

Fondamenti di
Comunicazione e
Internet

Introduzione

Nelle pagine seguenti una sintesi delle dispense con arricchimenti del libro di testo di riferimento del corso.

Gli argomenti affrontati a lezione saranno:

- *Introduzione e architetture*
- *modelli funzionali*
- *sistemi di comunicazione*
- *protocolli applicativi*
- *livello di trasporto*
- *livello di networking*
- *inoltro ed instradamento*
- *reti locali e livello di linea*
- *intranet*

Mentre durante le ore di laboratorio (ed alcune di esercitazione) si approfondiranno le seguenti tematiche:

- *sniffer di rete e comandi (ping, traceroute, ecc.)*
- *python e scripting per analisi di rete*
- *programmazione socket in python*
- *configurazione e simulazione rete con Cisco Packet Tracer*

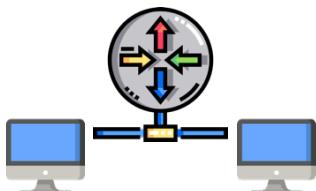
Nota: negli esercizi, o calcoli in generale, non usare le unità di misura della memoria; in questo corso 1Kbit/s non sono 1024bit/s, ma equivale a 10^3 bit/s.

Bibliografia

A. Pattavina, *Internet e Reti 3^aedizione*, Pearson Education.
P. Parolari, *slides delle lezioni*, a.a. 2023/2024.

Concetti di base

Internet è un'infrastruttura fisica con una precisa *architettura* ed un servizio di comunicazione che segue dei *protocolli*.



L'infrastruttura fisica è formata, essenzialmente, da:

- *nodi di rete*, come i router che operano sui pacchetti, ne esistono differenti tipi.
- *host*, dispositivi connessi alla rete in grado di ricevere/inviare informazioni.
- *link*, canali di comunicazione (in diversi materiali, come il rame, fibra, ecc.), differiscono anche per tecnologia e velocità di trasmissione.

Per accedere ad internet si possono utilizzare differenti modalità, di seguito le principali ed il loro funzionamento.

Dialup: accesso diretto al router dell'ISP (Internet Service Provider, e.g. TIM) tramite circuito telefonico, utilizzando il *modem* per trasmettere informazioni digitali in banda fonica (compresa tra 0 e 4KHz).

- Velocità: fino a 56kbps.

ADSL: l'Asymmetric Digital Subscriber Line prevede l'accesso al router ISP mediante reti ad alta velocità, condividendo il doppino con la rete telefonica fino alla centrale (divisione di frequenza).

- Velocità: fino a 1Mbps in upload e 20Mbps in download.

Fibra ottica: prevede una sostituzione totale o parziale del doppino con cavi in fibra ottica, né esistono differenti tipologie, in base a “dove” arriva il collegamento. I collegamenti sono “passivi”, gli unici elementi attivi nella rete attuale sono i fotodiodi per tradurre i fotoni in segnali elettrici e viceversa.

- Fiber To The Home/Basement/Curb/Neighborhood (FTTH, FTTB, FTTC, FTTN).
- Velocità: nell'ordine dei Gbps, a seconda del tipo.

Rete cellulare: come mezzo trasmissivo viene utilizzato l'etere, dove le informazioni sono trasmesse all'ISP tramite appositi ripetitori detti *celle*, né esistono differenti tipologie con diverse velocità.

- Velocità: dai 200kbps (GPRS/EDGE) fino ai 20Gbps in download e 10Gbps in upload (5G).

Rete Wireless LAN: prevede l'accesso radio condiviso (tra terminali e router) attraverso un dispositivo detto *Access Point* o stazione base. Lo standard di comunicazione utilizzato è il 802.11b/g/nY/ac (WiFi).

- Velocità: 11/54/300/1000 Mbps.

È da tener presente che internet non è formato da un solo tipo di rete, piuttosto è un insieme eterogeneo di reti interconnesse. Ciò consente di connettere ulteriori reti sfruttando l'Internet Protocol; si avranno tante reti gestite ciascuna da operatori indipendenti (gli ISP).

Nota: un ulteriore tipologia di rete è la Ethernet.

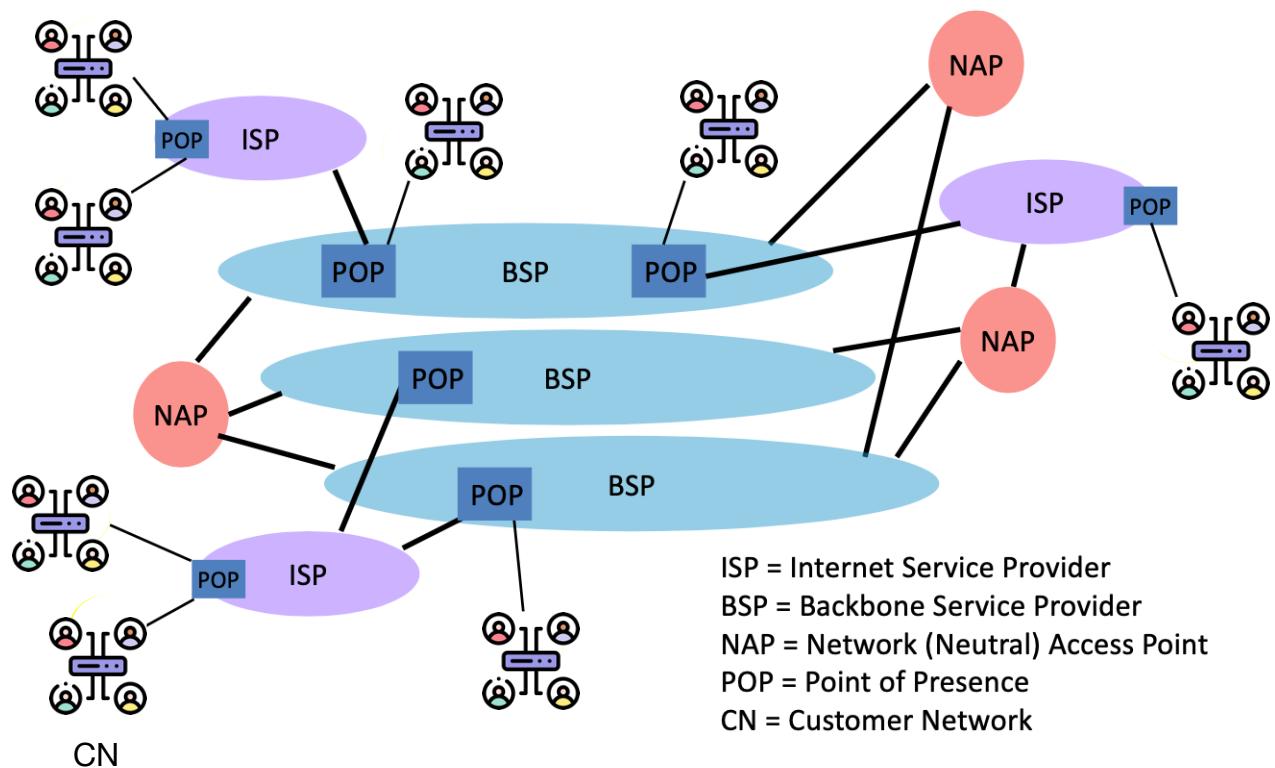
Una distinzione delle reti può essere fatta per base geografica, che vede le:

- LAN, o local area network, per aree limitate come edifici e campus.
- MAN, o metropolitan area network, collega quartieri e città (copre decine di km).
- WAN, o wide area network, hanno coperture illimitate, solitamente nazioni, ecc.

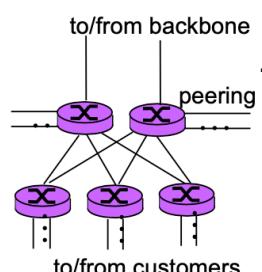
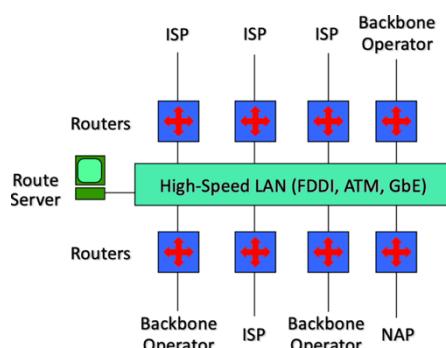
Gli ISP invece sono specializzati per specifici segmenti di reti, nello specifico:

- *Tier-3*, sono gli ISP locali, per gli utenti finali.
- *Tier-2*, sono ISP di estensione regionale/nazionale.
- *Tier-1*, hanno come clienti solo altri ISP, operano WAN intercontinentali (*backbone*).

Due ISP dello stesso tipo si scambiano traffico attraverso il *Peering Link*. Uno schema esemplificativo di architettura di interconnessione può essere il seguente.



Il POP, o Point Of Presence, può essere schematizzato così:
Mentre il modello sottostante rappresenta il NAP.



Content Distribution Network: o CDN, grandi reti private (amazon, google, ecc.) che collegano i loro datacenter con le reti pubbliche. All'interno di questa rete non si "pagano" i servizi degli ISP, poiché gestite dalle stesse aziende; questo permette di avvicinarsi agli utenti finali senza pagare tutti i servizi degli ISP.

Servizi di comunicazione: è fornito dalla rete alle applicazioni per il trasporto delle informazioni tra processi remoti, può essere di vari tipi in funzione della qualità di servizio (QoS) richiesta:

- Si usa un servizio di comunicazione non affidabile di messaggi corti, ad esempio, nel DNS e nello streaming.
- È preferito un servizio di comunicazione affidabile per sequenze anche lunghe di dati per la messaggistica (e-mail) oppure per il web/file transfer.

Le applicazioni richiedono alla rete il servizio più appropriato grazie ad apposite API (interfacce di programmazione).

Nota: parte del laboratorio si concentrerà proprio su quest'ultime, utilizzando il linguaggio Python.

Protocolli di comunicazione: insieme di regole per la formattazione, invio e ricezione dei messaggi, diversi servizi avranno differenti target di qualità (QoS, di ritardo, affidabilità, bit rate). Quelli utilizzati per la richiesta di connessione in internet sono di tipo *handshake*, dove viene richiesta la connessione dal client al server, il primo attenderà la risposta del server; se sarà accettata, inizierà la vera e propria comunicazione. Esistono due modelli di protocolli di comunicazione:

- *Client/Server*, client chiede servizio e server lo fornisce.
- *Peer-to-Peer*, tutti i client collaborano.

È bene osservare che la rete non fa distinzione tra i modelli, almeno nel modo in cui vengono trasportati i dati.

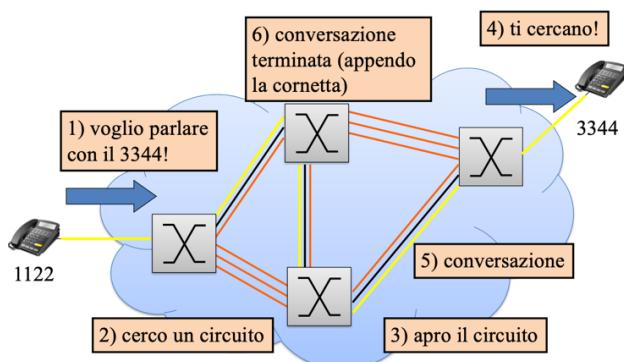
Autonomous Systems: reti *autonome* molto grandi, possono essere modellizzate come ISP, hanno un elevato numero di peering (collegamenti) permettendo con pochi collegamenti di raggiungere qualunque host. Nell'internet attuale, si può raggiungere un qualunque host, per un massimo di 10 gradi di separazione.

La Commutazione

È la modalità di trasferimento dell'informazione attraverso una rete, né esistono due tipologie: *commutazione di circuito* e *di pacchetto*.

Osservazione: le due tipologie di commutazioni sono rispettivamente *connection-oriented* e *connectionless*.

Commutazione di circuito: le risorse per la comunicazione sono riservate per la chiamata (e.g. rete telefonica) il cui funzionamento è espresso dal seguente schema.

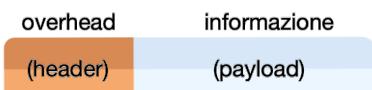


I collegamenti¹ sono suddivisi in "fette", ciascuna delle quali viene allocata ad un flusso continuo di informazioni, detto *circuito*. Le risorse rimangono inattive se non utilizzate. La risorsa trasmissiva sarà suddivisa (*multiplazione fisica*) per:

- Divisione di frequenza
- Divisione di tempo
- Divisione di lunghezza d'onda

Posso modellizzare il nodo come un dispositivo in cui la capacità dei canali di ingresso è pari a quella dei canali in uscita; inoltre, una volta che il circuito si è instaurato non è necessario memorizzare l'informazione.

Commutazione di pacchetto²: questo sistema di comunicazione è basato sulla suddivisione dell'informazione in parti più piccole dette *pacchetti*.



Data una rappresentazione della struttura di base di un pacchetto, sarà possibile descriverne il funzionamento. Nell'header è contenuto l'indirizzo del dispositivo di destinazione, che sarà letto dai router e inoltrato al router successivo, un procedimento che si ripete fino al destinatario che leggerà il contenuto del payload.

Il router sa' come instradare i pacchetti grazie alla tabella di routing (contiene indirizzi raggiungibili). In questo modo, i pacchetti di tutti gli utenti condividono le risorse di rete (utilizzano completamente canale per un certo intervallo t); pertanto tali risorse vengono utilizzate a seconda delle necessità, portando una maggiore flessibilità.

Il nodo è rappresentato da un *packet Router* o *Switch*, che avrà capacità dei collegamenti arbitraria, conflitti temporali per trasmissione. Proprio per evitare/risolvere questi ultimi, si memorizza temporaneamente una coda di pacchetti:

- All'ingresso, per analizzare l'indirizzo di destinazione.
- All'uscita, per gestire le contese.

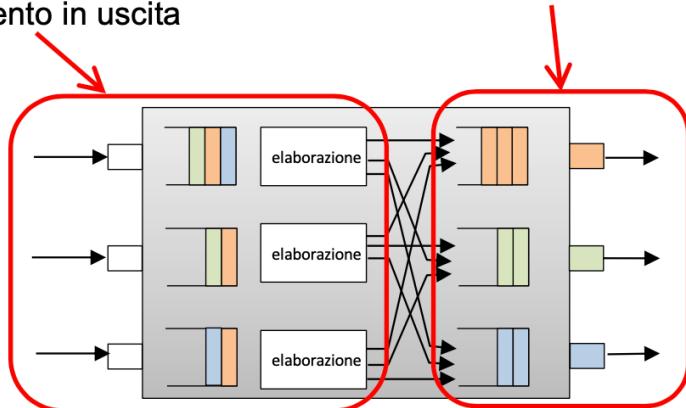
¹ Per collegamenti si intendono anche le risorse di rete

² È di tipo *best-effort*, cioè si suppone che la rete faccia del suo meglio per consegnare i pacchetti, senza effettivamente sapere se siano arrivati o meno (*connection-less*); si può avere comunicazione sicura con alcuni protocolli.

Di seguito uno schema che rappresenta la “contesa” per le risorse.

Store and forward (S&F): il commutatore deve ricevere l'intero pacchetto prima di poter cominciare a trasmettere sul collegamento in uscita

Multiplazione statistica: accodamento dei pacchetti, attesa per l'utilizzo del collegamento



I problemi derivati dalla multiplazione statistica riguardano il ritardo di accodamento e la perdita di UI³, dovuti all'arrivo asincrono dei pacchetti alle code d'uscita.

Rispetto alla commutazione di circuito, il sistema a pacchetti consente di sostenere comunicazioni con più utenti senza “perdita” di velocità, anche perché è improbabile che trasmetteranno tutti assieme nello stesso momento. Se ben dimensionata, la commutazione di pacchetto riduce anche i ritardi (rispetto quella di circuito). Questa maggiore efficienza porta ai problemi (introdotti dalle code) precedentemente descritti, come la perdita di pacchetti; pertanto, vi è l'esigenza di utilizzare protocolli per il trasferimento affidabile dei dati e per il controllo della congestione.

Esistono due tipologie di commutazione di pacchetto, analizzeremo quindi la *datagram* e *circuito virtuale*.

Datagram: commutazione dove la scelta della porta d'uscita viene operata sulla base del solo indirizzo IP di destinazione; i pacchetti dello stesso flusso (d'informazioni) sono inoltrati indipendentemente. La commutazione a pacchetti nasce con questa modalità.

Circuito virtuale: commutazione dove i nodi identificano i pacchetti di un flusso informativo sulla base di un identificativo di circuito virtuale (VCI, Label); tale circuito viene instaurato in una fase di *setup*, dopodiché tutti i pacchetti seguono lo stesso percorso, instradati sulla base del VCI. Non viene riservata la risorsa durante la comunicazione.

Confronto: in breve, la commutazione di circuito prevede le risorse trasmissive prefissate per ogni utente, operazione che prende il nome di multiplazione deterministica ed è realizzata al livello fisico. Mentre in internet si gestiscono pacchetti trasmessi tra coppie di terminali, la commutazione di pacchetto sfrutta la multiplazione statistica ⁴(flessibile rispetto alla deterministica).

³ Sono le unità di informazione (i pacchetti inviati).

⁴ Probabilità calcolabile con una binomiale, e.g. con 35 utenti la probabilità di averne più di 10 attivi contemporaneamente è inferiore allo 0,004%.

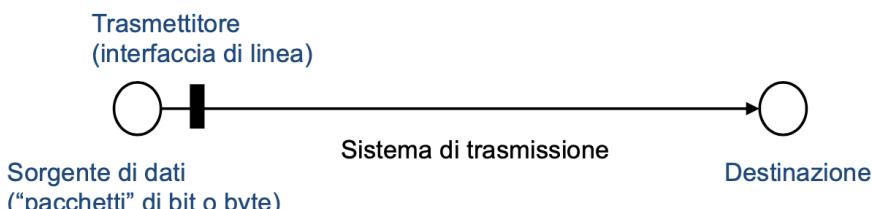
Sorgenti, servizi e prestazioni

I servizi di comunicazione possono essere monomediali o multimediali (voce, dati, immagini, ecc.) e devono avere dei requisiti di prestazione QoS (o Quality of Service). Mentre le sorgenti in grado di produrre l'informazione sono digitali (utilizzata in internet) o analogiche.

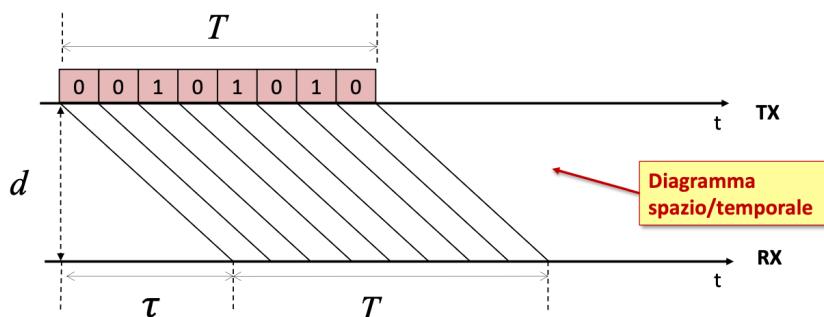
I servizi possono avere diverse configurazioni (punto-punto, multipunto, diffusivo o broadcast) e direzioni (bidirezionale simmetrico/asimmetrico, unidirezionale) del flusso informativo.

La qualità del servizio è garantita quando le prestazioni offerte dalla rete soddisfano i requisiti di prestazione, sarà il progettista della rete a calcolare i parametri.

Parametri di prestazione: (*ritardo, throughput, perdita*) sono offerti dalla rete, prendendo come modello il collegamento punto-punto, allora posso analizzare le velocità/ritardi di *trasmissione, propagazione e ricezione*.



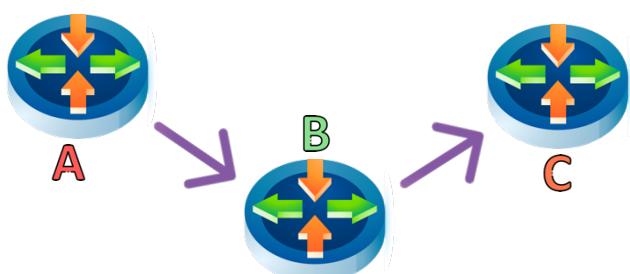
- *Velocità di trasmissione*, velocità con la quale trasmetto informazioni su una linea, l'unità di misura è bit/s.
- *Tempo di trasmissione*⁵, tempo necessario a trasmettere n-bit, dipende dalla velocità.
- *Ritardo di propagazione*⁶, tempo τ necessario affinché un impulso generato da TX raggiunga RX, dipende dalla distanza e velocità di propagazione della luce nel mezzo fisico.
- *Tempi di attraversamento del canale*, intervallo di tempo che intercorre fra la trasmissione del primo bit e la ricezione dell'ultimo, con $T_{totale} = T + \tau$.



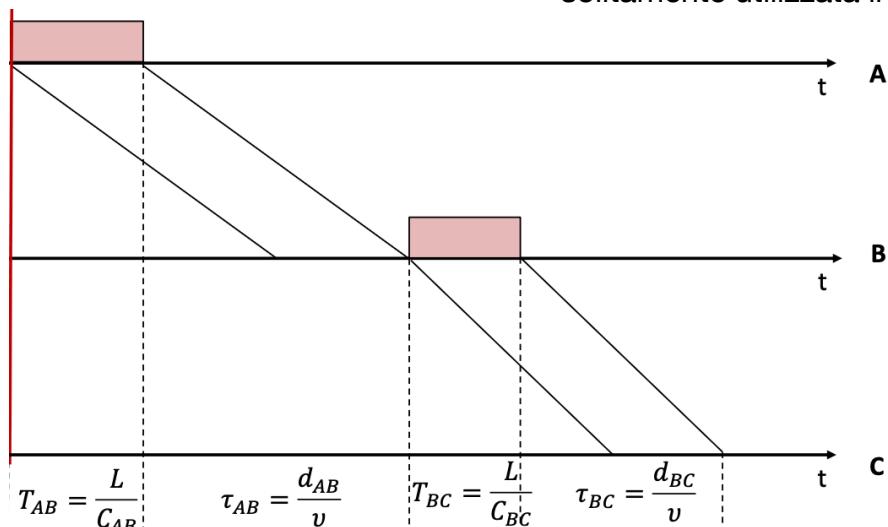
⁵ Si calcola come $T = \frac{n}{v}$, con n numero di bit e v velocità di trasmissione

⁶ Si calcola come $\tau = \frac{d}{v}$, con d distanza (da TX a RX) e v velocità di propagazione della luce nel mezzo fisico

Store and forward: è una tecnica (implementata nella commutazione di pacchetto) utilizzata nei nodi della rete (router) per la trasmissione dei dati, dove il pacchetto verrà inoltrato solamente una volta che sarà stato completamente ricevuto dal router.

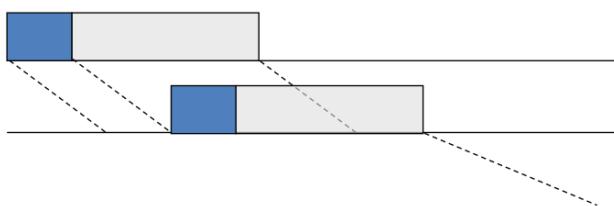


L'impiego di tale tecnica porta ad una maggior correttezza del contenuto informativo, grazie alle capacità dei router di riuscire a controllare la presenza di eventuali errori (e correggerli). Questa trasmissione "sicura" dei dati riduce la velocità della trasmissione. È la modalità solitamente utilizzata in internet.

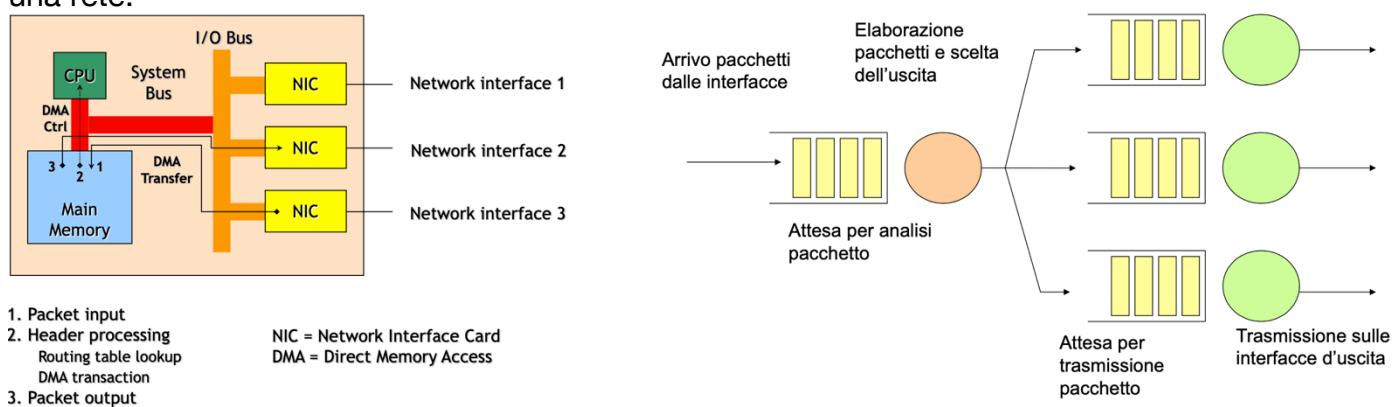


Osservazione: non sempre è fondamentale che i dati trasmessi siano "completi"; in alcune applicazioni la velocità di trasmissione è più importante, come nel gaming online o nel VoIP.

Cut-through: altra tecnica di trasmissione dei dati (usata nella commutazione a pacchetti) che prevede l'inoltro del pacchetto alla completa ricezione dell'header, può portare ad una perdita di informazioni, ma a beneficio di una maggiore velocità. È utilizzato solamente da alcuni protocolli di linea o data link.



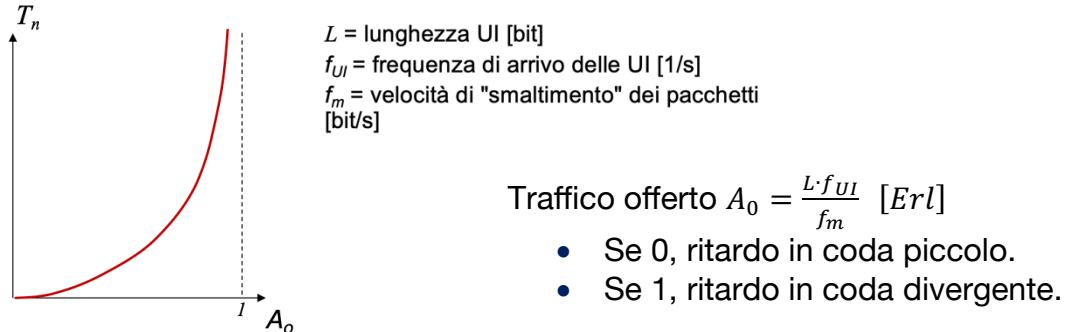
Di seguito l'architettura semplificata ed il modello di funzionamento di un nodo/router in una rete.



Ritardo di elaborazione: indicato con T_P , è un ulteriore parametro che va' tardare l'inoltro di un pacchetto, necessario ai nodi per elaborare un header e trasferire i dati all'uscita corretta; negli apparati ben dimensionati è pari a 0, solitamente trascurabile.

Ritardo di accodamento: tempo che ritarda l'inoltro di un pacchetto nel caso in cui la linea di uscita sia già occupata, pertanto si attenderà che la NI si liberi. Interfacce diverse hanno code di uscita separate e indipendenti; questo ritardo dipende dalla multiplazione statistica, dovuta all'arrivo asincrono dei pacchetti alle code d'uscita.

Dal ritardo di accodamento medio si hanno i seguenti modelli statistici⁷:



Poiché le code hanno dimensione limitata, in una situazione di congestione (coda piena), i pacchetti che arrivano vengono scartati: questi pacchetti “persi” potranno poi essere ritrasmessi a seconda del protocollo utilizzato.

Ritardo End – To – End: è il tempo necessario per instaurare la connessione da TX a RX, è diverso per:

- Comutazione di circuito, dove questo si aggiunge al ritardo totale per il trasferimento, è l'intervallo di tempo tra apertura connessione ad arrivo dell'ultimo bit a destinazione.
- Comutazione di pacchetto, dove non è presente il tempo di instaurazione della connessione, in ogni *hop* il ritardo è dovuto unicamente a *propagazione, trasmissione, elaborazione e accodamento*.
 - Vale anche in presenza di frammentazione, dove un pacchetto viene diviso se è troppo grande per passare sulla rete di destinazione/successiva, oppure per velocizzare la trasmissione (entro un certo limite).

Throughput: quantità di informazione trasportata con successo da una rete (o parte di essa) nell'unità di tempo⁸.

Traffico: simile al THR, dato un collegamento di capacità C, è una misura di efficienza che si calcola come $A_s = \frac{THR}{C}$; misurato in Erlang, assume valori nell'intervallo [0, 1].

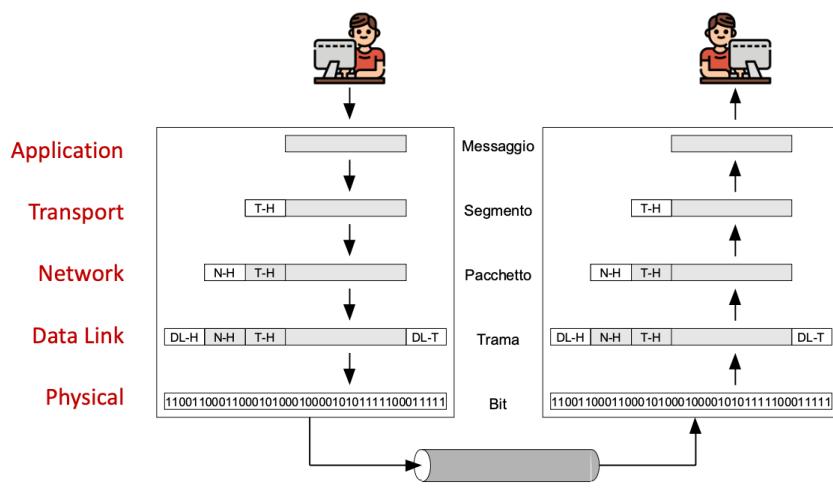
⁷ Basati sulla Teoria delle Code.

⁸ Si indica con THR, misurato in bit/s.

Perdita: indice della quantità di informazione persa durante la trasmissione, si riferisce a diverse entità, come i pacchetti o flussi di bit (in questo caso è detta *bit error rate*); determinata da ricezioni di pacchetti corrotti (errori di trasmissione) a causa del rumore, oppure dalla saturazione dei buffer nel percorso end – to – end. Si esprime come una probabilità, mentre si ottiene dalla seguente formula $\pi = \frac{bit_{lost}}{bit_{total}}$.

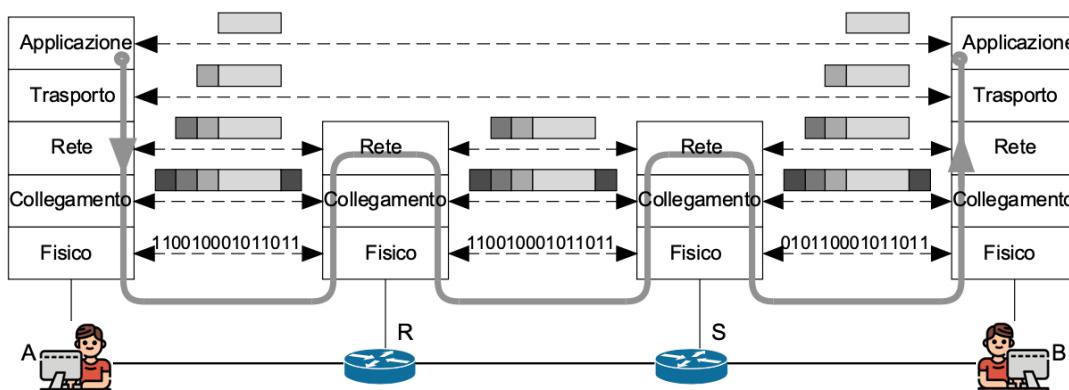
Modelli dei servizi e pile protocollari

L'architettura utilizzata è detta a *strati*, di seguito uno schema per illustrare come varia il pacchetto di informazioni tra uno strato, o livello, e l'altro.



Dal modello a sinistra, si può vedere come la dimensione dell'header del pacchetto cresca all'abbassarsi del livello, e viceversa.

Per capire come viaggiano i pacchetti nella rete, seguendo i protocolli illustrati nel precedente modello, sarà sufficiente inserire dei nodi al posto del cilindro, come di seguito.



I protocolli che operano nei rispettivi livelli sono: questa struttura consente ai vari livelli di evolvere in maniera piuttosto indipendente, senza intaccare il corretto funzionamento della rete.

