1- INTRODUZIONE

Con il termine *telecomunicazione* si intende la capacità di due o più individui (o dispositivi) di condividere informazioni attraverso collegamenti wireless o cablati.
Una ***rete di telecomunicazioni*** è quindi un insieme di dispositivi e dei rispettivi collegamenti (fisici o logici) che consentono la trasmissione e la ricezione di informazioni tra due o più utenti. Modalità di comunicazione:

- ***simplex***: il canale è percorribile solo in un verso (mittente – destinatario)
- ***half-duplex***: il canale è percorribile in entrambi i sensi ma non nello stesso istante
- ***full-duplex***: il canale è percorribile in entrambi i sensi anche contemporaneamente

Caratteristiche di una rete:

- *Prestazioni*: si riferiscono a *throughput* (utilizzo di un collegamento rispetto alla capacità massima) ed al *delay* (tempo che passa tra fine della trasmissione e fine della ricezione);
- *Affidabilità*: continuità del servizio della rete ;
- *Sicurezza*: protezione dalle intrusioni esterne.

Modalità di comunicazione:

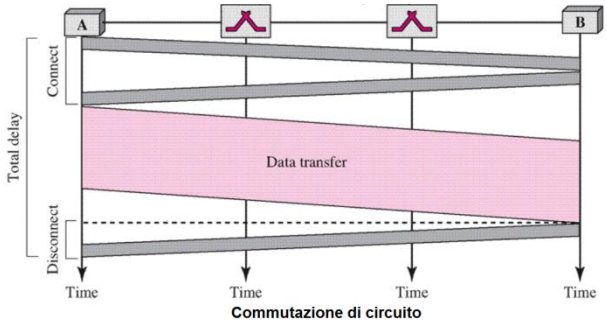
- *punto-punto*: comunicazione tra due soli dispositivi ;
- *multi-punto*: comunicazione multicast (o broadcast).

Topologie:

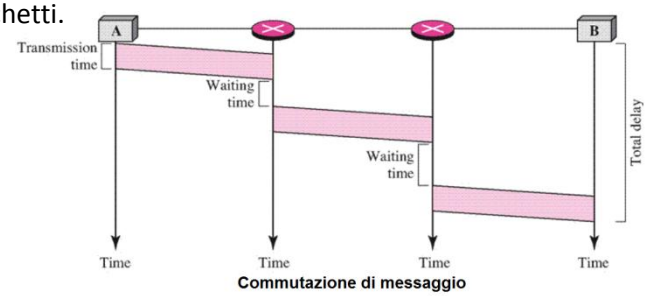
- *Maglia*: $O(n^2)$ collegamenti tutti dedicati ma scarsamente utilizzati, poco sensibile a guasti. Da usare quando le cose che contano sono l'affidabilità e le prestazioni.
- *Stella*: riduce il numero di collegamenti, resistente ai guasti ma ha un punto critico sul nodo centrale.
- *Bus*: supporta punto-punto e multi-punto in quanto tutti i terminali sono connessi ad un bus comune. Sensibile a guasti sul bus e non scalabile a causa dell'attenuazione.
- *Anello*: topologia cooperativa, ogni nodo aiuta a trasmettere i messaggi, sensibile a guasti su nodi e collegamenti (risolvibile con un doppio anello).

Commutazioni:

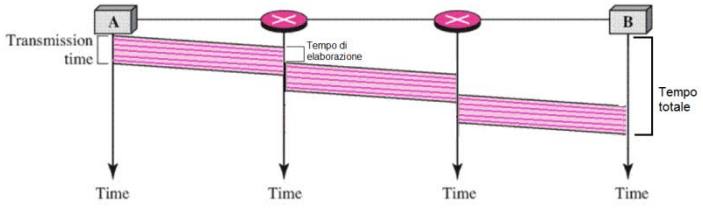
- **Circuito**: caratterizzato dalle fasi di *set-up* (individuazione del percorso), *utilizzo* (scambio dati dedicato) e *abbattimento* (rilascio risorse). Consigliata per comunicazioni di emergenza o massicce.
  Non è previsto lo *store&forward*, quindi deve esserci compatibilità fisica lungo tutto il percorso.
- **Messaggio**: introduce lo *store&forward*, ogni nodo memorizza il messaggio, ricalcola il percorso ottimo ed effettua l'inoltro del messaggio stesso in blocco. Si costruisce dunque il percorso *step-by-step*.
- **Pacchetto**: un messaggio è diviso in ***frammenti*** di dimensione fissa; ha le caratteristiche della commutazione di messaggio, ma è maggiormente immune agli errori.
  - ***Circuito virtuale***: si effettua una breve fase di *setup* per individuare il percorso che i pacchetti dovranno seguire; utile per servizi connection-oriented, ma svantaggioso in caso di cambio di stato della rete (ripetizione fase di setup).
  - ***Datagramma***: ogni nodo sceglie il miglior percorso per ogni pacchetto. Più flessibile ma può comportare disordine nell'arrivo dei pacchetti.



