ACK duplicato si pone  $SSTHRESH = \max\left(\frac{\text{FLIGHT-SIZE}}{2}, 2 \text{ MSS}\right)$
- 2) viene ritrasmesso il pacchetto perso.
- 3) si pone  $CWND = SSTHRESH + 3 \text{ MSS}$
- 4) per ogni ACK duplicato la cwnd viene incrementata di 1
- 5) vengono trasmessi nuovi segmenti se cwnd > rsvwnd lo permettono
- 6) appena arriva un ACK che riscontra i nuovi dati si uscita dalla fase di recovery e si pone  $CWND = SSTHRESH = \max\left(\frac{\text{FLIGHT-SIZE}}{2}, 2 \text{ MSS}\right)$

La versione che implementa i meccanismi sopra è la TCP RENO (la prima è TCP TAHOE)

## 5. LIVELLO DI RETE (A)

Nel livello di rete non abbiamo solo 1 solo tipo di protocollo ma due:

- gestione **PIANO DATI** (trasportare dati): IP
- gestione **PIANO DI CONTROLLO** (segnalazione e supporto del trasporto): ICMP, ARP, RARP, OSPF, RIP

Il livello di rete ha funzioni di:

- indirizzamento
- inoltro / forwarding
- indirizzamento / routing

I segmenti del livello di trasporto vengono trasportati "hop by hop" dall'host sorgente a quello di destinazione. Ogni segmento è incapsulato in un header specifico del livello. I pacchetti vengono chiamati datagram.

Il servizio di trasmissione è di tipo Best Effort e serve connessione. Ogni router che riceve il datagramma e, in base all'header e ad una tabella di indirizzamento, vengono forwardati fino a destinazione. Gli algoritmi di routing si occupano di scrivere le tabelle di routing e di creare un percorso end-to-end tra gli host.

### 5.1 INDIRIZZO IP (V4)

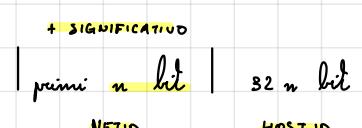
È un numero binario di 32 bit. La notazione standard è:

$$\begin{array}{c} \boxed{x.y.z.w} \\ \downarrow \\ 8 \text{ bit} \end{array} \rightarrow 128.14.3.31$$

Ogni indirizzo di rete è associata ad un'interfaccia di rete, non all'host. Perciò, di solito, un router ha più indirizzi, mentre gli host ne hanno uno.

Ogni gestore ha un blocco di indirizzi che distribuire alle interfacce

Un indirizzo IP è diviso in due parti: NETID e HOSTID



Il valore di  $n$  dipende dal tipo di rete.

### 5.2 CLASSLESS INTER-DOMAIN ROUTING (CIDR)

Introdotto negli anni '90. La notazione  $x.y.z.w/n$  significa che sono allocati  $2^{32-n}$  indirizzi dopo  $x.y.z.w$ . La  $n$  nella notazione non è altro che il numero di bit allocati all'host-id.

E.  $134.76.96.0/19 \Rightarrow$  da  $134.76.96.0$  a  $134.76.127.255$  (8192 ind.)  
 $\hookrightarrow \underline{01100000} = 96$

Un altro modo per indicare la stessa cosa è la netmask: inizia con  $n$  bit pari a 1 (n pari alla lunghezza di NETID) e ha i rimanenti  $32-n$  bit a 0.

Esempio: IP: 193.17.31.45, NETMASK: 255.255.255.0  $\in$  193.17.31.0

Ci sono anche altre modalità per indicare la NETID.

Una rete IP è un insieme di interfacce fisicamente interconnesse. È necessario che vi sia almeno un router con un'interfaccia collegata alla rete IP per comunicare con altre reti IP.

### 5.3 INDIRIZZI PRIVATI E SPECIALI

Gli indirizzi privati sono IP usati solo in reti private. Essi sono divisi in 3 blocchi:

- 1) 10.0.0.0 - 10.255.255.255
- 2) 172.16.0.0 - 172.31.255.255
- 3) 192.168.0.0 - 192.168.255.255

Ci sono anche indirizzi speciali:

- 1) host id = 0 → indirizzo della rete
- 2) host id = sub 1 → indirizzo di broadcast: vengono inviati pacchetti a tutte le interfacce delle reti.
- 3) indirizzo sub 1 → indirizzo di broadcast limitato: invia un pacchetto a tutti gli host della stessa rete
- 4) net id = 0 → host d cui indirizzo è contenuto nel campo host sulla rete militare
- 5) indirizzo sub 0 → il militare stesso del pacchetto
- 6) primo byte = 127 → indirizzo di loopback

Gli indirizzi vengono assegnati da autorità internazionale: la IANA. La IANA assegna blocchi di indirizzi a 5 regional internet registries (RIR) (solitamente indirizzi /8). Giacché RIR dà blocchi ai LIR (local i.R) che a loro volta possono assegnare prefissi ad altri clienti.

### 5.4 INDIRIZZAMENTO CLASSFUL

Gli indirizzi erano divisi in 5 classi in base alle dimensioni:

- reti grandi: NETID 7 bit, HOSTID 24 bit (A) (1...126)
- reti medio-grandi: NETID 14 bit, HOSTID 16 bit (B) (128...191)
- reti piccole: NETID 21 bit, HOSTID 8 bit (C) (192...223)
- end. multicom: x = [224; 239] (D)
- uso futuro: x = [240; 255] (E)

	1B	2B	3B	4B
A	0	/	/	/
B	10	/	/	/
C	110	/	/	/
D	1110	/	/	/
E	1111	/	/	/

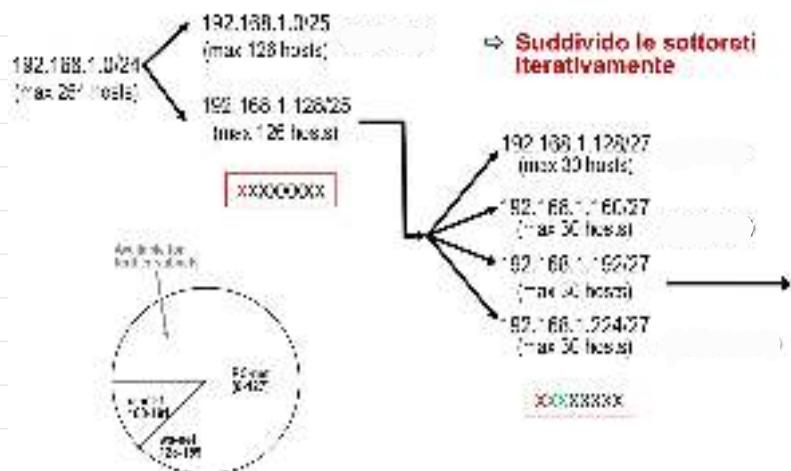
### 5.4 SUBNETTING

Alcune reti sono troppo grandi. Bisogna quindi essere in grado di creare nuovi sottoblocki. Ogni sottoretina corrisponde a una rete fisica. I sottoblocki sono suddivisi allo stesso modo delle reti, solo che il prefisso è più lungo.

NET ID	SUBNET ID	HOST ID
--------	-----------	---------

### 5.5 SUBNETTING A PREFISSO VARIABILE

Le maschere vengono applicate a cascata dalla più grande alla più piccola. Così posso usare reti delle dimensioni diverse. Le dimensioni, però, non sono arbitrarie ma sono vincolate a  $2^n - 2$  bites.



### 5.6 INOLTRO PACCHETTI

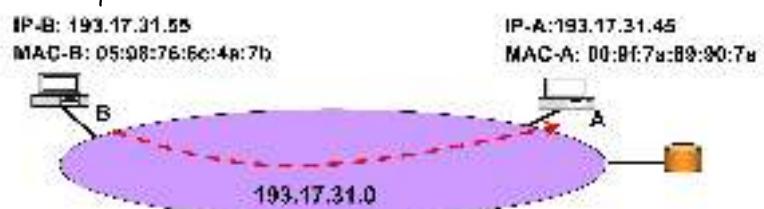
L'inoltro può essere di due tipi:

- **diretto:** destinazione è nella stessa rete
- **indiretto:** destinazione è fuori dalla rete

L'inoltro nelle reti locali si basa sugli indirizzi MAC e pacchetti di livello 2

#### 5.6.1 INOLTRO DIRETTO

La rete locale coincide con quella IP. In questo caso:

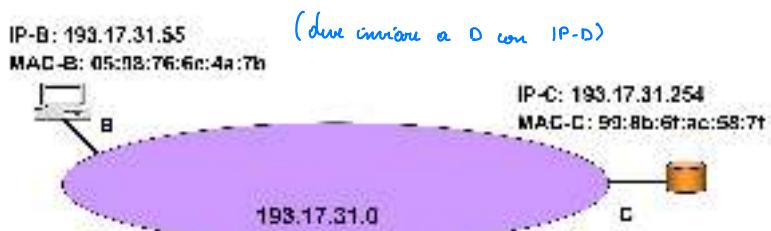


Come fa B a capire che deve fare inoltro diretto? B conosce già il suo IP che ha destinazione. B semplicemente confronta lo NET-ID del suo IP con la destinazione. Se essi coincidono, allora la destinazione è nella rete.

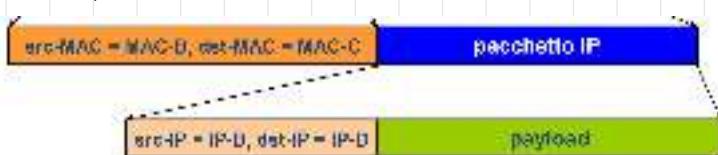
Per inviare il pacchetto, B ambasterà al livello 2 il pacchetto con il MAC preso da una tabella di corrispondenza.