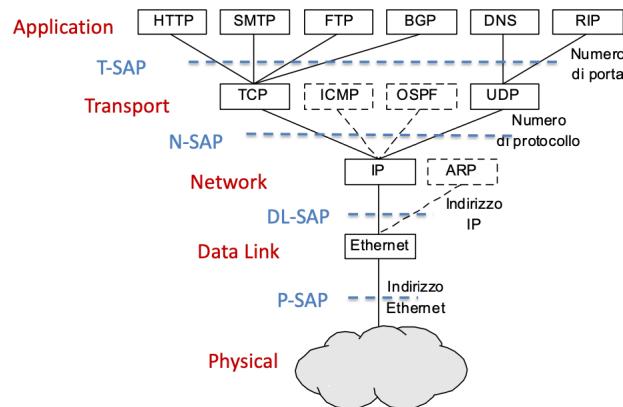
Applicazione	HTTP	SMTP	FTP	BGP	DNS	RIP		
	TCP				UDP			
	ICMP	OSPF						
	IP							
Collegamento	PPP	ARP		FDDI	Ethernet/Wi-Fi			

Ognuno dei livelli protocollari aggiunge valore ai servizi forniti dallo strato inferiore, fornendoli a quello superiore; tali servizi sono forniti indipendentemente dalle procedure realizzative, i cambiamenti in un livello sono trasparenti agli altri livelli. Ciascuno strato può svolgere diverse funzioni in base al tipo di servizio richiesto dal livello superiore, come:

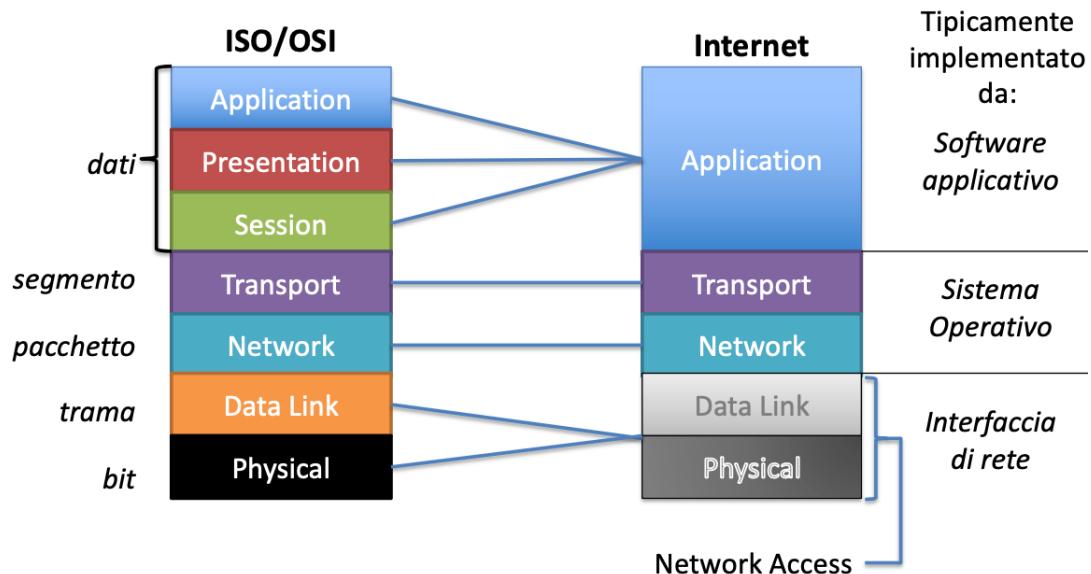
- Frammentazione/ricostruzione pacchetto
- Multiplazione
- Controllo errore
- Instradamento

- Riordino trame (di pacchetti)
- Funzioni di arricchimento e/o adattamento

Ogni strato ha un sistema di indirizzamento, per consentire l'identificazione di un Service Access Point.



Modello ISO/OSI: è una pila protocollare, un modello di riferimento per il funzionamento della trasmissione dei pacchetti, quello utilizzato realmente in internet è una sua semplificazione, di seguito uno schema che confronta le due modellizzazioni.



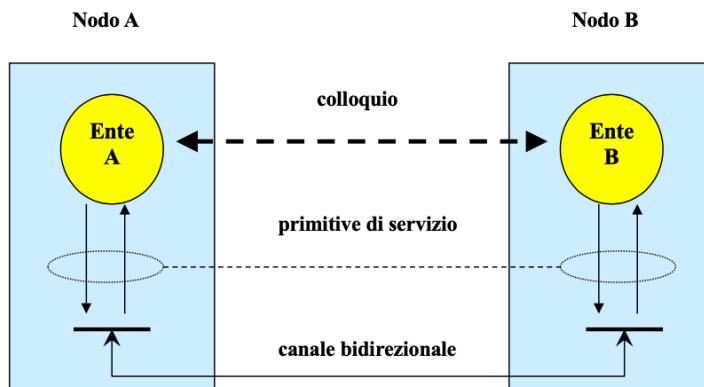
Modelli funzionali di servizio

Date due o più entità remote (host) che intendono scambiare informazioni tra loro, allora possiamo descrivere il servizio di comunicazione come un *fornitore del servizio di trasporto dell'informazione*. È un concetto generale applicabile a diverse situazioni, solitamente si usa come esempio il servizio postale ed il suo corrispettivo digitale, il servizio e-mail.



Servizio di comunicazione: gestisce lo scambio di informazioni fra due entità, è un servizio di trasferimento di UI; può essere descritto mediante le chiamate di servizio, dette *primitive di servizio*.

Primitive di servizio: necessarie non solo per descrivere il servizio, ma anche per richiederlo e ricevere informazioni sullo stesso dal fornitore. Inoltre, sono caratterizzate da parametri come, ad esempio, dati da trasferire, indicazioni del destinatario, caratteristiche del servizio, ecc.



Il servizio di comunicazione può essere sia *connection oriented* che *connectionless*, rispettivamente composti da 3 fasi (instaurazione, trasferimento, rilascio) ed una sola fase di connessione. Due entità che colloquiano tra loro sono dette *di pari livello*.

Protocollo: insieme di regole che gestiscono il *colloquio* tra entità, cioè che regolamentano la comunicazione; in particolare stabiliscono il formato dei messaggi, informazioni di servizio e che algoritmi di trasferimento da utilizzare.

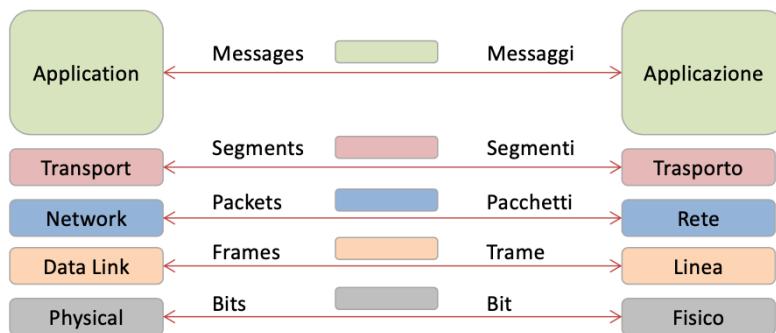
Packet Data Units: o PDU, sono le unità di trasferimento dati utilizzate dai protocolli; riprendendo lo schema di base dei pacchetti, l'header contiene le informazioni di servizio mentre l'informazione da trasmettere è contenuta nel payload.

Le entità che colloquiano in un servizio di comunicazione possono offrire quest'ultimo ad entità terze *di livello superiore*. Le entità di un livello collaborano con quelle "superiori" scambiandosi messaggi grazie al servizio offerto dal livello inferiore: il servizio di comunicazione del livello superiore è più complesso grazie alle *funzioni* implementate dal livello inferiore.

Architettura a livelli⁹: i servizi di comunicazione complessi possono essere articolati a livelli, partendo da uno specializzato al solo trasferimento di bit, fino ad uno dove sono definiti servizi complessi con diverse funzionalità.

- Questa tipologia di architettura agevola l'identificazione dei servizi, la gestione del servizio e l'aggiornamento.
- Ogni livello svolge una o più funzioni, alcune di queste sono le funzioni di multiplazione/demultiplazione, controllo d'errore e instradamento.

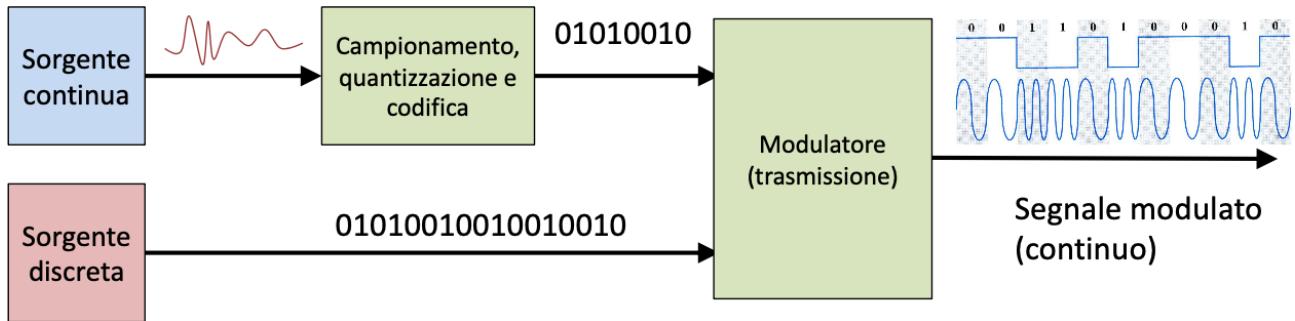
TCP/IP: è un altro modello di pila protocollare, una semplificazione dell'ISO/OSI, di seguito uno schema che ne illustra il funzionamento.



⁹ Si veda, ad esempio, la pila protocollare ISO/OSI caratterizzata da un architettura a livelli.

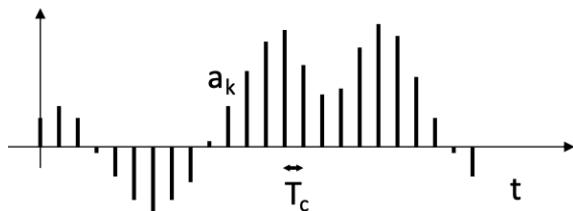
Physical layer – trattamento segnali

I segnali si distinguono in *fisici* con sorgenti continue e associate a grandezze fisiche continue nel tempo/ampiezze, oppure *logici* (o discreti) con sorgenti numeriche e discrete nel tempo/ampiezze.



I segnali si possono rappresentare come funzioni del tempo, in particolare potremo avere:

- *segnali continui*, $s(t)$ con $t \geq 0$
- *segnali discreti*, $s(t) = \sum_{k=0}^{\infty} a_k \cdot \delta(t - kT_c)$ con $t \geq 0$ e $\delta(x) = \begin{cases} 1 & \text{per } x = 0 \\ 0 & \text{per } x \neq 0 \end{cases}$



- *segnali periodici*, $s(t + T) = s(t)$ con $t \geq 0$, i segnali reali spesso sono lievi variazioni di segnali periodici, la loro analisi è più semplice rispetto ai non periodici.

Analisi di Fourier: consente di studiare un segnale scomponendolo in sinusoidi, dove quelle costituenti sono dette *armoniche* o *componenti spettrali*¹⁰. Ad esempio, i segnali periodici di periodo T possono essere scomposti in un numero discreto di sinusoidi (serie di Fourier) di frequenza multipla della *frequenza/armonica fondamentale* $f_0 = \frac{1}{T}$.¹¹



Qualunque $s(t)$ periodico nel dominio del tempo è del tutto equivalente alla somma delle armoniche sinusoidali, ognuna con la propria ampiezza e frequenza. Mentre $s(t)$ variabile nel tempo può essere rappresentato nel dominio delle frequenze dalle sue componenti in frequenza che costituiscono lo *spettro del segnale S(f)*.

¹⁰ ciascuna armonica ha la propria ampiezza $s_n = \sqrt{a_n^2 + b_n^2}$; è presente anche una componente continua a_0 .

¹¹ $s(t) = a_0 + 2 \cdot \sum_{n=1}^{\infty} [a_n \cos(2\pi n f_0 t) + b_n \sin(2\pi n f_0 t)]$

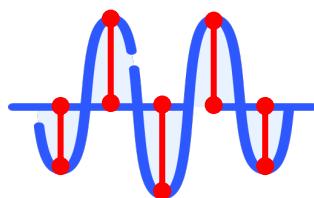
Trasformata di Fourier: generalizza l'analisi delle serie di Fourier al caso dei segnali non periodici, dove riesce a scomporre questi segnali in un insieme continuo di armoniche; ogni componente è generalmente moltiplicata per un coefficiente complesso che determina ampiezza e fase della sinusode.

- $X(f)$ è la funzione che descrive ampiezze e fasi delle sinusoidi componenti ed è lo spettro *in frequenza* di $x(t)$.

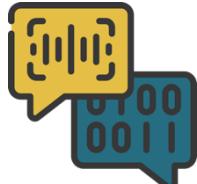
Banda: è l'insieme/intervallo continuo delle armoniche che costituiscono il segnale.

- *Stretta*, per i segnali che variano lentamente
- *Larga*, per i segnali che variano velocemente

Campionamento: operazione di misura dell'ampiezza del segnale in specifici istanti di tempo equispaziati tra loro.



Teorema del Campionamento (Nyquist): un segnale tempo variante è completamente determinato dai suoi campioni presi a distanza T_c tale che $T_c < \frac{1}{2f_{max}}$ con f_{max} = frequenza massima nello spettro del segnale. Posso quindi definire la *minima frequenza di campionamento* f_n ¹² come $f_c > f_n = 2 \cdot f_{max}$.



I campioni presi alla frequenza di Nyquist rappresentano il contenuto informativo (la banda B) del segnale, prendere troppi campioni non è necessario; è importante non scegliere una frequenza minore ad F_N altrimenti si avrà una perdita di informazioni.

Aliasing: fenomeno dovuto al sotto-campionamento di un segnale analogico, dove la bassa frequenza di campionamento porterà ad avere un segnale campionato che non rappresenterà più il segnale di partenza.

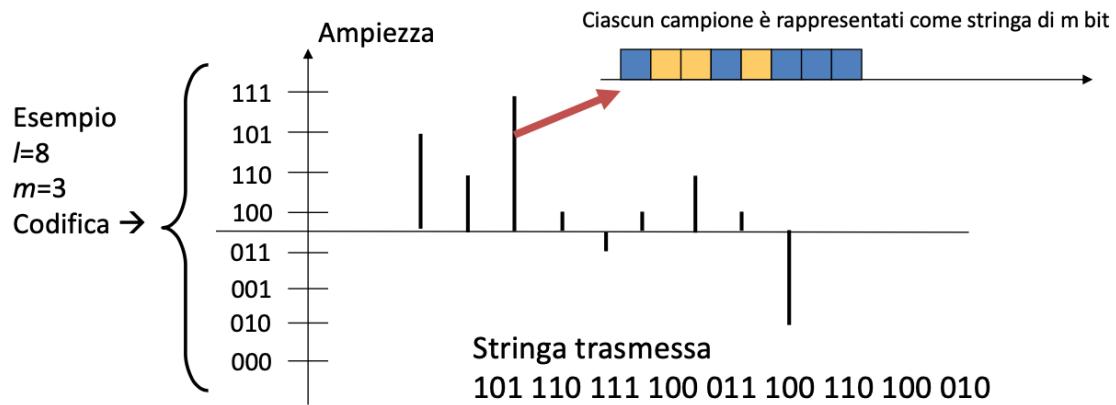
- Se la frequenza di campionamento è esattamente uguale a $2B$ (o F_N) potrei confondere la sinusode con la componente continua se il campionamento avviene negli istanti di attraversamento degli zeri.

Ricostruzione: a patto che sia stato rispettato il teorema del campionamento, allora a partire dai campioni potrà essere ricostruito il segnale al ricevitore, grazie all'azione di un filtro passa – basso che “taglia” le frequenze oltre la banda B.

Quantizzazione: è l'operazione con cui una grandezza che assume valori in un intervallo continuo è trasformata in un valore all'interno di un set discreto di valori, un'approssimazione che introduce un errore (detto *di quantizzazione*); più livelli discreti di valori avrò, maggiore sarà la precisione e meno influente l'errore, entro un certo limite.

¹² Anche detta *Frequenza di Nyquist*.

Codifica: associazione di uno specifico gruppo di bit per ogni livello discreto di quantizzazione; in altre parole, in funzione del numero di livelli l avrò il numero di bit m associato a ciascun campione, tale che $l = 2^m$.



I segnali analogici per essere trasmessi in formato digitale sono trasformati in flussi di bit astratti dai supporti fisici (segnali logici).

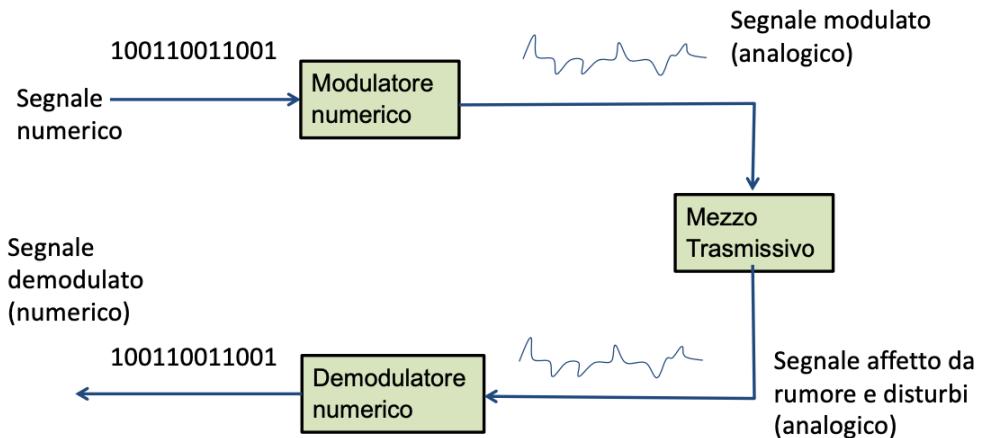
Formule: le formule richieste per svolgere gli esercizi di seguito.

Segnale	Banda $B \text{ Hz}$	Frequenza di campionamento $f_c > f_N = 2B \text{ Hz}$	Livelli di quantizzazione $l=2^b$	Flusso binario $R_b=2B*b \text{ bit/s}$
---------	-------------------------	---	--------------------------------------	--

Physical layer – trasmissione

Modulatore: dispositivo utilizzato per trasformare i bit in un opportuno segnale fisico che sarà poi trasmesso tramite collegamento fisico (e.g. fibra ottica, rame, onde elettromagnetiche).

La sequenza digitale viene utilizzata per *modulare* (cioè modificare) uno dei parametri del segnale fisico inviato nel mezzo trasmittivo. Uno schema di funzionamento è il seguente.



Modulazione: operazione di “traduzione” di segnale digitale in corrispettivo fisico, può avvenire in banda base o passante. Rispettivamente, avranno le seguenti caratteristiche.

- Nella banda base, i segnali da modulare hanno uno spettro contiguo rispetto l’origine. Prendendo come esempio la modulazione d’ampiezza in banda base, si utilizzerà la PAM¹³ dove il bit corrisponde ad un impulso di ampiezza positiva (1) o negativa/nulla (0).



- Mentre in banda passante/traslata, i segnali da modulare avranno uno spettro traslato su intervalli di frequenze non contigue all’origine. Si adopera un’onda elettromagnetica detta *portante/carrier* ad una determinata frequenza per traslare lo spettro del segnale intorno alla frequenza della portante. Ne è un esempio la modulazione d’ampiezza AM (analogica).



¹³ PAM o Pulse Amplitude Modulation.

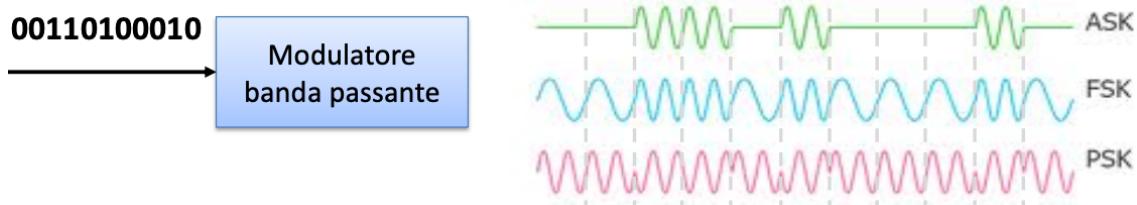


Si tenga presente che la modulazione analogica si usa quando anche il segnale da modulare è analogico, quando si trasportano bit si deve usare la modulazione numerica.

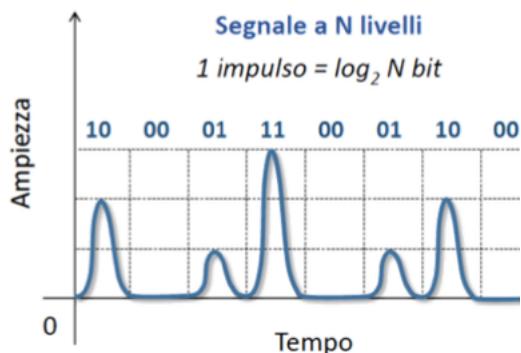
Propagazione di un segnale (con portante): un segnale può propagarsi nell'atmosfera o in un mezzo trasmittivo guidante attraverso la modulazione di un'onda portante¹⁴.

Modulazione numerica in banda portante: uno dei parametri della sinusode portante viene modulato dai valori che assumono i bit da trasmettere. Ne esistono differenti tipologie, alcune descritte di seguito.

- ASK, modulazione di ampiezza
- FSK, modulazione di frequenza
- PSK, modulazione di fase
- QAM, modulazione mista ampiezza/fase



Modulazione multilivello: tecnica di modulazione dove si incrementa l'ordine di modulazione per aumentare la capacità del canale.



Il flusso di bit in input è diviso in gruppi $\log_2 N$, si usano N livelli di ampiezza differenti e ad ogni simbolo/impulso corrispondono $n=\log_2 N$ bit.

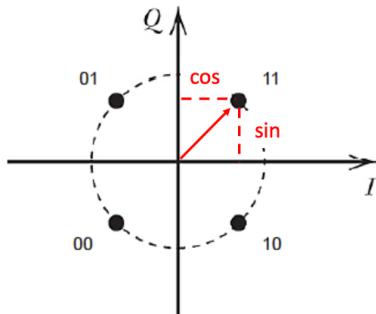
Posso determinare il bit rate come $R_b=R_s/\log_2 N$; di seguito un esempio con la PAM a 4 (=N) livelli di ampiezza.

Altri esempi di modulazioni multilivello di fase e ampiezza/fase sono le seguenti.

¹⁴ Nelle telecomunicazioni sono utilizzate in particolare le frequenze radio e le infrarossi.

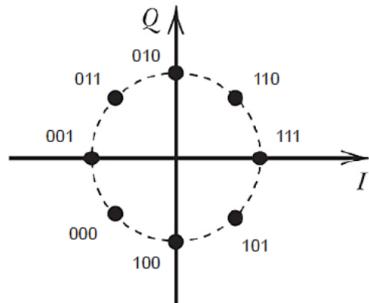
• **QPSK**

4 livelli di fase
2 bit per livello



8-PSK

8 livelli di fase
3 bit per livello

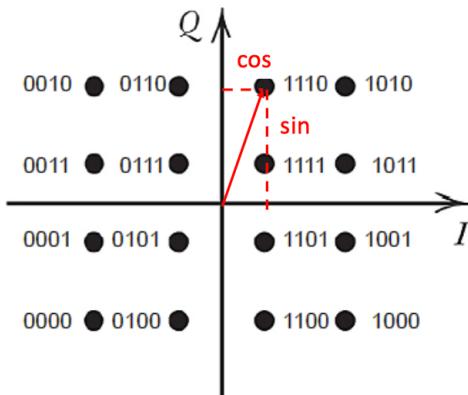


I: portante «in fase» (cosinusoide)
Q: portante «in quadratura» (sinusoide)

• **16QAM**

16 livelli di fase e ampiezza
(chiamati anche SIMBOLI)
4 bit per livello

Si parla anche di
«costellazione»



Capacità di canale: sfruttando la trasmissione multilivello posso incrementare la capacità trasmittiva di n volte, ma non la velocità. O almeno non aumentando arbitrariamente i livelli:

- Se mantengo inalterata la distanza tra i vari simboli → devo aumentare l'energia dell'impulso¹⁵.
- Se mantengo inalterata l'energia massima → la trasmissione è più sensibile al rumore¹⁶.

Un impulso (segnale) che attraversa un *canale trasmittivo* è soggetto ad alterazioni, come l'*attenuazione* e *dispersione*. Inoltre, il canale può introdurre rumore, che influisce su tutte le armoniche dell'impulso.

Attenuazione: riduzione della potenza del segnale in funzione della distanza percorsa e della frequenza del segnale stesso.

Dispersione: introduzione di un ritardo differente per ciascuna componente spettrale del segnale.

¹⁵ Energia dell'impulso necessaria per trasmettere i simboli più esterni della costellazione.

¹⁶ Rumore introdotto dal canale, può causare errore in ricezione.

Risposta in frequenza: funzione $H(f)$ del canale trasmittivo che può sintetizzare nel dominio delle frequenze le precedenti distorsioni.

Banda passante del canale (B): regione nel dominio delle frequenze in cui si ha la miglior risposta in frequenza, in termini di attenuazione e dispersione; il canale di comunicazione deve essere lineare, con lo spettro del segnale ricevuto pari al prodotto tra lo spettro del segnale trasmesso per la funzione di trasferimento. Per evitare distorsioni, il canale non deve modificare lo spettro del segnale.

- Banda passante del canale > banda occupata dal segnale.

Mezzi trasmittivi: distinti in due tipologie, guidati e non, dove nel primo caso ne fanno parte,

- *I mezzi elettrici*, si modula segnale che è associato a variazioni di tensione o corrente
- *Fibre ottiche*, si modula segnale sotto forma di impulsi luminosi

Mentre fanno parte dei mezzi trasmittivi non guidati le *onde radio*, dove il segnale è associato ad un'onda elettromagnetica che ha la capacità di riprodurre a distanza una corrente elettrica in un ricevitore. Ogni mezzo trasmittivo è caratterizzato da valori propri di banda passante, attenuazione, sensibilità al rumore, ecc.

Velocità di trasmissione: dato un canale con banda passante pari a $B[\text{Hz}]$, allora posso definire il *Baud rate* come $R_s = \eta_s B \left[\frac{\text{simboli}}{\text{s}} \right]$, con $\eta_s \begin{cases} = 2, \text{ impulsi ideali} \\ = 1, \text{ impulsi reali} \end{cases}^{17}$ che è l'efficienza spettrale.

- *Trasmissione binaria*, il bit rate massimo è $R_b = R_s \approx B \left[\frac{\text{bit}}{\text{s}} \right]$.
- *Trasmissione multilivello (n -bit)*, il bit rate massimo è $R_b = nR_s \approx nB \left[\frac{\text{bit}}{\text{s}} \right]$.

Attenuazione: dato un segnale in ingresso con potenza P_{IN} ed in uscita P_{OUT} , allora si definisce *attenuazione o guadagno* il rapporto $A = \frac{P_{OUT}}{P_{IN}}$, si esprime in decibel come $A_{dB} = 10 \log_{10} \left(\frac{P_{OUT}}{P_{IN}} \right)^{18}$. Posso esprimere l'*attenuazione del mezzo trasmittivo* per km α come $P_{OUT} = P_{IN} e^{-\alpha L} \left[\frac{dB}{km} \right]$, mentre calcolo l'attenuazione stessa con $\alpha_{dB} = (10 \log_{10} e) \cdot \alpha = 4,343 \cdot \alpha$.

Power budget: differenza in dB tra potenza media del trasmettitore e la potenza al ricevitore necessaria ad ottenere le prestazioni volute, in altre parole, è la potenza del sistema che si può “spendere” per supportare le attenuazioni *note* (perdita di linea e perdite A_{extra}) ed un margine di sistema M per gli imprevisti.

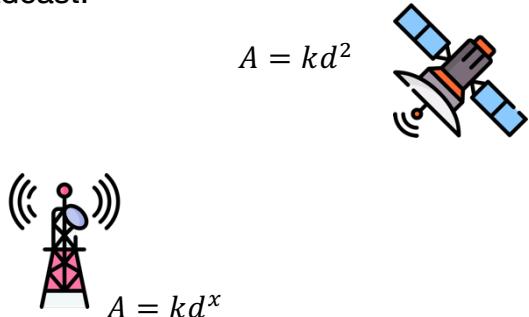
$$P_{TX} - P_{RX} = \alpha \cdot l + A_{extra} + M^{19}$$

¹⁷ Per impulsi reali si intendono particolari forme di questi, in grado di resistere al rumore del canale.

¹⁸ Il decibel viene utilizzato per indicare un valore relativo di tensione, corrente o potenza; si noti che il log è moltiplicato per 20 tranne nel caso della potenza, dove invece si moltiplica per 10.

¹⁹ Per l si intende la lunghezza di tratta in km.

Trasmissione wireless: dove le onde elettromagnetiche viaggiano nell'etere alla velocità della luce, e sono in grado di indurre corrente in dispositivi di ricezione anche molto distanti. Al salire della frequenza, le onde avranno un comportamento diverso (onde radio a bassa frequenza, le frequenze alte sono bloccate da ostacoli); realizzano una trasmissione di tipo broadcast.

- 
- L'attenuazione nello spazio libero aumenta col bit-rate e la distanza percorsa.
 - Ha una banda limitata, d_{sat} molto grande, mentre la d_{tower} è piccola.

La velocità di trasmissione è funzione dell'ampiezza della banda passante utilizzata, si sfrutta una banda traslata con modulazione di ampiezza/fase.

DENOMINAZIONE		SIGLA	FREQUENZA	LUNGHEZZA D'ONDA	USO
FREQUENZE ESTREMAMENTE BASSE		ELF	0 - 3kHz	> 100Km	Bande marittime
FREQUENZE BASSISSIME		VLF	3 - 30kHz	100 - 10Km	
RADIOFREQUENZE	FREQUENZE BASSE (ONDE LUNGHE)	LF	30 - 300kHz	10 - 1Km	
	MEDIE FREQUENZE (ONDE MEDIE)	MF	300kHz - 3MHz	1Km - 100m	Radio AM
	ALTE FREQUENZE	HF	3 - 30MHz	100 - 10m	Radio FM
	FREQUENZE ALTISSIME (ONDE METRICHE)	VHF	30 - 300MHz	10 - 1m	TV
MICROONDE	ONDE DECIMETRICHE	UHF	300MHz - 3GHz	1m - 10cm	
	ONDE CENTIMETRICHE	SHF	3 - 30GHz	10 - 1cm	
	ONDE MILLIMETRICHE	EHF	30 - 300GHz	1cm - 1mm	Satelliti

Osservazione: solitamente nelle telecomunicazioni si usano le onde decimetriche e centimetriche.

Doppino: coppia di conduttori di rame intrecciati ²⁰ e ricoperti da isolante, è utilizzato per le connessioni terminali del sistema telefonico, da casa alla centrale più vicina.

- B dipende dalla lunghezza
- Si distinguono in base alla velocità di trasmissione supportata
 - Categoria 3, dove si hanno 4 coppie contenute in una guaina, comune nei cablaggi interni agli edifici.
 - Categoria 5, dove si ha un avvolgimento più fitto con isolamento al teflon, si ha una migliore qualità sulle lunghe distanze, utilizzato nelle LAN.²¹

Cavo coassiale: meglio isolato rispetto al doppino, costituito da un conduttore centrale circondato da un isolante ricoperto da una calza metallica; usato per la tv via cavo.

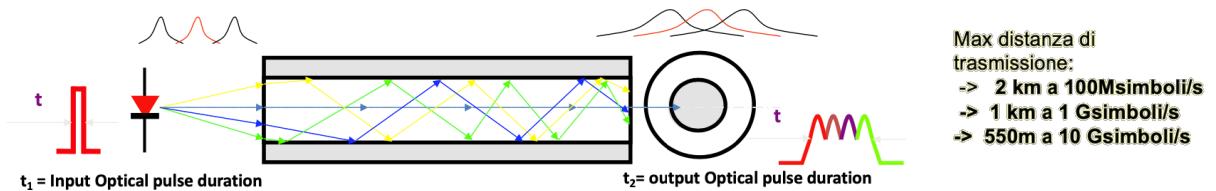
- Attenuazione di 5-8dB/100m (thick) e di 10-15dB/100m (thin)
- Velocità di circa 10Mb/s per km.

²⁰ Serve per minimizzare le interferenze, altrimenti i due cavi realizzerebbero un'antenna.

²¹ Attenuazione di circa 20dB/100m.

Fibra ottica: struttura guidante per segnali ottici, dove il salto di indice di rifrazione $n_1 - n_2$ tra nucleo e mantello determina il confinamento del fascio ottico, permettendone la propagazione.

- *Fibra multimodale MMF*, diametro del nucleo $50\mu m \leq x \leq 62\mu m$. La luce si propaga con diversi percorsi, la distanza di trasmissione è limitata dal fenomeno di dispersione modale, utilizzata per le reti locali e collegamenti nei datacenter. Opera tra la prima e la seconda finestra di portante ottica.

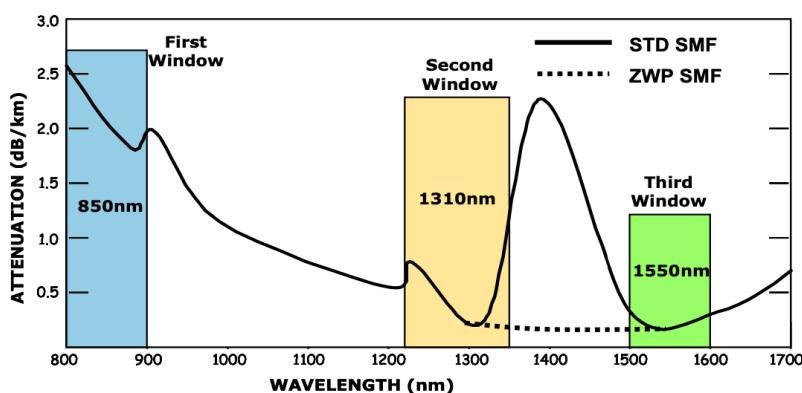


- *Fibra monomodale SMF*, diametro del nucleo $x < 10\mu m$. La luce si propaga in una sola direzione, pertanto si avrà una banda portante elevatissima (nell'ordine dei THz); per questo motivo sono utilizzate nei collegamenti a lunga distanza, dove questo mezzo riesce a garantire una velocità di 100Gbit/s per migliaia di km. Opera tra la seconda e la terza portante ottica.

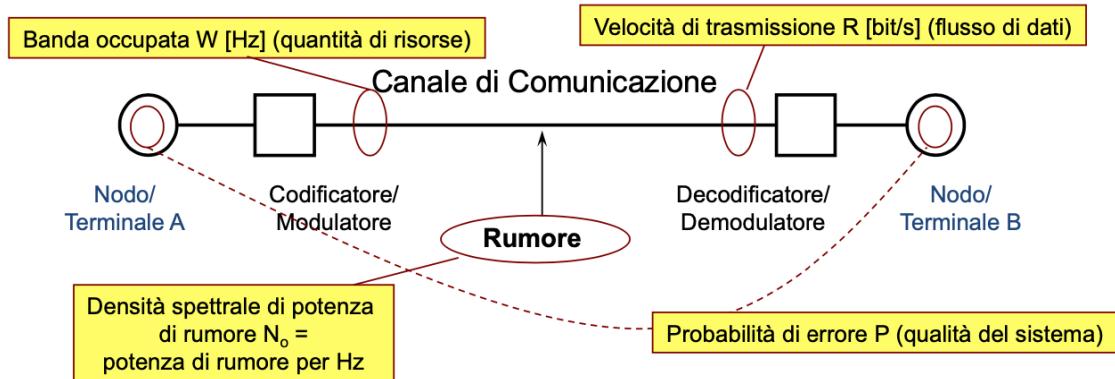


Mentre l'attenuazione nelle fibre di vetro è espressa dal seguente schema, sfruttando la distinzione nelle finestre.