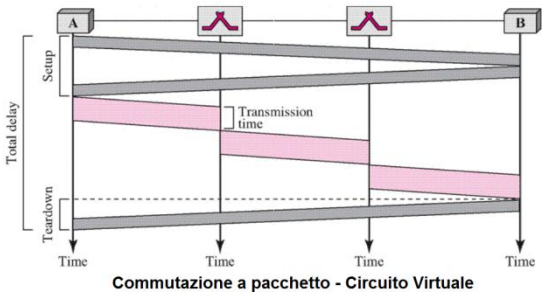
Commutazione di circuito



Commutazione di messaggio



Commutazione a pacchetto - Datagramma



Commutazione a pacchetto - Circuito Virtuale

2- RETE TELEFONICA

- *Reti pubbliche*: utilizzano *commutazione di circuito*. La topologia è a stella per i centri di basso livello (*Rete Urbana*, *Settore*) e a maglia per il nucleo della rete (*Distretto*, *Compartimento*, *Nazionale*).
- *Reti private*: riguardano applicazioni professionali (es *Enti pubblici*).

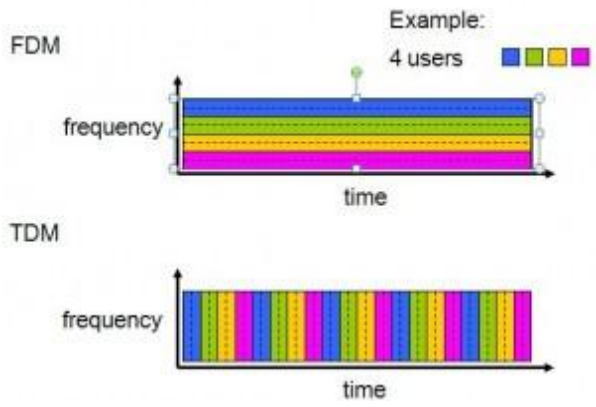
La **telefonia** può essere di due tipi:

- *Analogica*: banda [0 – 4]KHz modulata sul canale tramite traslazione in un intervallo di frequenze;
- *Numerica*: affinché da un insieme discreto di valori sia possibile ricostruire il segnale continuo originale, occorre basarsi sul *Teorema di Shannon* ($f_s \geq 2B$). Per un segnale fonico, si campiona con passo 125µs (fisso) e lo si quantizza con 8 bit per campione. La frequenza è 64 Kbit/s.

A parte la maggior banda richiesta la telefonia numerica permette una buona resistenza al rumore (si tratta solo di distinguere 0 da 1), bassi costi, riservatezza e flessibilità per varie applicazioni (voce, video ecc...).

Multiplexing : permettono la condivisione di un canale fisico tra più utenti (segnali) distinti. Tecniche:

- *Divisione di frequenza* (**FDM**): Si divide il ***canale*** in ***sottocanali*** assegnati ai singoli utenti; si implementa con una modulazione in trasmissione e un filtro in ricezione.
- *Divisione di tempo* (**TDM**): Il tempo è diviso in ***frame*** a loro volta divisi in ***slot*** assegnati ai singoli utenti.
- *Divisione di lunghezza d'onda*: si trasmettono su fibra ottica impulsi di luce con diversa lungh. d'onda.