#### 5.6.2 INOLTRO INDIRETTO

Il caso indiretto parla come quello diretto. Alla stessa maniera B capisce che la destinazione è fuori dalla sua rete. B userà un router come intermediario. Cognitivo ha, infatti, un default router preconfigurato.



Dalla tabella di corrispondenza recuperare il MAC del router di default e crea il pacchetto:



### 5.6.3 INOLTO NEI ROUTER

Nel corso dei router vengono create tabelle di routing dove è indicata la rotta giusta per le varie reti.

L'inoltro usa esclusivamente la NET-ID della destinazione e ha queste caratteristiche:

- **DESTINATION BASED:** l'inoltro si basa solo sulla destinazione
- **NEXT-HOP ROUTING:** per ogni rete nella tabella è indicato solo il prossimo router verso la destinazione

Tutti gli host della stessa rete vengono "aggregati" sotto una singola entry, il prefisso di rete. Per questo è necessario che i protocolli di routing devono essere in grado di inviare dei mark se vogliono fare routing.

#### 5.6.3.1 TABELLE DI INSTRADAMENTO



#### 5.6.3.2 INSTRADAMENTO DIRETTO E INDIRETTO NEI ROUTER

Per verificare su quale interfaccia inoltrare, eseguo un AND con le netmask (N.B. le netmask sono associate alle interfacce e non stanno nel pacchetto). Se non si trovano interfacce, si passa all'inoltro indiretto.

Il confronto riga per riga della tabella di routing viene effettuato allo stesso modo (la netmask è nella tabella). Se ci sono più esiti, si prende quello con il prefisso più lungo.

network	netmask	first hop
131.175.21.0	255.255.255.0	131.17.123.254
131.175.16.0	255.255.255.0	131.17.78.254
131.56.0.0	255.255.0.0	131.17.15.251
131.156.0.0	255.255.0.0	131.17.16.254
0.0.0.0	0.0.0.0	131.17.123.254

interfaccia eth0	
IP address	131.17.123.1
netmask	255.255.255.0

interfaccia eth1	
IP address	131.17.78.1
netmask	255.255.255.0

interfaccia eth2	
IP address	131.17.15.12
netmask	255.255.255.0

#### default router:

il confronto dà sempre esito positivo ma la netmask è lunga 0 bit

## Esercitazione

R1: 32 H R8: 5 IP: 201.129.287.0 /24

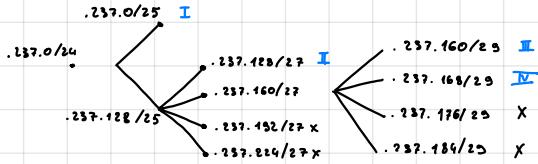
R2: 18 H R4: 4

R1: 192 più vicina  $\rightarrow$  2 bit  $\rightarrow$  /25

R2: 32 / /  $\rightarrow$  5 bit  $\rightarrow$  /27

R2: 8 / /  $\rightarrow$  8 bit  $\rightarrow$  /29

R4: 8  $\rightarrow$  3 bit  $\rightarrow$  /29



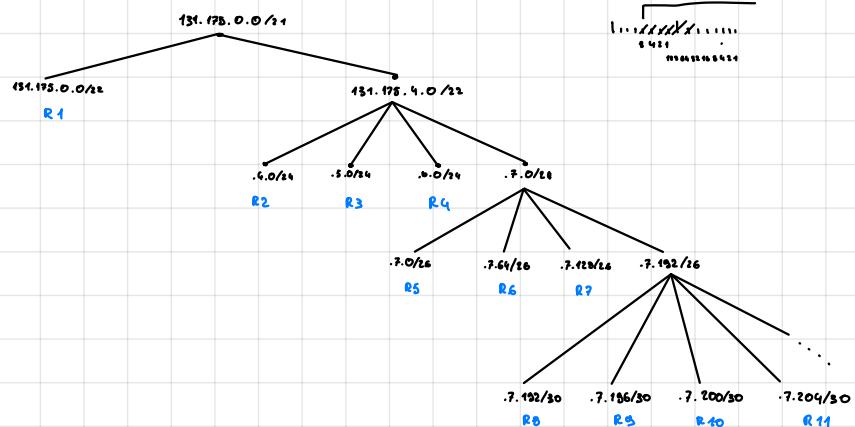
IP: 131.175.0.0/21

R1: 1000H  $\rightarrow$  10b /22

R2: R3: R4: 220H  $\rightarrow$  8 /24

R5: R6: R7: 56H  $\rightarrow$  6 /26

R8: R9: R10: R11: 2H  $\rightarrow$  2 /30



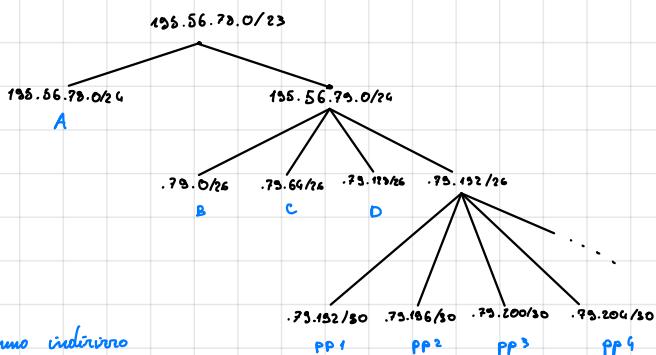
A: 210H  $\rightarrow$  8b  $\rightarrow$  /24

B: 55H  $\rightarrow$  6b  $\rightarrow$  /26

C: 57H  $\rightarrow$  6b  $\rightarrow$  /26

D: 61H  $\rightarrow$  6b  $\rightarrow$  /26

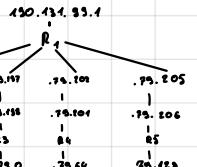
PP\*: 2H  $\rightarrow$  4b  $\rightarrow$  /30



Le interfacce interne dei router di rete hanno indirizzo broadcast - 1

Tabelle di routing:

R1	135.156.70.0	/24	135.56.75.193
	135.156.75.0	/26	135.56.75.197
:	:	:	:
	0.0.0.0	/0	190.131.99.1 (I)



...

### 5.7 SUPERNETTING

Il subnetting può causare la crescita sproporzionale delle tabelle di routing dei router (principalmente del core della rete, non della periferia). Il problema viene risolto con la supernetting. Essa non è altro che l'inverso di quello che fa la subnet mask. La riduzione viene effettuata principalmente su base topologica.

È ovvio che se un router ha interfacce vicine a una rete, non può fare supernetting in quella stessa rete. Il supernetting ha senso principalmente nei grandi router nel centro della rete.

Il supernetting permette una distribuzione gerarchica delle reti. I router, infatti, possono comunicare e dire ad altri router il range di indirizzi su cui hanno controllo.

### 5.8 FORMATO TABELLE

Netmask	Destination	Next Hop (Gateway)	Flag	Metric	Use	Interface
255.0.0.0	1.24.0.0.0	14.0.0.1/25	LG	4	20	E0/1
.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....

- Flag:
  - Unicast/Allcast
  - C. una lista di finali da le reti (host/gateway)
  - H. destination è un host specifico (unicast)
  - U. indirizzo unico da partecipante in un ring o nei rottorini ICMP
  - M. un bit indicante se il pacchetto è un messaggio di richiesta della CCN
- Reference Count: numero connessioni attive per quella rete
- Use: numero pacchetti verso il destinatario (numero di occup)
- Interface: nome l'una fonda di uscita.



## 6 LIVELLO DI RETE (B)

### 6.1 IP (DATA PLANE)

È un protocollo che offre un servizio molto semplice:

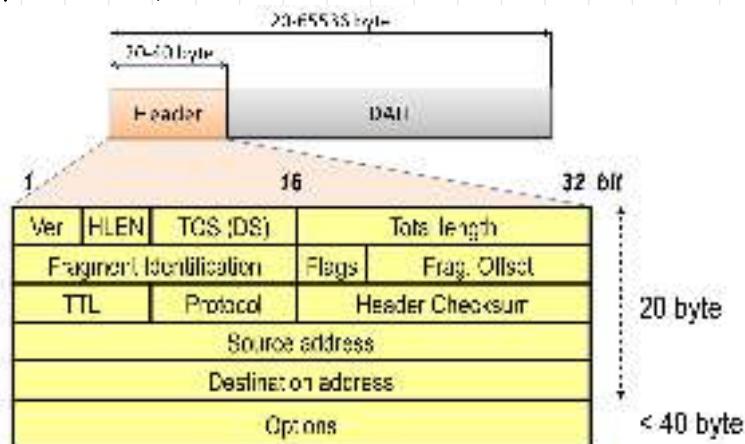
- connectionless: paradigma datagram
- non affidabile: consegna best-effort senza garanzia di successo.

Offre servizio di:

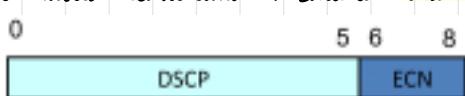
- indirizzamento: viene assegnato un indirizzo univoco non toccato da altri
- frammentazione / deframmentazione su richiesta di livelli inferiori (gira intorno alla restrizione dell'MTU del livello 2)

[20; 40] b

I pacchetti IP si chiamano datagram e ha la solita struttura header + payload. I pacchetti hanno una maximum lifetime (dopo essere circolati per troppo vengono dropati) e sono checksummati.