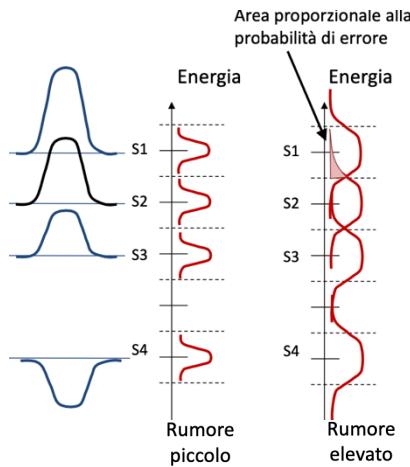
Attenuazione tipica:
 1° finestra 850 nm → **2 dB/km**
 2° finestra 1310 nm → **0.4 dB/km**
 3° finestra 1550 nm → **0.2 dB/km**



Per analizzare gli errori che possono influenzare la trasmissione attraverso un generico canale di comunicazione, verrà utilizzato il seguente modello.



Errori in ricezione: dovuti alla ricezione di una sequenza di bit diversa da quella trasmessa, a causa del *rumore casuale*, cioè quel disturbo introdotto dai mezzi trasmittivi e altri dispositivi lungo il percorso; tale rumore impatta la probabilità di ricevere gli impulsi nel modo corretto. Entra in gioco il *decisore*²², che elabora il campione (del segnale ricevuto) misurandone l'energia e confrontandola coi risultati attesi, deciderà che è stato trasmesso il simbolo il cui livello atteso è più vicino all'energia ricevuta misurata.



- Se l'alterazione causata dal rumore è grande rispetto alla differenza tra i livelli, aumenta la probabilità di commettere errore.
- A parità di rumore, la probabilità d'errore diminuisce all'aumentare della differenza tra i livelli.
- *Codici correttori*²³, adottati per abbassare le probabilità di errore, prevedono di aggiungere dei bit di ridondanza (parità) in modo che gli errori possano essere corretti, se limitati in numero.
- *Ritrasmissione*, se non si riesce a correggere un errore, può comunque essere rilevato dal sistema (come nella commutazione di pacchetti) e richiedere la trasmissione del solo pacchetto errato; si utilizza ARQ o Automatic Repetition Request.

Capacità massima di canale: esiste un limite teorico per la velocità di un canale, scoperta e dimostrata da Shannon, è espressa dalla seguente equazione.

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

C: capacità di canale [bps]
B: banda del canale [Hz]
S: potenza del segnale ricevuto [W]
N: potenza del rumore ricevuto [W]

²² In maniera analoga a quanto visto in Fondamenti di Elettronica con la conversione analogico digitale.

²³ Possono correggere fino a $c = \frac{n-1}{2}$ errori, con n numero di volte in cui si ripete il bit da trasmettere, n-1 sono le cifre di parità.

Application layer – introduzione

Un'applicazione di rete²⁴ è un software che può essere eseguito su diversi terminali e comunica con altri applicativi remoti attraverso la rete. Inventare una nuova applicazione non richiede di sostituire il software della rete, anche perché i suoi nodi non hanno software applicativo.

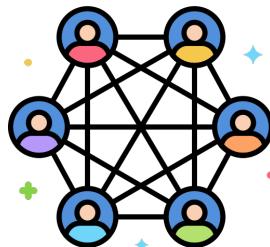
Le applicazioni di rete hanno generalmente due tipologie di architetture differenti, la *client-server* e la *peer-to-peer*.

Client – Server: architettura dove i dispositivi coinvolti nella comunicazione implementano o solo il processo *client* o solo quello *server*; in particolare, posso distinguere le due macchine che realizzano l'architettura nel seguente modo.



- *Server*, possono solo rispondere alle richieste dei client, si tratta di un host sempre attivo con un indirizzo IP permanente, è in grado di gestire molte richieste (da altrettanti client) anche grazie alla possibilità di utilizzare tali macchine in *cluster*.
- *Client*, possono solo eseguire richieste ai server (non può comunicare direttamente con altri client) inviandone anche più di una, possono cambiare indirizzo IP ed essere connessi in modo discontinuo.

Peer-To-Peer: architettura dove i dispositivi coinvolti implementano tutti sia il processo *server* sia il processo *client*. Caratterizzata dall'avere dei terminali/peer che comunicano direttamente, utilizzando un collegamento intermittente e con IP non fisso; non ci sono server sempre connessi.



Alcuni esempi di questa architettura sono *torrent*, *Tor* (che usa il protocollo onion) e altri sistemi di download p2p.

Nonostante le differenze precedentemente descritte, le applicazioni di rete condividono alcuni elementi fondamentali. Tra questi, gli *host* che sono i dispositivi degli utenti, i *processi* cioè il programma in esecuzione (su un host) e la *comunicazione interprocesso*. In particolare, quest'ultima coinvolge tecnologie software che consentono a diversi processi di comunicare tra loro, con meccanismi più semplici per processi locali, mentre per quelli remoti saranno necessarie funzionalità di rete.

²⁴ Tali applicazioni sono solo nei terminali, possono essere facilmente sviluppate e diffuse.

Comunicazione tra processi remoti: è necessario un sistema di indirizzamento dei processi, essenzialmente costituito da indirizzo di processo + indirizzo host su cui è in esecuzione. Inoltre, si dovrà utilizzare un opportuno protocollo di scambio dati, per decidere “come” e “quando” ricevere i messaggi/pacchetti.

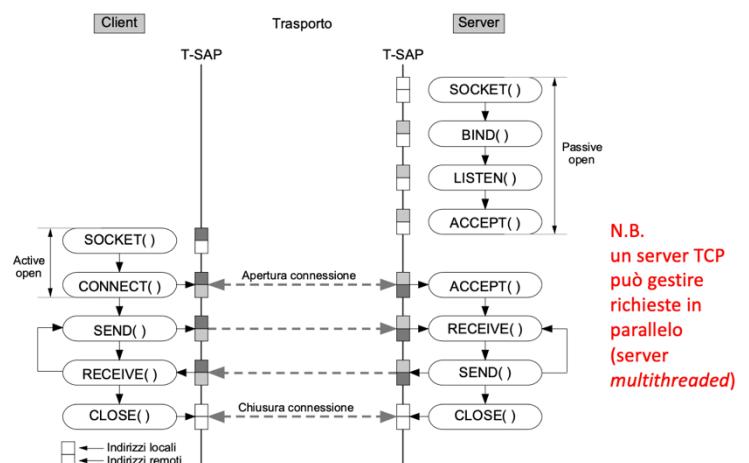
- *Indirizzamento*, la comunicazione tra processi è supportata dai servizi dei livelli inferiori e avviene attraverso i *SAP* (*Service Access Point*); ogni processo è associato ad un SAP, al quale si associa il *Socket*²⁵.
L'indirizzo di un processo in esecuzione è formato da indirizzo IP + socket.
- Più complesso è il discorso dei protocolli da utilizzare, che saranno approfonditi più avanti, in generale possiamo comunque definire i requisiti che porteranno alla scelta del protocollo per l'applicazione.
 - *Affidabilità*, alcune applicazioni possono tollerare perdite parziali di dati, mentre altre necessitano di completa affidabilità.
 - *Ritardo*, alcune applicazioni richiedono basso ritardo, e.g. gaming interattivo.
 - *Banda*, alcune applicazioni richiedono una velocità di trasferimento minima, mentre altre si adattano a quella disponibile.

Servizi di trasporto: utilizzati dalla rete, sono essenzialmente due, il *Transmission Control Protocol (TCP)* e lo *User Datagram Protocol (UDP)*.

- TCP, caratterizzato dall'essere connection-oriented, implementa il trasporto affidabile, si ha controllo di flusso, di congestione e degli errori. D'altro canto, non fornisce garanzie di ritardo e banda.
- UDP, si distingue per essere connectionless, il trasporto non è affidabile, non si ha alcun controllo se non quello di errore (opzionale), ma è molto veloce.

Questi servizi vengono richiesti, sia da server che da client, attraverso i socket utilizzando le *primitive di servizio*. Nel caso del TCP, si compiono operazioni iniziali per instaurare la connessione, dove il server attua *passive open*, il client invece *active open*; di seguito si utilizzeranno le primitive descritte sotto.

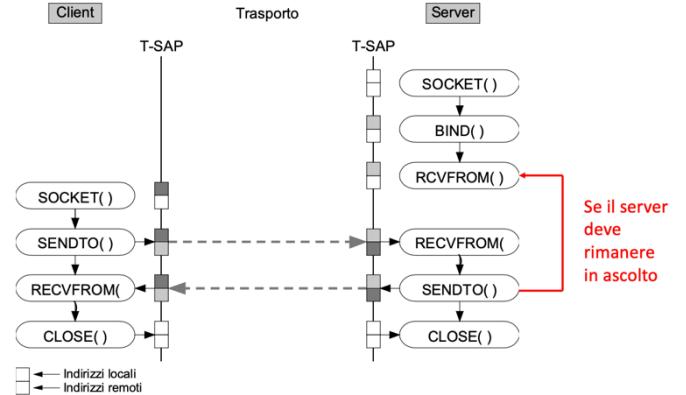
Primitiva	Server/Client	Funzione
SOCKET()	Entrambi	Genera una nuova socket
BIND()	Server	Genera un indirizzo locale che viene associato a una socket
LISTEN()	Server	Predisponde la ricezione di richieste di connessione dal client
CONNECT()	Client	Richiede l'instaurazione di una connessione con un server
ACCEPT()	Server	Blocca il server fino all'arrivo di una richiesta di connessione
SEND()	Entrambi	Invia dati
RECEIVE()	Entrambi	Blocca il processo in attesa di ricevere dati
CLOSE	Entrambi	Termina la comunicazione



²⁵ Il Socket, o porta software, è l'informazione di indirizzamento del processo nell'host.

Nel caso dell'UDP non si avranno più le primitive per instaurare una connessione, ma solo quelle necessarie all'interazione tra applicazioni e socket.

Primitiva	Server/ Client	Funzione
SOCKET()	Entrambi	Genera una nuova socket
BIND()	Server	Genera un indirizzo locale che viene associato a una socket
SENDTO()	Entrambi	Invia dati a una specifica socket remota
RECVFROM()	Entrambi	Blocca il processo in attesa di ricevere dati da una specifica socket remota
CLOSE	Entrambi	Termina la comunicazione



RFC: *request for comment*, è seguito da un numero e indica un protocollo internet; un protocollo può averne associati più di uno, poiché potrebbero anche indicare la versione di un certo protocollo.

Application layer – web browsing

Il servizio di web browsing permette di “navigare” tra le pagine web²⁶, che sono identificate da un apposito indirizzo detto *Uniform Resource Locator (URL)*, come ad esempio <http://github.com/Vinello28>.



HTTP: l'*Hyper Text Transfer Protocol* è il servizio utilizzato per la trasmissione delle pagine web, è *stateless* (non tiene traccia delle richieste), sfrutta il protocollo TCP. Di seguito una sintesi del funzionamento.

- Client http inizia connessione TCP verso la porta 80 del server
- Il server http accetta la connessione
- Client e server si scambiano le pagine web e messaggi di controllo
- Terminata la trasmissione si chiude la connessione tra client e server

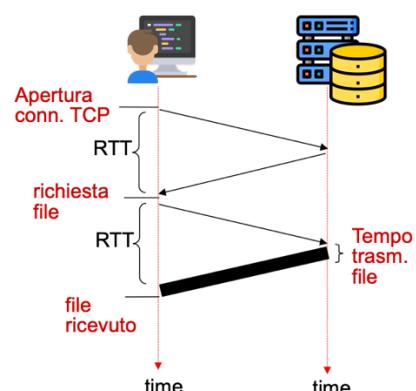
Modalità di connessione: l'HTTP sfrutta due tipologie di connessione, persistente e non.

- *Persistente*, rimane aperta e può essere utilizzata per trasferire più oggetti della stessa pagina o più pagine web.
 - *senza pipelining*, richieste HTTP inviate in serie (si aspetta prima di effettuare successiva).
 - *Con pipelining*, non si attende la risposta a richiesta precedente.
- *Non persistente*, una sola sessione che server chiude una volta inviato l'oggetto, si ripete per tutti gli oggetti da trasmettere; più sessioni possono operare in parallelo per minimizzare ritardo.

Round Trip Time RTT: tempo necessario per trasferire un messaggio da client a server e ritorno, più il relativo ACK²⁷; si calcola con le seguenti formule.

$$T_{non-persistente} = \sum_{i=0}^n (2RTT + T_i)$$

$$T_{persistente} = RTT + \sum_{i=0}^n (RTT + T_i)$$
²⁸

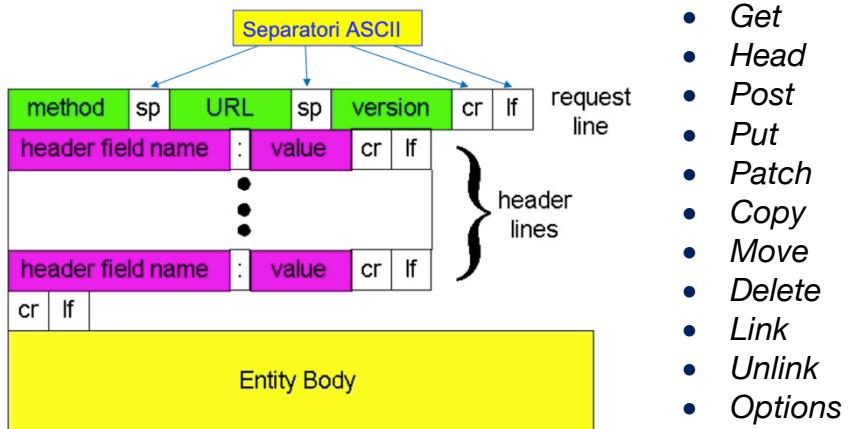


²⁶ Sono risorse in rete costituiti da oggetti come file HTML, immagini, applet, ecc. Hanno generalmente una pagina di base che chiama le successive quando richiesto (solitamente PHP o HTML).

²⁷ Messaggio di conferma della ricezione.

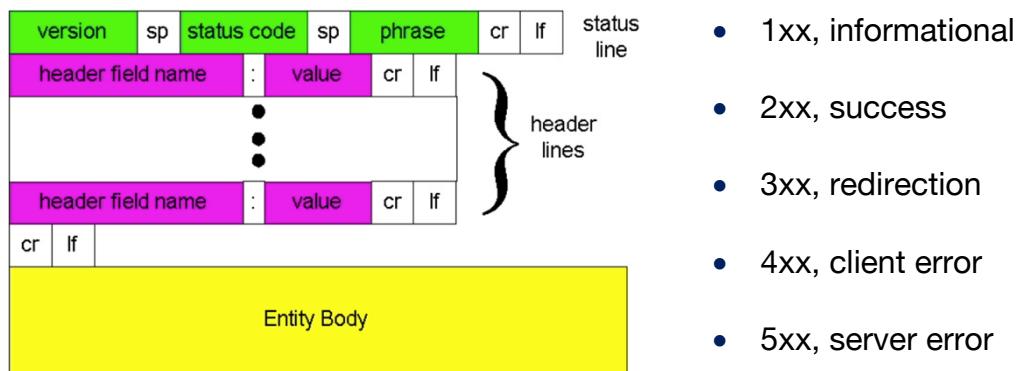
²⁸ La n indica il numero di oggetti da trasmettere.

I messaggi http sono codificati in ASCII (human readable), i metodi utilizzati dal protocollo sono:



- Get
- Head
- Post
- Put
- Patch
- Copy
- Move
- Delete
- Link
- Unlink
- Options

Gli header permettono di scambiare informazioni di servizio aggiuntive, è flessibile poiché si possono inserire più campi header all'interno dello stesso messaggio; la sintassi utilizzata prevede *header_name : header_value*. Mentre le risposte http hanno una struttura leggermente diversa, di seguito il modello. Alcuni status code sono:



- 1xx, informational
- 2xx, success
- 3xx, redirection
- 4xx, client error
- 5xx, server error

Curiosità: non solamente l'URL, gli oggetti trasmessi con l'http sono univocamente identificati anche dall'ETag, assegnato dal filesystem.

Conditional GET: permette di non inviare un oggetto richiesto se già presente nel *client*, si inserisce nella richiesta http la data dell'oggetto presente in cache locale tramite l'header, la risposta non conterrà l'oggetto se la copia contenuta nel client sarà aggiornata.

Cookie: è un file utilizzato per ovviare il problema dello *stateless*, viene settato da un messaggio http per poi essere mantenuto su una lista nell'host (cache del browser) ed anche in un database di cookies nel sito web. In questo modo si salvano nel cookie quelle informazioni che si vogliono “salvare” per i futuri accessi al sito, o anche fini di pubblicità personalizzata.

Proxy http: è un server che permette di rispondere alle richieste http senza coinvolgere il server http, dove il client invia tutte le richieste http ad un proxy, se l'oggetto richiesto è disponibile nella cache di questo server allora il proxy risponde con l'oggetto; altrimenti il proxy recupera l'oggetto dal server http.

- È un application gateway, cioè un instradatore di messaggi di livello applicativo (un host *fittizio*)
- Deve essere sia client che server
- Maschera gli utenti poiché i server vedono arrivare tutte le richieste dal proxy

Standard HTTP/2: permette di ridurre la latenza di caricamento delle pagine web, è in formato binario (trasferisce frame) sfrutta una connessione TCP per *stream*²⁹ multipli (*multiplazione*), gli header sono compressi, è presente il servizio di server push e si appoggia su TLS.

- Con il *push*, il server invia informazioni utili senza che il client le richieda esplicitamente, tale funzionalità è richiesta dal client.
- Il protocollo HTTPS è la versione “sicura” dell’http, grazie al *Secure Socket Layer* ed al *Transport Layer Security* che aggiungono confidenzialità, integrità ed autenticazione alle connessioni TCP.

Connessioni SSL/TLS: connessione sicura del protocollo http, composta da 3 fasi

- Handshake, dove client e server si accordano per trasferimento e cifratura da utilizzare. È in questa fase che si ha lo scambio di certificati tra server e client (e viceversa) per autenticarsi; un certificato contiene la chiave pubblica dell’entità certificata, informazioni aggiuntive e la firma digitale della *certification authority*. Poi si generano e scambiano le chiavi simmetriche per la cifratura del trasferimento dati, su un’apposita connessione a sua volta cifrata con chiavi asimmetriche.
- Trasferimento dati, dove i dati sono suddivisi in PDU, ciascuno cifrato secondo l’algoritmo precedentemente deciso.
- Chiusura connessione, si utilizza un apposito messaggio per una chiusura sicura.

²⁹ Lo stream è una sequenza di logica di frame.

Application layer – servizio eMail

Il servizio di posta elettronica opera tra *User Agent* (client, e.g. outlook) che comunicano con i *Mail Servers*³⁰ sfruttando i protocolli *SMTP* e *POP3/IMAP*.

Mail server: per ogni client controllato ne contengono una *coda di mail in ingresso* privata (client vede solo la propria) ed una *coda di mail in uscita* condivisa tra tutti i client. Hanno anche duplice funzione,

- Server, quando ricevono le e-mail da tutti i client che controllano
- Client, quando inviano ad altri mail server tutte le mail destinate ai client controllati da questi

Inoltre, utilizzano diversi protocolli

- SMTP, con altri mail server e client in uplink
- POP3/IMAP, con client in downlink

Formato dei messaggi: RFC822 indica lo standard di formattazione di un messaggio e-mail, costituito da un *header* contenente mittente-destinatario-oggetto e dal *body*, cioè il contenuto della mail in formato ASCII.

- *MultipurposeInternetMailExtension* (MIME), è un'estensione del formato RFC822, indicato da RFC2045/2046 supporta l'invio di contenuti multimediali non ASCII 7bit. Viene indicato nell'header il tipo di contenuto *ContentType* ed il tipo di codifica; permette trasferimento di più oggetti come parte dello stesso messaggio.

SimpleMailTransferProtocol (SMTP): applicativo client/server *push*, usa il TCP sulla porta 25 e l'interazione client -server con questo protocollo è di tipo comando-risposta (di tipo testuale) usa l'ASCII a 7bit; In particolare, la comunicazione avviene nel seguente modo.

- Apertura connessione TCP sulle porte 25 tra client (UA/MTA) e server (MTA)
- client avvia “presentazione” verso server stabilendo connessione a livello applicativo
- trasferimento del messaggio
 - invio *header* messaggio
 - invio *body* messaggio
- richiesta chiusura connessione (a livello applicativo)
- chiusura connessione TCP

Campo	Significato
From:	Indirizzo e-mail dell'autore del messaggio
To:	indirizzo e-mail del destinatario del messaggio
Cc:	Indirizzi e-mail di altri destinatari in copia del messaggio
Bcc:	Indirizzi e-mail di altri destinatari del messaggio nascosti ai precedenti destinatari
Date:	Data e ora di generazione del messaggio pronto per l'invio
Sender:	Indirizzo e-mail di origine del messaggio
Subject:	Identificazione dell'argomento del messaggio
Message-Id:	Identificatore univoco del messaggio
Reply-To:	Identificatore del messaggio originario cui si risponde
Received:	Identificatore del MTA lungo il percorso e data/ora di attraversamento
Return-Path:	indicatore di un percorso di ritorno verso il mittente

Tra questi messaggi possiamo distinguere comandi e risposte, di seguito elencati e descritte brevemente in tabelle.

³⁰ Include il *Mail Transfer Agent*, un processo necessario al trasferimento dei messaggi.

Comando	Parametro	Significato
HELO	Dominio client	Richiesta di apertura della connessione con identificazione del client SMTP verso il server SMTP
MAIL FROM	Indirizzo mittente	Identificazione del mittente del messaggio
RCPT TO	Indirizzo destinatario	Identificazione del destinatario del messaggio
DATA		Richiesta di autorizzazione a inviare il messaggio
QUIT		Richiesta di chiusura della connessione
RSET		Segnalazione di interruzione della transazione in atto
VRFY	Stringa	Richiesta di verifica di identificazione di utente o di mailbox

Sopra i comandi SMTP, a destra le risposte.

Per accedere alla propria *mailbox* virtuale, un certo client utilizzerà dei protocolli diversi dall' SMTP, si tratta del POP3 (RFC1939) e IMAP(RFC1730).

Codice	Parametro	Significato
220	Dominio server	Servizio pronto
221	Dominio server	Servizio in fase di chiusura del canale di trasmissione
250		Comando richiesto completato
251		Utente non locale; messaggio da inoltrare
354		Autorizzazione a invio mail che termina con <CRLF>.<CRLF>
421	Dominio server	Servizio non disponibile, canale in fase di chiusura
450		Comando richiesto ignorato: mailbox non disponibile (occupata o bloccata temporaneamente)
451		Comando interrotto: errore locale
452		Comando richiesto ignorato: memoria insufficiente
455		Server non disponibile ad accettare i parametri
500		Errore di sintassi, comando non riconosciuto
501		Errore di sintassi nei parametri o negli argomenti
502		Comando non implementato
503		Sequenza di comandi errata
504		Parametro del comando non implementato
550		Comando richiesto ignorato: mailbox indisponibile
551		Utente non locale
552		Comando ignorato: memoria allocata insufficiente
553		Comando ignorato: nome della mailbox non corretto
554		Transazione fallita
555		Parametri dei comandi MAIL FROM/RCPT TO non riconosciuti o non implementati

POP3: il *PostOfficeProtocol v3* è un applicativo client/server *pull* che permette ai client di accedere al proprio mail server e scaricare i messaggi; l'interazione client/server è di tipo comando-risposta, di seguito ne è descritto il funzionamento.

- Instaurazione connessione TCP sulla porta 110 (del server)
- Autenticazione del client
- Richiesta lista messaggi
- [Optional] richiesta download e/o cancellazione messaggi
- Chiusura comunicazione con il server
- Chiusura connessione TCP

Le risposte del POP3 sono essenzialmente di tre tipi, OK | ERR | consegna del messaggio, mentre i comandi sono elencati e descritti nella tabella sottostante.

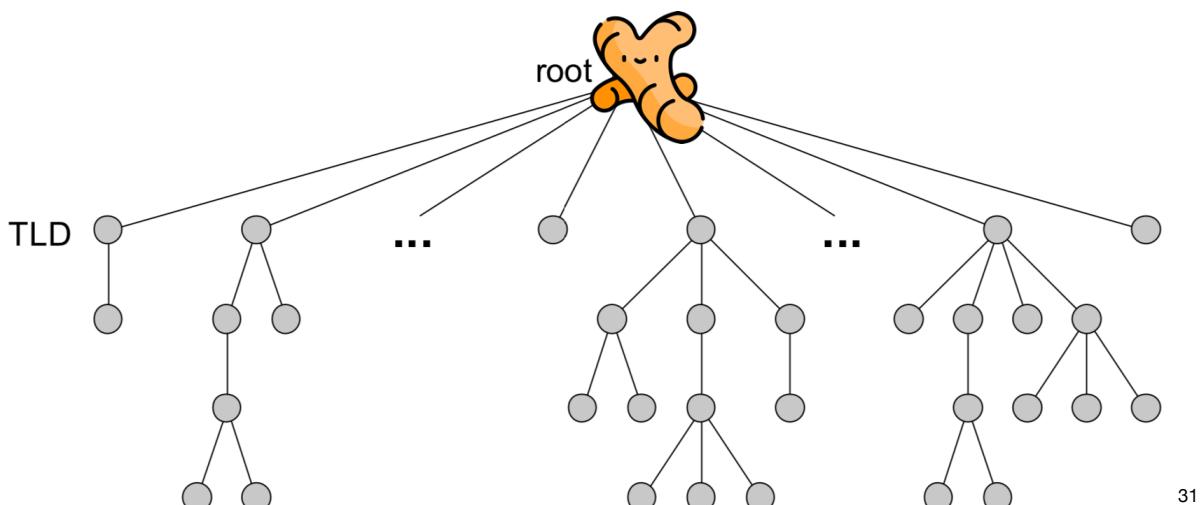
Comando	Parametro	Significato
USER	Nome	Richiesta di accesso alla mailbox
PASS	Stringa	Invio della password associata al "nome" precedente
STAT		Interrogazione sul numero di messaggi e occupazione nella mailbox
LIST	[Numero messaggio]	Interrogazione sui messaggi presenti nella mailbox
RETR	Numero messaggio	Richiesta di recupero di un messaggio dalla mailbox
DELE		Richiesta di cancellazione di un messaggio dalla mailbox
RSET		Richiesta di annullamento di precedenti ordini di cancellazione nella mailbox
NOOP		Nessuna operazione
QUIT		Richiesta di chiusura della comunicazione

Application layer – DNS

Gli indirizzi IP a 32 bit sono ottimi per essere letti/processati dalle macchine ma non per gli applicativi; in questo caso si usano gli *indirizzi simbolici*. Pertanto, l'operazione di *risoluzione di un nome* sarà la mappatura tra il nome simbolico e indirizzo numerico.

Domain Name System DNS: è il protocollo applicativo che realizza la risoluzione dei nomi, di tipo client/server, utilizza l'UDP sulla porta 53 del server; è strutturato come un database distribuito, con molti *name server* combinati in un'organizzazione gerarchica. Inoltre, consente

- *Host aliasing*, associazione tra vero nome host ed i suoi alias
- *Load distribution*, distribuzione del carico di richieste di mapping tra molti name servers distribuiti ma con uguali informazioni di mapping



31

Il nome di dominio è una sequenza di etichette che arriva fino alla radice, con una struttura fortemente dipendente dalla struttura dell'albero gerarchico; ogni livello ha una diversa profondità di informazione, con uno o più server per ogni dominio (nodo del grafo).

Local Name Server: ogni ISP ne ha uno, sono direttamente collegati con gli host, che lo contattano per risolvere un indirizzo simbolico; nel caso in cui non fosse disponibile la mappatura, allora contatterà altri *name server*.

Authoritative Name Servers: è un name server responsabile di un particolare nome simbolico.

L'associazione *indirizzo simbolico* \Leftrightarrow *indirizzo IP* sfrutta l'indirizzo dell'NS locale configurato in ogni *host*

- Le applicazioni richiedono l'associazione usando le funzioni DNS
- Il local NS reperisce l'informazione e restituisce la risposta
 - Subito se è disponibile nel suo database
 - Dopo aver interrogato altri NS (ci sono due modi, *iterativo* e *ricorsivo*)

³¹ Modello dell'organizzazione gerarchica dei name server, dove i TLD sono i domini di primo livello (TopLevelDomain), a seguire avrà quelli di secondo, terzo, ecc.

Modalità iterativa: di seguito sono descritti i passaggi.

- Client DNS sull'host contatta LNS
- L'LNS contatta RootNS
- RootNS segnala a LNS il TLD server responsabile del dominio cercato
- LNS contatta il TLD server
- Il TLD segnala al LNS il server autoritativo per l'indirizzo richiesto
- LNS contatta ANS
- L'ANS segnala al LNS l'indirizzo IP corrispondente
- L'LNS restituisce l'IP al client

Modalità ricorsiva: di seguito sono descritti i passaggi.

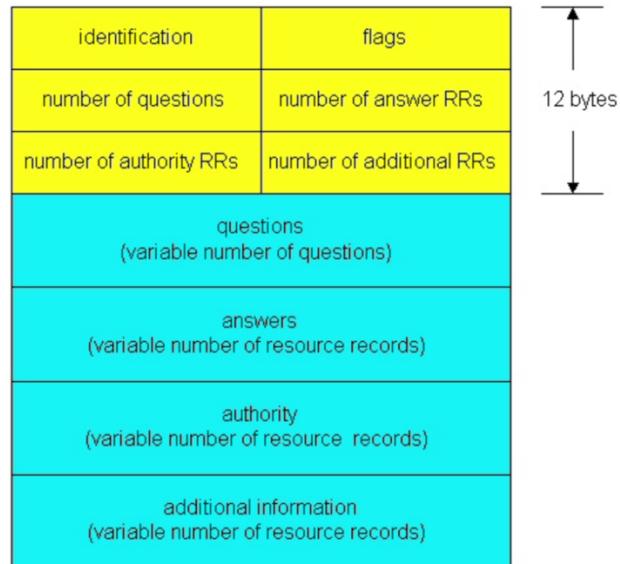
- Client DNS sull'host contatta LNS
- LNS contatta RootNS
- RootNS contatta TLD server responsabile del dominio cercato
- TLD contatta ANS per l'indirizzo simbolico cercato
- ANS segnala al TLD server l'indirizzo IP cercato
- L'informazione segue percorso inverso fino al client

DNS Caching: permette ad un server la memorizzazione temporanea³² di un indirizzo per cui non è authoritative dopo averlo reperito, all'arrivo di una nuova richiesta può fornire l'informazione senza risalire fino all'ANS. Le informazioni memorizzate comprendono *name*, *value*, *type*, *TTL*.

- Type =A, allora [*name* = hostname] e [*value* = indirizzo IP]
- Type =NS, allora [*name* = domain] e [*value* = nome server con informazioni]
- Type =CNAME, allora [*name* = alias di un host] e [*value* = nome canonico]
- Type =MX, allora [*name* = dominio di mail] e [*value* = nome del mail server]

³² Per un tempo definito *Time To Live* (TTL), deciso dal server authoritative, utilizzato dai non-ANS per decidere il timeout scaduto il quale cancellano l'indirizzo memorizzato; ha un valore compreso da pochi secondi a settimane.

Formato dei messaggi DNS:



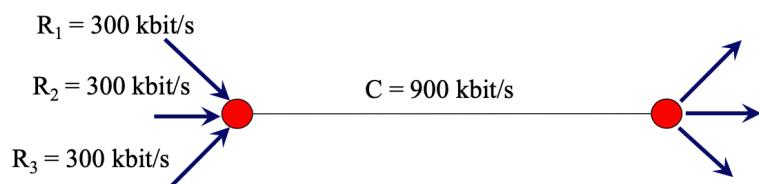
Per aggiungere un dominio (supponendo che non sia già stato preso), questo dovrà essere registrato presso uno dei *DNS Registrars* fornendogli i nomi simbolici e relativi indirizzi IP degli ANS; eventualmente il DNSR scriverà un record MX per l'indirizzo simbolico creato.

Application layer – Condivisione

La velocità di trasferimenti di un file dipende da fattori come velocità di trasmissione, propagazione, accodamenti e tipologia del protocollo utilizzato, si veda ad esempio la differenza tra connection-oriented e connection-less.

In condizioni ideali, il meccanismo di controllo del TCP è in grado di:

- Consentire un'equa suddivisione della capacità di ciascun link tra i flussi che lo attraversano.
- Limitare le congestioni della rete³³.



Prendendo come riferimento il modello a sinistra, nella realtà i valori dei rate indicati sono delle *medie* che valgono in condizioni ideali e a regime.

Inoltre, il ritmo di trasmissione è variabile in una situazione reale; pertanto, la *condivisione in condizioni reali non può essere equa*.

³³ Argomento approfondito nella sezione dedicata al protocollo TCP.

Transport layer – introduzione

È un livello protocollare presente solo nei terminali, consente il collegamento logico tra processi applicativi, mascherando alle applicazioni il trasporto fisico delle UI. Svolge operazioni di multiplexing/demultiplexing per applicazioni poiché più applicazioni possono necessitare del servizio di trasporto.

Collegamento logico: necessita di un indirizzo solitamente indicato dal socket = *IP + #porta*; hanno una lunghezza di 16bit, i numeri di porta variano tra 0 e 65535.

Indirizzamento: le funzioni di de/multiplexing sono gestite mediante gli indirizzi precedentemente descritti, dove le porte utilizzate si dividono in tre categorie.

- *Well-known ports [0, 1023]*, assegnate ad importanti applicativi dal lato server.
- *Numeri registrati [1024, 49151]*, assegnati a specifiche applicazioni da chi ne fa richiesta.
- *Numeri dinamici [49151, 65535]*, assegnati dinamicamente a processi applicativi lato client.

Buffering: i protocolli di trasporto sono implementati nei sistemi operativi, che quando associano una porta ad un processo, vengono assegnate due code (ingresso e uscita) alla porta stessa e sono dette *buffer*; il loro utilizzo è necessario per “assorbire” i rallentamenti del processing dei livelli adiacenti³⁴.