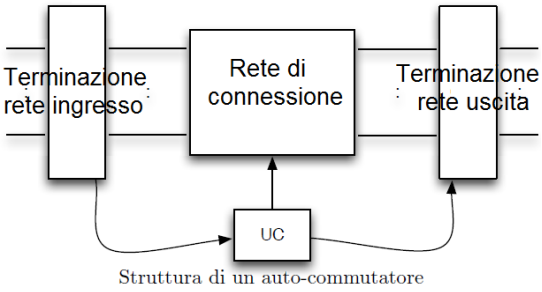
Tecnologie xDSL

Le linee telefoniche tradizionali possono essere utilizzate per la trasmissione di dati verso le postazioni utente inviando tali dati su un collegamento telefonico tradizionale (inizialmente grazie ai modem).
Venne poi sviluppata la tecnologia **DSL** per meglio sfruttare le capacità di trasporto delle linee in doppino di rame: tale tecnologia prevede quattro varianti che complessivamente vengono indicate come xDSL.

- *ADSL* (*Asymmetric Digital Subscriber Line*): Assegna una velocità maggiore al download; il canale condiviso non permette un bit-rate costante.
- *HDSL* (*High-bit-rate Digital Subscriber Line*): Assegna la stessa banda a upload e download. Vi è un doppino di rame dedicato per ogni utente.
- *SDSL* (*Symmetric Digital Subscriber Line*): Simile a HDSL ma con un'unica linea.
- *VDSL* (*Very-high-bit-rate Digital Subscriber Line*): Disponibile in versione asimmetrica o simmetrica, utilizza mezzi di trasmissione di buona qualità e permette velocità di decine di Mbps.

3- COMMUTATORI

Un **commutatore** è un dispositivo che si occupa dell'*inoltro* all'interno della rete. E' una struttura intelligente in quanto deve decidere a quale delle sue uscite collegare un certo ingresso in relazione all'interpretazione della richiesta di connessione. Il commutatore deve poter *costruire*, *mantenere* e *abbattere* specifici collegamenti.



Struttura di un auto-commutatore

Struttura di un Autocommutatore:
- *Terminazione rete ingresso*: separa il flusso delle informazioni d'utente dal flusso di segnalazione (*demultiplexing*);
- *Rete di connessione*: stabilisce l'interconnessione tra linee di ingresso e linee di uscita;
- *Terminazione rete uscita*: riunifica i due flussi secondo le regole di inoltro dettate dall'unità centrale UC (*multiplexing*).

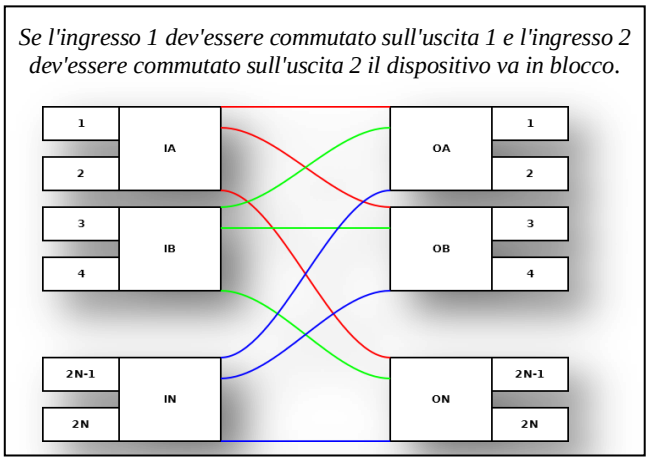
A seconda del numero di linee di ingresso e di uscita, i commutatori sono classificati come:
- *Concentratori*: # ingressi > # uscite (verso collegamenti più capaci)
- *Espansori*: # ingressi < # uscite (verso collegamenti meno capaci)
- *Distributori*: # ingressi = # uscite.