- VER: versione
- HLEN: lunghezza header in multipli di 32
- TOS: ha subito molti cambiamenti. Indica come deve trattarlo il pacchetto



DSCP: priorità pacchetti

ECN: regolazione congestione (usato in TCP)

- **TOT LENGTH:** lunghezza Ideale pacchetto
- **IDENTIFICATION:** identifica un frammento in modo univoco (nello dell' ip del frammento)
- **FLAGS:** usati nella frammentazione:
  - **M:** pari a 0 solo sull'ultimo frammento
  - **D:** "do not fragment": non frammentare MAI (se si deve frammentare, il pacchetto viene scartato e viene mandato errore; questi errori possono essere usati per determinare la MTU delle reti (IP v6, PATH MTU DISCOVERY))
- **FRAG. OFFSET:** indica il primo byte sul frammento (i byte del datagram originale sono numerati)
- **TTL:** tiene lo life; indica il numero massimo route attraversati (se scade manda l'errore ICMP "TIME EXCEEDED")
- **PROTOCOL:** identifica il livello 4 (TCP/UDP)
- **CHECKSUM:** come in UDP (niente pseudo-header)

Le opzioni oggi non sono usate e tipicamente vengono ignorate dai router in quanto rischio per la sicurezza. Il loro lunghezza massimo 40 byte e sono usate per testing / debugging.

...

## 6.2 ICMP (CONTROL PLANE)

È un protocollo usato per trasmettere messaggi di controllo tra router e host. Può essere considerato come parte dell'IP. I messaggi ICMP vengono trasportati da datagram IP. Esso è quindi anche un utilizzatore di IP. In IPv6, la maggior parte delle funzionalità di ICMP sono state spostate in IP.

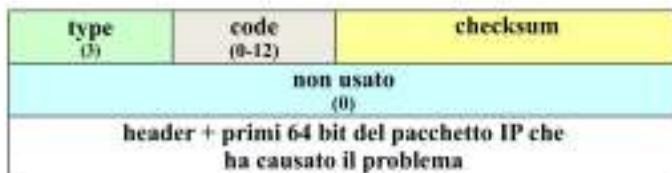


Type indica il tipo del messaggio:

- ERRORE: Destination Unreachable, Time Exceeded...
- QUERY: Address Mask Request ...

Il ICMP non fixa errori, ma li segnala alla sorgente. I messaggi di errore contengono header + primi 8 byte del pacchetto problematico.

### 6.2.1 DESTINATION UNREACHABLE



Il router scatta un pacchetto per qualche motivo. Il campo code specifica il codice d'errore.

- 0 network unreachable
- 1 host unreachable
- 2 proto unreachable
- 3 port unreachable
- 4 fragmentation needed and DF set
- 5 source route failed

### 6.2.2 TIME EXCEEDED

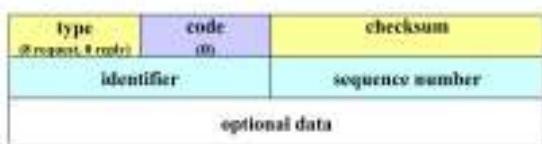


Mandato dal router sul corso TTL arrivi a 0. I.e.:

- code 0: un router ha scattato TTL a 0
- code 1: non tutti i frammenti sono arrivati all'host entro un tempo limite.

### 6.2.3 ECHO REQUEST

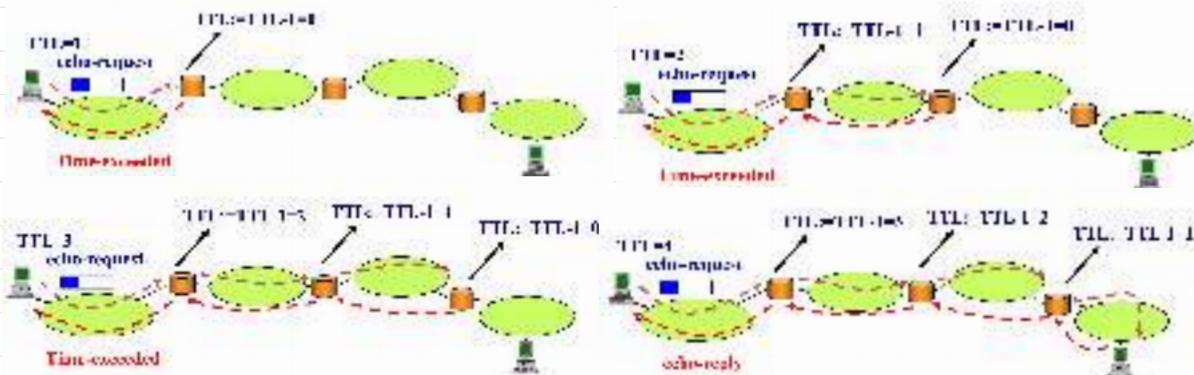
È un messaggio di diagnosi. Segue il paradigma Domanda/Risposta. Un host/router che riceve una Echo Request risponde con una Echo Reply.



Il campo **ttl** definisce un limite dal munitore. La risposta contiene lo stesso identificativo della domanda. Più richiede consultare possono avere diverso TTL ma sequenze number diverso. Il munitore può aggiungere una sequenza di dati che verrà rimandata uguale per verificare errori.

I principali utenti delle Echo Requests sono "ping" e "traceroute".

Traceroute usa le Echo requests e TTL per tracciare il percorso dei pacchetti IP. Usa un incremento iterativo del TTL dei pacchetti IP con dentro uno Echo Request per sondare i vari router sul percorso.



Una alternativa è usare pacchetti UDP: la destinazione risponderà con un errore ICMP Port unreachable.

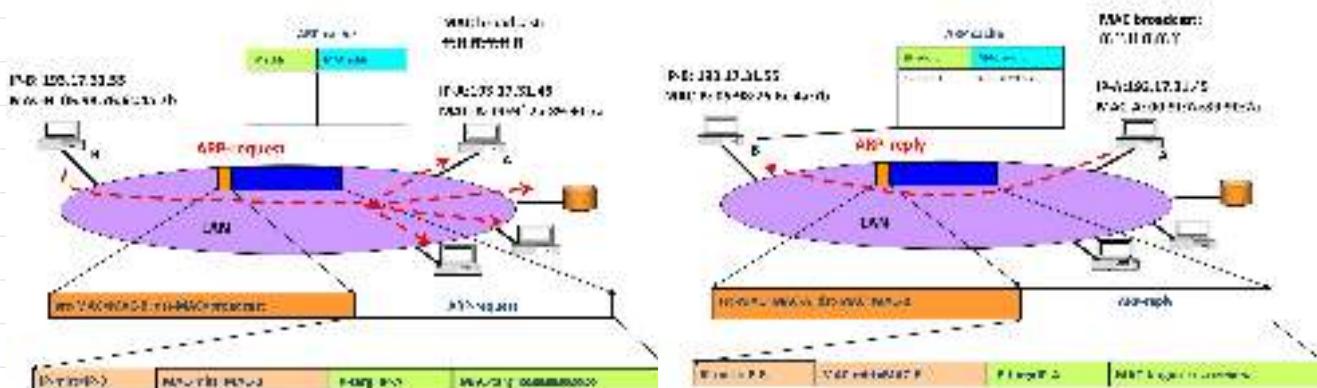
#### 6.2.4 ADDRESS MASK REQUEST

Un host lo invia al router locale per conoscere la propria netmask. È solitamente sostituito da DHCP e viene usato solo in fase di setup.

#### 6.3 ARP e RARP

Nell'invito direttamente abbiamo rapporto l'esistenza di una tabella di corrispondenza IP-MAC. Questa Tabella è generata da ARP. ARP si basa sul broadcast delle reti IP. Se nella ARP-cache (tabella IP-MAC) non è presente un MAC allora verrà broadcastata una ARP-request. Ogni host controlla se il MAC corrisponde al proprio e, se sì, risponde con una ARP-REPLY.

I messaggi ARP vengono incapsulati in frame di livello 2. Gli indirizzi MAC sono a 48 bit in notazione esadecimale. La ARP-reg viene mandata al MAC broadcast FF:FF:FF:FF:FF:FF.



Il RARP esegue l'operazione inversa: associa un IP a un MAC. Oggi è sostituito da DHCP. Il RARP viene usato per il booting in rete di macchine diskless.

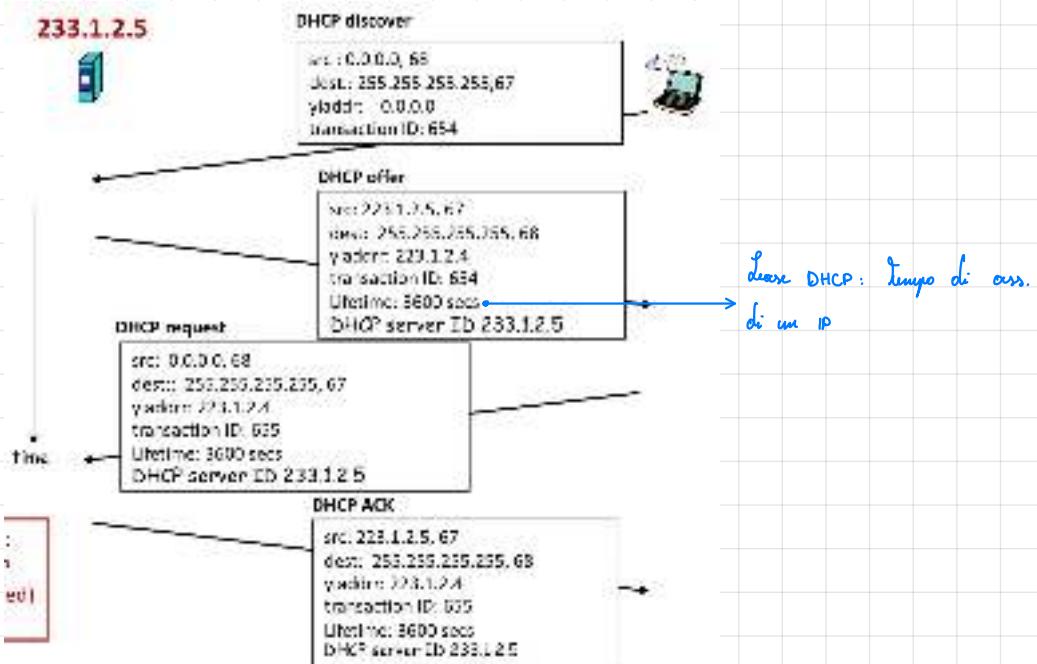
#### 6.4 DHCP

Assegnare IP fissi ai vari host è scomodo e poco flessibile. È meglio usare un server che gestisce molti indirizzi e li allochi in modo dinamico. Ci sono 3 tipi di associazioni:

- **dinamica**: gli IP sono nuovi degli host, quindi si riutilizzano IP quando possibile.
- **statica**: il server ha una tabella IP-MAC che viene consultata ad ogni query.
- **autonoma**: il server automatizza la procedura di corrispondenza sulla tabella

Il **DHCP** risolve questo problema. È un protocollo client/server. Quando un host necessita IP manda in broadcast un messaggio di **DHCP DISCOVER**. Il server risponde con una offerta **DHCP OFFER**. Se il client accetta l'offerta manda una **DHCP REQUEST**. Il server conferma con **DHCP ACK**, mandando i parametri di configurazione. Oppure il client ha finito manda una **DHCP RELEASE**.