Poiché il servizio di rete nasce come non affidabile, e per alcune tipologie di trasmissione è richiesto un certo grado di affidabilità, il servizio di trasporto mette a disposizione degli appositi protocolli per un trasporto sicuro (ma lento) oppure veloce (ma inaffidabile) dei dati. Inoltre, l’interazione tra entità di trasporto avviene seguendo l’architettura Client/Server, che possono essere entrambi eseguiti in modalità parallela o seriale, a seconda del protocollo utilizzato.

- Nel caso dell’UDP, i server sono *seriali*, dove i pacchetti che arrivano con delle richieste vengono accodati attendendo il loro turno, dopodiché il server li esamina e genera risposte sequenzialmente.
- Se utilizzano il TCP, i server sono *parallel*, rispondono simultaneamente a più richieste, usando una connessione aperta verso ciascun processo client per il tempo necessario allo scambio richieste/risposte.

³⁴ Si intendono i livelli di Rete e Applicazione.

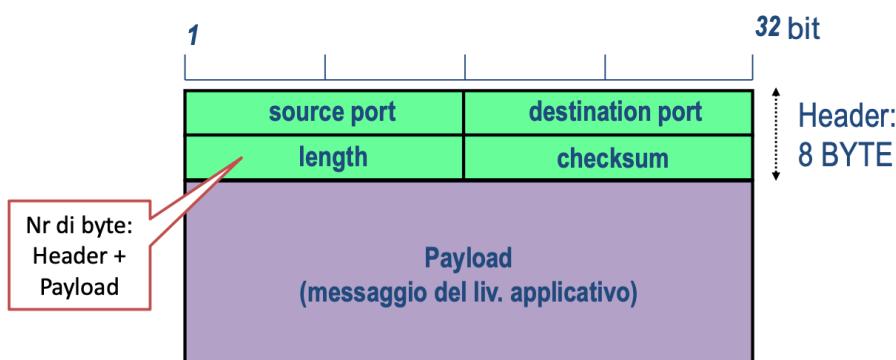
Transport layer – RFC768 UDP

È un protocollo connection-less che fornisce, *multiplazione* per mezzo dell'indirizzamento di livello 4, e optionalmente la *rivelazione d'errore*. Non implementa la consegna in sequenza, rilevamento delle perdite e controllo di flusso/congestione, per questo motivo è detto *inaffidabile*.

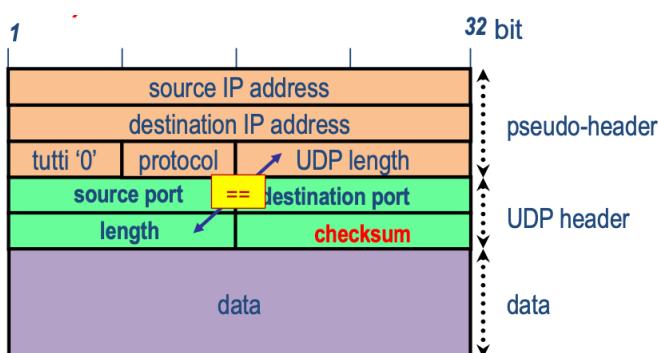
Vantaggi: ha una minore latenza (non occorre stabilire connessione), è più semplice poiché non si tiene traccia dello stato di connessione e si implementano poche regole ed ha un minore overhead (con un header UDP più piccolo del TCP).

Incapsulamento: il messaggio applicativo viene incapsulato in esattamente 1 messaggio UDP e viene trasportato da 1 pacchetto IP; frammentazione del pacchetto eventualmente svolta dal livello di rete.

Formato del messaggio: è definito e riassunto dal seguente schema, dove si possono osservare i campi introdotti dall'UDP per la de/multiplazione e rivelazione d'errore.



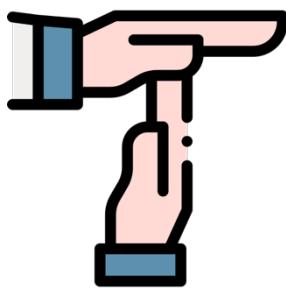
Checksum: campo del messaggio utilizzato per il controllo d'integrità, contiene informazione ridondante e viene calcolato dal trasmettitore; il ricevitore invece effettuerà il calcolo sull'intero messaggio, se corretto accetterà il segmento altrimenti verrà scartato.



- Calcolo effettuato considerando un messaggio UDP con pseudoheader.
- *Trasmettitore* → l'insieme di bit è diviso in blocchi da 16 bit, il checksum è inizializzato a 0. Poi, tutti i blocchi vengono sommati con complemento a 1, il risultato complementato viene inserito nel campo di *checksum* del segmento inviato.
- *Ricevitore* → l'insieme di bit è diviso in blocchi da 16 bit, tutti i blocchi vengono sommati con complemento a 1; osservando il risultato complementato, se tutti i bit sono a 0 è accettato, viene scartato altrimenti.

Transport layer – protocolli ARQ

Il recupero d'errore può essere effettuato utilizzando tecniche di *correzione d'errore*, che introducono però un elevato overhead nei pacchetti, oppure utilizzando i *protocolli di ritrasmissione*: dove ciascuna UI ricevuta correttamente viene riscontrata positivamente con un messaggio ACK (*acknowledgement*), che se non ricevuto entro un certo *timeout*, porterà alla ritrasmissione del pacchetto corrispettivo mancante.³⁵



L'utilizzo di questi protocolli a livello di trasporto permette il recupero *end-to-end* degli errori, i pacchetti UI possono anche essere persi nei buffer dei router lungo la rete (per questo sono utili i timeout). In breve, gli ARQ³⁶ hanno come obiettivo l'integrità delle UI e la loro consegna in sequenza, senza duplicazione, utilizzando i messaggi ACK/NACK e meccanismi come *timeout / finestra di trasmissione*.

Di seguito le descrizioni, in sintesi, dei principali protocolli ARQ utilizzati nei livelli di trasporto delle reti.

Stop&Wait: utilizza ACK e timeout, ogni messaggio ricevuto correttamente è riscontrato con un ACK, mentre quando il mittente trasmette un pacchetto, avvia il suo timeout e si mette in attesa; se il primo evento successivo è la ricezione dell'ACK, allora trasmette il pacchetto successivo e si itera il processo, altrimenti, alla scadenza del timeout si ritrasmette il pacchetto corrente.

- *Tempo di trasferimento*, di N_f UI senza errori è pari alla seguente formula.

$$T_{tot} = N_f T_1 = N_f (T_f + T_a + 2T_p + 2\tau) = N_f \left(\frac{L_f}{C} + \frac{L_a}{C} + 2T_p + 2\tau \right)$$

- *Efficienza* η , è la frazione di tempo in cui il canale è usato per trasmettere informazione utile in assenza di errori, si calcola come nella formula sottostante.

$$\eta = \frac{T_f}{T_{tot}} = \frac{T_f}{T_f + T_a + 2T_p + 2\tau} \quad ^{37}$$

- T_f : tempo di trasmissione di una UI (s)
- T_a : tempo di trasmissione di un riscontro (s)
- T_p : tempo di elaborazione (s)
- $\tau = d/v$: ritardo di propagazione (s)
 - d : distanza tra trasmettitore e ricevitore (m)
 - v : velocità di propagazione (m/s)
- C : capacità del canale (bit/s)
- L_f : lunghezza (costante) della UI (bit)
- L_a : lunghezza (costante) del riscontro (bit)

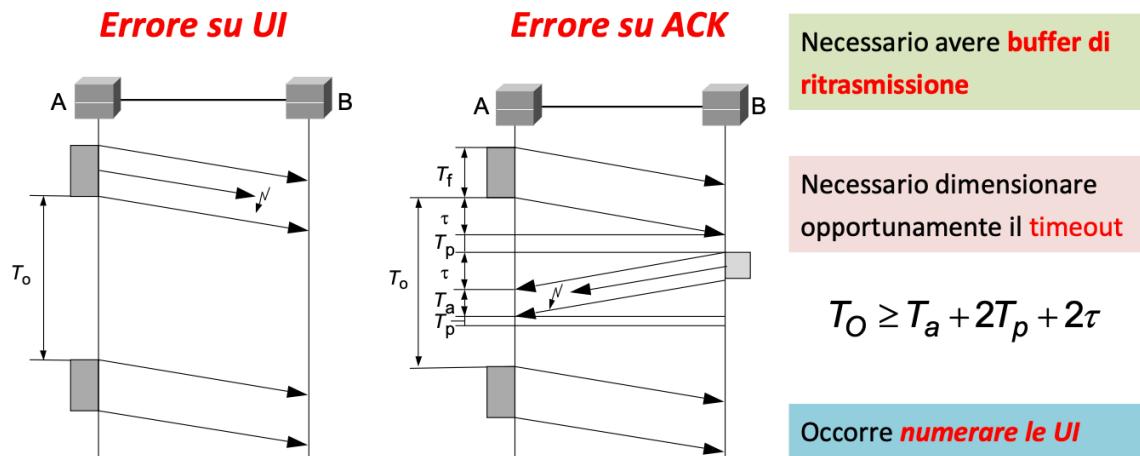
- *Throughput*, è il grado di utilizzo del collegamento, calcolato come $THR = \eta \cdot C \quad \left[\frac{\text{bit}}{\text{s}} \right]$.

³⁵ È necessario un canale di ritorno per il trasporto di ACK/NACK; inoltre, si deve tener presente che anche questi messaggi di conferma possono essere affetti da errori.

³⁶ Acronimo che indica i protocolli che svolgono funzioni di *Automatic Repeat reQuest* o ARQ.

³⁷ T_a e T_p sono trascurabili. Se $T_f \ll \tau$ avrà efficienza bassa.

- Errore su UI e/o ACK richiede ritrasmissione con timeout T_0 .

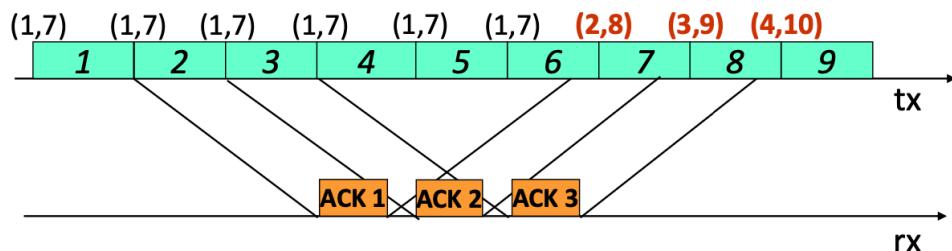


Numerare le UI permette di distinguere eventuali pacchetti replicati nel caso di mancata ricezione dell'ACK ed il loro riordino all'arrivo. Per il dimensionamento del timeout anche i messaggi di riscontro sono numerati. L'operazione di numerazione introduce overhead, però aiuta a garantire un trasporto affidabile dell'informazione.

- Il timeout si dimensiona solitamente al valore minimo $T_O = T_a + 2\tau$, ma il tempo di elaborazione non è controllabile e la stima dell'RTT non sempre è valida, pertanto vengono numerati anche i riscontri e si evitano ambiguità.

Continuous ARQ (Go-back-n): è una variante dello Stop&Wait, dove si possono trasmettere UI (appartenenti ad una stessa “finestra”) senza aspettare riscontri. Utilizza una numerazione ciclica modulo $N = 2^b$ (in S&W $b=1$), comporta overhead di b -bit, il TCP usa $b=32$ bit.

- *Efficienza*, molto alta in assenza di errori.
- *Tempo di trasferimento*, di N_f UI con $T_p = 0$ è pari a $T_{tot} = N_f T_f + T_a + 2\tau$.
- Ad ogni istante, il mittente tiene traccia della sua *finestra di trasmissione*, dove il riscontro della prima UI della finestra, fa “scorrere” (idealmente) la stessa di una posizione.



- La ritrasmissione avviene allo scadere di un timeout, oppure se ricevitore invia un riscontro negativo (NACK)³⁸; a livello di trasporto (TCP) viene principalmente utilizzato il limite di tempo.

³⁸ Poco utilizzato nei sistemi moderni.

Go-back-n: variante del Continuous ARQ, si distingue perché al verificarsi di un errore, ritrasmette solamente la finestra a partire dal primo pacchetto non riscontrato. Il trigger utilizzato per la ritrasmissione è il timeout³⁹. Questo meccanismo può portare alla ritrasmissione di pacchetti ricevuti correttamente ma semplifica il funzionamento poiché i pacchetti fuori sequenza vengono ignorati e scartati (sequenza mantiene l'ordine), mentre i pacchetti ignorati non inoltrano alcun ACK.

- Se non sono presenti UI fuori sequenza, ACK può essere inviato cumulativamente (e.g. “riscontro la ricezione di tutti i pacchetti fino al numero x”); a patto che non scadano timeout.
- *Dimensionamento della finestra*, deve avere una dimensione W_s ottimale, la condizione per evitare perdite di efficienza è espressa dalla seguente formula.

$$W_s T_f \geq T_f + T_a + 2\tau \text{ da cui ricavo la finestra ottima } W_s = \left\lceil \frac{T_f + T_a + 2\tau}{T_f} \right\rceil$$

Se i tempi di trasmissione e propagazione non sono noti, si ovvia al problema con una finestra grande, che in caso di errore può aumentare il rischio di inutili ritrasmissioni; oppure si effettuano stime sull'RTT e si adatta di conseguenza la finestra e/o timeout.

- *Utilizzo del NACK*, può abbreviare i tempi di ritrasmissione evitando di aspettare la fine della finestra, ma per inviare tale riscontro di deve conoscere il numero del pacchetto che se è andato perso non è possibile sapere; quindi, si ipotizzerà che sarà il precedente a quello arrivato fuori sequenza. È un ragionamento che non si può applicare nei meccanismi dove non è garantita la consegna in ordine.

Protocolli bidirezionali: in una comunicazione TCP, il trasferimento dati/riscontri avviene in modo bidirezionale utilizzando *piggyback*; infatti, gli ACK possono anche essere inseriti negli header dei pacchetti che viaggiano in direzione opposta (*piggybacking*).



- SN indica il numero di sequenza del pacchetto trasmesso, mentre RN è il numero di sequenza del pacchetto atteso in direzione opposta⁴⁰.
- Date N =dimensione finestra, N_{last} =ultimo riscontro ricevuto, N_c =numero corrente disponibile. Allora posso definire le seguenti regole per il **trasmettitore** e per il **ricevitore**.

Ogni nuova UI viene messa in attesa se $N_c \geq N_{last} + N$, altrimenti, se $N_c < N_{last} + N$ UI viene trasmessa con $SN = N_c$, viene inizializzato il timeout e incrementato di 1 il valore di N_c .

Ogni riscontro RN ricevuto $N_{last} = RN$.

Le UI nella finestra trasmesse senza vincoli di temporizzazione.

Con la scadenza di *timeout* la ritrasmissione parte da $UI[N_{last}]$.

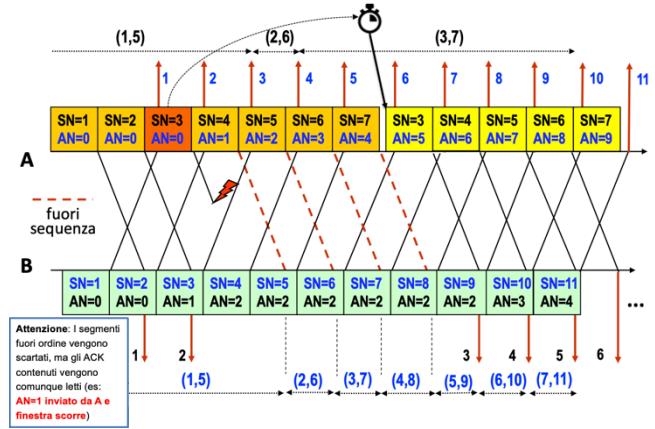
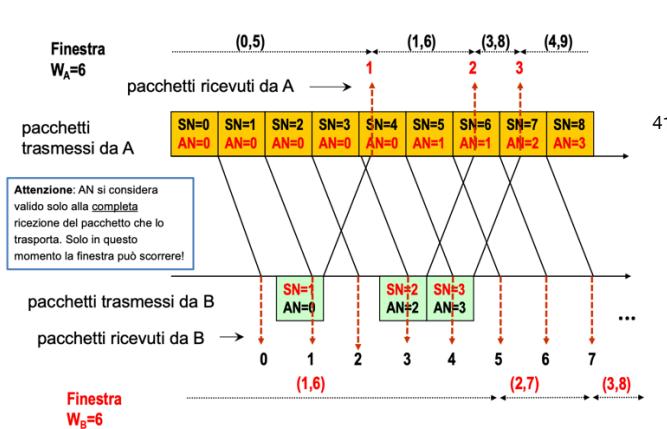
³⁹ Proprio di ogni pacchetto, viene cancellato quando si riceve il relativo riscontro positivo.

⁴⁰ Valido come riscontro cumulativo dei pacchetti fino a RN-1.

Se UI correttamente ricevuta con $SN = RN$ allora viene inoltrata ai livelli superiori e si pone $RN = RN + 1$.

Ad istanti arbitrari, ma con ritardo finito, RN ritrasmesso al mittente con le UI in direzione opposta.

AN è l'ACK Number, ossia il numero di sequenza del pacchetto riscontrato (può essere cumulativo); nel protocollo TCP ha significato di *prossimo byte atteso*. I numeri SN e AN devono essere inizializzati in entrambe le direzioni, cioè deve esistere un momento di inizio non equivocabile in cui scambiare informazioni per l'inizializzazione



⁴¹ Modello di trasferimento Go-back-n senza errori a destra, mentre a sinistra quello con errori.

Transport layer – controllo di flusso

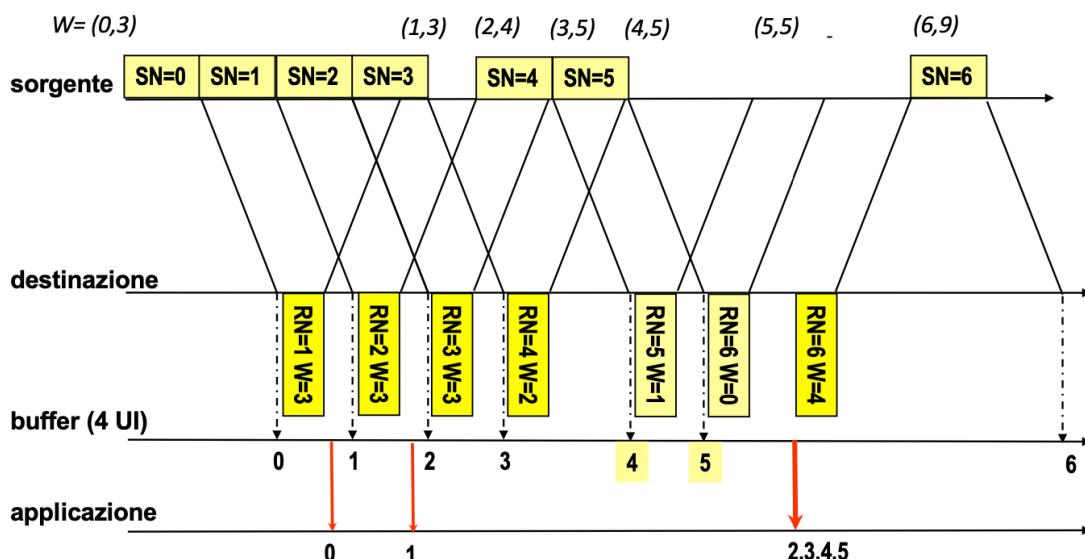
L'obiettivo è regolare il ritmo di invio, così da evitare che i pacchetti vadano persi a causa del buffer pieno. Questo accade per il numero di posizioni W_{RX} limitate del *buffer di ricezione*; infatti, il processo applicativo ricevente ha una velocità di lettura delle UI variabile e non sempre corrispondente a quella di trasmissione.



È possibile utilizzare un meccanismo a *finestra*⁴², questo porterà ad avere problemi relativi al dimensionamento della finestra ed eventuale lentezza nella lettura delle UI causerà un altrettanto lento svuotamento del buffer.

Soluzione: scorrelare il controllo di errore, cioè l'invio dei riscontri, dal controllo di flusso; si avrà un *controllo di flusso a credito esplicito con finestra mobile*. Si inserisce nei riscontri un campo *Window* ($\approx W_{RX}$) che indica quante UI (o byte) si possono ancora ricevere, in altre parole indica lo spazio rimanente nel buffer in ricezione; è la soluzione adottata dal protocollo TCP.

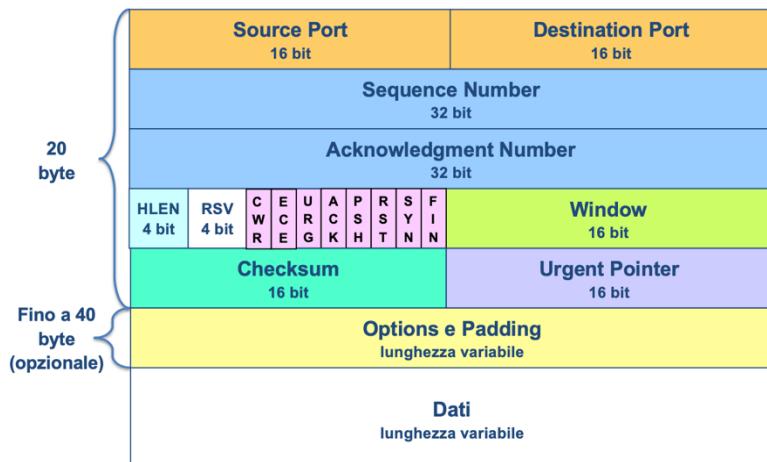
- Gestione della finestra del controllo di flusso, la dimensione rimanente comunicata non è quella effettiva, viene tenuto un margine di sicurezza oppure si comunica che è vuoto (anche se non lo è). Si potrebbe aspettare che si riempia significativamente prima di comunicarlo al trasmettitore, così da evitare l'invio di segmenti troppo brevi. Si possono usare dei meccanismi adattivi appositi.



⁴² Analogico a quelli visti per i precedenti protocolli di ARQ.

Transport layer – RFC793 TCP

È un protocollo connection-oriented che utilizza connessioni bidirezionali *full duplex* (simmetria nel traffico non certa), implementa un servizio di trasporto *affidabile*, sfruttando il Go-back-n nella sua versione base. Il flusso di dati è organizzato in gruppi di byte, contenuti nei segmenti⁴³; inoltre, fornisce *multiplazione* per mezzo dei socket, *consegna in sequenza*, *controllo di flusso e degli errori*, *controllo di gestione*.



- *source/dest port*, sono gli indirizzi del socket sorgente e di destinazione
- *sequence number*, n°sequenza del primo byte nel payload
- *ACK number*, numero di sequenza del prossimo byte da ricevere
- *HLEN*, lunghezza complessiva header TCP

- *Window*, valore della finestra di ricezione
- *Checksum*, calcolato come nell'UDP
- *Flags*, costituito da 8 sottocampi con valori 0 o 1
 - *Congestion window reduced (CWR)*
 - *Explicit congestion notification echo (ECE)*
 - *Urgent pointer (URG)* ➔ se vale 1 considera campo *Urgent Pointer*
 - *Acknowledge number (ACK)* ➔ se vale 1 ACK valido
 - *Push (PSH)*
 - *Reset (RST)*
 - *Setup (SYN)* ➔ utilizzato durante il setup della connessione
 - *Chiusura connessione (FIN)*
- *Options and Padding*, con i due sottocampi
 - *Options*, indica massima dimensione dei segmenti (MSS), impostato di default a 536byte
 - *Padding*, riempimento fino a multipli di 32 bit, con uso di no-op e end-of-op

Funzione PUSH: funzionalità del TCP che prevede l'inoltro immediato dei dati da parte del TCP ricevente all'applicazione ricevente, attivo settando l'apposito flag PSH. Generalmente non implementato nella primitiva send(...) delle interfacce TCP, ma viene automaticamente settato dal sistema operativo nell'ultimo segmento che svuota il buffer.

⁴³ Sono le UI a livello di trasporto, hanno dimensione variabile (in termini di byte).

Dati urgent: settando la modalità del TCP in urgent, il dato contenuto nel campo *urgent pointer* conterrà

- RFC 793 (1981), puntatore all'inizio del primo byte dopo i dati urgenti.
- RFC 1122 (1989), puntatore all'inizio dell'ultimo byte dei dati urgenti.

Nonostante sia stato adottato l'RFC1122, alcune implementazioni di TCP seguono ancora lo standard precedente.⁴⁴

Opzioni: aggiunte all'header TCP, quella da 1 byte servono per avere header multiplo di 32 bit, mentre le opzioni lunghe hanno un apposito campo *length* che indica la dimensione (e.g. MSS).

Incapsulamento e frammentazione: il messaggio dell'applicazione può essere frammentato in più segmenti per fare controllo di flusso e di congestione, l'IP può frammentare nuovamente il messaggio se necessario. MSS ha valore tale da far corrispondere un segmento ad un datagramma, la dimensione degli header TCP è calcolata come una stima (no valore certo).

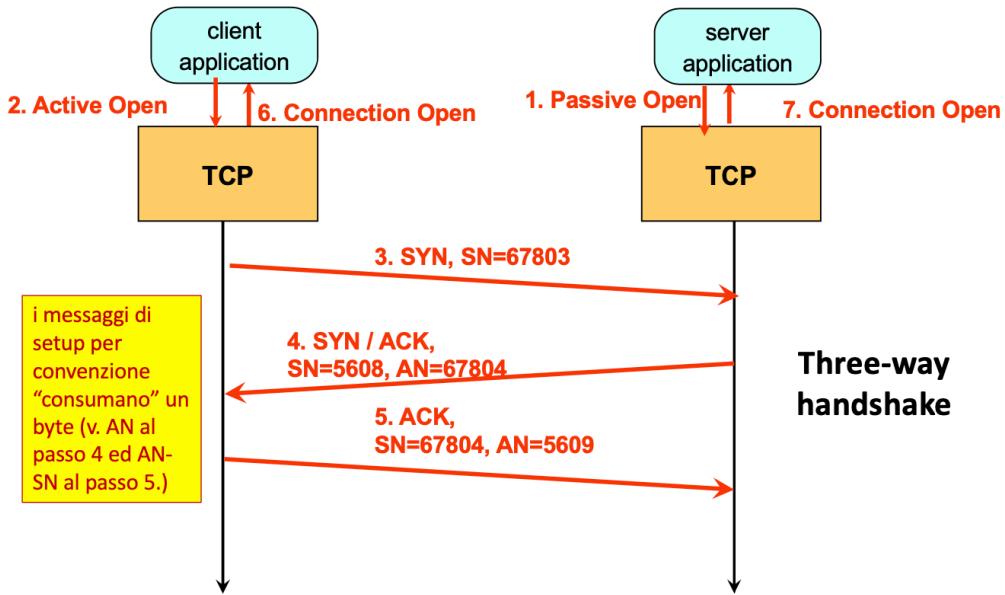
Fattore di scala della finestra: descritto dall'RFC7323, definisce il fattore di scala della finestra di ricezione, settato in fase di *open*, dove fa sì che il campo *Window* venga scalato per ottenere il valore reale della finestra di ricezione.

$$Rwnd = W \cdot 2^{fs}$$

Setup della connessione: le applicazioni comunicano con il software TCP locale, poi seguono i passaggi sottostanti. È una tecnica anche detta *handshake a 3 vie*.

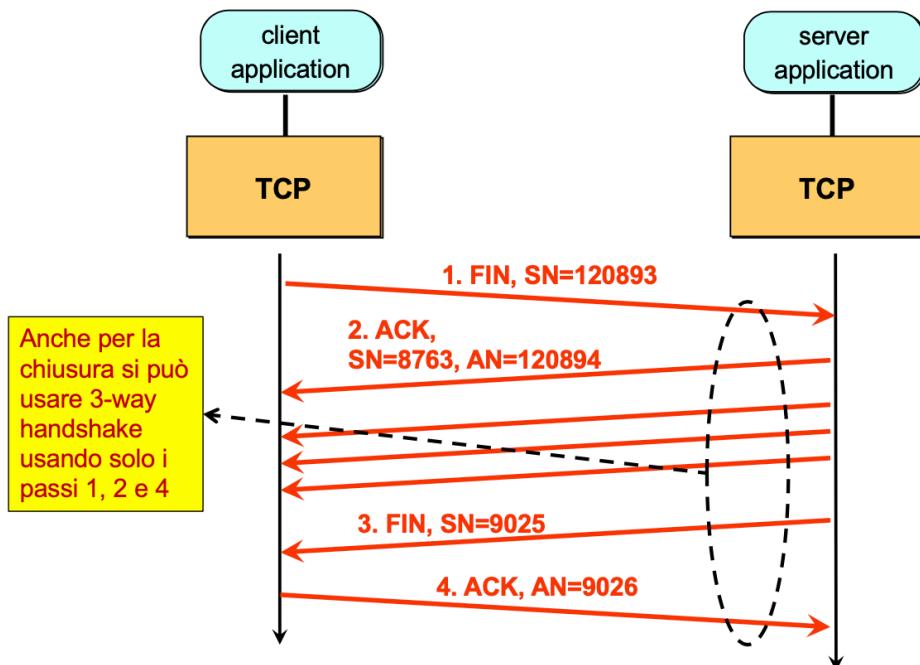
1. Il server fa una *passive open*, comunica a TCP locale che accetta nuove connessioni
2. Client fa un'*active open*, comunica a TCP locale che intende effettuare connessione verso socket remoto
3. Il client TCP estrae numero di SN e manda un segmento di tipo SYNchronize (SYN=1) contenente l'SN estratto e altri parametri di connessione (MSS, fattore di scala fs)
4. Alla ricezione del SYN il TCP server estrae SN e manda un segmento SYN/ACK per riscontro contenente anche un AN. Una volta che arriverà al client si aprirà la connessione
5. TCP client riceve SYN/ACK e invia ACK, nel payload inserisce i primi dati della connessione con numero di sequenza del primo byte ed anche la dimensione della finestra (*window*).
6. Il TCP client notifica all'applicazione l'apertura della connessione
7. Quando TCP server riceve ACK del TCP client, notifica all'applicativo (server) che la connessione è aperta.

⁴⁴ Sistemi operativi differenti implementano in maniera differente le politiche di gestione dei dati urgent.



Tear down della connessione: di seguito una descrizione dei passaggi utilizzati dal TCP per la chiusura della connessione.

1. Il TCP che chiude la connessione invia segmento FIN con gli ultimi dati
2. TCP dall'altra parte invia un ACK, connessione attiva *half duplex* per invio dei dati
3. Appena completata la trasmissione, TCP chiude la connessione anche nell'altra direzione, inviando segmento FIN
4. TCP che aveva già chiuso connessione in direzione opposta, invia ACK per confermare, chiudendo la connessione.



Collisione di richieste: se due *open* hanno uguali socket sorgenti e di destinazione (in ordine inverso) si instaura una sola connessione.

Reset connessione: se si setta RST a 1 nel segmento, allora si interrompe la connessione in entrambe le direzioni, senza inviare dati.

Controllo di flusso: Il controllo di flusso in TCP implica che il ricevente controlla il flusso del trasmittente. Il lato ricevente ha un *buffer di ricezione*, mentre il lato trasmittente ha un *buffer di trasmissione*⁴⁵. La finestra di trasmissione/ricezione si muove in sincronia tra gli host A e B. La dimensione della finestra di ricezione di B (Rwnd) è determinata dal buffer di ricostruzione di B e viene comunicata ad A per adattare la finestra di trasmissione. Se un segmento non viene riscontrato, viene copiato in un buffer di ritrasmissione e ritrasmesso dopo il timeout.

- Alcuni problemi con le finestre includono il "silly window syndrome" (lato ricevente), dove il destinatario svuota il buffer lentamente e comunica una finestra piccola, causando overhead. Questo può essere risolto con l'algoritmo di Clark. Il "silly window syndrome" (lato trasmittente) si verifica quando l'applicazione trasmittente genera dati lentamente, causando segmenti piccoli con overhead. L'algoritmo di Nagle risolve questo problema.
- La persistenza, quando il destinatario imposta la finestra di ricezione a 0, causando l'interruzione della trasmissione da parte del trasmittente TCP. Si utilizza un timer di persistenza che coincide con il timeout di ritrasmissione. Quando arriva un segmento con finestra=0, si attiva il timer. Allo scadere del timer, viene inviato un segmento di sonda. Se viene riscontrato positivamente, la trasmissione riprende; altrimenti, si riavvia il timer e si ripete il processo.

Meccanismo di ritrasmissione: è di tipo Go-back-n con timeout e senza NACK, dove le ritrasmissioni avvengono solo per scadenza di timeout; i byte vengono accettati anche fuori sequenza purché all'interno della stessa finestra di ricezione. Quando arrivano segmenti ritrasmessi, la finestra scorre fino al prossimo byte atteso in sequenza; viene poi mandato un ACK che riscontra tutti i segmenti precedenti.