A seconda della *tecnologia realizzativa* si ha:
- ***Strutture a divisione di spazio*** (**S**): composti da una *matrice*, permettono di *cambiare linea e lasciano inalterato lo slot nel frame*. Il costo è il prodotto tra numero di linee in ingresso e numero di linee in uscita.
- ***Strutture a divisione di tempo*** (**T**): memorie elettroniche, c*ambiano slot all'interno di un frame e lasciano inalterata la linea*. La commutazione può avvenire in fase di scrittura sulla memoria (scrittura casuale, lettura sequenziale) che in fase di lettura (scrittura sequenziale, lettura casuale). Il costo è dato dal tempo di accesso, che dev'essere < $\frac{125\mu s}{2N}$ (N accessi in scrittura e N accessi in lettura).

Potendo combinare assieme le strutture S e T si ha la classificazione:
- *omogenee* & *non-omogenee*: a seconda se sono formati da componenti dello stesso tipo o meno;
- *bloccanti* & *non-bloccanti*: a seconda se è sempre possibile connettere una linea di ingresso ad una qualsiasi linea di uscita libera.

Strutture multistadio

- Struttura **SS**: è composta da due stadi S organizzati in modo che:
  "*l'uscita i dell'elemento j del primo stadio sia collegata all'entrata j dell'elemento i del secondo stadio*";
  ha un costo inferiore rispetto ad un'equivalente S, ma è *bloccante*. Se si rende la struttura non bloccante si perde il vantaggio del costo.



- Struttura **TS**: utilizzata per combinare le funzionalità dei due tipi di commutazione. *Bloccante* dato che è impossibile portare slot diversi di una stessa linea su slot uguali di linee diverse.

- Struttura **ST**: utilizzata per combinare le funzionalità dei due tipi di commutazione. *Bloccante* dato che

è impossibile portare slot uguali di linee diverse su slot diversi della stessa linea.

- Struttura **SSS**: prevede un ulteriore stadio interno. Nel caso semplificato di N linee in ingresso e N linee in uscita si hanno:
  - primo stadio: N/n dispositivi con n ingressi e k uscite (per ogni blocco) ;
  - secondo stadio: k dispositivi con N/n ingressi e N/n uscite (per ogni blocco) ;
  - terzo stadio: N/n dispositivi con k ingressi e n uscite (per ogni blocco).

  L'espressione del costo è $k \cdot N\left(2+\frac{N}{n^2}\right)$.

  Nell'***analisi di Clos*** , cercando di collegare *l'ingresso i del blocco A del primo stadio con l'uscita j del blocco Z del terzo stadio*, nel *caso peggiore* si ha che:
  -> tutti i restanti ingressi di A e tutte le restanti uscite di Z sono occupati/e.
  -> le richieste in ingresso ad A non sono destinate alle uscite del blocco Z.
  -> i blocchi del secondo stadio interessati da A (n-1) sono diversi dai blocchi del secondo stadio interessati da Z (n-1) .

  La condizione di non blocco è $k=(n-1)+(n-1)+1=2n-1$, ovvero ci devono essere $2n-1$ blocchi S al secondo stadio. Sostituendola nella formula del costo si ottiene $\bar{C}=4\sqrt{2}N^{\frac{3}{2}}$

- Struttura **TST**: l'obiettivo è combinare le funzionalità di cambiamento di linea e di slot all'interno del frame. Come nel caso SSS, è possibile studiare la condizione di non blocco nel caso peggiore:
  Cercando di portare *lo slot i della linea A nello slot j della linea Z* si ha:
  -> tutti gli altri slot di A e di Z sono occupati.
  -> le richieste sulla linea A non sono rivolte alla linea Z (tranne ovviamente quella considerata).
  -> gli slot del secondo stadio interessati da A (n-1) sono diversi da quelli interessati da Z (n-1).

  La condizione di non blocco è $k=2n-1$ (analoga a quella della struttura SSS).

Analisi di Lee

Si basa su un modello probabilistico del traffico; l'obiettivo è quello di definire strutture di commutazioni per le quali l'evento probabilità di blocco sia estremamente raro e allo stesso tempo abbiano un costo inferiore rispetto alle equivalenti strutture basate sulla teoria di Clos. Per l'analisi di una struttura SSS i passi sono:

- a = probabilità che una linea di ingresso sia occupata.
- $\frac{1}{k}$ = probabilità che una richiesta in ingresso del primo stadio scelga un'uscita specifica.
- $p=\frac{n\cdot a}{k}$ = probabilità che una linea in uscita al primo stadio sia occupata.
- $(1-p)^2$ = probabilità che un cammino dal primo al terzo stadio sia libero;
- $1-(1-p)^2$ = probabilità che un cammino dal primo al terzo stadio sia occupato;
- $[1-(1-p)^2]^k$ = probabilità di blocco (tutti i cammini sono occupati).