Il **DHCP** usa delle porte nede (67 server, 68 client) e tutti i messaggi sono mandati in broadcast.



È possibile che in una rete ci siano più server. È possibile anche avere un **DHCP relay**, una specie di proxy DHCP, permettendo al vero server di stare su un'altra rete IP da quella del client.

DHCP è un protocollo di livello 5; è trasportato da UDP e lavora con:

- SORGENTE: 0.0.0.0
- DESTINAZIONE: 255.255.255.255
- P. SORGENTE: 68
- P. DESTINAZIONE: 67

I parametri di configurazione sono principalmente:

- IP
- Netmask
- Gateway (router di default)
- DNS server

#### 6.5 NAT

Il **NAT** traduce indirizzi privati in pubblici. È un servizio offerto a livello router.



Venne assegnato a ogni indirizzo privato un indirizzo pubblico da un pool. Le associazioni vengono tenute in una tabella.

Le associazioni hanno un tempo limitato pari alla durata della connessione. Nel caso del TCP è facile, ma nel caso di UDP è più difficile. Poiché alcune applicazioni trasportano IP (ASCII/BIN) nei loro messaggi, è necessaria la

(fornita dal firewall/Other)

funzionalità di Application Level Gateway (ALG) che traduce anche i modelli indirizzi leggendo dentro i vari pacchetti. Gli ALG sono, precisamente, come dei proxy con la sola differenza dell'essere trasparenti. Gli ALG sono necessari per il corretto funzionamento del NAT.

Il NAT tradizionale (detto anche allorno NAT) permette l'inizio di sessioni solo dall'interno. Le informazioni di routing possono essere distribuite solo dall'interno all'interno. Ci sono due tipi di NAT Tradizionali: **BASIC** e **NAPT**:

- **BASIC NAT:** viene tradotto solo l'IP e c'è una corrispondenza diretta tra IP pubblici e privati. Possono esserci blocchi a causa della scarsa presenza di IP pubblici.
- **NAPT:** viene tradotta la coppia indirizzo-porta, in modo da permettere allo stesso IP pubblico di corrispondere a più IP privati.

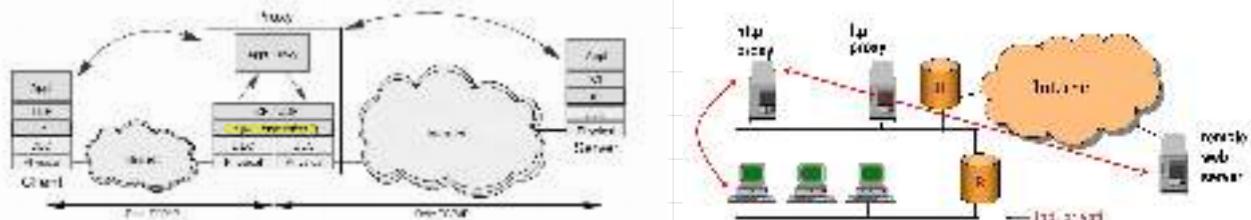
Il NAT bidirezionale permette di usare un server pubblico con indirizzo privato. Per ottenere ciò, si usa il DNS che usa un unico spazio e collabora con il NAT. Questo tipo di DNS si chiama **dynamic-DNS**. Per far corrispondere IP:PORTA ai corrispondenti pubblici si usa il cosiddetto **port-forwarding**.

## 6.6 INTRANET E ALTERNATIVE AL NAT

- Le reti private si sono evolute grazie alla tecnologia IP e sono passate da grandi reti collegate a livello 2 (bridge) a reti collegate con router IP
- Una intranet non è altro che una rete privata che utilizza tecnologia di interconnessione IP, dotata degli stessi servizi dell'INTERNET come server www, server e posta ecc.

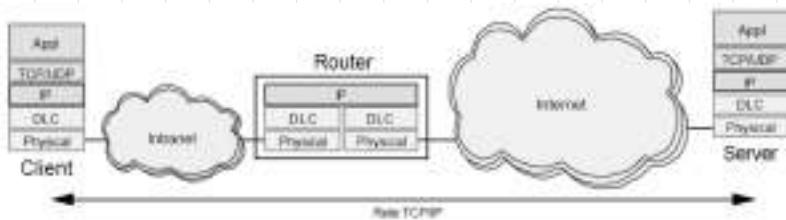
- L'evoluzione di servizi e protocolli ha però reso le Intranet strutturalmente differenti dalle reti pubbliche
  - Problemi di sicurezza
  - Problemi di gestione degli indirizzi
  - Problemi di distinzione tra servizi offerti ai soli utenti della Intranet e servizi offerti anche agli utenti di INTERNET

### 6.6.1 APPLICATION PROXY



### 6.6.2 ROUTER

Può funzionare solo se la Intranet usa indirizzi pubblici:



Non è una buona soluzione poiché l'intranet, di fatto, viene inglobata in internet, rendendola tutto poco sicuro.

## 7 LIVELLO RETE (C)

L'indirizzamento unicast consente a due nodi A e B di essere collegati direttamente per comunicare. Esistono anche indirizzamenti broadcast e multicast. Le entità di livello 3 eseguono forwarding alla prossima entità basandosi sulle tabelle d'indirizzamento. Per ora abbiamo supposto che le tabelle fossero scritte e mantenute da uomini. Nella realtà esse sono gestite da algoritmi di routing (politiche di routing). Gli algoritmi di routing determineranno il percorso attraverso la rete, mentre la tabella si occuperà del forwarding locale.

I protocolli di routing racchiudono due funzionalità:

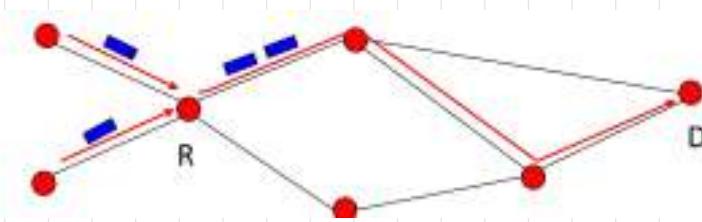
- scambio fra router di informazioni di raggiungibilità
- costruzione di tabelle d'indirizzamento

Il protocollo, formalmente, è solo la parte che descrive lo scambio di messaggi. L'algoritmo è implementazione dettagliata.

Il routing è legato alle capacità delle reti: il routing può suddividere il carico su un link in modo che tutti vadano più veloce. La capacità totale di una rete, quindi, dipende dal routing.

Il tipo di invito IP (next-hop, destination-based) ha delle importanti conseguenze sulle politiche di routing:

- i pacchetti da A a B che arrivano in R seguiranno lo stesso percorso indipendentemente dal link d'ingresso



L'insieme dei cammini da una sorgente verso una destinazione sarà un'albero.

Per il calcolo dei cammini si usa il metodo dei cammini minimi con metrica generale (non lunghezza). Il calcolo avviene in modo distribuito tramite scambio di informazioni. I cammini minimi risalgono alla ristrutturazione sopra pochi tutti i sotto-cammini di un cammino minimo sono anch'essi minimi.

La metrica può essere il numero di salti, lunghezza delle code ecc...

La rete viene rappresentata come un albero completo. Il calcolo di questo albero è distribuito ed è basato su algoritmi efficienti.

### 7.1 ALGORITMO DI BELLMAN-FORD

Ci sono poi sia positivi che negativi. Non ci deve essere nessun ciclo a lunghezza negativa. Da come scopo quello di trovare i cammini minimi tra una sorgente a tutti gli altri nodi e riceverla.

La variabile  $D_i^h$  indica la lunghezza del cammino minimo tra i e il nodo i composto da un numero di archi inferiore ad h.

In ciascuna iterazione aggiorniamo  $D_i^{h+1} = \min(D_i^h, \min_j(D_j^h + d_{ji}))$  dove j sono i "vicini" di i. L'algoritmo termina in  $N-1$  passi. La complessità è di  $O(N^3)$

```

 $h=0;$ 
 $D_i^h = 0; \forall i;$ 
 $D_i^h = \infty \forall j \neq i;$ 
repeat
     $h = h + 1;$ 
     $D_i^h = \min\{\min_j(D_j^{h-1} + d_{ji}), D_i^{h-1}\};$ 
until  $D_i^h = D_i^{h-1} \forall j \neq i$ 

```

Si può dimostrare che l'algoritmo converge anche in modalità distribuita. Periodicamente i nodi inviano ai nodi vicini le proprie stime e aggiornano la propria seconda i parametri ricevuti. L'ordine di aggiornamento è ininfluente.

Nella pratica l'algoritmo è implementato usando delle dichiarazioni per ogni nodo contenenti:

- $n$ : primo nodo nel cammino minimo
- $L$ : lunghezza del cammino.