- Valore del timeout stabilito dal TCP utilizzando l'RTT (continuamente misurato) per effettuare stime. È utilizzato un filtro a media mobile, con l'RTT misurato \forall segmento al ricevimento del rispettivo riscontro⁴⁶. Di seguito, rispettivamente la stima corrente e la stima di variabilità.

$$SRTT = RTT_{av}^i = (1 - \alpha) \cdot RTT_{av}^{i-1} + \alpha \cdot RTT_{last}, \quad \text{con } 0 < \alpha \leq 1$$

$$RTT_{dev}^i = (1 - \beta) \cdot RTT_{dev}^{i-1} + \beta \cdot |RTT_{last} - RTT_{av}^{i-1}|, \quad \text{con } 0 < \beta \leq 1$$

Generalmente, $\alpha = 0.125$ mentre $\beta = 0.25$, i valori iniziali invece sono $RTT_{av} = 0$ e $RTT_{dev} = 1.5 [s]$. Il timeout viene effettivamente determinato dall'*algoritmo di Jacobson*, dove si avrà

$$T_o = RTT_{av} + n \cdot RTT_{dev}, \quad \text{solitamente } n = 4$$

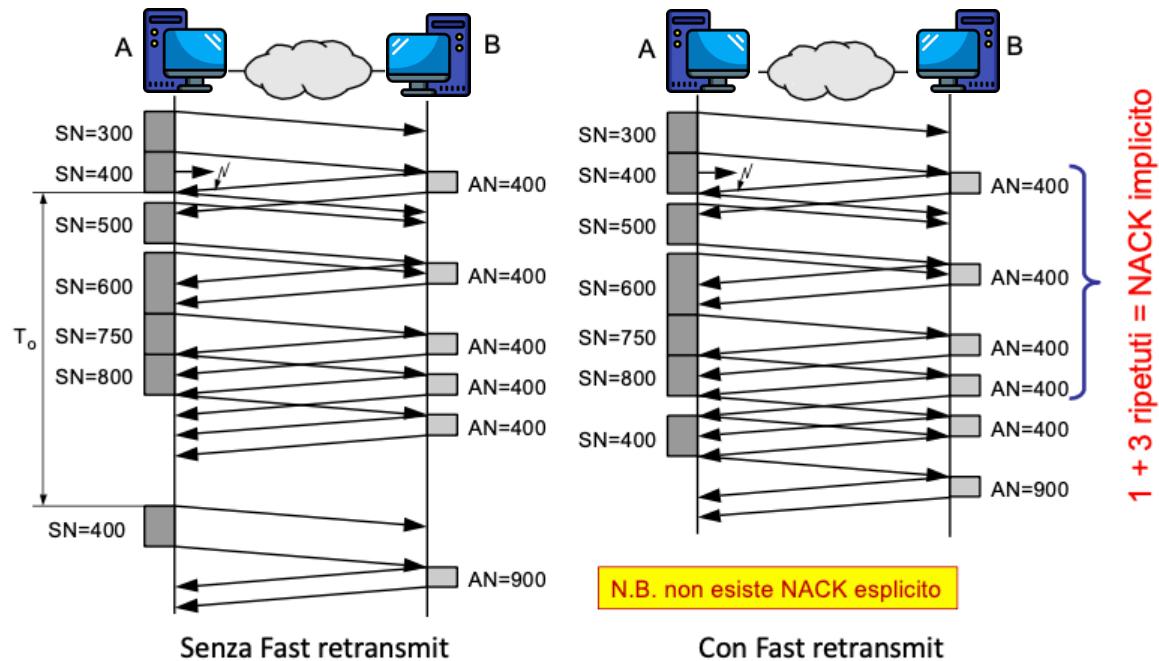
Per evitare che il timeout non venga più aggiornato in caso di ritrasmissione di tutti i pacchetti (dovuto ad improvvisa congestione della rete) si utilizza l'*algoritmo di Karn* o di *timer backoff*, dove ad ogni ritrasmissione dovuta a scadenza di timeout,

⁴⁵ Saranno inseriti i segmenti in attesa del consenso (del ricevitore) per essere trasmessi.

⁴⁶ La misura dell'RTT del segmento corrente è detta RTT_{last} .

questo viene aumentato; generalmente con moltiplicazione e fino ad una certa soglia.

- *ritrasmessione standard*, prevede che ricevitore mantenga buffer con segmenti fuori sequenza e ad ogni nuovo segmento invia ACK con AN che indica segmento mancante (ripete ACK); tutto ciò una volta che timeout è scaduto.
- *Ritrasmissione veloce*, si distingue dal precedente poiché alla ricezione di 3 ACK consecutivi ripetuti⁴⁷, procede a ritrasmettere anche prima che scatti il timeout (equivalente a NACK implicito).



Controllo di congestione: nel TCP è di tipo *end-to-end*, si ha congestione in un link quando la somma dei rate di trasmissione dei flussi che lo attraversano è maggiore della sua capacità; un modo semplice per evitare che ciò accada è regolare la finestra di trasmissione, dove trasmettitore mantiene *congestion window* Cwnd (variabile in base a timeout e ACK) e $Swnd = \min(Rwnd, Cwnd)$.

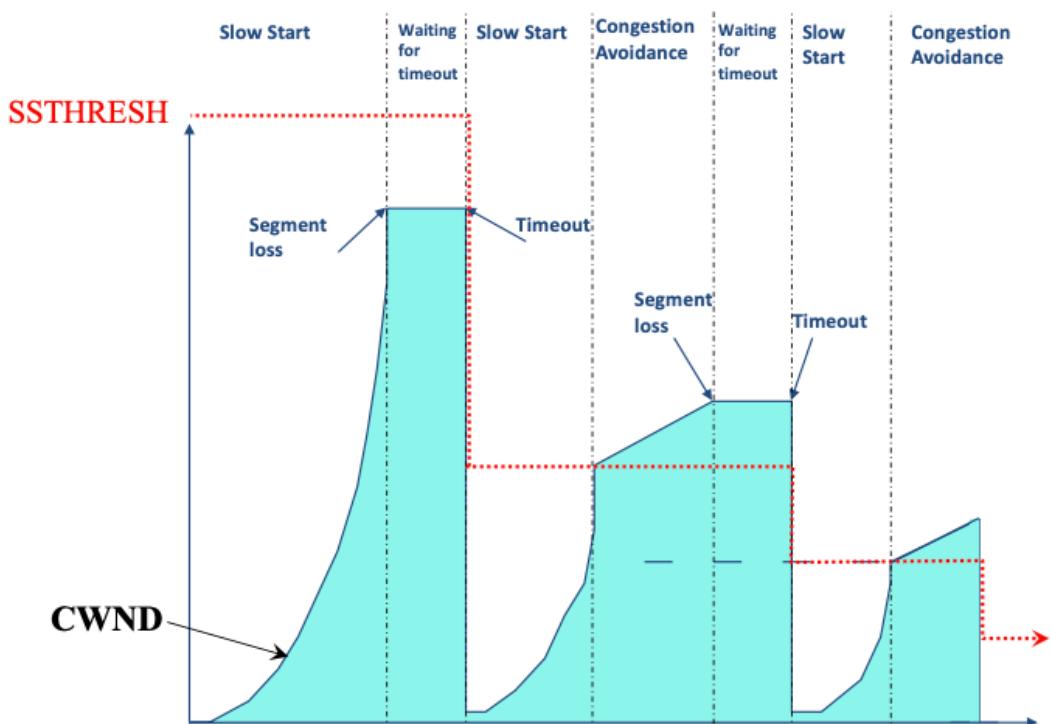
- Interpreta la perdita di un segmento come un evento di congestione, reagisce riducendo l'ampiezza della Cwnd
- L'aggiornamento della finestra di connessione avviene in due fasi, distinte da una soglia detta *Ssthresh*
 - *slow start*, all'inizio trasmettitore mette $Cwnd = 1MSS$ e $Ssthresh = high | Cwnd < Ssthresh$. Dopodiché, la Cwnd viene aumentata (esponenzialmente, e.g. 1, 2, 4, 8, ecc.) di 1MSS ∀ ACK ricevuto.⁴⁸ L'incremento continua fintanto che non si entra in *congestion avoidance*, si

⁴⁷ Dopo il primo non ripetuto; inoltre, richiede che il *delayed ACK* sia disabilitato.

⁴⁸ In realtà, la finestra è espressa in numero di byte, non MSS, che utilizziamo comunque per semplicità.

verifica il primo evento di congestione oppure $Cwnd \geq Rwnd$, anche se in quest'ultimo caso $Cwnd$ continua ad aumentare ma la $Swnd$ è vincolata a $Rwnd$.

- Congestion avoidance, in questa fase si attua un incremento lineare della $Cwnd$ pari a circa $\frac{1}{Cwnd} \text{ACK}$ ricevuto; dalla formula osservo che aumenterà molto più lentamente.
- Assieme alla finestra aumenta rate di trasmissione, $R = \frac{Cwnd}{RTT} \left[\frac{\text{byte}}{\text{s}} \right]$.
- Se scade un timeout sia in *slow start* che in *congestion avoidance*, allora la finestra non cresce più e viene modificato il valore di $Ssthresh$ e di $Cwnd$ come segue, $Ssthresh = \max\left(2MSS, \frac{\text{FlightSize}^{49}}{2}\right)$ e $Cwnd = 1$ (*TCP Tahoe*). Si noti che la $Ssthresh$ non viene modificata se la scadenza del timeout è relativa ad una ritrasmissione (RFC 5681). Quindi si tornerà alla fase di slow start, dove il segmento inviato per primo sarà quello a cui è scaduto il timeout. Con le versioni moderne di TCP(Reno) il ricevitore accetta fuori sequenza e poi invia ACK cumulativo appena riceve segmenti mancanti, mentre con le versioni base (Tahoe) il trasmettitore invia tutti i segmenti successivi al mancante anche se già trasmessi e correttamente ricevuti (destinatario li ha eliminati).
- Il valore di $Ssthresh$ corrisponde a stima della finestra ottimale, che eviterebbe future congestioni.



⁴⁹ Rappresenta il numero dei "byte in volo"; solitamente sono pari a $Swnd = \min(Rwnd, Cwnd)$.



Condivisione equa delle risorse: in condizioni ideali, il meccanismo di controllo del TCP è in grado di limitare la congestione in rete e consentire l'equa multiplazione della capacità dei link tra i diversi flussi; queste condizioni *ideali* sono alterate da differenti RTT (per diversi flussi) e da buffer nei nodi minori del prodotto banda-ritardo.

TCP Tahoe: versione base del protocollo TCP, non implementa il *fast retransmit* e accetta fuori sequenza solo all'interno della numerazione della *Rwnd*.

TCP Reno: *fast retransmit/fast recovery* in grado di accettare i fuori sequenza dopo 3 ACK ripetuti, con $Ssthresh = \max\left(2MSS, \frac{FlightSize}{2}\right)$ e $Cwnd = Ssthresh$ (così si salta la fase di *slow start*).

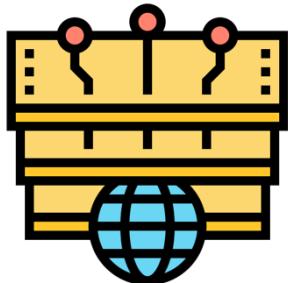


TCP Vegas: cerca di evitare la congestione invece di reagire con i metodi precedentemente descritti.

Altre versioni: ne esistono altre, anche quelle utilizzate per contesti wireless come *Westwood* e *Tibet*.

Network layer – introduzione

È il livello che si occupa di trasferire i dati tra gli host che ospitano due processi comunicanti, attraverso i nodi di rete.



I protocolli dello strato di rete sono implementati sia in host che router, dove il network layer trasferisce i segmenti dello stato di trasporto dall'host sorgente al destinatario: i segmenti sono incapsulati alla sorgente in datagrammi IP, inoltrati hop-by-hop dai router fino a destinazione, dove saranno letti dallo strato di trasporto ricevente.

Inoltre, offre servizi di *indirizzamento*, *inoltro* e *instradamento* (possibile con algoritmi di routing); in particolare, il servizio di instradamento prevede l'utilizzo delle *tabelle di routing* che contengono gli indirizzi dei prossimi router o del destinatario.



Network layer – indirizzamento IPv4

È un numero binario di 32bit, per facilitarne la lettura viene suddiviso in 4 campi da 8bit ciascuno con valori che vanno da 0 a 255. È univocamente associato ad un’interfaccia di rete di un host/router⁵⁰; deve avere valenza universale (in tutta la rete) poiché il routing è basato sull’indirizzo IP dell’host di destinazione.

Struttura indirizzo IP: ogni blocco di indirizzi è associato ad una rete e caratterizzato dall’avere i primi n bit (prefisso) identici, questo perché i primi n bit identificano la rete, mentre i rimanenti $32 - n$ bit sono usati per un host specifico (interfaccia) nella rete.

CIDR: il *classless inter-domain routing* è un sistema di indirizzamento che prevede l’allocamento di indirizzi contigui ad una rete, senza distinzione di classe (A,B,C,D, ecc.); dato l’indirizzo di esempio 192.168.0.1/16 il valore ’16’ indica quanti bit sono stati riservati alla *net mask*, il prefisso di rete che identifica la sottorete nella quale si trova l’host (contestualmente alla sua appartenenza ad una rete più grande, come internet stessa).

- La *netmask*, è un numero binario a 32 bit associato ad una rete IP, ha n bit posti a 1, mentre tutti gli altri sono 0. È utilizzata per ottenere un indirizzo di rete a partire dall’indirizzo di un host e la relativa netmask con l’*AND logico bit a bit*.
- Una rete IP è identificata da una *netmask* ed è un insieme di interfacce fisicamente interconnesse, deve essere presente almeno un router con un apposita interfaccia collegata alla rete per comunicare con altre reti IP.
- *Indirizzi speciali*, sono indirizzi non utilizzabili per identificare gli host.
 - *Network address*, ha i bits dell’hostId tutti a 0, e.g. 192.168.0.0/16, identifica la rete.
 - *Broadcast address*, ha i bits dell’hostId tutti a 1, e.g. 192.168.255.255/16, è utilizzato per mandare messaggi a tutti gli host della rete.⁵¹
 - *Host corrente-rete corrente*, è l’indirizzo 0.0.0.0/X (indirizzo sorgente) si utilizza quando l’host non conosce il proprio indirizzo IP; viene usato fintanto che non gli viene assegnato un IPv4 valido.
 - *Unicast*, è 0.0.X.X/16 dove solo il NetId è impostato a 0, indica uno specifico host nella rete corrente ed è usato nel campo destinatario di un pacchetto IP.
 - *Loopback*, è 127.X.X.X utilizzato solitamente per la comunicazione tra processi diversi, i pacchetti contrassegnati con questo indirizzo tornano al mittente senza uscire dall’host.

⁵⁰ Non è associato ad un intero host poiché può avere più interfacce di rete, come ad esempio ethernet e WiFi.

⁵¹ Esiste anche il *limited broadcast*, dove l’indirizzo 255.255.255.255 utilizzato nel campo destinatario di un pacchetto IP indica che il messaggio deve essere mandato a tutti gli host all’interno della rete.

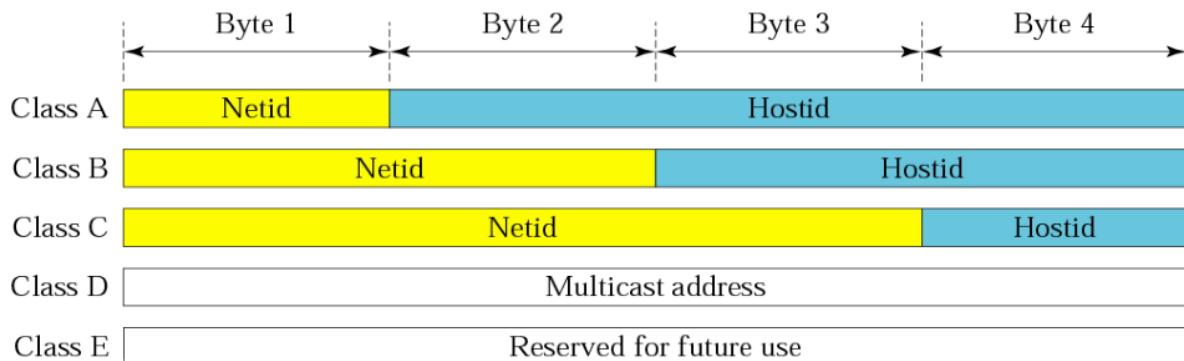
Indirizzi privati: utilizzabili in ambito privato, non univoci in internet, sono tre blocchi, $10.0.0.0 \rightarrow 10.255.255.255$, $172.16.0.0 \rightarrow 172.31.255.255$, $192.168.0.0 \rightarrow 192.168.255.255$.

Un router non può inviare un pacchetto che ha come destinazione un IP privato su un'interfaccia con IP pubblico, deve prima utilizzare un NAT⁵².



Curiosità: gli indirizzi IP vengono assegnati dalla IANA, organizzazione no profit che coordina e pianifica a livello mondiale l'assegnazione di indirizzi IPv4 e IPv6; i blocchi di indirizzi vengono mandati a 5 *regional internet registries (RIRs)* che a loro volta possono delegare (parzialmente) ai *local internet registries/ISP*. Tutti gli indirizzi IPv4 sono stati esauriti nel 2011, per questo è stato introdotto lo standard IPv6, che permette di identificare in modo univoco 10^{38} hosts, almeno a livello teorico.

Indirizzamento classfull: per indirizzi IPv4, prevede la distinzione in gruppi di questi, dette classi, come si può osservare nel seguente schema.



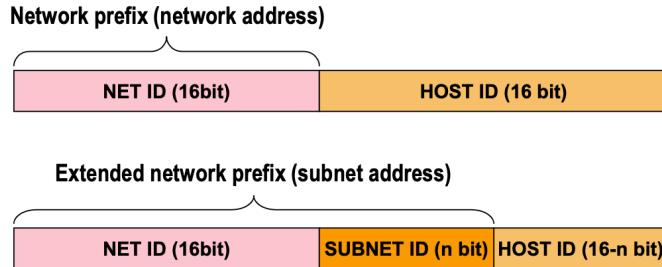
Ma quelle utilizzate nella sistematica sono solitamente le prime tre, rispettivamente le classi A, B e C.

Class	Mask in binary	Mask in dotted-decimal
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

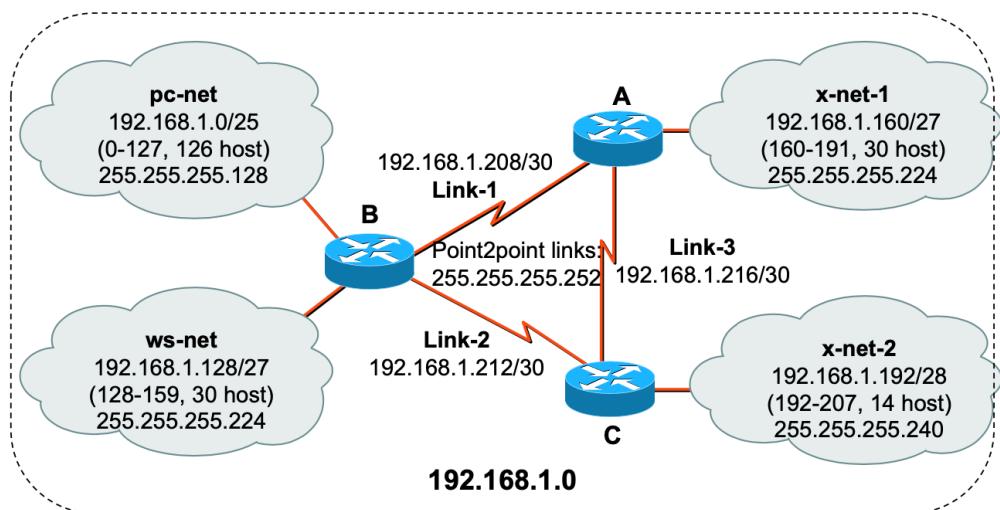
Rispetto al CIDR, permette di non utilizzare la *netmask* poiché il router riesce a dedurla dall'indirizzo stesso; tuttavia l'utilizzo del sistema classfull porta ad uno spreco o ad una carenza di indirizzi, un problema risolto invece nel CIDR che elimina la distinzione tra classi.

⁵² NAT è l'acronimo che sta per *Network Address Translation*, sarà approfondito di seguito.

Subnetting: suddivisione delle grandi reti IP in sottoreti/*subnet* che corrispondono ciascuna ad una rete fisica diversa; ciò è possibile utilizzando alcuni dei bit destinati agli host per identificare la sottorete, con un risultato simile a come si può vedere nel grafico sottostante.



- Qualora il subnetting classico non bastasse, allora si può utilizzare la VLSM⁵³ applicando due *subnetmask* in cascata. A titolo esemplificativo si osservi la rete seguente.



⁵³ VLSM è l'acronimo che sta per *Variable Length Subnet Mask*, è uno standard che permette di aggiungere subnet mask in cascata, ad esempio posso aggiungere due sottoreti a .../24 utilizzando .../25.

Network layer – inoltro IPv4

Una rete IP è un insieme di host interconnessi, identificata da un NetID; solitamente coincidono con reti di livello 2.⁵⁴ In questo tipo di rete gli host sono identificati attraverso il MAC Address, proprio di ciascuna scheda di rete.

IP: tecnica di internetworking che sfrutta la capacità di inoltro delle varie reti locali tra loro interconnesse, dove vengono trasmessi pacchetti encapsulati in trame di livello 2, usa il MAC address.

- *inoltro diretto*, se la destinazione è nella stessa IP net.
 - B deve spedire un pacchetto ad A
 - B conosce IPv4 della propria interfaccia e confrontandolo con quelle di A capisce che sono nella stessa rete
 - B consulta tabella di corrispondenza tra IP e MAC per reperire il MAC address
 - B passa il pacchetto al livello 2 che crea una trama e la invia a MAC di A
- *Inoltro indiretto*, se la destinazione non è nella rete locale.
 - *B deve spedire un pacchetto a D (fuori dalla sua rete)*
 - B conosce IP della propria interfaccia e dal confronto con IP-D capisce che non sono nella stessa rete
 - B inoltra quindi il pacchetto al router designato (solitamente di default)
 - B recupera MAC del router nella tabella di corrispondenza e passa il pacchetto al livello 2

Dual homing: non solo i router, questo fenomeno indica la possibilità per un host di avere più interfacce di rete, cioè appartiene di fatto a IP net differenti. Non è possibile, invece, assegnare due interfacce della stessa macchina alla stessa rete IP.

Router: dispositivo di internetworking con differenti interfacce di uscita; inoltre, anche essi seguono le regole definite precedentemente per l'inoltro diretto e indiretto.

Inoltro nei router: l'operazione di inoltro sfrutta le tabelle di routing, inoltre, ha le seguenti caratteristiche.

- L'inoltro fino a destinazione avviene fra router posti tra reti IP (*next-hop routing*)
- Pacchetti inoltrati solamente su base NetID destinatario

⁵⁴ Ne sono un esempio le LAN Ethernet.

- Tutti gli host connessi alla rete sono espressi nelle tabelle di inoltro del router come un'unica entry
- Nelle tabelle di routing è necessario specificare il prefisso di rete, utilizzando la *netmask*, per ciascuna riga
- I *protocolli di routing* devono trasportare l'info sul prefisso in ciascun *route advertisement* (annuncio di rete)

Come si può capire se il destinatario appartiene alla stessa rete del mittente?

Prima dell'inoltro del pacchetto si compiono due operazioni: AND tra indirizzo interfaccia e netmask interfaccia e AND tra indirizzo destinatario e netmask interfaccia. Se i precedenti risultati coincidono allora si trovano entrambi nella stessa rete, altrimenti si dovrà ricorrere all'inoltro indiretto.

Anche nel caso dell'indirizzamento indiretto si procede allo stesso modo per tutti gli indirizzi presenti nella tabella di routing; così, quando si avrà un riscontro si sceglierà di inviare il pacchetto su quella rete, nel caso di più riscontri si sceglie sempre quello con la netmask a maggior numero di bit a 1 (*longest prefix match*).

Supernetting: tecnica per definire indirizzi utilizzando una subnet mask arbitraria magari per permettere di avere altre sottoreti (al suo interno) oppure per evitare sprechi di indirizzi.

Route aggregation: i blocchi di indirizzi sono assegnati su base geografica; inoltre, la crescita delle tabelle di routing è mantenuta sotto controllo.

Exception route: dovuto al caso precedentemente accennato dove si usa il *supernetting*, quando una delle sottoreti potrà essere raggiunta solamente da una strada differente dalle altre. Si ricade in questa casistica nell'eventualità in cui via sia un trasferimento di un ramo d'azienda oppure un cambio di ISP.

Riduzione della RT: in generale le regole da seguire per l'aggregazione possono essere riassunti nelle seguenti quattro.

1. Si possono aggregare gruppi di reti contigue che hanno lo stesso next-hop. Ovviamente il numero di reti deve essere una potenza di 2 (gruppi di 2, 4, 8, ... reti). Il gruppo è sostituito da un'unica riga che contiene l'aggregato, ovvero la supernet, ed è ottenuto accorciando la netmask.
2. Si possono aggregare reti contigue come nella prima regola anche se per alcune il next-hop è diverso. In questo caso il gruppo è sostituito da un'unica riga che contiene l'aggregato, più una riga per ciascuna delle righe del gruppo con diverso next-hop (exception route) che sono lasciate inalterate.

3. Si possono aggregare reti contigue come nella prima regola anche se mancano nella tabella alcune reti. In questo caso il gruppo è sostituito da un'unica riga che contiene l'aggregato, più una riga per ciascuna delle reti mancanti con next-hop pari a quello della rotta di default.
4. Si possono eliminare tutte le reti con next-hop pari alla rotta di default, a meno che non ricadano nello spazio di indirizzamento di una supernet che aggrega altre subnet ma con diverso next-hop rispetto alla rotta di default.

Nota: la riduzione risulta particolarmente utile per una delle tipologie di esercizi presenti all'esame.

Network layer – Network Address Translation NAT

Le reti private, o *intranet*, utilizzano tecnologia di interconnessione IP e sono dotate dei medesimi servizi dell'internet.



L'evoluzione dei servizi e dei protocolli della rete globale ha reso le reti private strutturalmente differenti rispetto l'internet. Tra queste, anche problematiche relative alla sicurezza, gestione degli indirizzi e distinzione tra servizi offerti fra utenti privati e pubblici.

Indirizzamento privato: necessario per garantire la separazione dei servizi tra host interni ed esterni.

Ma gli utenti privati possono accedere ad Internet ed ai suoi servizi?

Proxy: è un application gateway, qualunque richiesta è inviata al proxy che la inoltra con il proprio IP (che sarà pubblico), ne occorre uno per applicazione.

NAT: il *network address translation* table (RFC2663/3022) realizza il mapping 1:1 tra indirizzi pubblici e privati. Per ridurre il numero di indirizzi pubblici necessari è stato introdotto il *network address port translation* che sfrutta il numero di porta per individuare la connessione, associando ad un host interno una stessa coppia di indirizzi pubblici.

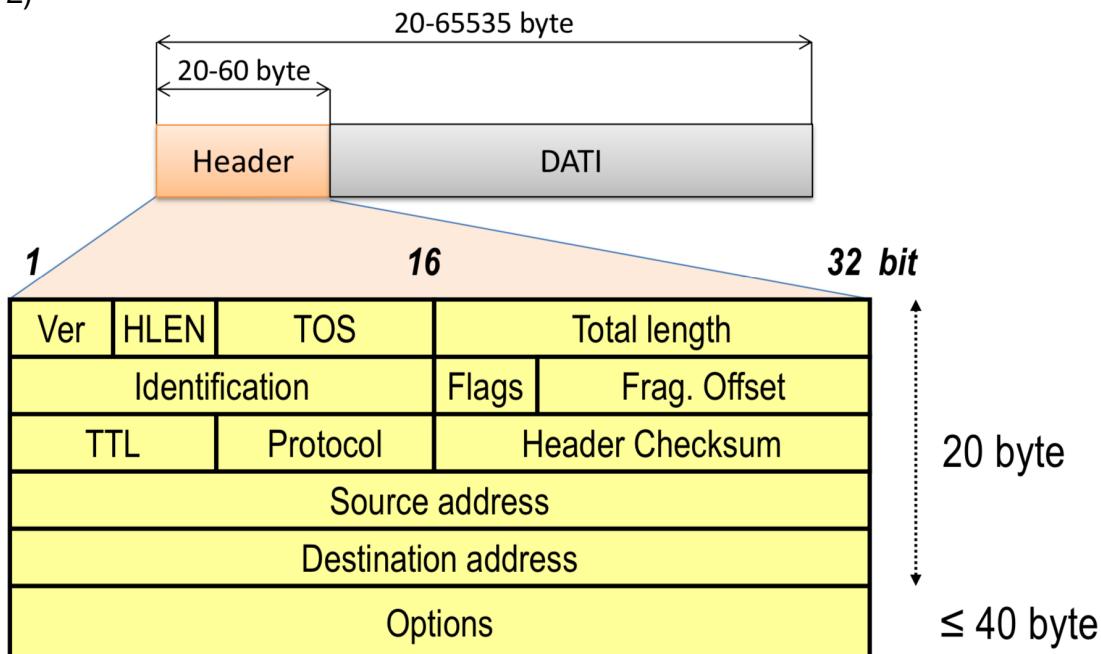
Limitazioni: con un solo IP pubblico non è possibile avere più server dello stesso tipo nella intranet, poiché i server usano porte dipendenti dal protocollo applicativo. Esistono modi per aggirarlo come l'utilizzo della *connection reversal* oppure del protocollo *Universal Plug and Play*.

Network layer – Protocollo IP

Il servizio di comunicazione offerto dall'IP si occupa di trasmettere datagrammi/pacchetti IP, è connectionless e utilizza la commutazione di pacchetto (ciascuno inviato indipendentemente dagli altri). Poiché utilizza una consegna *best effort* e la ritrasmissione è affidata ai livelli superiori, è un servizio *non affidabile*.

Svolge anche funzioni di:

- *Indirizzamento*, con indirizzi IP universalmente riconosciuti
- *Rivelazione di errori*, svolta sull'header
- *Max – lifetime*, datagram eliminati quando finisce il TTL (*Time To Live*)
- *Frammentazione/riassemblaggio*, (dei pacchetti) se richiesto dalla rete locale (livello 2)



IP source/destination address: a 32 bit ciascuno, contengono l'IP address del mittente e del destinatario.

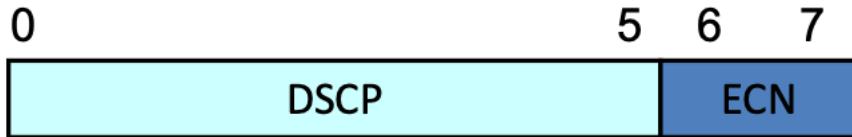
Ver: campo a 4 bit che indica la versione del protocollo (IPv4 o IPv6) utilizzato.⁵⁵

HLEN: campo a 4 bit, indica la lunghezza dell'header.

Total length: a 16 bit, indica la lunghezza totale del pacchetto in byte ($l_{max} = 2^{16} - 1 = 65535$).

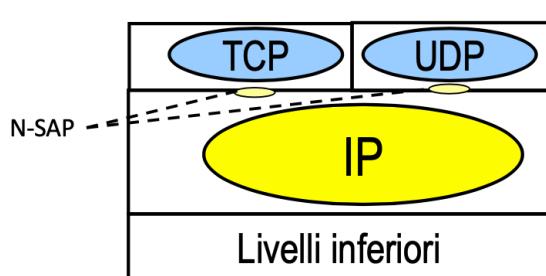
⁵⁵ Se la versione del protocollo utilizzata dal pacchetto non corrisponde a quella implementata dal router ricevente, allora il pacchetto viene scartato.

TOS: campo a 8bit, il *Type Of Service* ha assunto diversi significati nel tempo, attualmente i primi 6 bit si occupano di DSCP (*Differentiated Services Code Point*) per la scelta dei servizi da utilizzare, mentre gli ultimi due sono per l'ECN (*Explicit Congestion Notification*) che segnala al router ricevente un imminente congestione (con conseguente drop di pacchetti).



Time To Live (TTL): a 8 bit, scarta i datagrammi che “superano” il massimo tempo di vita, è espresso in numero di *hop* (salti tra un nodo della rete ed un altro); decrementato di 1 per ogni salto compiuto, viene scartato non appena $TTL = 0$.⁵⁶

Protocol: a 8 bit, indica il protocollo di livello superiore; si tenga presente che più protocolli *superiori* possono utilizzare IP, grazie alla *multiplazione*.



Valore	Protocollo
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Header checksum: a 16 bit, aggiunge protezione al solo header IP.

Options: ha dimensione variabile da 0 a 40 byte, inizialmente pensato per funzioni di testing e debugging, ora viene generalmente ignorato dai router.

Padding: usato per ottenere header IP di lunghezza multipla di 32 bit.

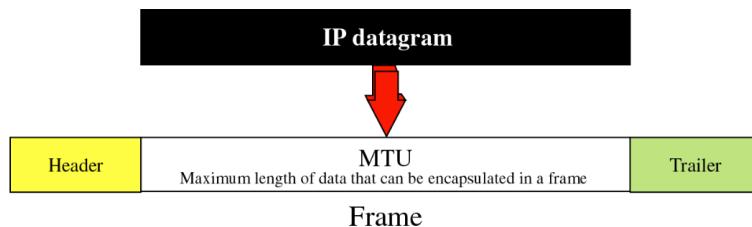
Frammentazione e riassemblaggio: i link impongono una dimensione massima alle trame, detta *Maximum Transmission Unit* (MTU), che potrebbe comportare la suddivisione di un singolo datagram in più trame da riassemblare una volta giunte a destinazione.

- Porta un maggiore overhead
- Ogni frammento viene gestito come datagram indipendente

⁵⁶ TTL viene decrementato in uscita dal router, pertanto se arriva con valore 1 all'ultimo router, riuscirà ad arrivare a destinazione. Nello scarto viene generato un messaggio di errore che viene inoltrato alla sorgente.

- Gestito da IP, nonostante sia un vincolo stabilito al livello 2

Rete	MTU (byte)
Maximum	65.535
Default	576
FDDI	4352
Ethernet/Fast Ethernet	1500
PPPoE	1492
PPP (low delay)	296



Identification: a 16 bit usato nella frammentazione, identifica univocamente tutti i frammenti dello stesso datagram.

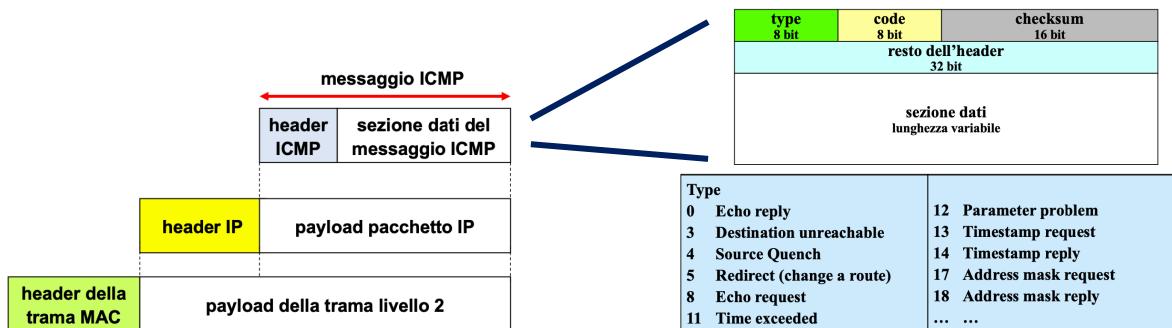
Fragment Offset: a 13 bit, riporta l'offset del primo byte trasportato nel frammento rispetto al riferimento 0 del pacchetto originale (in parole da 8 byte).

Flags: a 3 bit, dove sono utilizzati solo il secondo ed il terzo bit, $M = 0$ solo nell'ultimo frammento (indica *More fragments*); mentre $D = 1$ quando si vuole che non venga applicata la frammentazione (indica *Do not fragment*).

Network layer – Protocolli di Controllo

Sono protocolli utilizzati in aggiunta all'IP, nelle pagine seguenti una descrizione delle funzioni svolte e del funzionamento di ICMP, ARP/RARP e DHCP.