L'analisi Lee non è esatta: sostituendo al posto di k il valore ottimo di Clos (blocco impossibile) non si ottiene 0.

Commutatori veloci a pacchetto

Usati per evitare che la commutazione sia un collo di bottiglia nello scambio di informazioni (a causa della lentezza con cui è eseguita). Tipicamente per rendere veloce l'operazione di commutazione, questa viene implementata il più possibile su base hardware. Le principali tecnologie sono:

- ***Crossbar***: molto simile alla tecnologia S.
- ***Banyan***: processa l'indirizzo di destinazione in modo sequenziale. Composto da switch elementari. Si fanno contemporaneamente le operaz. di creazione del percorso e invio del pacchetto (step-by-step);
- ***Batcher-Banyan***: ordina i pacchetti in ingresso in modo da evitare conflitto sulla richiesta di una stessa uscita, risolvendo dunque i conflitti presenti nelle strutture Banyan.

4- RETI PER TRASMISSIONE DATI

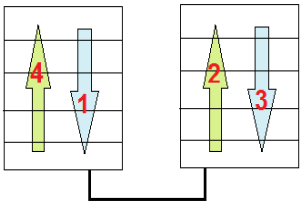
Un *flusso dati* è un flusso di bit che tramite tecniche di trasmissione dati (modulazione, codifica,ecc…) viene messo in grado di attraversare un canale di comunicazione per connettere sorgente e destinazione.
Le caratteristiche di un traffico dati sono:

- **Intermittenza temporale**: lo scambio di informazioni non è sempre attivo, vi sono momenti di "silenzio";
- **Asimmetria**: il flusso di info può andare in entrambe le direzioni, anche se spesso una è privilegiata;
- **Integrità**: il flusso dati ricevuto deve essere privo di errori o comunque rispettare una tolleranza.

Le tipologie di traffico possono essere **sincrono** se è abbastanza continuo nel tempo (file di grandi dimensioni), **asincrono** (classica comunicazione intermittente) o **isocrono** (ha bisogno di un riferimento temporale preciso).
Una rete solitamente è composta da un insieme di regole (***protocolli***), e data la complessità è organizzata in *livelli*. In generale *ogni livello fornisce dei servizi a quello immediatamente superiore* tramite un'interfaccia definita da uno standard. Ogni livello è in *comunicazione logica* diretta con uno paritario.

I **servizi** (*primitive*) che fanno parte di un'interfaccia tra livelli sono:

- ***request***: Un livello chiede un servizio al livello sottostante;
- ***indication***: Un livello invia un messaggio al livello superiore;
- ***response***: Un livello risponde alla richiesta di servizio verso il livello sottostante;
- ***confirm***: Un livello informa il livello superiore che il servizio è stato fornito.



I servizi possono essere **connectionless** se i pacchetti trasmessi possono seguire percorsi differenti all'interno della rete per raggiungere la destinazione permettendo più reattività ma non garantendo che i pacchetti arrivino in ordine, oppure **connection-oriented** se il percorso, determinato a priori, è seguito da tutti i pacchetti che arriveranno in ordine.
Un servizio può essere **affidabile** se il mittente richiede un riscontro dell'invio/ricezione del pacchetto, cosa che può introdurre ritardi, oppure **non affidabile**, quando il mittente non è certo che la comunicazione sia andata a buon fine.
Un'architettura di rete può essere **aperta**(componenti provenienti da più produttori, ottimizzabile solo nelle singole parti) o **proprietaria**(componenti certificati, ottimizzabile in modo globale).