Le dichiarazioni vengono aggiornate guardando le dichiarazioni dei vicini. Quando le dichiarazioni smettono di cambiare si ricostituisce l'albero seguendo le dichiarazioni.

## 7.2 ALGORITMO DI DIJKSTRA

Ricorda un'assunzione in più di quello di B.F.: ordini con passi positivi. Lo scopo è lo stesso di B.F. Le variabili sono le stesse di B.F., con l'unica differenza che  $d_{i,j} = \infty$  se l'arco tra  $i \rightarrow j$  non esiste.

```

dist[s] ← 0
forall v in V \ {s}
    dist[v] ← ∞
S ← ∅
Q ← V
while Q ≠ ∅
    u ← mindist(Q,dist)
    Q ← Q \ {u}
    S ← S ∪ {u}
    forall v in Neigh(u)
        if dist[v] > dist[u] + w(u,v)
            then d[v] ← d[u] + w(u,v)

```

• Nodo s è radice e ha coste 0

- Inizializzo gli altri nodi a costo  $\infty$

S è l'insieme dei nodi a etichetta permanente

Nodo u è il nodo con etichetta

- non permanente a distanza minima

• Etichetta di u diventa permanente

Aggiorno i vicini di u con la nuova distanza passando da u solo se migliora della distanza precedente

Nella pratica funziona similmente a B.F. con l'unica differenza essendo l'uso di etichette temporanee e permanenti. L'algoritmo di Dijkstra ha  $O(n^2)$  e termina in massimo  $N-1$  iterazioni

## 7.2 INFORMAZIONE GLOBALE E DECENTRALIZZATA

I protocolli di routing si possono classificare in base al tipo di informazioni disponibili su ciascun router:

- **GLOBALE:** Tutti i router vedono la topologia totale e hanno informazioni sul costo di ciascun link → alg. **LINK STATE**
- **DECENTRALIZZATA:** i router vedono solo i percorsi dei vicini e costruiscono la tabella comunicando esclusivamente coi vicini  
↳ alg. **DISTANCE VECTOR**

## 7.3 INSTRADAMENTO DISTANCE VECTOR

Ha come output la tabella di instradamento annotata con la minima distanza ad ogni altro nodo. In forma distribuita, ogni nodo riceve la lista delle distanze dai suoi vicini, somma la sua distanza al vicino e sceglie la distanza minima verso ogni altro nodo. Il distance vector (DV) viene inviato periodicamente o a causa di un cambio nella rete. Ogni nodo calcola e invia il nuovo DV se riceve un DV diverso da quello salvato o se avviene un evento di modifica della rete.

Pseudo-codice per la ricezione di un DV da un vicino:

- 1 Incrementa la distanza delle destinazioni specificate da costo del link in ingresso
- 2 Ricavi per ogni destinazione specificata nel DV
  - Se la destinazione non è nella tabella di routing
    - Aggiungi a destinazione-distanza
  - Altrimenti
    - Se l'attuale costo di destinazione è superiore al minimo del DV
      - sostituisci l'informazione della tabella di routing con quella nuova
    - Altrimenti
      - Aggiorna la tabella di routing nel DV in base al costo minimo
      - Eseguisci l'aggiornamento della tabella di routing con quel DV
- 3 Torna

Ogni nodo si attiva contemporaneamente (cold start) e conosce i link ai quali è connesso direttamente. Inizialmente le tabelle di routing hanno una sola entry con il nodo stesso.

Il principale vantaggio del distance vector sta nella sua semplicità. Tutti gli svantaggi risiedono nell'implementazione:

- i buchi o convergenza (proporzionale al numero di nodi)
- limitate dal nodo più lento
- problemi di stabilità su reti grandi con tanti gradi / cambi:
  - cicli
  - counting to infinity

### 7.3.1 COUNTING TO INFINITY

Può accadere quando un nodo manda informazioni corrette perché non è a conoscenza di una modifica nella rete. Invece allora un ciclo infinito fra i nodi dove le distanze dei nodi raggiungibili vengono incrementate fino all'infinito mettendo la rete in uno stato incoerente.

Per mitigare il problema si può:

- limitare il counting a un max offsetterro basso.
- SPLIT-HORIZON: il nodo omette nel DV ogni informazione sulle destinazioni che raggiunge tramite quel link
- POISON REVERSE: il nodo include nel messaggio tutte le destinazioni ma pone a distanza infinita quelle tramite quel link

Un problema dello split-horizon è che può non funzionare per tutte le topologie. Il rimedio è di due timer:  $T_{HALF}$  e  $T_{FLUSH}$ .

Un router segna la rete come instabile (hold down) quando:

- riceve un DV con distanza infinita per la rotta
- non riceve un DV che segnala la rotta dal nodo del primo hop per un tempo  $T_{HALF}$ .

Le rotte in hold down non vengono annunciate nei DV, non vengono considerati validi i DV con metrica peggiore e possono uscire da hold down se viene ricevuto un DV migliorativo. Dopo un tempo  $T_{FLUSH}$  la rotta è cancellata.

I due timer vanno tarati in modo da favorire la propagazione nella rete.

Un altro modo è segnalare i cambiamenti con un trigger update.

• • •

### 7.3 INSTRADAMENTO LINK STATE

Ogni nodo conosce la topologia completa della rete. Non ci è quindi un algoritmo distribuito in stile B.F.

I nodi mandano agli altri router dei Link State Packets (LSP) contenenti informazioni sulla topologia locale di vicinno nodo. I LSP vengono mandati in flooding a tutti i nodi della rete. Ogni nodo costruisce, quindi, un database di LSP e una mappa completa della rete. Il grafo così costruito viene usato per il calcolo dei cammini minimi.

VANTAGGI: più flessibile; LSP mandati solo in caso di cambiamento; Tutti i router sono informati dei cambiamenti

SVANTAGGI: richiede un protocollo dedicato per mantenere info sui vicini (Hello); uso del flooding; richiede riscontro degli LSP inviati; consumo

#### 7.3.1 FLOODING

Ogni pacchetto viene ritrasmesso su ogni link eccetto quello di niente. Un problema è la possibile presenza di cicli che esplodono causando una "broadcast storm". Ciò si evita con:

- numeri di sequenza + database di SV in modo da non ritrasmettere lo stesso pacchetto una seconda volta
- controllore di loop (~ TTL IP)

#### 7.3.2 GESTIONE INOLTRO LSP

All'arrivo di un LSP se:

- ha SV maggiore o non è nel database LSP → save e flood
- ha SV uguale → drop
- ha SV minore → viene mandata una copia dell'LSP aggiornato al niente

#### 7.3.3 OSSERVAZIONI

- entrambi i tipi di algoritmo convergono alla stessa soluzione in situazioni statiche

- possono essere implementati sia distribuiti che centralizzati

- DISTANCE VECTOR ha:

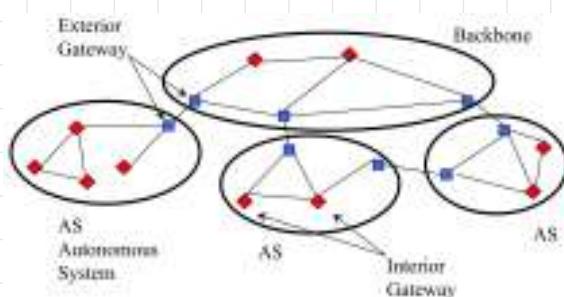
- convergenza più lenta in situazioni dinamiche
- ogni nodo sa solo cosa vedono i vicini

- LINK STATE ha:

- ogni nodo vede "tutta" l'intera rete

## 8 LIVELLO DI RETE (D)

Applicare DV o LS su tutta internet è impossibile. Diversi protocolli di routing lavorano invece su pezzi di internet gestiti dalla stessa autorità: un pezzo di rete gestito da una sola entità è detto Autonomous System. Ogni autonomous system gestisce solo il suo di routing.



Un router al bordo di un AS viene detto exterior gateway. Quello all'interno della AS si chiama interior gateway. Le reti di diversi operatori si incontrano nei NAP (mix).

All'interno degli AS, gli interior gateway si scambiano informazioni topologiche complete usando un Interior Gateway Protocol. Gli exterior gateway scambiano informazioni tramite un exterior gateway protocol.

L'EGP comunica all'interno dell'AS informazioni di raggiungibilità esterno. I vari exterior gateway scambiano tra di loro informazioni simili che di raggiungibilità.

In un AS possono essere configurati più IGP. Un routing domain (RD) è una porzione di AS che implementa lo stesso IGP. Alcuni router fanno da frontiera tra gli RD e devono fare redistribuzione di pacchetti traducendo da un protocollo all'altro. La traduzione può avvenire anche tra IGP ed EGP.

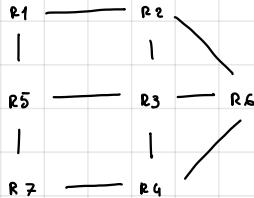
#### 8.1 TIPI DI INOLTRE

- **DIRETTO:** i net-id coincidono e l'indirizzamento è eseguito a livello 2
- **INDIRETTO:** i net-id non coincidono ma appartengono allo stesso AS e l'indirizzamento avviene tramite IGP
- **INDIRETTO GERARCHICO:** sorgono e dest. appartengono a AS diversi; l'indirizzamento avviene con IGP fino all'exterior gateway, con EGP fino all'exterior gateway dell'AS di dest. e con IGP fino a destinazione.