Internet Control Message Protocol (ICMP): RFC792, trasporta messaggi di servizio da router/host verso host; i messaggi sono trasportati da IP datagram, che avrà indicato nel campo *protocol* dell'header IP che si tratta di un pacchetto ICMP.



In particolare, si possono distinguere i messaggi di *error reporting* 3-4-5-11-12 e di *diagnistica* 0/8-13/14-17/18-10/9.

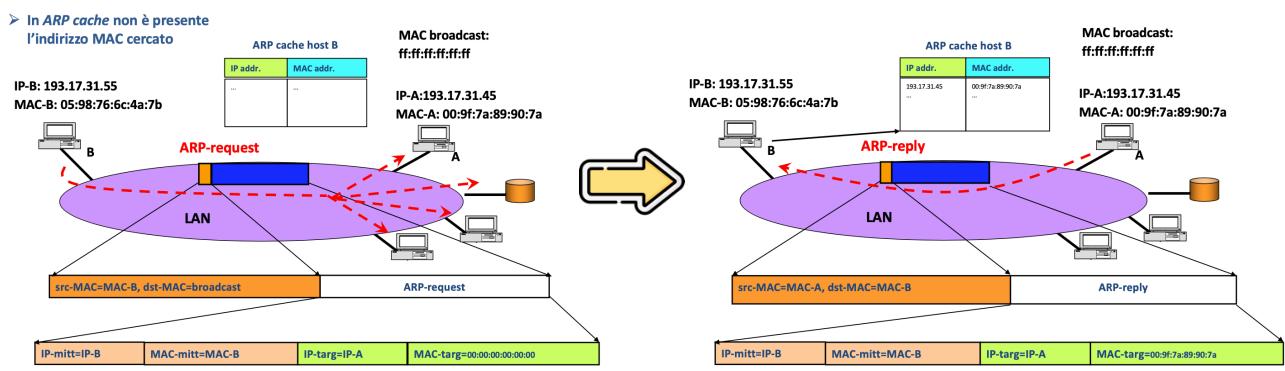
- *Error reporting*, dove l'evento di errore è notificato alla sorgente del pacchetto che ha generato l'errore, gestisce eventi di:
 - *destination unreachable* (3), nel campo *code* si indica la causa dell'errore, generato solamente se il router è in grado di accorgersi del problema.
 - *time exceeded* (11), se *code* = 0 è stato inviato dai router quando *TTL* = 0, altrimenti è stato inoltrato dalla destinazione se non tutti i frammenti sono arrivati entro un tempo massimo.
 - *parameter problem* (12), se *code* = 0 l'header ha un'incongruenza in qualcuno dei suoi campi, parte dell'header del messaggio è usata come *pointer* al byte che ha causato l'errore (del pacchetto inviato); se *code* = 1 allora un'opzione richiesta non è implementata o manca qualche parte del campo opzioni.
 - *redirection* (5), messaggio utilizzato quando si vuole che la sorgente usi un diverso router per la destinazione.
- *Diagnistica*, prevede coppie di messaggi di tipo domanda/risposta, saranno analizzate due delle quattro tipologie previste:
 - *Echo Request/Reply* (8,0), utilizzati per verificare la raggiungibilità/lo stato di un host/router; una volta che il destinatario riceve il messaggio, risponde inoltrando un messaggio al mittente con le informazioni richieste. Il campo *identifier* è scelto dal mittente e ripetuto nella risposta dal destinatario; richieste diverse possono avere stesso *identifier* ma diverso *sequence*

number. Se inserito campo optional data dovrà essere riportato identico dal destinatario nella risposta.⁵⁷

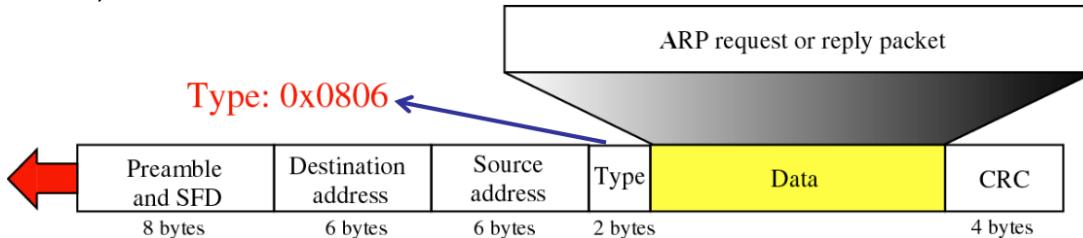
- *Address Mask Request/Reply (17/18)*, utilizzato per conoscere la netmask di un host/router, il campo *address mask* viene riempito dal destinatario.

Inoltre, il payload di questi messaggi contiene l'header del pacchetto che li ha generati + i primi 8 byte di dati.

Address Resolution Protocol (ARP): RFC826, utilizzato (da una stazione) per mappare un indirizzo di livello 3 in uno di livello 2. In particolare, si preoccupa di creare ed aggiornare una tabella contenente indirizzi IP e corrispettivi MAC address, sono generate da ciascun host sfruttando questo protocollo. Le richieste sono inviate in broadcast.



Nonostante sia un protocollo di servizio IP, i messaggi ARP sono incapsulati in una trama (UI di livello 2).

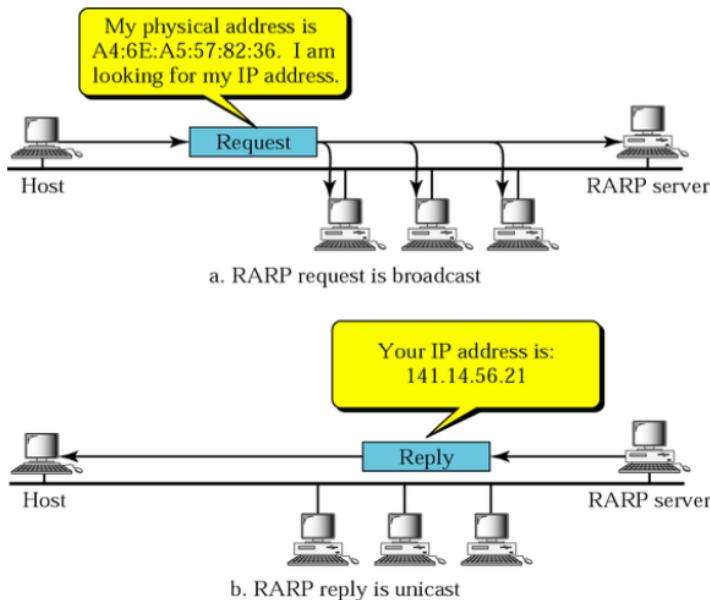


I casi d'uso del protocollo ARP sono essenzialmente 4,

- Un host ha un pacchetto da mandare ad un host nella stessa rete
- Un host ha un pacchetto da mandare ad un host su un'altra rete (passa per un router)
- Un router riceve un pacchetto da inoltrare ad un host su un'altra rete passando per un router di destinazione
- Un router riceve un pacchetto da consegnare ad un host nella stessa rete.

⁵⁷ Un messaggio di tipo ECHO è utilizzato nel *Traceroute*, dove vengono inviati messaggi consecutivi con TTL crescente fino alla ricezione di una risposta da parte del destinatario.

Reverse Address Resolution Protocol (RARP): RFC903, utilizzato dagli host che non conoscono il proprio IP address; ciascuna rete prevede un *Server RARP* per assegnare l'indirizzo IP su richiesta. Tali richieste sono inviate in modalità broadcast, attualmente è stato sostituito da protocolli come il DHCP.



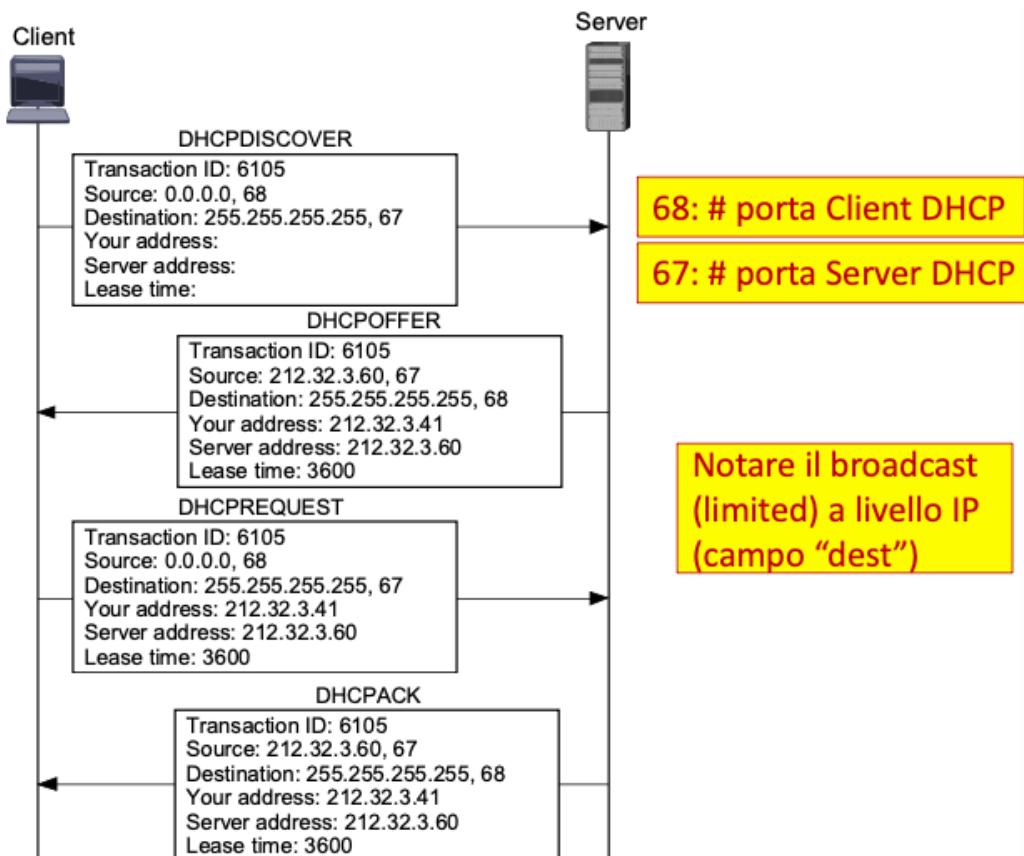
58

Dynamic Host Configuration Protocol (DHCP): RFC2131, assegna dinamicamente gli indirizzi IP agli host. Viene adottato per contesti dove gli host sono spesso inattivi o utilizzano IP solo per rari scambi di informazioni, è necessario utilizzare un apposito server che memorizzi le configurazioni di indirizzi sfruttando l'associazione automatica.

- L'associazione è temporanea, si utilizza un timeout oppure si implementano procedure di rilascio esplicito.
- È possibile che all'arrivo di una nuova richiesta non vi siano indirizzi disponibili, in questo caso viene rifiutata.
- È possibile avere più server DHCP, oppure utilizzare un DHCP relay, nel caso in cui il server DHCP non si trovasse nella stessa LAN del client.
- La procedura di assegnazione segue delle tappe ben definite,
 - Client che deve configurare il stack IP invia messaggio DHCPDISCOVER in broadcast, contenente il proprio MAC
 - Server risponde con un messaggio DHCPOFFER con il proprio identificativo e un indirizzo IP da utilizzare
 - Client può accettare l'indirizzo inviando DHCPREQUEST in BD che contiene l'identificativo del server

⁵⁸ Lo schema descrive in modo sintetico il funzionamento del protocollo RARP.

- Server crea l'associazione con l'indirizzo IP e manda al client un messaggio DHCPACK con tutte le informazioni di configurazione.⁵⁹
- L'indirizzo viene rilasciato dal client con un messaggio DHCPRELEASE
- I messaggi DHCP sono incapsulati in un segmento UDP (simile a protocollo applicativo); di seguito uno schema che descrive in linea di massima lo scambio di messaggi tra client e server DHCP.

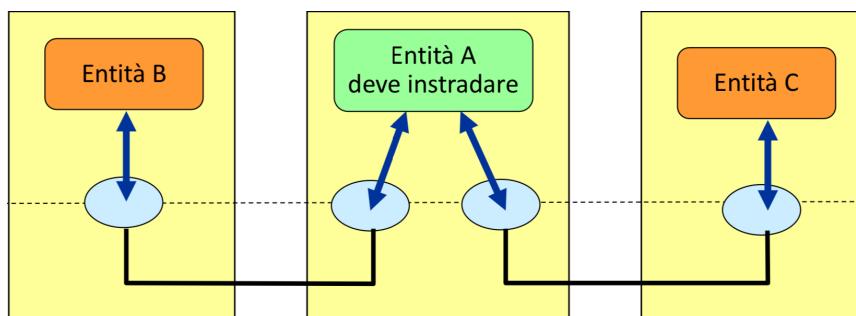


⁵⁹ I parametri di configurazione solitamente sono: *IP address, netmask, default gateway, DNS server*.

Network layer – Algoritmi di instradamento

L'instradamento, o *routing*, consente a due nodi non direttamente collegati di comunicare sfruttando gli altri nodi della rete.

Il cammino è basato sulla commutazione (*forwarding*) verso il SAP d'uscita, sulla base di un indirizzo o di un'etichetta posta sul pacchetto; la corrispondenza tra indirizzo e SAP d'uscita è salvata dai nodi in apposite *tabelle di routing*.



Algoritmi di instradamento: definiscono i criteri di scelta del cammino nella rete, servono per costruire le tabelle di routing consultate dai router per il forwarding dei pacchetti; anche la tipologia di rete adottata (datagram o circuito virtuale) ha un peso nella scelta dell'algoritmo più adatto.⁶⁰ Solitamente si utilizzano i grafi per effettuare i dovuti calcoli, costituiti da router (nodi) e link (archi) alle reti.



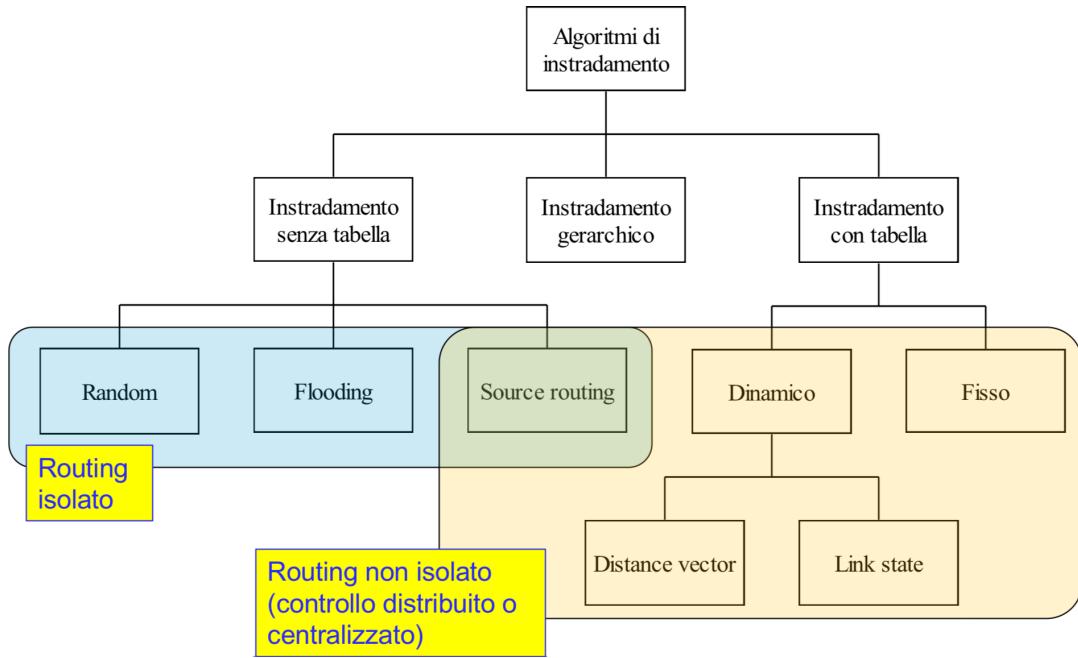
Osservazione: da un lato il protocollo regola lo scambio di informazioni sulla raggiungibilità fra router, mentre l'algoritmo si preoccupa della costruzione delle tabelle di routing/scelta del percorso migliore su base delle informazioni scambiate.

Requisiti: un algoritmo d'instradamento deve rispettare le seguenti caratteristiche.

- *Semplicità*
- *Robustezza*
- *Stabilità*
- *Ottimalità*

Le informazioni scambiate sono essenzialmente pacchetti: di segnalazione per i servizi a circuito virtuale e di dati in servizi datagram. Invece, per determinare la soluzione di instradamento si possono vi sono due opzioni, dove nella prima gli algoritmi sono centralizzati ed un unico centro (di controllo) prende decisioni, nell'altra si hanno algoritmi distribuiti dove tutti i nodi cooperano per determinare il miglior instradamento (\forall nodo).

⁶⁰ In alcuni casi, l'inoltro indiretto può essere effettuato senza ricorrere a tabelle di routing, sfruttando l'inoltro randomico (anche detto *flooding*).



Algoritmi di instradamento con tabella: basati su instradamento a distanza minima, motivo per cui è necessario definire una metrica di costo⁶¹; le tabelle utilizzate indicano per ogni destinazione il nodo successivo verso cui instradare il pacchetto.

- *Instradamento statico*, è il più semplice poiché non richiede segnalazione tra router e le tabelle sono compilate manualmente dall'amministratore della rete. Per questo la creazione di una nuova rete prevede una *fase di design* (progettazione delle rotte tra le sottoreti) ed una *fase di configurazione* (compilazione delle tabelle di routing con appositi comandi).
 - è preferibile utilizzare questa tecnica alla “periferia” della rete, nei sistemi a bassa connettività (e.g. ISP) o per necessità di *load-balancing*.
- *Instradamento dinamico o adattivo*, sono quelli utilizzati in Internet dove le *entry* delle tabelle cambiano in funzione di:
 - *Guasti/failures* nei link
 - *Cambiamenti* nella topologia di rete (e.g. aggiunto un router)
 - *Carico di rete e congestione*, quando un link è poco usato fa in modo di sfruttarlo di più

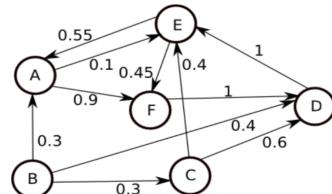
Nelle prossime pagine saranno analizzati gli algoritmi basati sui grafi, i cui calcoli sono effettuati sui modelli delle reti; ogni arco/link ha associato un peso opportunamente scelto (*metrica di costo*). I cammini minimi verso una destinazione formano un grafo detto *albero dei cammini minimi* (*minimum spanning tree*, *MST*); esistono algoritmi per calcolare MST che possono essere eseguiti in modo distribuito dai router.

⁶¹ Alcuni esempi sono il numero di salti(hop), capacità dei link, ritardo dei link e numero medio dei pacchetti in coda sull'intero percorso.

Per semplicità, di seguito è riportata una diapositiva riassuntiva sulle convenzioni e nomi da adottare nell'analisi dei grafi.⁶²

Cenni sui grafi

- **digrafo** (grafo diretto) $G(N,A)$
 - N : insieme dei **nodi**
 - $A=\{(i,j), i\in N, j\in N\}$: insieme degli **archi** (coppia ordinata di nodi)
 - Se i link sono bidirezionali si parla di "grafo non diretto"
 - **percorso**: (n_1, n_2, \dots, n_l) insieme di nodi con $(n_i, n_{i+1}) \in A$
 - **cammino**: percorso senza nodi ripetuti
 - **ciclo**: percorso con $n_1 = n_l$
 - **digrafo connesso**: per ogni coppia i, j esiste almeno un cammino da i a j
 - **digrafo pesato**: d_{ij} peso associato all'arco $(i,j) \in A$
 - **costo (lunghezza)** di un cammino (n_1, n_2, \dots, n_l) :
- $$d_{n_1, n_2} + d_{n_2, n_3} + \dots + d_{n_{l-1}, n_l}$$



Distance vector: sfrutta l'instradamento a distanza minima, dove ogni tabella contiene la minima distanza da ogni altro nodo (e quale usare come *next hop*).

- *Forma distribuita*, dove ciascun nodo riceve dai suoi "vicini" la stima delle distanze (array delle distanze, da cui il nome, *distance vector*), somma la sua distanza dal vicino e scopre la distanza minima verso ogni altro nodo.

L'algoritmo DV invia il *distance vector* ai soli nodi adiacenti, periodicamente oppure in seguito ad un cambiamento nella topologia di rete: ∀ nodo ricalcola DV se cade/nasce una nuova linea a cui è connesso.

→ appena viene ricevuto un DV, il nodo aggiunge a ciascuna destinazione inclusa nel vettore il costo del link verso il vicino; per ogni destinazione,

- Se non era già inclusa nella tabella, aggiunge la coppia destinazione-distanza
- Se il next hop nella tabella corrisponde al mittente del DV, sostituisce l'informazione precedente con la nuova
- Se la nuova distanza è minore di quella scritta in tabella, allora aggiorna la tabella con i nuovi dati

⁶² (Proprietà) Se il nodo k è attraversato dal cammino a costo minimo da i a j , il sotto-cammino fino a k è anch'esso a costo minimo.

(Curiosità) Il problema del cammino minimo è di complessità polinomiale dove al crescere del numero di nodi in \mathbb{N} , il numero di operazioni per ottenere i cammini minimi cresce come un polinomio in \mathbb{N} .

- *Forma centralizzata*, applicazione dell'algoritmo di *Bellman-Ford* per il calcolo (centralizzato) dell'MST. Applicare l'algoritmo DV in forma distribuita corrisponde all'applicazione del Bellman Ford in ogni nodo.

d_{ij} = costo del link da i a j ($d_{ij} = \infty$ senza link diretto)

D_j^h = Costo della via a minimo costo da s a j con max h salti

s = sorgente,

h = salto/hop

1 $h = 1$

$$D_j^h = d_{sj} \quad \forall j \neq s$$

2 $h = h + 1$

$$D_j^h = \min_i \{D_i^{h-1} + d_{ij}, D_j^{h-1}\} \quad \forall j \neq s$$

3 If $D_j^h = D_j^{h-1} \forall j \neq s$

$$\text{then } h_{\max} = h - 1$$

else go to 2

In un grafo diretto,
funziona anche con
link di costo
negativo, a patto che
nella rete non
esistano cicli di costo
totale negativo

stop

Mentre lo pseudo-codice relativo all'algoritmo ed un esempio di funzionamento sono:

d_{ij} = costo del collegamento diretto da i e j
($=\infty$ in assenza del collegamento)

D_j^h = costo del percorso a minimo costo
da s a j con max h hop

$h=0$;

$D_s^h = 0$;

$D_j^h = \infty \quad \forall j \neq s$;

repeat

$h = h + 1$;

$D_j^h = \min_i \{D_i^{h-1} + d_{ij}, D_j^{h-1}\}$;

until $D_j^h = D_j^{h-1} \quad \forall j \neq s$

• Inizializzazione

– $D_s^h = 0$

– $D_1^0 = \infty$

– $D_2^0 = \infty$

• Prima iterazione

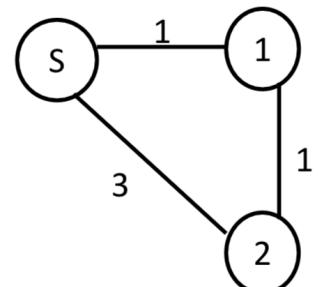
– $D_1^1 = \min(D_1^0, D_s^0 + 1) = 1$, NH:S

– $D_2^1 = \min(D_2^0, D_s^0 + 3) = 3$, NH:S

• Seconda iterazione

– $D_1^2 = \min(D_1^1, D_2^1 + 1) = 1$, NH:S

– $D_2^2 = \min(D_2^1, D_1^1 + 1) = 2$, NH:1



– Hp: costi unitari



– Nuovo nodo D raggiungibile in rete



		Node A		Node B		Node C		Events
Step	Dest.	Cost	Next hop	Cost	Next hop	Cost	Next hop	
Init								
1 ⁻	D							C-D = 1
1	D							C → B
2	D	3	B	2	C	1	D	B → A
3	D	3	B	2	C	1	D	

Problematiche: l'algoritmo *Distance Vector* ha propagazione veloce in caso di aggiunta di nodi alla rete, mentre la rimozione dei nodi può causare problemi all'algoritmo.

Count to Infinity: come precedentemente descritto, la rimozione di un nodo può comportare problemi, di fatto si avrà una distanza infinita nella tabella di routing (nella riga corrispondente alla destinazione del nodo mancante).

Come posso risolvere il Count to Infinity?

Split Horizon: è un rimedio al problema presentato, dove la stima della distanza verso una destinazione non viene trasferita sul collegamento utilizzato per raggiungere quella destinazione.

- *Split Horizon with Poisonous Reverse*, è una variante dove la stima è inviata a tutti i vicini, ma viene posta a ∞ se il collegamento è quello non funzionante.
- È un sistema che potrebbe non funzionare con alcune topologie, dipende dalla successione dei vettori inviati e ricevuti.

In breve, il Distance Vector è facile da implementare ma presenta alcuni svantaggi, come la bassa velocità di convergenza, il limite imposto dal nodo più lento, cicli necessari per la propagazione di un cambiamento nella topologia, difficilmente mantiene un comportamento stabile nelle grandi reti e presenta il problema del *count to infinity*.

Link State: algoritmo dove ogni nodo misura il costo di collegamento (con valori positivi) verso i propri vicini, inviando un vettore *link state*⁶³; la distanza viene comunicata con il flooding (*topology discovery*). Poiché ogni nodo ha completa visibilità sulla rete, ciascuno costruisce i propri percorsi a distanza minima.

- È necessario inviare nuovi LSP ogni qualvolta si verifichino variazioni od intervalli prefissati
- *Flooding*, ogni pacchetto in arrivo viene ritrasmesso su tutte le uscite eccetto quella da cui proviene. Per evitare i loop si utilizza numero di sequenza e si mantengono salvati gli SN dei precedenti LSP (non vengono ritrasmessi una seconda volta) oppure si utilizza un contatore di hop, come TTL.
 - Se LSP non è stato ricevuto o SN nuovo è maggiore di quello salvato, allora si salva LSP e lo si ritrasmette sulle uscite, altrimenti si ritrasmette quello in memoria al mittente.
 - Ogni nodo ha un database (degli LSP) in cui è descritta una mappa/grafico della rete.

⁶³ LSP, anche detto *link state packet*.

- L'instradamento a minima distanza è effettuato seguendo l'*algoritmo di Dijkstra*, che si applica a partire da un generico nodo sorgente, indicato con s .

N = nodi della rete

M = insieme dei nodi del MST corrente

$V(M)$ = nodi "vicini" all'insieme M

d_{ij} = costo della via diretta da i a j ($d_{ij} = \infty$ in assenza del link $i - j$)

D_j = costo della via a minimo costo da s a j

P_j = nodo predecessore del nodo j

$$1 \quad M = \{s\}$$

$$D_j = d_{sj} \quad \forall j \in N - \{s\}, D_s = 0$$

$$P_j = s \quad \forall j \in N - \{s\}$$

$$2 \quad \text{Select } k \in V \setminus M \quad | \quad D_k = \min_{i \in V \setminus M} D_i$$

$M = M \cup \{k\}$, connect k to P_k in MST

if $D_j > D_k + d_{kj}$ then $P_j = k \quad \forall j \in V \setminus M$

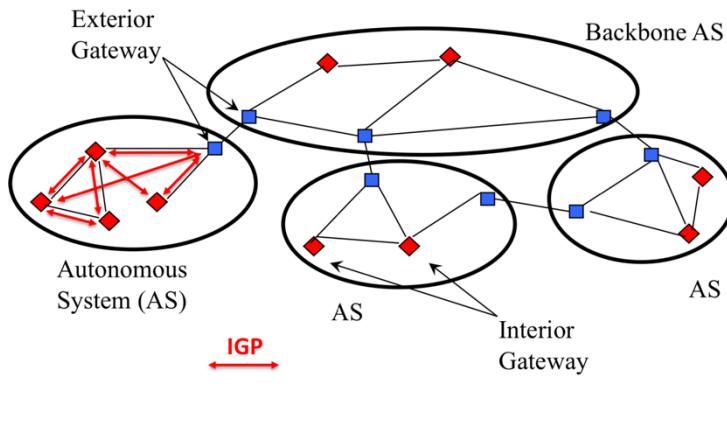
$$D_j = \min \{D_j, D_k + d_{kj}\} \quad \forall j \in V \setminus M$$

In breve, il link state è più flessibile (nodo ha mappa completa \rightarrow routing ottimale), si possono inviare LSP solo in seguito a cambiamenti e tutti i nodi vengono prontamente informati di eventuali variazioni topologiche. D'altro canto, è necessario un protocollo per mantenere l'informazione sui vicini (*Hello*), si deve utilizzare il flooding e sono necessari riscontri dei pacchetti di routing inviati; tutti questi meccanismi portano ad una maggiore complessità di implementazione.

Osservazione: gli algoritmi precedentemente presentati convergono alla stessa soluzione in condizioni statiche, distinguendosi in condizioni dinamiche, dove il primo è più lento ma semplice da implementare; il secondo invece converge rapidamente ma è di difficile implementazione.

Network layer – Instradamento IP

Le soluzioni precedentemente introdotte non sono adatte all'utilizzo sull'intera Internet, da motivi pratici fino a quelli relativi alla sicurezza, è preferibile invece rendere indipendente il routing all'interno delle singole reti autonome⁶⁴ che compongono la rete globale.



Interior gateway: router interno ad una rete autonoma. Si scambiano informazioni utilizzando un *Interior Gateway Protocol* (IGP); all'interno di un AS si ha completa condivisione delle informazioni topologiche.

Exterior gateway: router al "bordo" di una rete autonoma. Si scambiano informazioni usando un *Exterior Gateway Protocol* (EGP), ha un approccio diverso da DV/LS

Per giungere a destinazione, i pacchetti possono subire differenti tipologie di inoltro, in particolare si potrà incorrere nei routing:

- *Diretto*, quando NETID di $host_{sorgente} = host_{destinatario}$, inoltro mediante rete di livello 2 ($Host_ID \Leftrightarrow MAC$)
- *Indiretto*, quando NETID di sorgente e destinazione sono diversi ma appartengono allo stesso AS, inoltro attraverso interior gateways utilizzando IGP
- *Indiretto gerarchico*, quando sorgente e destinazione sono in AS differenti, il pacchetto viene instradato con IGP fino all'exterior gateway, che utilizzando EGP lo inoltrerà fino all'AS di destinazione e da qui fino al destinatario sfruttando nuovamente l'IGP.

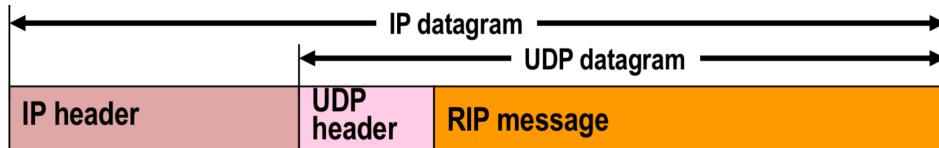
Routing Domain: indicato con RD, è una porzione di AS dove è implementato un solo IGP, può coincidere con l'AS stesso; alcuni router appartengono a più RD, permettendo la ridistribuzione delle informazioni di un dominio nell'altro (e viceversa).

Routing Information Protocol (RIP): standard RFC1058, è un IGP che si affida a UDP utilizzando la porta 520. Basato su Distance Vector con metrica di costo *hop-count*⁶⁵ e gli aggiornamenti sono inviati ogni 30 secondi circa; facile da implementare ma ha elevata complessità ($\sim O(n^3)$, $n = \text{numero nodi della rete}$), si possono verificare loop e *count-to-infinity*.

⁶⁴ Anche dette *Autonomous Systems* o AS; all'interno di una rete autonoma si possono configurare più IGP, a patto che sia garantita la consistenza del routing.

⁶⁵ La distanza massima è pari a 15 *hop*, mentre il valore 16 *hop* corrisponde ad una rete irraggiungibile.

- I messaggi RIP sono datagram UDP incapsulati in IP packets, dalla dimensione massima di $512\text{ Byte} = 500_{\text{Payload}} + 12_{\text{header}_{\text{UDP}}} + 12_{\text{header}_{\text{RIP}}}$, solitamente sono necessari diversi datagrammi per una singola tabella di routing.



- I messaggi RIPv1 hanno invece il seguente formato, dove
 - command*, indica il tipo di messaggio (request/responde, ecc)
 - version*, assume valore 1 o 2
 - address family*, stack protocollare usato (2=TCP/IP)
 - IP address*, indirizzo rete destinataria (4B per IPv4)
 - Metric/distance*, hop count da router che annuncia fino a destinazione (da 1 a 15)

0	7 8	15 16	31
Command (1-6)	Version (1)	0	
Address family (2)	0		
IP address			
0			
0			
Metric			

Si possono inserire fino a 24 rotte in più con lo stesso formato

Funzionamento RIP: di seguito le tappe e le operazioni svolte con l'esecuzione del protocollo.