Modello ISO/OSI:

Modello di rete *aperta* proposto da ISO. Prevede 7 livelli gerarchici e ne definisce le funzionalità di base.
I livelli superiori (dal 4 al 7) operano su base E2E, mentre quelli inferiori (dall'1 al 3) operano su base L2L.
Si noti che nei nodi di transito (*router*) sono implementati solamente nei primi (tre) livelli della pila.
Il passaggio dell'info tra livelli avviene mediante l'***incapsulamento successivo***: ogni livello inserisce in testa al messaggio (ricevuto dal livello sovrastante) un *header*, ovvero info di controllo che consente di comunicare con il corrispondente livello paritario nella pila di destinazione (solo il livello 2 può mettere una testa e una coda).

Applicazione	Permette ad utenti di cooperare	End to End
Presentazione	Compressione, traduzione e gestione della protezione (crittografia)	
Sessione	Gestione della sessione: apertura, utilizzo e chiusura del collegamento	
Trasporto	Fornisce, se richiesto, un servizio connection-oriented e divide l'informazione in pacchetti	
Rete	Si occupa di inoltro e instradamento, nonché di rendere compatibili reti eterogenee	Tutti i nodi
Collegamento	Trasferisce dati tra nodi adiacenti controllando errori, accesso al mezzo, framing e riscontri	
Fisico	Trasmette/riceve i bit dal canale utilizzando tecniche di modulazione	

TCP/IP

Suite protocollare alla base di Internet .

Applicazione	Simile ad Applicazione di OSI
Trasporto	Simile a Trasporto di OSI
Internet	Scambia pacchetti tra nodi connessi anche in modo eterogeneo, corrisponde a Rete di OSI
Host to Network	Maschera ai livelli più alti le caratteristiche fisiche della rete

- **Livello Rete :** ***IPv4*** e ***IPv6*** sono i due protocolli principali a livello Internet.

L' header **IPv4** prevede 13 campi:

Nome	**Bits**	**Descrizione**
Version	4	Ha valore 4
Internet Header Lenght	4	Lunghezza dello header
Type of Service	8	Tipo dei dati contenuti nel pacchetto
Total Length	16	Lunghezza del pacchetto in byte
Identification	16	Identifica i frammenti di un pacchetto
Flags	3	1) Non usato 2) Pacchetto non frammentabile 3) Il pacchetto è un frammento
Fragment Offset	13	Offset di un frammento espresso in blocchi da 8 bytes
Time to Live	8	Decrementato ad ogni hop; se scende a 0 il pacchetto è scartato
Protocol	8	Protocollo usato al livello trasporto (TCP/UDP)
Header Checksum	16	Controllo degli errori
Source Address	32	Indirizzo mittente
Destination Address	32	Indirizzo destinatario
Options	...	Informazioni per datagrammi particolari

La **frammentazione** entra in atto se un nodo della rete non può trasmettere un pacchetto perché di dimensioni troppo elevate. I frammenti creati (tranne l'ultimo) avranno il terzo flag dell'header settato ad 1.

Gli indirizzi IP sono espressi in forma ***decimale puntata***; ogni sezione può assumere valori da 0 a 255.
La gestione degli indirizzi è stata modificata spesso in quanto il numero di indirizzi disponibili non copre il numero di interfacce esistenti. Si noti che un indirizzo IP non identifica in maniera univoca un dispositivo, ma solo la sua interfaccia con il collegamento fisico vero e proprio. Vi sono state diverse fasi temporali:

1. I primi 3 byte identificano la rete, l'ultimo l'host. In una rete potevano essere identificati solo 256 terminali.
2. **Classful Networking** (vedi tabella): le classi A,B,C usano lunghezze diverse di byte per identificare la rete (*netid*) e il dispositivo connesso (*hostid*). Vi è spreco di indirizzi IP quando un router gestisce più reti locali.
3. **CIDR**: indirizzi del tipo *a.b.c.d/y* , la *y* (*prefisso di rete*) indica il numero di bit associati alla rete, ed identifica dunque la ***subnet mask***: essa indica quanti bit sono stati riservati per l'indirizzo di rete.

Alcuni indirizzi IP sono riservati:

- ***Indirizzi privati*** (visibili solo all'interno di una rete locale): ***10.0.0.0/8***, ***172.16.0.0/20*** e ***192.168.0.0/16***
- ***Indirizzo loopback*** (identifica l'interfaccia corrente): ***127.0.0.0/8***
- ***Indirizzo broadcast***: I bit dell'host sono tutti ad 1; ***255.255.255.255*** è un indirizzo broadcast globale.