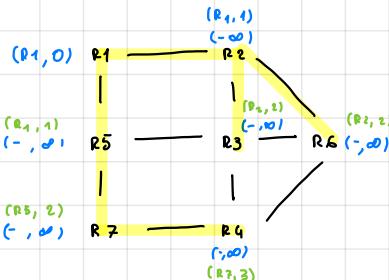
## ESERCITAZIONE

### ESERCIZIO: REGOLE PER IL SUPERNETTING

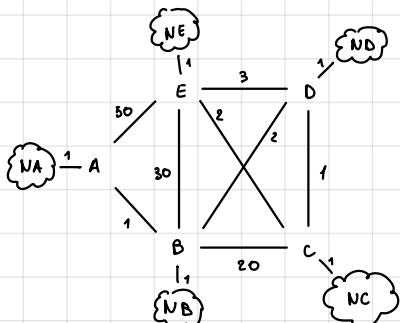
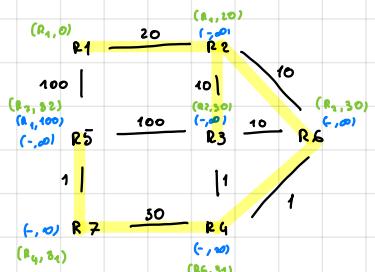
- Si presume di avere un gruppo di link che hanno lo stesso step. Sovraccarico il numero dei link con uno prefisso? (esempi: 2, 4, 8, ..., 1) per permettere un'unica regola che contiene l'oggetto, ovvero la superrete, ed è comunque accettabile altrui.
- Si presume di avere più link con lo stesso step. Quale deve essere la rete? Il next-hop è diverso. In questo caso il gruppo è suddiviso da un'altra riga di contenuto l'oggetto, più una riga per riunire delle regole in gruppo con diverse next-hop (swapping route) che sono le loro intolleranze.
- Si possono disporre nell'ordine come nella prima regola anche le manzane. Nella tabella alcune righe specificano il gruppo stesso, altre righe che controllano l'oggetto, più una riga per riunire delle regole manzane e una per i quelli che servono di esclusione.
- Si possono eliminare tutte le righe con next-hop parallela come di seguito:



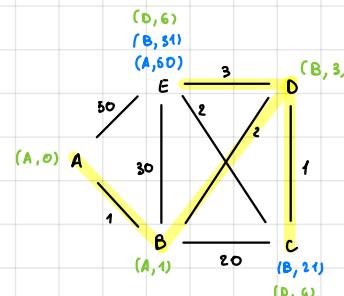
Cammino minimo tra R1 e altri (Rightbox)?



Si ripeta il calcolo considerando il peso dei link pari a 100% (C è la capacità)?



Calcola il grafo di C.M.:  
(escludi le reti)



Come sono i D.V. mandati da A?

(NA, 1), (NB, 3), (NC, 5), (ND, 4), (NE, 7)

Con S.H., come sono i D.V. mandati da A?

(NA, 1)

...

## 8.2 RIP

Protocollo IGP di tipo distance vector. Usa l'algoritmo di B.F in modo distribuito. Usa come metrica il numero di hop (max 16) e comunica in UDP con indirizzo 255.255.255.255 su porta 520.

L'header contiene:

- Command: 1 per domanda, 2 per risposta
- Version: versione RIP

Command	Version	Reserved
Family	AllS	
Network address		
AllR		
AllS		
Distance		

Giocano ovv i formato da:

- Family: tipi di indirizzi (255.255.255.255)
- Address: indirizzo di destinazione
- Distance: hop count dalla rete di destinazione

Le rivelate possono venire da: un router appena attivato, in scadenza di validità di una rotta. Le risposte sono le solite risposte D.V. con massima lunghezza di 25 DV.

Il flow dei messaggi viene gestito dai timer:

- routing update timer: intervallo di invio dv (30 s)
- route invalid : intervallo dopo il quale, se non si ricevono dv dalla stessa interfaccia, invalida la rotta (180 s)
- route flusso : intervallo dopo cui una rotta viene eliminata (60-120 s)

Il RIP supporta il triggered update

Il RIP ha varie limitazioni:

- metrica troppo sempistica
- limitazione di 15 hop lo rende adatto solo a reti medio-piccole
- convergenza lunga dovuta ai timer

## 8.3 RIP v2

Versione migliorata che aggiunge supporto per il classless routing, indicazioni esplicativa del next-hop, autenticazione e multicasting (224.0.0.9 come indirizzo)

Command	Version	Reserved
Family	Routing	
Network address		
Number routes		
Next hop address		
Router ID		

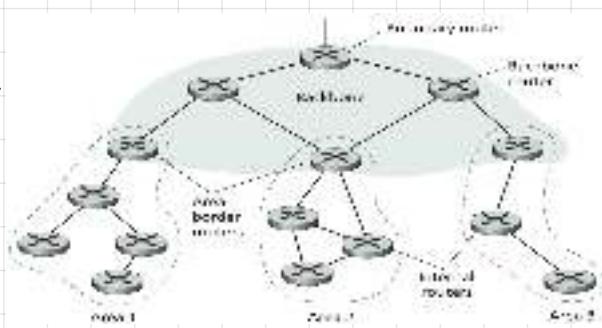
## 8.4 OSPF

È un protocollo link-state con supporto per il routing gerarchico. Usa una metrica generica configurabile dall'admin. Usa un sottoprotocollo di Hello per neighbour-discovery. Supporta anche il load-balancing (multiple rotte eguali) ed autenticazione. È trasportato da ip (protocollo 89) e implementa un meccanismo di ACK.

OSPF consente la creazione di aree. Un'area sarà sempre backbone ed è quella contenente i boundary routers. Ogni area comunica con il backbone attraverso gli area-border routers.

L'osso supporta diversi tipi di reti:

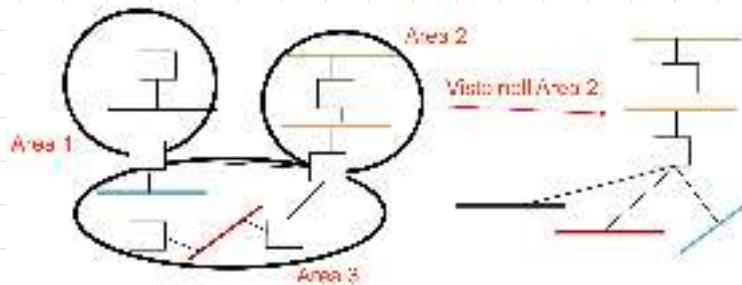
- 1) collegamenti PTP
- 2) reti con singolo router
- 3) reti con più router



Le rute 2 e 3 conterranno oramai 1 router designato che si occupa di mandare pacchetti link-state sulla rete, riducendo il numero di pacchetti mandati.

OSPF può mandare diversi tipi di pacchetti:

- Hello: pacchetti di Hello
- Database description: raccolto dell'intero DB di rete (utile per copiare il DB da un router ad un altro appena con)
- Link-state request: richiesta di informazioni per una rotta
- Link-state update: messaggi link-state
  - Router link
  - Network link
  - Summary link to network: mandato dai border-router nelle loro rute per comunicare le destinazioni esterne. Escludo "summary" contiene solo le destinazioni e niente sulla topologia
  - Summary link to AS boundary router: mandato tra border-router e contiene le destinazioni delle proprie aree.
  - External link
- Link-state ACK: messaggio di ACK



Esempio su come funziona la suddivisione in aree

• • •

### 8.5 BGP

Il più diffuso EGP. Il routing tra AS è molto diverso dal routing interno:

- i criteri di scelta del percorso sono difficilmente traducibili in metriche
- i gestori AS hanno bisogno di poter scegliere il percorso in base alla loro politica
- modello sulla rete può essere fatto conoscendo l'intero percorso

Perciò sia DV che LS non sono adatti a questo tipo di routing in quanto: il DV non fornisce informazioni sul percorso e il LS fornisce informazioni scambiate sulla topologia.

Per soluzioone è il PATH VECTOR. Esso è un DV modificato: non contiene la distanza dal destinazione ma bensì l'intero percorso verso la destinazione. Possono essere raccolte altre informazioni.

In BGP, i PV da scambiare contengono attributi. Esistono attributi obbligatori (tutte le implementazioni BGP devono riconoscerli) e facoltativi. Tra gli attributi obbligatori abbiamo:

- ORIGIN: protocollo IGP di provenienza
- AS-PATH: sequenza di AS attraversati
- NEXT-HOP: prossimo router

Ogni router manda il proprio PV ai vicini tramite TCP (porta 179). I messaggi BGP scambiati sono:

- OPEN: apre la connessione e i router si autenticano
- UPDATE: annuncia / cancella una rete
- KEEP ALIVE: mantiene attiva la connessione in corso di UPDATE (visto come ACK per OPEN)
- NOTIFICATION: notifica errori in messaggi precedenti (può chiudere la connessione)

La scelta dell'intradamento viene lasciata all'amministratore di rete (policy based routing): un router BGP può scegliere se salvare o inviare ai vicini un path vector ricevuto.

Ad ogni AS è assegnato un AS number dalla IANA.

### 9 LIVELLO DI LINEA

È il primo livello logico nella modalità a pacchetti. La sua principale funzionalità è il framing: identificare logicamente gruppi di bit scambiati al livello fisico. Altre funzionalità sono:

- regolazione e correzione di errori
- multiplexing
- accesso multiplo

Il livello di linea è normalmente parte della scheda di rete (NIC). Di solito i componenti innesti al livello fisico in un chip dedicato (controller). Alcune funzionalità sono implementate in software degli host.

I collegamenti possono essere di 3 tipi

- PUNTO - PUNTO
- BROADCAST: es: WIFI
- COMMUTATO: vicinanza del P2P ma con altri elementi di rete locale

#### 9.1 COLLEGAMENTI P2P

La prima funzione da voi eseguita è il framing. Come separiamo le frame? In alcuni casi il livello fisico fornisce i limiti, in altri si usano delimitatori di frame.

Un esempio è l'HDL: la frame inizia con dei flag (01111110). Per evitare che la frame sia conclusa erroneamente da una sequenza di bit uguali ai flag si usa il bit stuffing: prima di trasmettere si aggiunge uno '0' dopo 5 '1' consecutivi; in ricezione vorremo rimuovere.