- Inizializzazione*, i router inviano speciali richieste su ogni interfaccia, con parametri
 - command* = 1, *address family* = 0, *metric* = 16, cioè richiede ai router vicini le loro tabelle di routing, così si scoprono identità e distanze dei router adiacenti
- Una volta terminata l'inizializzazione si possono effettuare le seguenti operazioni.
 - Request*, richiesta di informazioni di routing su specifici indirizzi di rete; inviate da un router appena connesso oppure con entry in scadenza, si possono richiedere entry specifiche o intera tabella

Com: 1	Version	Reserved
Family	All 0s	
Network address		
All 0s		
All 0s		
All 0s		

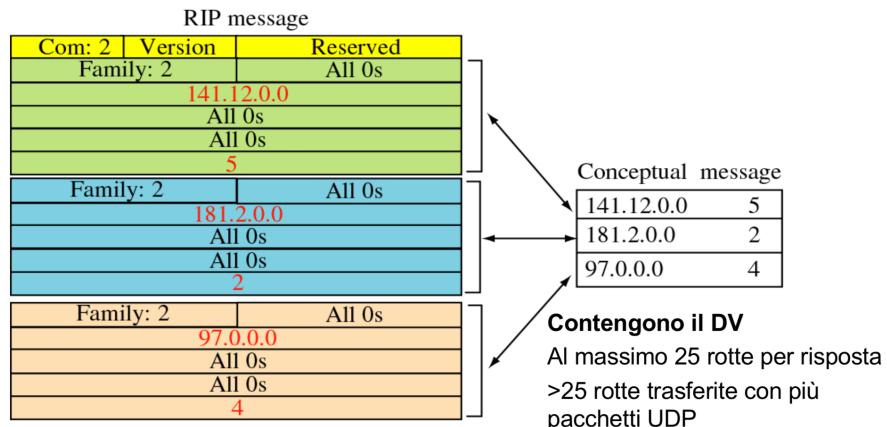
a. Request for some

Com: 1	Version	Reserved
Family	All 0s	
All 0s		

b. Request for all

- Response*, restituisce lista di indirizzi e relativo costo, se un router non ha informazioni per un indirizzo, questo avrà costo 16. Possono essere *solicited*

(triggered update) quando inviate in risposta a una Request, oppure *unsolicited* (regular update) se inviate periodicamente (30s)



- *Regular update*, ogni 30s i router inviano la tabella o parte di essa ai vicini, se una entry della tabella non viene aggiornata da 6 cicli (180s) viene posta a costo 16⁶⁶
- *Triggered update*, inviati non appena si verifica una variazione di costo di una rotta, trasmette solo informazioni sulle reti che hanno subito un cambiamento; funziona sia con *split horizon* e *poisonous reverse*

I problemi del RIPv1 riguardano sia la metrica di costo, troppo semplice, che la lenta convergenza e la limitazione alle piccole reti (due nodi a distanza maggiore di 15 sono irraggiungibili).

RIPv2: è un miglioramento della precedente che sfrutta i campi ex-vuoti in RIPv1, mantenendo la compatibilità con i router che lavoravano con la prima versione; introduce l'autenticazione, mentre i nuovi campi sono i seguenti.

0	7 8	15 16	31
Command (1-6)	Version (2)	0	
Address family (2)	Route tag		
IP address			
Subnet mask			
Next hop IP address			
Metric			

20 bytes

- *route tag*, ID dell'autonomous system
- *subnet mask*, aggiunge supporto al CIDR
- *next hop IP address*, indica IP address dell'interfaccia verso cui instradare i pacchetti diretti alla specifica sottorete

Open Shortest Path First (OSPF): è un IGP basato su Link State che utilizza l'algoritmo di Dijkstra, dove gli LSP sono inviati con flooding a tutti gli altri nodi all'interno dell'RD; gli LSP possono essere inviati periodicamente o in seguito a cambiamenti del costo dei link.⁶⁷ A differenza del precedente si adotta una metrica di costo generica e definita dall'amministratore di rete (in funzione di ritardo, costo, bit rate, ecc.), il costo è visto

⁶⁶ Nella realtà, per essere sicuri dell'effettiva irraggiungibilità, si aspettano altri 60s prima di settare la rete come irraggiungibile.

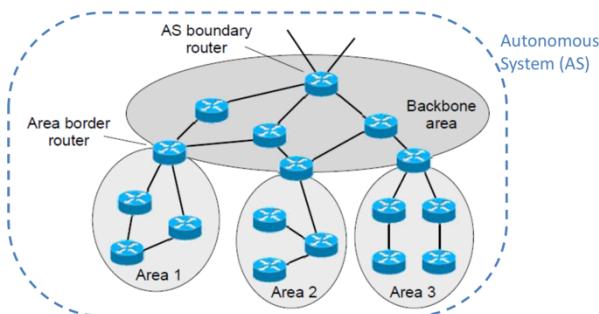
⁶⁷ Gli LSP contengono solo le informazioni sui link adiacenti a differenza di quanto accade nel RIP.

indipendentemente da ciascuno dei router adiacenti; inoltre, è definito sull'interfaccia di uscita e corrisponde ad un grafo con pesi non simmetrici su ogni link.

- *Load balancing* dinamico, ottenuto specificando rotte multiple, usato quando +rotte hanno stesso costo. Evita di caricare troppo una linea ma la consegna in sequenza dei pacchetti non è garantita.



- Supporta il routing gerarchico, con suddivisione in aree e redistribuzione dell'informazione di routing.



Gli area border router diffondono in ciascuna area informazioni riassuntive delle altre aree, indicando solo destinazione raggiungibili (non i router da attraversare). Prevede l'adozione dell'inoltro indiretto gerarchico.

- I messaggi sono incapsulati direttamente in IP (*protocol* = 89), implementa funzioni di livello trasporto (e.g. *messaggi ACK*). Esistono differenti tipi di messaggi, che hanno tutti identico header

Bit			
0	8	16	31
Version	Type	Message length	1
		Source router IP address	2
		Area ID	3
	Checksum	Authentication type	4
		Authentication (octets 0-3)	5
		Authentication (octets 4-7)	6

- *Hello*, testa la raggiungibilità dei vicini
- *Database description (LSDB)*, annuncia update del sender
- *Link state request (LSR)*, richiede informazioni su certo link
- *Link state update (LSU)*, fornisce costo dei link vicini (al sender)
- *Link state ACK (LSA)*, riscontro di LSU

Border Gateway Protocol (BGPv4): standard RFC4271, è un EGP che consente a router di diversi AS di scambiarsi informazioni di routing; si basa su un algoritmo *path vector*, dove i gestori decidono il routing in funzione delle proprie politiche (conoscendo i percorsi usati per raggiungere altri AS).

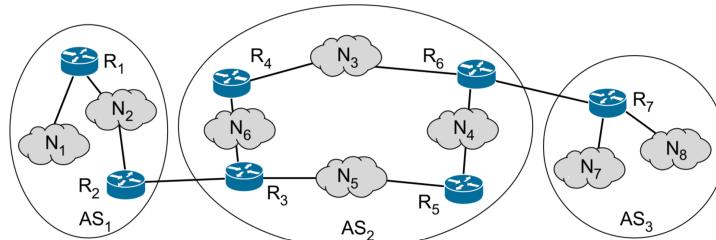


È indipendente dai protocolli IGP dell'AS e ne esistono due tipologie, la *eBGP* e la *iBGP*.⁶⁸

- *Path vector*, simile al *distance vector*, si discosta perché nelle informazioni scambiate non viene indicata la distanza dalla destinazione, ma l'intero percorso verso la destinazione, cioè una sequenza di AS da attraversare. Questo è possibile perché ogni AS è identificato da un *Autonomous System Number (ASN)* unico a livello globale (assegnato da IANA).

Anche se in realtà il messaggio contiene degli attributi aggiuntivi che si distinguono in obbligatori e facoltativi; tra gli obbligatori vi sono:

 - ORIGIN, protocollo IGP da cui proviene l'informazione
 - AS_PATH, sequenza di AS attraversati
 - NEXT_HOP, prossimo router
- Ogni BGP router invia il proprio path vector ai BGP router vicini sfruttando una connessione TCP (router to router, porta 179).
- Le tipologie di messaggio sono riassunte come segue:
 - OPEN, apre connessione TCP e gestisce autenticazione tra router
 - UPDATE, annuncia nuova rottura/eliminazione della vecchia
 - KEEPALIVE, mantiene connessione attiva in assenza di UPDATE⁶⁹
 - NOTIFICATION, notifica errori in messaggi precedenti⁷⁰



Messaggi eBGP iniziali inviati tramite connessioni TCP

- $R_2 \rightarrow R_3$: N₁, N₂, R₂, AS₁
- $R_3 \rightarrow R_2$: N₃, N₄, N₅, N₆, R₃, AS₂
- $R_6 \rightarrow R_7$: N₃, N₄, N₅, N₆, R₆, AS₂
- $R_7 \rightarrow R_6$: N₇, N₈, R₇, AS₃

Messaggi iBGP all'interno degli AS

- $R_2 \rightarrow AS_1$, $R_3 \rightarrow AS_2$, $R_6 \rightarrow AS_2$

Messaggi eBGP per redistribuire le nuove informazioni

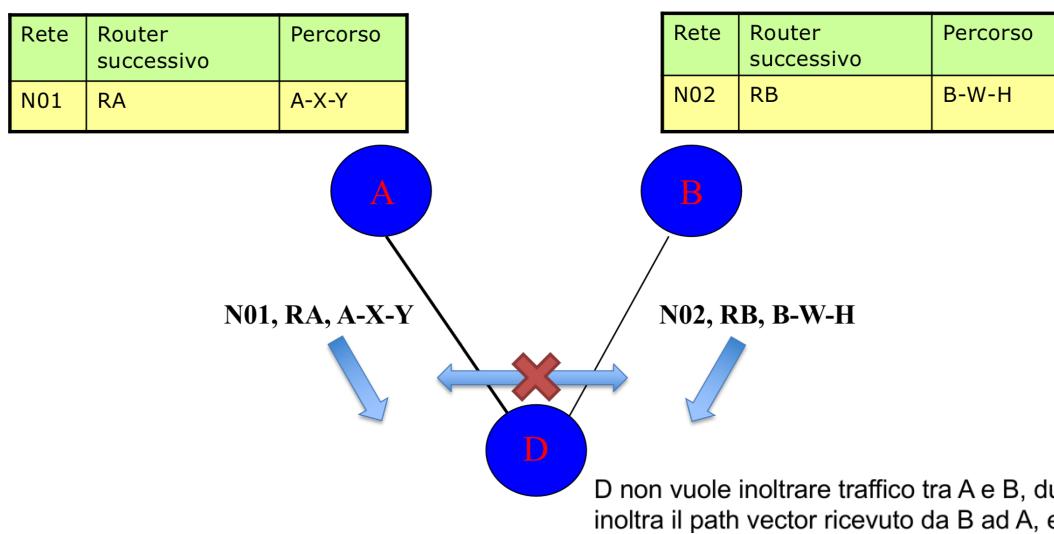
- $R_3 \rightarrow R_2$: N₇, N₈, R₃, AS₂, AS₃

⁶⁸ eBGP è usato da border router di AS diversi, mentre iBGP è usato da un border router per propagare l'informazione di routing dentro l'AS stessa.

⁶⁹ Viene anche utilizzato come ACK ai messaggi OPEN.

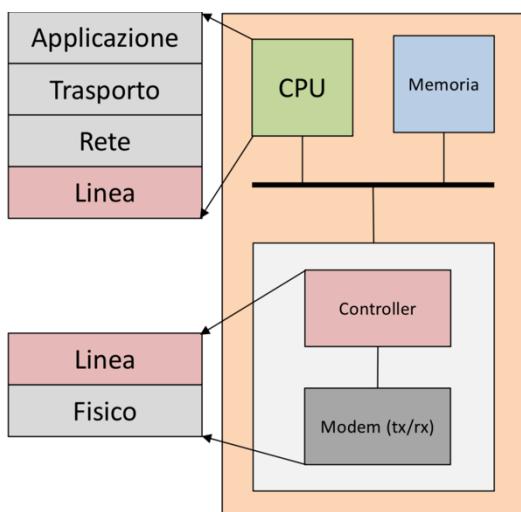
⁷⁰ Viene utilizzato anche per chiudere la connessione.

- *Policy based routing* prevede che sia l'amministratore scegliere come fare il routing e se propagare l'informazione ad altri AS; quindi, un router BGP che riceve un path vector da un peer, può:
 - Aggiungere alla propria tabella la destinazione specificata da PV
 - Inoltrare PV ai suoi vicini
- A seconda della politica di routing localmente implementata, di seguito un esempio.

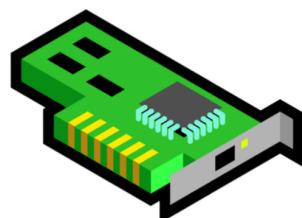


Data Link layer – generalità

Le funzionalità introdotte a questo livello riguardano *framing* (delimitazione trame), *gestione degli errori*, *indirizzamento*, *multiplazione* e *accesso multiplo* (su mezzo trasmissivo condiviso).



È parte della NIC⁷¹ e implementato (insieme al livello fisico) su apposito chipset (controller dedicato), mentre alcune funzionalità come gestione degli indirizzi e preparazione della trama sono svolte dal software dell'host.



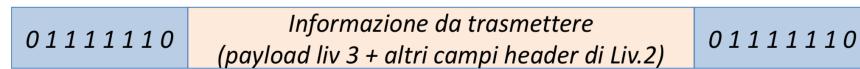
I collegamenti di linea sono solitamente di 3 tipi, *point-to-point*, *broadcast* e *commutati* (*variante p2p*).

⁷¹ NIC o *Network Interface Card*, è un componente fisico che si occupa di gestire trasmissione e ricezione in rete.

Data Link layer – collegamenti P2P

Framing: operazione di costruzione della trama dove in ricezione si riceve una serie di bit (dal livello fisico); prima si individua il significato logico dei bit appena scambiati, raggruppandoli in una struttura dati detta *trama*.

- Per identificare inizio e fine di una trama, si usano appositi *delimitatori di trama*, cioè dei *flag* costituiti da una particolare stringa di bit. Un esempio è l'HDLC, dove il flag corrisponde a 01111110, di seguito uno schema.

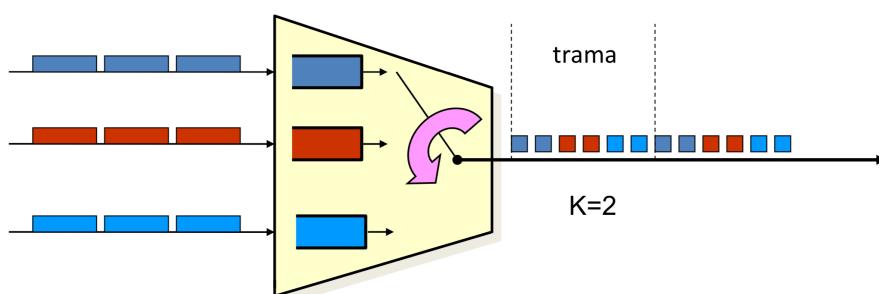


Per impedire una casuale presenza del flag nella trama si utilizza il *bit stuffing (HDLC)*, che -prima di aggiungere il flag- inserisce un bit 0 dopo aver osservato cinque bit a 1 consecutivi. Con l'operazione opposta (*bit destuffing*) verranno scartati gli 0 aggiunti con HDLC.

Gestione degli errori: la correzione degli errori è possibile con l'utilizzo dei codici correttori di tipo FEC (*forward error correction*), l'error detection già visto al livello trasporto, ha come obiettivo principale il recupero dei segmenti persi; diversa è la ritrasmissione, dove a questo livello l'ARQ viene usato per recuperare gli errori di livello fisico: a questo proposito si usa un apposito campo della trama detto *Frame Check Sequence (FCS)*.

Multiplazione: necessaria per permettere a più host di comunicare attraverso lo stesso canale trasmissivo, né esistono differenti tipologie.

- *SDM*, divisione di spazio, e.g. cavi con diverse fibre ottiche.
- *FDM*, divisione di frequenza, equivalente a WDM, dove la banda disponibile viene suddivisa in più canali, ciascuno utilizzabile in maniera indipendente dagli altri.⁷²
- *TDM*, divisione di tempo, l'intera risorsa è messa a disposizione degli utenti che hanno ciascuno un certo slot temporale in cui possono utilizzarla; e.g. rete telefonica digitale. I bit di N flussi sono raccolti in code e trasmessi sul canale d'uscita a gruppi di K (*interlacciamento*); ogni gruppo è trasmesso in un dato slot temporale.



Si noti che $durata_{trama} = t$ in cui su singolo canale in entrata arrivano numero di bit pari a quelli trasmessi nello slot della trama dedicato a quel flusso.

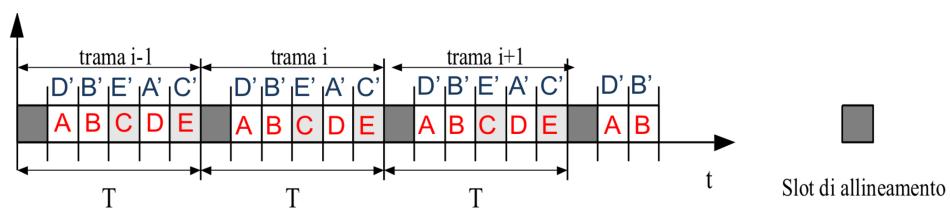
⁷² Si mantiene sempre un margine tra un canale e l'altro, per evitare interferenze.

- *WDM*, divisione di lunghezza d'onda/wavelength.
- *CDM*, divisione di codice.

Simplex: la trasmissione simplex è un canale di comunicazione unidirezionale.

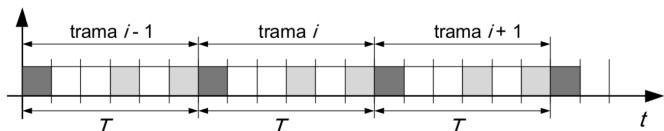
Duplexing: dato un canale di comunicazione bidirezionale, dove la trasmissione può essere *half* se avviene alternativamente tra le due direzioni, oppure *full* se avviene simultaneamente in entrambe le direzioni ⇒ il *duplexing* è la multiplazione utilizzata per condividere il mezzo fisico nelle due direzioni (FDD-TDD).

Trama (TDM): insieme di slot organizzati tali da avere stessa lunghezza, anche a livello temporale hanno la stessa “durata” T . Sono necessarie opportune strategie di allineamento, con “stringhe” di bit in posizione prefissata. Poiché la divisione è data dal tempo/periodo di trasmissione, si avranno la delimitazione implicita delle UI (no flag) e l’indirizzamento隐式 UI (impostazione write/read o mux/demux).



Di seguito la diapositiva riassuntiva sulla trasmissione delle trame in multiplazione TDM, con relative formule.

- T : durata trama [s]
- L_c : nr. bit per slot [bit]
- f_c : capacità del tributario [bit/s]
- N : numero slot di utente per trama [adimensionale]
- L_a : nr. bit per trama aggiunti per servizio (allineamento + segnalazione) [bit]
- f_a : capacità addizionale per servizio [bit/s]
- L_m : nr. tot. bit per trama ($L_m = N \cdot L_c + L_a$) [bit]
- f_m : capacità canale multiplato [bit/s]



Frequenza di cifra del segnale multiplato

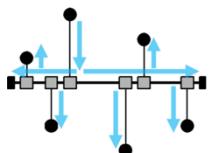
$$f_m = \frac{L_m}{T} = \frac{N \cdot L_c + L_a}{T} = N \cdot f_c + f_a$$

- Assegnazione canali
 - A slot singolo (a ciascun utente è assegnato 1 slot/trama): $f_c = L_c/T$ (es. 64 kbit/s)
 - A slot multiplo (sovramultiplazione): $f_c = n \cdot L_c/T$ con $1 \leq n \leq N$ (es. 384 kbit/s)
 - A frazione di slot (sottomultiplazione): $f_c = n/T$ con $1 \leq n \leq L_c$ (es. 8 kbit/s)

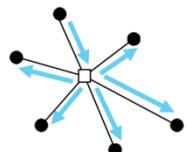
Data Link layer – collegamenti BC

Fin dalle prime versioni di Internet, le LAN hanno avuto una velocità maggiori, questo perché le trasmissioni avvenivano in modalità *broadcast* (BC): una rapidità derivata dal mancato utilizzo delle funzioni di rete come commutazione e/o switching. Ad oggi i mezzi fisici che usano broadcast sono i *WiFi* (canali radio a corto raggio) e le *passive optical networks* (fibra ottica, accoppiatore ottico o star coupler).

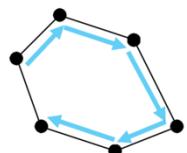
Topologie: di seguito le principali topologie di reti broadcast.



- *Bus*, trasmissione half/full duplex dove le stazioni sono solo sorgenti/destinazioni di UI



- *Stella*, trasmissione bidirezionale dove le stazioni sono solo sorgenti/destinazioni di UI, mentre il centro stella (HUB/stazione radio base) agisce come snodo.



- *Anello*, trasmissione unidirezionale con stazioni attive, che ritrasmettono anche le UI delle altre stazioni.

Problemi: sono propri della trasmissione broadcast, infatti sarà necessario un opportuno sistema di indirizzamento e un modo per gestire l'accesso multiplo degli utenti allo stesso mezzo fisico.⁷³

Come si possono risolvere?

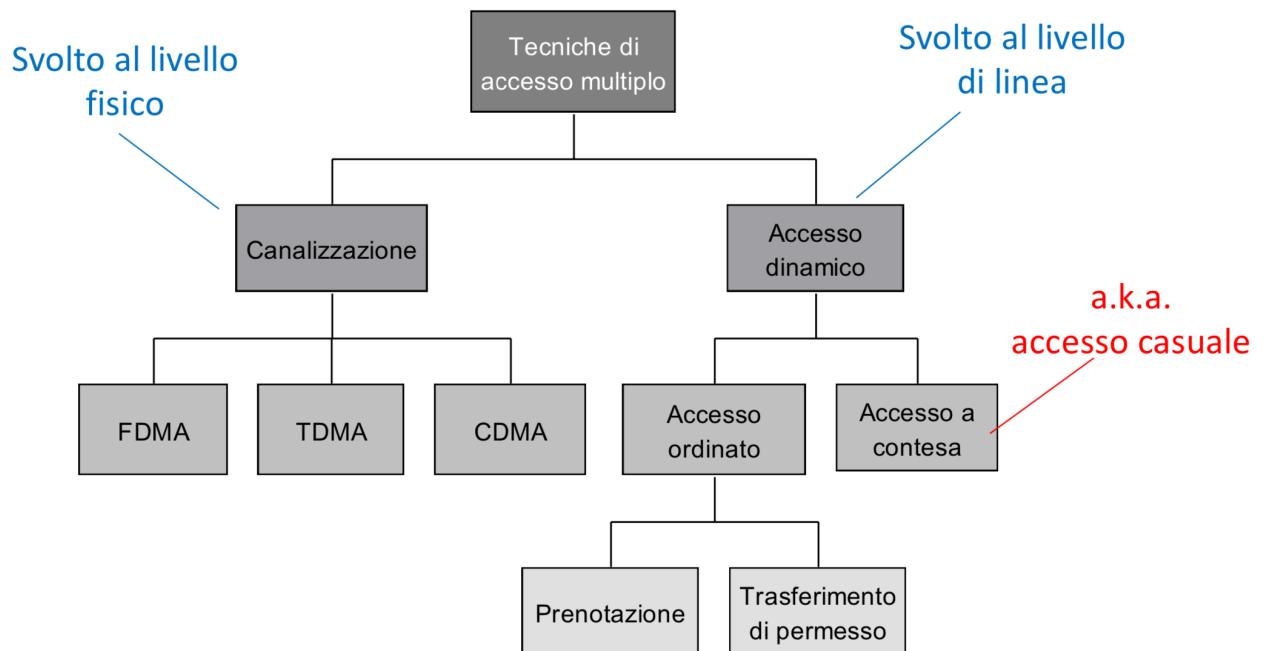
La scelta di accettare o meno una UI dipende dall'indirizzo MAC di destinazione, mentre per le collisioni (trasmissioni contemporanee), saranno necessari sistemi e tecniche di gestione più complessi.

L'accesso multiplo: consente di regolare l'accesso al canale (evitare collisioni) e può essere implementata sia a livello 1 che 2.

- A livello fisico, si dividono staticamente le risorse tra gli host (*protocolli di canalizzazione*)
- A livello del protocollo di linea, gestendo l'accesso pacchetto per pacchetto (*accesso dinamico*)

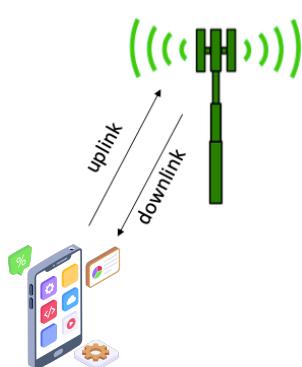
⁷³ Nella broadcast, tutti ricevono il segnale ma solo il destinatario analizza il contenuto della trama; viene utilizzato un mezzo trasmissivo (con possibili contese/conflicti e congestioni).

Quindi è possibile riassumere nel seguente schema, i protocolli e tecniche utilizzabili per evitare le collisioni.



Accesso multiplo con canalizzazione: l'accesso multiplo fisico è equivalente alla multiplazione ma con sotto-canali gestiti da trasmettitori differenti, le differenti tipologie sono descritte di seguito.

- *Frequency Division Multiple Access* o FDMA, a divisione di frequenza, e.g. canali WiFi o cellulari
- *Time Division Multiple Access* o TDMA, con slot temporali per la trasmissione delle diverse stazioni, il flusso di bit è generalmente asincrono⁷⁴; prima di inviare lo slot successivo, si aspetta per un tempo pari a τ (detto *tempo di guardia*)
- *Code Division Multiple Access* o CDMA, dove la distinzione avviene tramite opportuni codici (chip) detti *ortogonal*, è più complesso dei precedenti sistemi, ma porta ad un maggiore utilizzo delle risorse



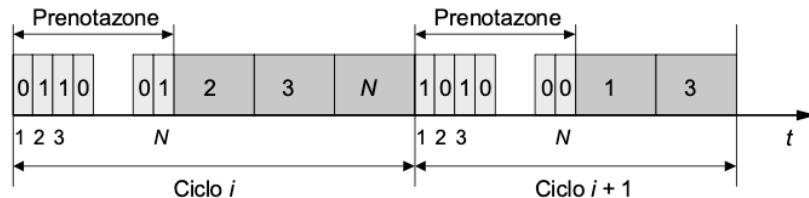
Duplexing e accesso multiplo: analogo a *duplexing point2point*, dove la capacità trasmissiva nei due versi può essere suddivisa per spazio, frequenza (*FDD*) o tempo (*TDD*).

⁷⁴ Il ricevitore deve sincronizzarsi su un particolare flusso, per questo motivo è necessario adottare dei *tempi di guardia* tra gli slot, che impediscono di “confondere” i flussi informativi di diverse stazioni. In altre parole, si aspetta che si liberi il canale dal messaggio precedente.

Ma l'accesso multiplo nella pagina precedente è realizzato dal livello fisico; invece, il livello data link si preoccupa di svolgere l'accesso *dinamico*, che sarà descritto di seguito.

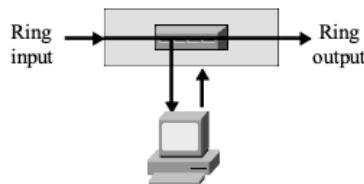
Accesso dinamico ordinato:

- Accesso a *prenotazione*, usato nelle topologie a BUS, prevede N stazioni con canale condiviso in modalità half-duplex, con l'asse temporale suddiviso in cicli di durata variabile.

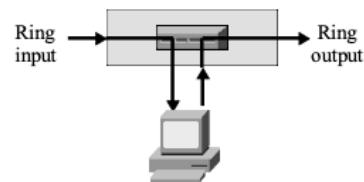


Un ciclo comprende N minislot per prenotazione e tanti slot di durata fissa quante prenotazioni.⁷⁵

- Accesso con *trasferimento di permesso*, usate nelle topologie ad Anello, prevedono un funzionamento in due stadi
 - Anello chiuso/ascolto, dove l'interfaccia di stazione riceve e ritrasmette i dati



- Anello aperto/trasmissione, ora l'interfaccia riceve i dati in ingresso e prima di ritrasmetterli inserisce i pacchetti che vuole inviare, riprendendo poi il normale funzionamento in anello chiuso



Accesso dinamico a contesa: ha l'obiettivo di regolare l'istante di inizio trasmissione delle singole trame, dove il coordinamento può essere gestito da un'entità centrale oppure in modo distribuito dalle stazioni stesse (ciascuna stazione decide autonomamente); in questo caso, il livello di linea è suddiviso in *Medium Access Control* (per accesso multiplo) e *Logical Link Control* (per altre funzioni di data link).

Il principio alla base del funzionamento è la ritrasmissione dopo un tempo casuale in seguito ad una collisione⁷⁶.

⁷⁵ È possibile utilizzare slot a durata variabile facendo in modo che un minislot comunichi la lunghezza dello slot dati prenotato.

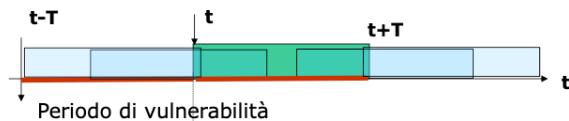
⁷⁶ Tale casualità conferisce, al sistema, una buona probabilità che la collisione non si ripresenti.

- ALOHA, protocollo semplice che utilizza la tecnica sopra descritta, dove la prima trasmissione avviene senza osservare il canale; per il rescheduling (in seguito a collisione) casuale delle UI viene utilizzato l'algoritmo di *backoff*: si sceglie di ritrasmettere dopo un multiplo i del tempo base T , tale multiplo è scelto casualmente in un intervallo, serve per distribuire il carico e ridurre il rischio di nuove collisioni.
 - Collisione, rilevata alla mancata ricezione ACK entro un certo periodo (e.g. $1RTT = 2\tau$)
 - T è il tempo di trasmissione \forall trama, τ è il tempo di propagazione tra le stazioni più lontane
 - L'analisi dell'efficienza è simile alla versione slotted, con differenza nel fatto che le collisioni sono possibili con sovrapposizione parziale dove il periodo di vulnerabilità è pari a $2T$,

$$P_S = (1 - p)^{2(N-1)}$$

E quindi:

$$S = Np(1 - p)^{2(N-1)} \quad S = G \left(1 - \frac{G}{N}\right)^{2(N-1)}$$



Dove si ha che $\lim_{N \rightarrow \infty} S = G \cdot e^{-2G}$ con massima efficienza $S = \frac{1}{e} \approx 0.37$.

- Slotted ALOHA, variante con asse dei tempi suddiviso, ogni slot ha $T = \frac{L_{trama}}{c_{link}}$; inoltre, tutte le stazioni sono sincronizzate, se 2+ stazioni trasmettono insieme avranno trasmissioni completamente sovrapposte.
 - Collisione, l'attesa dell'ACK sarà pari a $T_{attesa} = 2\tau + \epsilon$, l'errore è aggiunto perché il primo termine potrebbe non essere un multiplo di T
 - Come il precedente, con pochi host è sicuramente efficiente; poiché le collisioni si verificano solo se si trasmette nello stesso slot, il periodo di vulnerabilità coincide con T . Considerando invece uno scenario con N stazioni, ognuna con probabilità p di trasmettere,⁷⁷

\Rightarrow la probabilità che una stazione trasmetta con successo è $P_S = (1 - p)^{N-1}$

Se invece si intende il successo di uno slot generico $\Rightarrow P_S = p \cdot (1 - p)^{N-1}$.

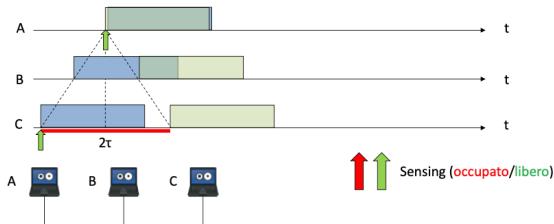
$$\text{Throughput} = S = N \cdot p \cdot (1 - p)^{N-1}, \quad \text{Traffico} = G = N \cdot p, \quad \Rightarrow p = \frac{G}{N}$$

⁷⁷ Come si osserva dalla formula, la probabilità che una stazione trasmetta con successo è pari alla probabilità che le altre $N - 1$ stazioni non trasmettano. (probabilità congiunta di eventi indipendenti)

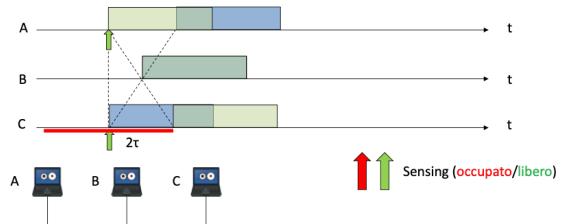
$S = G \cdot \left(1 - \frac{G}{N}\right)^{N-1}$ è il numero medio di successi (S) in funzione del numero medio di trasmissioni (G) e del numero di stazioni (N); in altre parole è una misura dell'efficienza.⁷⁸

- Carrier Sense Multiple Access (CSMA), a differenza degli ALOHA verifica che il canale sia libero prima di trasmettere⁷⁹, in caso positivo viene trasmessa la trama, altrimenti si aspetta un certo tempo (casuale) e si ripete il procedimento. Esistono versioni dette *persistenti* dove la stazione resta in ascolto e ritrasmette non appena il canale si libera.
 - Collisioni, anche in questo caso sono possibili, legate al τ , ad esempio si può avere una situazione dove

- C trova libero → trasmette
- A trova il canale libero → trasmette → collisione in B e A

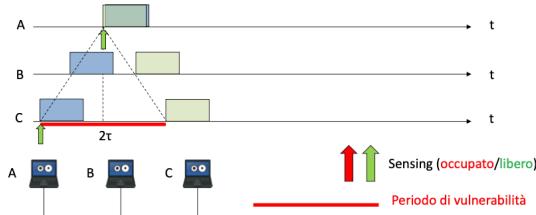


- C trova libero → trasmette
- A trova il canale libero → trasmette → collisione in C, A e B

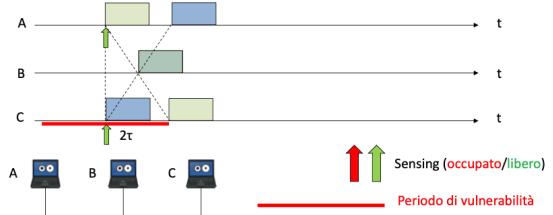


Questo avviene per $T > \tau$, mentre per il caso in cui $T < \tau$ si potrà avere

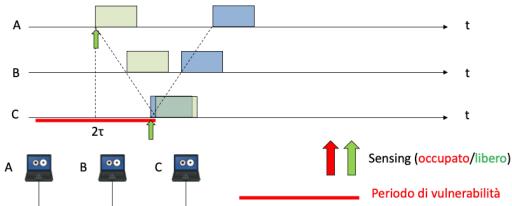
- Collisione in A



- Collisione in B

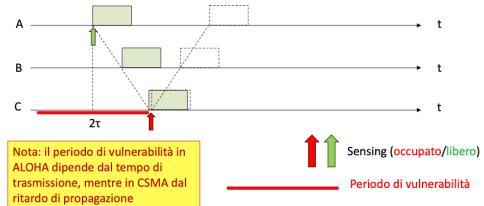


- Collisione in C



- Collisione evitata

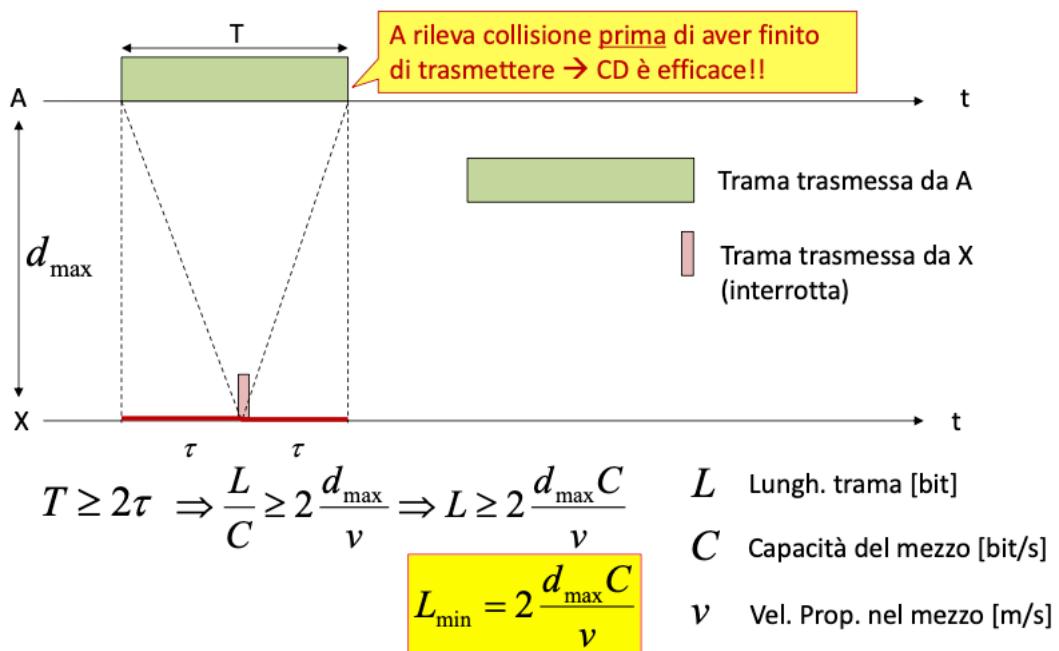
• E' come prima, ma questa volta il periodo di vulnerabilità è molto più grande rispetto al tempo di trasmissione → la probabilità di evitare collisioni col CSMA è molto più piccola



⁷⁸ $\lim_{N \rightarrow \infty} S = G \cdot e^{-G}$, mentre la massima efficienza si ha con una sola stazione che trasmette ⇒ $S = \frac{1}{e} \approx 0.37$.