Per assegnare un indirizzo IP ad un host vi sono due modi:

- *Manuale*: l'host ottiene un indirizzo IP fisso.
- *Dinamica*: l'host ottiene un indirizzo tramite server **DHCP** (*Dynamic Host Configuration Protocol*).

Per risolvere la carenza di indirizzi IP si è utilizzato il metodo *Network Address Translation* (**NAT**). Esso interfaccia una rete privata con la rete pubblica: alla prima è assegnato un solo IP pubblico; quando gli host richiedono di uscire dalla rete il NAT modifica i pacchetti inserendo come indirizzo IP sorgente l'indirizzo di rete e come numero di porta un valore scelto dal dispositivo; tramite una tabella associa *(IP privato sorgente, porta sorgente)* a *(IP destinazione, porta esterna)*.
Si noti che l'utilizzo della tecnica NAT funziona solo se la comunicazione è iniziata da un host appartenente ad una rete privata: il NAT ha una corrispondenza biunivoca indirizzo-privato/indirizzo-pubblico solo se il primo pacchetto è stato inviato da un terminale all'interno della rete privata. Il tipo di comunicazione è quindi "client-server" dove l'utente all'interno della rete privata contatta un server esterno: un host all'interno della rete non può fungere da server perché il suo indirizzo IP univoco non è visibile all'esterno.
Una soluzione al NAT è stata quella di utilizzare le ***socket*** = ***indirizzi IP + numero porte*** (quest'ultima identifica il processo destinatario sul relativo host, ed è un concetto del livello di trasporto).

IPv6: Introdotto principalmente per risolvere la carenza di indirizzi IP. Gli indirizzi sono espressi come 8 gruppi di 4 cifre esadecimali. Ciascun header IPv6 prevede:

Nome	**Bits**	**Descrizione**
Version	4	Ha valore 6
Traffic Class	8	Simile a TOS di IPv4
Flow Label	20	Identifica il flusso di datagrammi (abbinato a Traffic Class)
Payload Length	16	Lunghezza del carico in byte
Next Header	8	Indica l'intestazione successiva a quella corrente
Hop Limit	8	Simile a TTL di IPv4
Source Address	**128**	Indirizzo mittente
Destination Address	**128**	Indirizzo destinatario

IPv6 comprende 3 tipi di indirizzamento:

- *Multicast*: Il messaggio è inoltrato ad un gruppo di interfacce che può essere esteso fino a diventare un messaggio Broadcast.
- *Unicast*: L'indirizzo unicast identifica una sola interfaccia nella rete tramite il suo MAC address;
- *Anycast*: Indirizzo che corrisponde a più interfacce; utile per distribuire il traffico.

NB: gli indirizzi vengono assegnati alle interfacce connesse alla rete, non ai singoli nodi.

Per far convivere *IPv4* con *IPv6* nel momento di transizione vi sono 2 metodi:

- *Tunneling*: Il datagramma IPv6 diventa carico di un datagramma IPv4 quando entra in una rete IPv4; all'uscita di essa viene tolto lo header IPv4.
- *Dual stack*: Il dispositivo di ingresso nella rete IPv4 implementa entrambi i protocolli e sostituisce lo header IPv6 con uno IPv4. In questo caso si possono però perdere funzionalità.