Il controllo d'errore di livello 2 è il recupero degli errori di livello fisico. Esiste anche nei collegamenti broadcast.

La multiplexione, nei P2P, avviene in collaborazione tra livello fisico e di linea: un canale viene diviso in sottocanali. Questa operazione viene detta multiplexione fisica. I canali vengono divisi in sotto-canali di capacità fisso. La divisione può avvenire in diversi modi: spazio, frequenza, tempo, codice e lunghezza d'onda.

- DIVISIONE SPAZIO: un esempio sono i canali che trasportano diverse fibre ottiche.
- DIVISIONE FREQ: si suddivide la banda passante del canale in diverse sotto-bande utilizzate dai vari sottocanali.
- DIVISIONE TEMPO: i flussi vengono raccolti in N code e trasmessi sul flusso d'uscita a grappi di K (intrecciamiento di K bit). Il periodo di "rotazione" viene chiamato frame (non è legata all'altra frame (PDU)). Il tempo di trasmissione in uscita deve essere perfettamente N·K volte più piccole di quella di ingresso. La frame deve, quindi, essere lunga tanto quanto ci impiega ad inviare K bit sul singolo canale.

$v$  = vel. entrata       $w$  = flussi in ingresso (tralasciati)

$w$  = vel. multiplex

$K$  = grado di intrecciamento

$T_f$  = durata frame       $T_s$  = durata slot

$$T_s = \frac{T_f}{N}$$

$$K = w \cdot T_s \Rightarrow T_f = N \cdot T_s = N \frac{K}{w} = \frac{K}{v}$$

$$v = \frac{K}{T_f} = \frac{w}{N}$$

### 9.2 COLLEGAMENTI BROADCAST

La funzione di rete, in passato, era molto onerosa portando le velocità a poche decine di Kbps. Le reti locali arrivavano anche ai Mb/s. Il trucco era evitare la funzione di rete e usare il broadcast: i processori erano più lenti dei link. Le reti broadcast sono usate ancora oggi (vedi WI-FI).

Esempi di tecnologie broadcast:

- ETHERNET (oggi non lo è più, è a commutazione)
- ALOHANET
- WI-FI
- PASSIVE OPTICAL NETWORKS

} anni '70  
} OGGI

## 9.2 CANALE BROADCAST

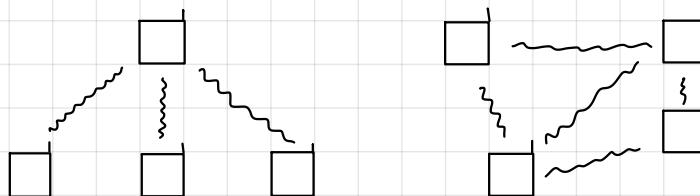
Il canale broadcast invia a tutti indiscriminatamente. I vari host decidono se ricevere o no la trasmessa in base ad un altro criterio. In un canale broadcast, inoltre, possono avvenire collisioni (contro di trasmissioni contemporanee). La funzione di accesso multiplo regola l'accesso contemporaneo alle risorse.

La funzione di accesso multiplo può essere effettuata:

- livello fisico: divisione statica delle risorse tra gli host
- livello di protocollo di linea: un protocollo gestisce l'accesso pacchetto per pacchetto

### 9.2.1 ACCESSO MULTIPLO FISICO

È simile alla multiplexazione fisica ma è relativo al caso in cui diversi sottocanali sono gestiti da trasmettitori diversi. Esempio:



Per separare le trasmissioni viene usato il FDMA. Esso è equivalente allo FDM (mult. fisica divisa per freq.). Un esempio è il Wi-Fi e la rete cellulare.

Venne usato anche il TDMA (analogo del TDM). La situazione è, però, più complessa: vengono definiti degli slot temporali dedicati alla trasmissione di ogni emittente. Tra gli slot sono inseriti dei tempi di guardia per compensare il ritardo di propagazione intrinseco al nesso di trasmissione.

Il duplexing è la modalità con cui si ricevano due canali in senso opposto da un nesso. In alcuni casi si può trasmettere contemporaneamente in entrambe le direzioni (full duplex fisico). Nel caso in cui il canale non sia full-duplex, bisogna ricorrere a tecniche di divisione delle capacità.

### 9.2.2 ACCESSO MULTIPLO A LIVELLO DI LINEA (LOGICO)

Il controllo può essere effettuato centralmente o in modo distribuito. Il livello di linea è diviso in MAC (il "riga") e LLC (il resto).

Un meccanismo molto utilizzato è l'accesso casuale. Ogni trasmettitore trasmette in base a come "vede" il link. Se avviene una collisione, il trasmettitore ritrasmette dopo un tempo casuale. La casualità rende poco probabile una collisione. Funziona molto bene per reti poco trafficate. Questo metodo era usato da ALOHANET (protocollo ALOHA). Questo metodo presupone che una stazione sia in grado di accorgersi delle collisioni.

#### 9.2.2.1 ALOHA E SLOTTED ALOHA

Se consideriamo un'unione fra TDMA e ALOHA (slotted ALOHA) una stazione ha probabilità di trasmettere con successo:

$$P = (1-p)^{n-1}$$

N: numero canali

p: probabilità di trasmettere in uno slot

La probabilità di successo in uno slot arbitrario è  $P = p(1-p)^{n-1}$ . La probabilità che una qualsiasi stazione riesca è il throughput:

$$\mathcal{S} = Np(1-p)^{n-1}$$

Il numero medio di trasmissioni nel canale è  $G = Np$ . Duplicando nel throughput si ha l'efficienza del canale.

$$S = G \left(1 - \frac{G}{N}\right)^{N-1}$$

Tendendo ad infinito  $N$ , otteniamo  $S = Ge^{-G}$ . Questa funzione ha massimo in  $G=1$ , quindi per  $S \approx 0,37$ . Ciò significa che slotless aloha ha un'efficienza del 37%.

La variante non slotless di aloha è ancora meno efficiente:  $S \approx 0,17$ .

### 9.2.2.2 CSMA

Il protocollo CSMA adotta la politica "ascolta prima di parlare". Esso trannele solo se il canale è libero. Anche in questo caso le collisioni avvengono a causa del ritardo di propagazione che può portare a falsi positivi. Il tempo  $2\tau$  viene detto periodo di vulnerabilità in quanto è quello dove possono sorgere collisioni.

Il throughput è:

$$S = \frac{Ge^{-\alpha G}}{G(1+2\alpha) + e^{-\alpha G}}$$

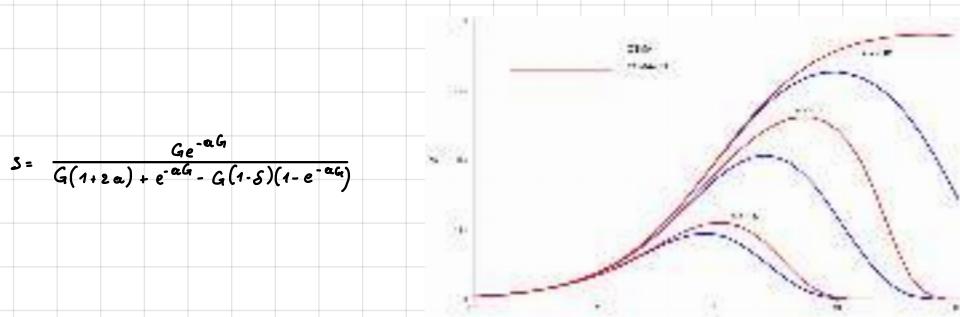
$$\alpha = \frac{\tau}{T}$$

$T$  Tempo di trasmissione.

L'efficienza è molto alta se  $\alpha \ll 1$ .

### 9.2.2.3 CSMA-CD

Una variante del CSMA introduce il meccanismo di Collision Detect: una stazione continua ad ascoltare il canale anche mentre trasmette per rilevare subito la collisione. La trasmissione viene subito interrotta (dopo un piccolo intervallo  $\delta$ ) per risparmiare tempo. Il meccanismo è usato da Ethernet. Il throughput sarà:

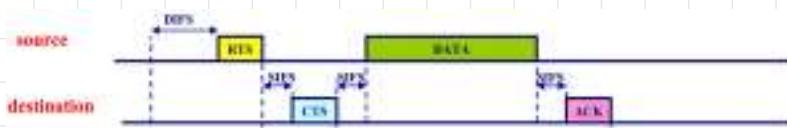


Studiando il grafico, si nota che CSMA-CD non porta numerosi vantaggi. Il vantaggio è l'assenza di un meccanismo di ACK necessario per ALOHA/CSMA che anch'esso può essere soggetto a collisione.

Il CD funziona solo se il trasmittitore rileva la collisione prima della fine della trasmissione. Il tempo di trasmissione minimo deve essere  $2\tau$ . Questo implica una velocità massima e una distanza massima.

Il CD, inoltre, si basa sulla bassa attenuazione di Ethernet che aiuta i trasmittitori a sentire le collisioni. Nelle reti radio, però, ciò non è possibile in quanto l'attenuazione del segnale è alta.

### 9.2.2.4 CSMA-CA



RTS e CTS sono chi "pacchettini" di servizio che regolano il traffico:

- RTS: ready to send

- CTS: clear to send

L'ACK torna solo quando è necessario.

Questo è il sistema utilizzato da WiFi. Esso può essere visto come un adattamento di Ethernet per reti radio.

## 9.3 TECNOLOGIE DI RETI LOCALI

La IEEE regola la standardizzazione delle reti locali. Differenti tecnologie condividono lo IEEE (standard 802.x). Il MAC e gli altri livelli sono diversi.