⁷⁹ A livello fisico, è possibile rilevando la presenza di segnale (carrier o portante).

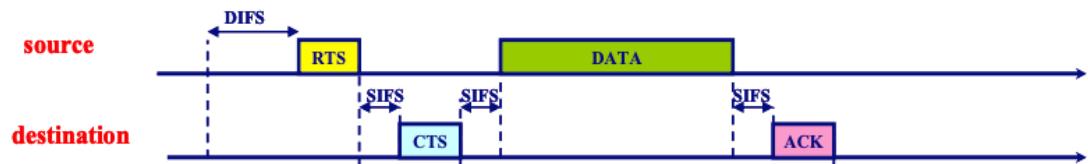
- Come nella ALOHA, anche CSMA può essere slotted, mentre come già anticipato, ha una versione persistente (rescheduling backoff oppure check del canale) che se slotted, ritenta allo slot successivo.
- Efficienza data da $\alpha = \frac{\tau}{T}$, mentre il throughput è $S = \frac{G \cdot e^{-\alpha G}}{G \cdot (1+2\alpha) + e^{-\alpha G}}$, si noti che per $\alpha \ll 1$ probabilmente si avrà un'efficienza elevata
- CSMA with Collision Detection (CSMA/CD), dove le stazioni possono identificare una collisione anche mentre trasmettono, in tal caso si interrompono e inviano un segnale di *jam* di durata γ (slot); a patto che $\tau < T$. Questo sistema riduce l'intervallo di collisione ed elimina la necessità di avere ACK ("no collisione no problemi").⁸⁰
 - [curiosità] si può calcolare il throughput come $S = \frac{G \cdot e^{-\alpha G}}{G \cdot (1+2\alpha) + e^{-\alpha G} - G \cdot (1-\gamma) \cdot (1-e^{-\alpha G})}$ mentre il throughput massimo sarà $S_{max} \approx \frac{1}{1+5\alpha}$.
 - Per consentire l'effettivo funzionamento del sistema del collision detection, è necessario stabilire delle relazioni tra i parametri, in particolare si avrà



Però il *collision detection* si basa sul fatto che nelle reti locali cablate l'attenuazione è piccola; quindi, il livello del segnale ricevuto dalle altre stazioni è simile (è facile accorgersi di una collisione). Per lo stesso motivo non è utilizzabile nelle reti wireless.

⁸⁰ È il sistema utilizzato dalle reti Ethernet.

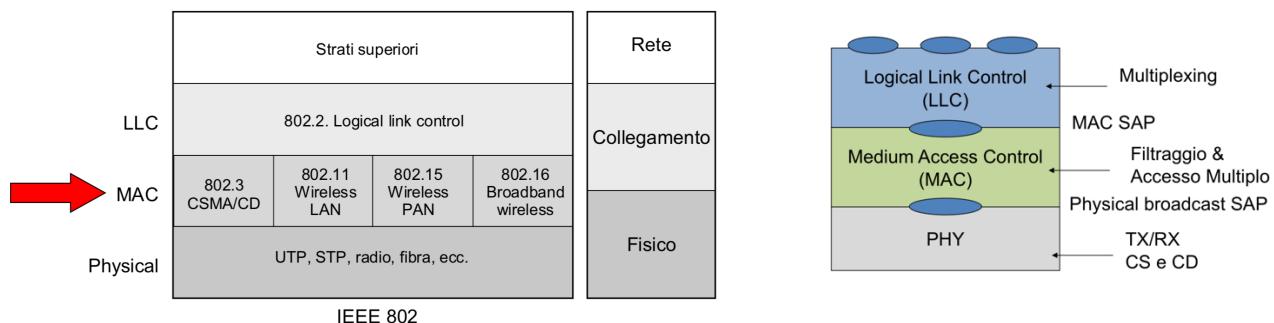
- CSMA with Collision Avoidance (CSMA/CA), resolve il problema appena descritto, il funzionamento può essere riassunto dallo schema seguente.



- RTS: Request to Send
- CTS: Clear to Send
- La collisione può avvenire solo su RTS
- Se si riceve il CTS si prosegue con la trama dati

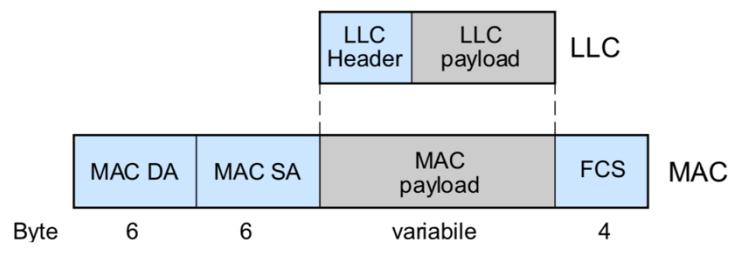
Data Link Layer – LAN specs

L'IEEE è il principale organismo che si occupa della standardizzazione delle tecnologie e dei protocolli di reti locali, di seguito due schemi che riassumono gli standard definiti.



Strato MAC: consente la condivisione del mezzo trasmissivo, poiché a livello di linea -le LAN- utilizzano mezzi fisici broadcast.

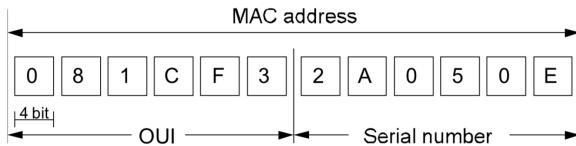
- *Trasmissione*, utilizza un protocollo di accesso multiplo, e.g. CSMA, CSMA/CD, CSMA/CA
- *Ricezione*, si usano indirizzi a livello MAC che trasformano le trasmissioni nel mezzo condiviso in comunicazioni *point-to-point*, *point-multipoint* e *broadcast*
- Le UI hanno la struttura seguente, dove
 - DA, destination address
 - SA, source address
 - FCS, campo a 32bit⁸¹
 - Può avere tre tipi di indirizzi di destinazione, Unicast, Multicast e Broadcast⁸²
- La scheda MAC che riceve una trama ne verifica l'integrità, analizza l'indirizzo e trasferisce la trama ai livelli superiori solo nei casi in cui abbia
 - MAC-DA broadcast
 - MAC-DA unicast con *indirizzo* = *indirizzo_{scheda}*
 - MAC-DA multicast con *indirizzo* ∈ *gruppo*



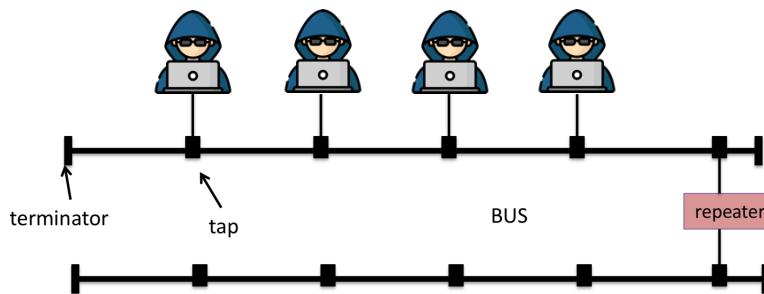
⁸¹ È il campo *Frame Check Sequence* (FCS) per il controllo d'errore a livello data link.

⁸² L'indirizzo MAC di Broadcast è FF:FF:FF:FF:FF:FF, viene utilizzata la codifica esadecimale.

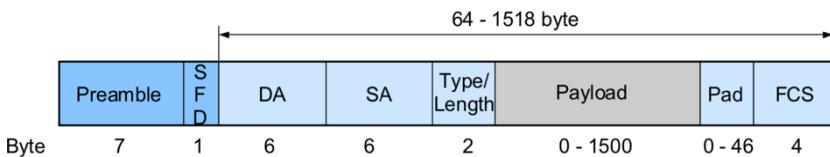
- Indirizzo MAC, attualmente è possibile riconfigurarlo (inizialmente univoco per ciascuna NIC), a 6 Byte è con salvato nella ROM della scheda di rete con 3B dedicati al codice del costruttore (OUI) e gli altri 3B per la numerazione progressiva della scheda



Ethernet IEEE 802.3: caratterizzata da una topologia a BUS⁸³ con capacità nell'ordine dei $10Mbps$ e che utilizza come metodo d'accesso il CSMA/CD. Di seguito le altre caratteristiche principali dell'Ethernet nella sua versione di base.



- Lo standard IEEE prevede anche un formato delle trame, con

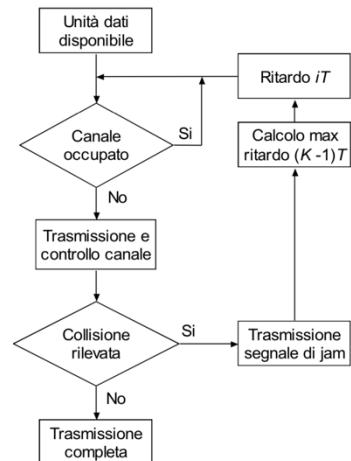


- Preamble*, consente sincronizzazione in ricezione
 - Start frame delimiter SFD*, flag
 - Destination address DA*
 - Source address SA*
 - Length*, lunghezza campo payload
 - Data*, lunghezza variabile
 - Pad*, garantisce la lunghezza minima della trama
 - Frame check sequence FCS*, usato per il controllo d'errore



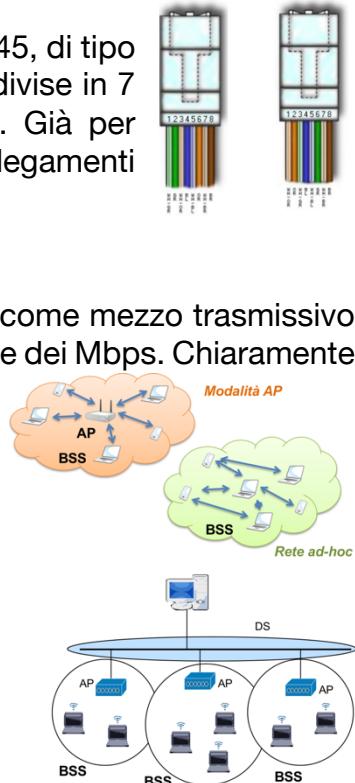
⁸³ In realtà a livello fisico può essere una rete sia a BUS che a Stella, poi vengono entrambe gestite a livello logico come topologie a BUS.

- Il metodo di accesso al canale è il CSMA/CD, dove per la gestione delle collisioni, il MAC interrompe la trasmissione non appena rileva collisione → genera jamming di 32 bit per segnalare l'avvenuta collisione e rischedula la trasmissione con l'algoritmo di back-off (per un massimo di 16 volte)
- Nella trasmissione delle trame, il MAC riceve UI dal livello superiore, quindi genera una stringa seriale da trasmettere sul mezzo fisico. Mentre nella ricezione riceve una stringa seriale sul mezzo fisico; quindi, fornisce l'UI al livello superiore. Inoltre, è necessario un *Inter-Frame Gap* tra le trame trasmesse⁸⁴
- Il campo FCS viene generato per le trame da trasmettere, viene controllato nelle trame ricevute per verifica della correttezza
- Il Preamble e l'SFD sono generati per le trame da trasmettere, mentre vengono rimossi nelle trame ricevute
- Inizialmente il mezzo trmissivo adottato è stato il cavo coassiale passivo⁸⁵ al quale venivano connesse le stazioni. La rete a BUS è stata sostituita dalle topologie a Stella (anni '90), che utilizzano il doppino in rame (twisted pair) e si basano sull'utilizzo di ripetitori di segnale multi porta detti HUB.
 - Il cavo in rame utilizzato è quello con connettore RJ45, di tipo UTP (non schermato) e che ha diverse varianti suddivise in 7 categorie, dove la 5 e 5E⁸⁶ sono le più utilizzate. Già per queste due categorie si possono anche utilizzare collegamenti in fibra ottica o rame ma solo per corto raggio.



Wireless LAN 802.11: caratterizzata dall'utilizzo dei canali radio come mezzo trmissivo a diverse bande (900MHz, 2.4GHz e 5GHz) con capacità nell'ordine dei Mbps. Chiaramente non ha una topologia ben definita, però ha diverse strutture.

- Basic Service Set (BSS)*, la più semplice struttura di comunicazione con stazioni mobili in mutua comunicazione
- Extended Service Set (ESS)*, diversi BSS messi in comunicazione tramite un *Distribution System* (DS, wired)



⁸⁴ Le trame più corte della lunghezza minima (64Byte) vengono scartate.

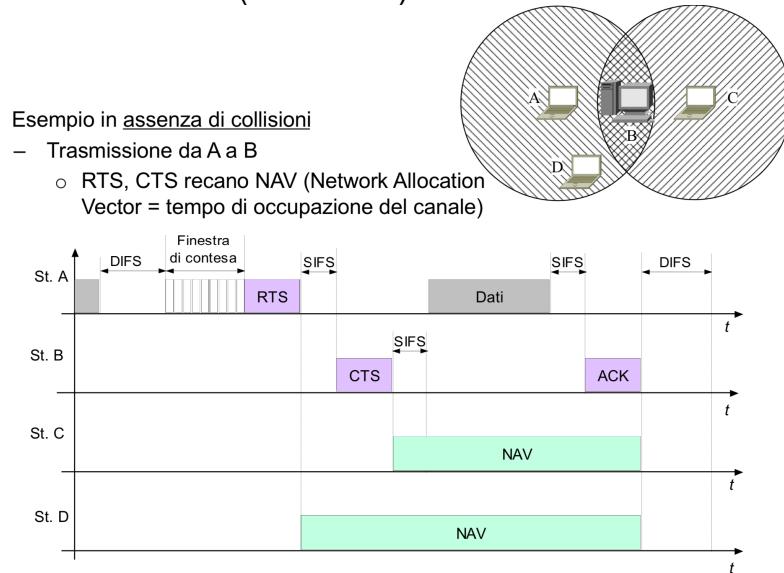
⁸⁵ In particolare, venivano usati i cavi RG-213 e RG-58, con denominazione XBaseY, con $X = Bit_{Rate}$ [$\frac{Mb}{s}$], Base = trasmissione in banda base (codifica Manchester) e $Y = \max(L)$ [$10^2 \cdot m$].

⁸⁶ Sono usate, rispettivamente, per la Ethernet 100BaseTX e per la Gigabit Ethernet 1000BaseTX.

o wireless); l'Access Point (AP) consente la comunicazione tra stazioni anche in BSS diversi⁸⁷

Ogni stazione ha un certo raggio di copertura, per cui il CSMA/CD non è più sufficiente. Come anticipato si utilizza il CSMA/CA, un protocollo di accesso dove il mittente invia esplicita richiesta di autorizzazione a trasmettere (RTS) → il destinatario risponde con esplicita autorizzazione (CTS) ed è richiesto ACK per ogni trama ricevuta correttamente. Non è l'unico modo per evitare/risolvere le collisioni; infatti, vi sono due modalità per regolare l'accesso al canale.

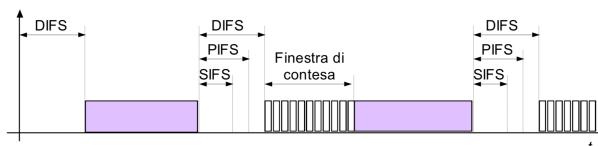
- *Distributed Coordination Function (DCF)*, modalità che adotta CSMA/CD, di seguito un esempio di funzionamento (in accesso)



- *Point Coordination Function (PCF)*, modalità d'accesso coordinato dagli Access Point senza collisioni

Una stazione agisce da Point Coordinator (PC)

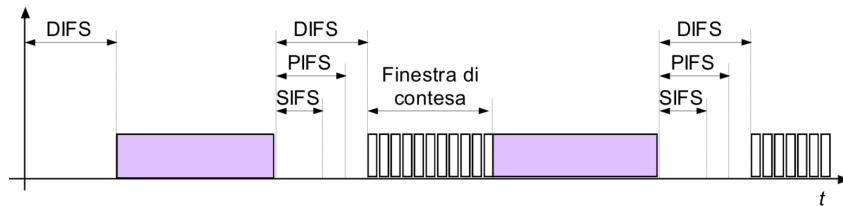
- Controlla centralmente l'accesso al mezzo trasmissivo dando alle varie stazioni l'autorizzazione esplicita a trasmettere
- Attiva la modalità PCF con invio di trama speciale dopo un tempo PIFS
 - o Ciò garantisce che non possa iniziare la finestra di contesa per trame "normali" (per queste è richiesta l'attesa di un intervallo di tempo DIFS)
- Interroga (poll) le singole stazioni che rispondono, sempre con un tempo SIFS



- La priorità d'accesso al canale è gestita con *InterFrame Spacing (IFS)* di durata crescente

⁸⁷ Può essere visto come una sorta di HUB wireless (si veda paragrafo sulle reti Ethernet).

- D-IFS, adottato da DCF per inviare il primo frammento e seguito dalla finestra di contesa del CSMA/CA
- P-IFS, adottato in modalità PCF
- S-IFS, tempo di attesa minima, usato per trame ad alta priorità (CTS, ACK), di risposta ad interrogazioni su AP e per frammenti successivi di trame con altri frammenti già ricevuti (con successo)



Formato trama nell'802.11: nuovo formato definito dall'IEEE, che prevede diversi campi.

	F C	Du/ ID	Address 1	Address 2	Address 3	Se Ctr	Address 4	QS Ctr	HT Ctr	Information	FCS
Byte	2	2	6	6	6	2	6	2	4	variabile	4
Bit	2	2	4	1	1	1	1	1	1	+HTC Order	

- *Frame Control (FC)*, che a sua volta contiene 11 sotto campi
 - *Protocol version*
 - *Type*, classe della trama (gestione 00, controllo 01, dati 10)
 - *Subtype*, specifico tipo di trama
 - *To DS*, *From DS*, indicano significato 4 campi address⁸⁸
 - *More frag*, altri frammenti da trasmettere
 - *Retry*, trama ritrasmessa
 - *Power management*, risparmio energetico della stazione
 - *More data*, altre trame indirizzate al destinatario
 - *Protected frame*, trama crittografata
 - *+HTC/Order*, indica che elaborazione in ricezione deve avvenire secondo ordine di ricezione
- *Duration/ID*, occupazione prevista in μs (NAV) del canale, oppure riporta ID stazione che sta inviando una trama in risposta a interrogazione
- *Address 1-2-3-4*, rispettivamente gli indirizzi di stazione mobile sorgente, stazione mobile destinazione e indirizzo del ricevitore/trasmettitore (AP del BSS di transito).
- *Sequence control*, numero frammento nella trama e numero di trama (primi 4 e ultimi 12)

⁸⁸ Nello specifico, da stazione a stazione nel BSS (00), uscente dal DS (01), entrante nel DS (10), da BSS a BSS (11).

- *QoS/HTcontrol*, informazioni per qualità servizio
- *Information*, fino a 2312Byte
- *FCS*, per controllo d'errore

Data Link Layer – interconnessione tra LAN

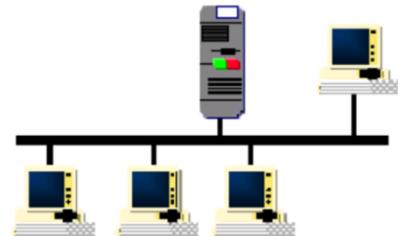
Non solo con livello di linea broadcast, le LAN odierne sono *commutate/switched* cioè connesse tra loro a diversi livelli: in ognuno dei quali opera un certo dispositivo.

Livello 1 (fisico)

Repeater: dispositivo che interconnette a livello fisico 2+ spezzoni di rete, con funzionalità essenzialmente di livello 1 come amplificazione, rigenerazione e temporizzazione dei segnali.

Hub: è essenzialmente un repeater multi porta, dove la banda disponibile è condivisa da tutte le stazioni ∈ *dominio di collisione*.

Collision Domain: porzione di LAN dove due stazioni che trasmettono simultaneamente vanno in collisione.

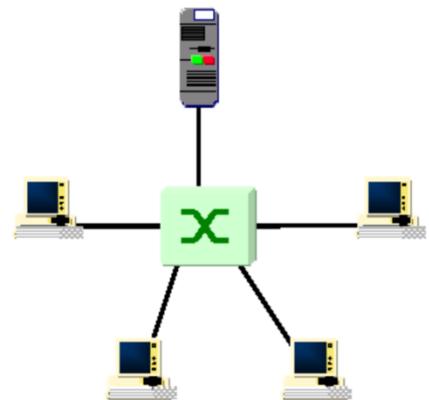


Broadcast Domain: porzione di LAN in cui tutte le stazioni ricevono una trama con MAC di broadcast (trasmessa da generica stazione).

I dispositivi precedentemente introdotti non separano i domini di collisione e non influenzano i domini di broadcast, servono piuttosto per superare le limitazioni fisiche per l'estensione delle LAN. Quindi, due parti di reti connesse da repeater o hub fanno parte degli stessi domini di collisione e broadcast.⁸⁹

Livello 2 (datalink)

Switch (bridge): dispositivo con funzioni di filtering e relay, dove nella prima le trame vengono inoltrate in modo “intelligente” non in broadcast (e.g. se trama ricevuta da LAN1 è indirizzata a stazione di LAN1, viene scartata e non inviata a LAN2); mentre si intende la capacità di forwarding verso LAN adiacenti a quella di provenienza (o comunque esterne). Inoltre, questo dispositivo funziona con store&forward delle trame.



Forwarding Database (FDB): (locale) tabella consultata per decidere se filtrare/inoltrare una trama, ha salvato il MAC address e la porta di uscita.⁹⁰

⁸⁹ Mentre i dispositivi di livelli superiori vanno ad influire sui domini citati, ad esempio, due parti di rete separate da un bridge apparterranno a differenti domini di collisione.

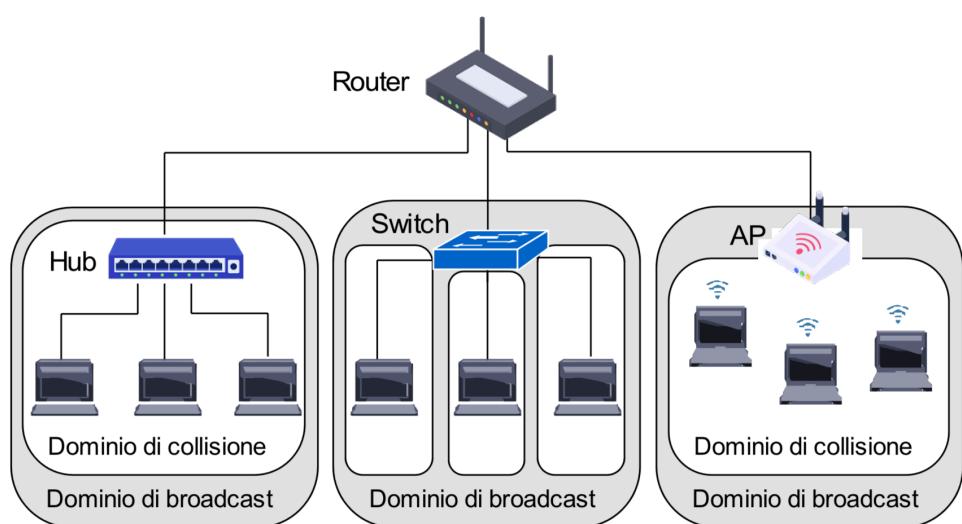
⁹⁰ È anche detta *switching table*.

In questa tipologia di interconnessione, ogni stazione può usare tutta la banda disponibile sul collegamento con lo switch; si noti che ogni coppia stazione-porta dello switch forma un dominio di collisione diverso, mentre non influiscono sui domini di broadcast.

Livello 3 (rete)

Router: dispositivo con funzioni di livello 3 (opera su datagrammi IP)

- rimuovono l'header di livello 2 se MAC è broadcast o uguale a quello dell'interfaccia del router
- esaminano header di livello 3 per inoltrare i datagrammi
- modificano header di livello 3 (e.g. decrementano TTL)
- inseriscono nuovo header di livello 2
- operano (implicitamente) con domini di broadcast separati (uno per interfaccia, anche i domini di collisione saranno \neq)



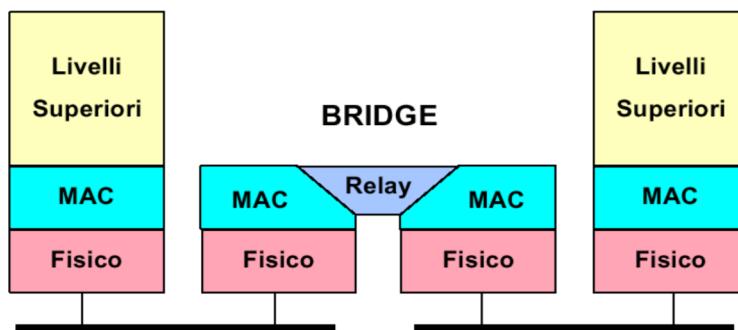
Livello 5 (applicazione)

Gateway: operano a livello applicativo della pila TCP/IP, idealmente sono un punto di accesso e uscita dalla rete.

La trattazione dei gateway non sarà approfondita ulteriormente nel corso, piuttosto ci si focalizzerà su forwarding/stp/VLAN.

Learning&Forwarding, SPT e VLAN

Nel capitolo precedente è stato introdotto lo switch, che svolge funzioni, tra cui la separazione dei domini di collisione e quella di *relay*.



Come apprende verso quale interfaccia si raggiunge una certa destinazione?

Transparent Bridging: la presenza dello switch è trasparente alle stazioni (che implementano le stesse funzioni di linea), non ha un indirizzo MAC che compare nelle trame; la tabella viene compilata automaticamente.

Tabella di switching	
MAC address 1	Porta A
MAC address 2	Porta B
MAC address 3	Porta C

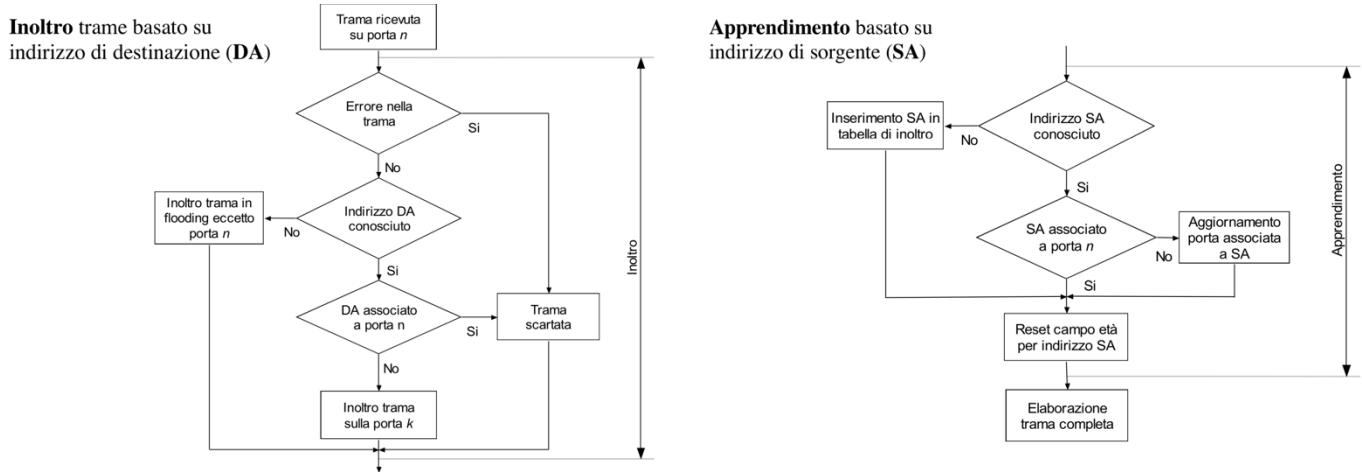
La tabella presentata ha solo entry *statiche*, in realtà alcune come l'*ageing time* sono entry *dinamiche*⁹¹; le entry sono ricavate e aggiornate con algoritmo *backward learning*.

Backward learning: algoritmo suddiviso in due fasi, si basa sui MAC sorgenti dei pacchetti in arrivo.

- *Learning (fase 1)*, l'indirizzo sorgente è inserito in tabella associato alla porta d'ingresso, se non già presente come entry statica (oppure la aggiorna) e aggiunge il timeout *ageing time* inizializzato a zero
- *Forwarding (fase 2)*, se l'indirizzo è in tabella lo inoltra alla porta indicata, altrimenti usa inoltro broadcast

È un algoritmo che funziona solo su topologie ad albero, ma le altre possono essere “adattate” sfruttando lo *Spanning Tree Protocol*.

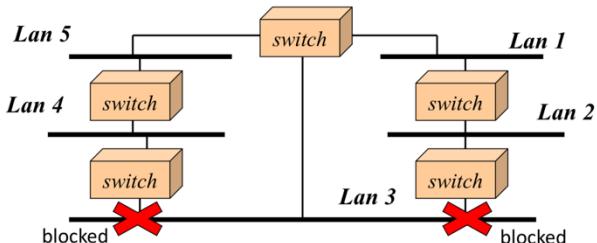
⁹¹ È una particolare riga della tabella che ha una “età”, il campo *ageing time* è utilizzato per eliminare l'entry dinamica più vecchia quando la tabella si sarà riempita.



Nota: l'*ageing time* risulta fondamentale per ridurre il più possibile gli errori di inoltro sulla porta sbagliata; in questo modo gli errori vengono eliminati il prima possibile.

Poiché la maggior parte delle reti non ha topologia ad albero ma ne usa una “magliata” per garantire ridondanza in caso di malfunzionamento di 1+ nodi, si può incorrere nel fenomeno di *broadcast storm*: si incapperebbe in un continuo flusso di trame che continuerebbero ad “intasare” le tabelle dei bridge. Il problema si può evitare trasformando la topologia da rete magliata ad albero che includa tutti i nodi (spanning tree).

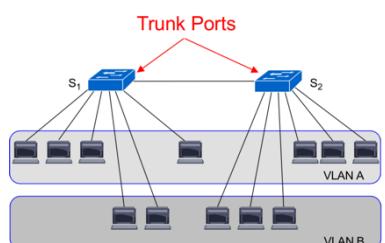
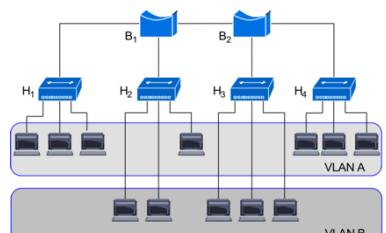
Spanning Tree Protocol (STP): protocollo distribuito che calcola lo spanning tree e lo aggiorna in caso di guasti; in breve, permette di ricavare una topologia logica ad albero a partire da una fisica magliata.



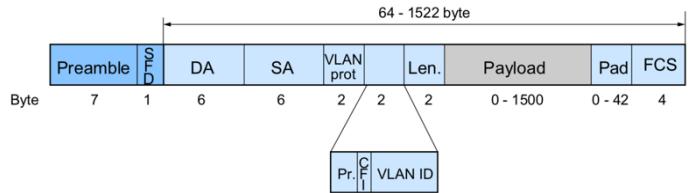
Si realizza ponendo in stato di blocco alcune porte (simula albero), lasciando passare solamente i messaggi di STP.

Virtual LAN (VLAN): tecnica che consente la creazione di LAN logicamente separate su un'unica LAN fisica, solitamente usate per distinguere il traffico tra diversi host nella stessa LAN.

- Realizzata connettendo in cascata bridge e hub, dove si avrà l'inoltro selettivo solamente nei bridge mentre gli hub consentono ad host di altre VLAN di ricevere le trame
- I domini di broadcast si separano a livello di trama con *VLAN tagging*
- Può essere realizzata utilizzando solamente gli switch, per una maggiore separazione tra VLAN; sarà ugualmente necessario il *tagging* sulle *trunk port*



- Formato della trama con VLAN ID (IEEE 802.1Q) prevede una struttura con alcuni campi aggiuntivi
 - *VLAN prot*, 0x8100
 - *Priority*
 - *Canonical Format Identifier*, identifica il formato del campo indirizzo
 - *VLAN ID*



L'associazione *stazione* \Leftrightarrow *VLAN* avviene per bridge/switch, MAC di stazione e tipo di protocollo (livello 3) utilizzato.