### 9.3.1 ETHERNET

È lo standard IEEE 802.3 wa. Il segnale era inizialmente su cavo coassiale passivo. Le stazioni si connettono con dei trascinitori ad un singolo BUS. I cavi sono classificati come XBASET:

- x: bit rate in Mb/s
- BASE: trasmetto in banda base
- y: lunghezza massima (m x 100 (km))

Successivamente si è passati dalla configurazione a BUS con cavo coassiale ad una configurazione a stella con hub. Il segnale diventa di doppio in rame (twisted pairs) (10 BASE T)

Il Fast Ethernet è l'Ethernet a 100 Mb/s. Si misura, inoltre, ad usare la fibra ottica (10 BASE FX). Con F.E. inoltre l'interconnessione avviene tramite switch e non più hub.

Oggi si è arrivati al Gigabit Ethernet che permette velocità fino a 100 Gb/s.

Quando Ethernet è nato, non era diviso in MAC/LLC ma era un unico livello.

Le trame Ethernet hanno forma:



- **Sync:** sincronizzazione livello fisico
- **Indirizzi:** indirizzi di 48 bit definiti dal produttore (nic)
- **Type:** usato per multiplexazione
- **Data:** campo per payload (max 1500 byte, min 46 byte)
- **Fcs:** trama check sequence per controllo d'errore.

Gli indirizzi sono chiamati MAC. I primi 3 byte identificano il destinatario, gli ultimi 3 identificano la sorgente. L'indirizzo con tutti i byte a 1 è l'indirizzo broadcast.

### 3.3.2 WiFi

La tecnologia è identificata dal standard IEEE 802.11. Può funzionare in 2 modalità:

- **AP:** un terminale comunica con un access point
- **ad-hoc:** i terminali comunicano direttamente tra loro

La modalità AP prevede la presenza di un Distribution System collegato da vari AP. Un'area ricoperta da un AP è il Basic Service Set. Ogni BSS ha un id (SSID) con cui si annuncia.

Ogni trama WiFi ha fino a 4 indirizzi diversi usati nelle varie modalità. Gli indirizzi sono anche utilizzati per identificare gli AP.

### 3.4 LAN COMMUTATE

Il modo switch (o bridge) è un modo di commutazione. Lo switch è ciò che differenzia le lan commutate dal resto.

La differenza del hub, che è un semplice ripetitore fisico, lo switch opera a pacchetto: ogni schiaccia di rete effettua store-and-forward.

Lo switch ha funzioni di:

- **filtraggio:** se una trama ricevuta da LAN1 è indirizzata ad una sorgente di LAN1 è scartata
- **relay:** se una trama ricevuta da LAN1 è indirizzata ad uno sorgente di LAN2 essa viene forwardata.

Pur di filtrare e inoltrare viene compilata una tabella di imbradamento della FDB.

Cd ogni porta dello switch è connesso un dominio. Lo switch accierra sia l'ingresso verso il dominio di destinazione che il broadcast su tutti i domini (maudato su tutti i domini tranne quello sorgente). Lo switch segnala i domini di accesso multiplo in più domini broadcast, aumentando l'efficienza.  
(domini di collisione)

La tabella di switching è compilata automaticamente: la presenza dello switch è trasparente in quanto lo switch non ha MAC virtuale. Il metodo di apprendimento è semplice se la rete è ad albero:

- 1) La tabella viene inizializzata vuota
- 2) All'arrivo di un pacchetto l'indirizzo viene aggiunto.
- 3) All'ingresso se il MAC non è in tabella la trama viene broadcastata

Le righe della tabella hanno una durata limitata (di solito 300 s) rinfattata all'arrivo di un pacchetto.

Se la rete non è ad albero, il meccanismo sopra viola causando una Broadcast Storm: i pacchetti

vengono broadcastati all'infinito poiché uno switch riceve il broadcast dell'altro. La soluzione è lo spanning tree: gli switch disponibilano alcune porte per trasformare la topologia maggiore in un albero. Un protocollo distribuito permette agli switch di comunicare fra loro e calcolare tramite un algoritmo i rami da tagliare.

Usando la modalità full duplex dell'Ethernet è possibile creare reti completamente connesse, rimuovendo la parola di collision detection. I bridge/switch possono funzionare anche se vengono utilizzate tecnologie diverse.

Una LAN fisica può essere separata in più VLAN. Due VLAN appartenenti a domini broadcast diversi non possono comunicare a livello 2. Possono, però, comunicare a livello 3.

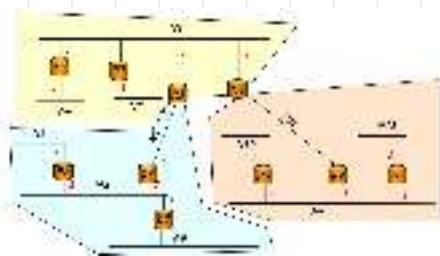
I vari domini di broadcast vengono separati fisicamente su base porta assegnando a ciascuna VLAN le sue porte (access ports).

I link di collegamento tra gli switch sono detti trunk e trasportano i dati di tutte le VLAN. Le trame di ogni VLAN sono separate tramite dei tag.

Anche nei link fra router e switch i pacchetti vengono taggati in base alla VLAN. I router useranno queste etichette per creare una interfaccia virtuale per ogni VLAN. I router, quindi, si comporteranno come se molte VLAN fossero LAN finali.

## ESERCITAZIONE

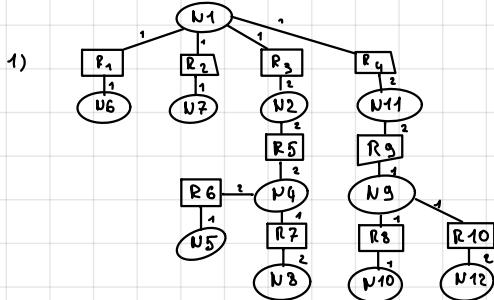
### ESERCIZIO 1



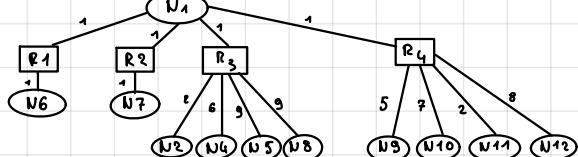
Usciamo OSPF.

1) Grafo nel caso delle sia una sola area.

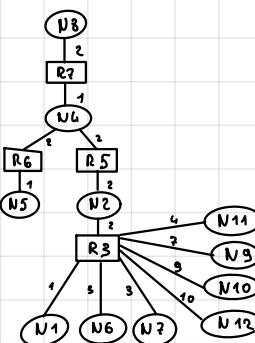
2) Considera le aree in figura. Rappresenta il grafo come visto da R1, R7 e R10



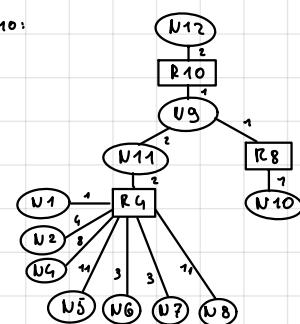
2) R1:



R7:



R10:



### ESERCIZIO 2

$$N = 10 \text{ slot}$$

$$W ?$$

$$K = 128 \text{ bit}$$

$$T_T ?$$

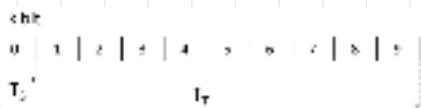
$$V = 64 \text{ Kb/s}$$

$$T_s ?$$

$$T_T = \frac{K}{V} = \dots = 2 \text{ ms}$$

$$W = V \cdot N = 64 \cdot 10 = 640 \text{ Kb/s}$$

$$T_s = \frac{T_T}{N} = 0,2 \text{ ms}$$



### ESERCIZIO 3

$$W = 2048 \text{ Mb/s}$$

$$N = ?$$

$$N = \frac{W}{V} = \dots = 32$$

$$K = 8 \text{ bit/slot}$$

$$T_T ?$$

$$T_T = \frac{K}{V} = \dots = 125 \mu\text{s}$$

$$V = 64 \text{ Kb/s}$$

$$T_s$$

$$T_s = \frac{T_T}{N} = \dots = 3,90 \mu\text{s}$$

### ESERCIZIO 4

$$N = 10$$

$$W ?$$

$$K = H + D = 200b$$

$$T_g = 200 \mu\text{s}$$

$$V ?$$

$$T_s = T_D + T_g = \frac{T_T}{N} = 1 \text{ ms} \rightarrow T_D = T_s - T_g = 0,8 \text{ ms}$$

$$D = 180 \text{ b}, H = 20b$$

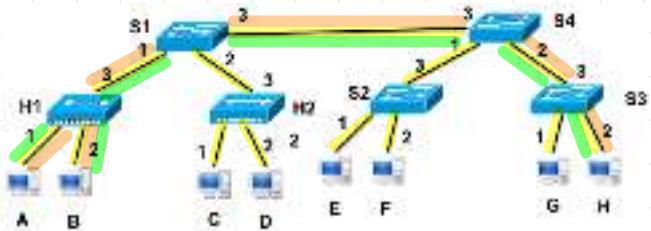
$$W = \frac{K}{T_D} = \frac{200b}{0,8 \text{ ms}} = 250 \text{ Kb/s}$$

$$T_T = 10 \text{ ms}$$

$$V = \frac{D}{T_T} = \dots = \frac{180b}{10 \text{ ms}} = 18 \text{ Kb/s}$$



ESERCIZIO 5



3 trame:

F1: A - BROADCAST

F2: H - A

F3: A - H

Tabelle pre-compilate

1) Come gli switch inviano i pacchetti? Cosa cambiano le tabelle? Seguono la tabella; Non cambia

2) Indovinare sorg./dest. dei pacchetti tra S1 e S2? Gli host che hanno sul link.

3) Cosa ricevono B e G? B: F1, F2, F3 ; G: F1