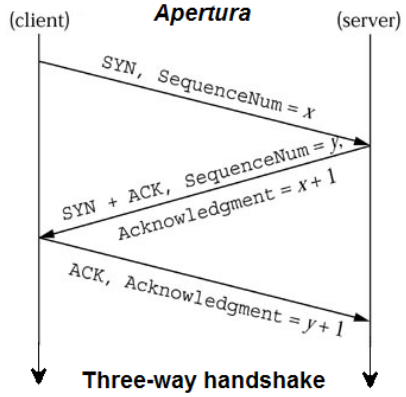
➢ Livello Trasporto

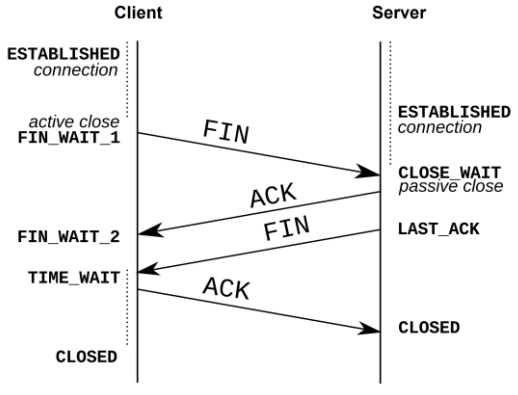
UDP: lo header UDP comprende *numero porta sorgente* e *destinatario*, *lunghezza del campo dati* e *checksum* (opzionale). Supporta comunicazioni connectionless non prevedendo una fase di setup; non è inoltre affidabile.

TCP: Protocollo che supporta le comunicazioni connection-oriented. L' header comprende:

- Numeri di porta sorgente e destinazione (16 bit)
- *Numero di sequenza* (32 bit): Identifica il primo *byte* del pacchetto
- *Numero di riscontro* (32 bit): Indica il prossimo *byte* che ci si aspetta di ricevere
- Lunghezza header (4 bit)
- *Flags*: **ACK** indica riscontro, **FIN**, **SYN**, **RST** gestione della connessione, **URG** presenza di dati urgenti.
- Finestra (16 bit) numero di byte disponibili nel buffer; usato nel *controllo del flusso*.
- Checksum (16 bit)
- Puntatore a dati urgenti: Indica dove sono i dati che TCP deve subito consegnare al processo applicativo
- Opzioni

TCP, a differenza di UDP, permette la segmentazione dei dati.
L'apertura di una connessione TCP avviene e con una **Three-way handshake**, la chiusura può avvenire anche con la **Four-way handshake** (quest'ultima prevede soltanto un controllo preventivo , prima della sua attivazione).





IEEE 802

Il comitato IEEE 802 ha definito un modello a strati composto di 3 livelli:

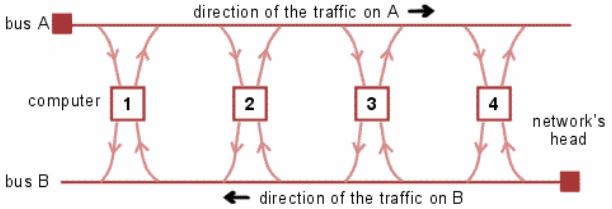
- **Logical Link Control** (**LLC**): Definisce uno standard che si può interfacciare con tutti i MAC di livello inferiore e i protocolli di rete superiori. Esso si occupa di sovrintendere al *trasferimento di pacchetti tra due nodi adiacenti* effettuando controllo di flusso, integrità ecc.. Può essere *connectionless senza riscontro*, *connectionless con riscontro, o connection-oriented*.
- **Medium Access Control** (**MAC**): Gestisce l'accesso al canale condiviso da tutti i dispositivi: conseguenza è una trasmissione broadcast; dunque tutti i dispositivi ricevono l'unità informativa anche se non ne sono i destinatari. Può utilizzare *metodi ordinati* (regole per accedere alla rete evitando collisioni) o *casuali* (libertà assoluta per i nodi che può portare alla collisione).
- **Physical Layer**: riguarda il tipo di rete ed il mezzo fisico di cui si dispone.

DQDB

Standard per le MAN, prevede una rete *multi-accesso* con protocollo ordinato realizzata con due ***bus unidirezionali*** a 150 Mbps. (NB) *L'accesso si assegna in modo FIFO e su base TDM*.
I pacchetti comprendono due campi di header **S** (*pacchetto dati*), **P** (*prenotazione*) ed un ***campo dati***.
Un nodo è caratterizzato(oltre che da un buffer dati) da due contatori: ADD/DROP indica il numero di nodi che hanno un prenotazione in corso (incrementa se sul bus passa un pacchetto P=1, decrementa se passa un pacchetto S=0) , e DROP (decrementa se il nodo intercetta sul bus di trasmissione un pacchetto S=1).

Se un nodo A vuole comunicare con un nodo B effettua le seguenti operazioni:

- Sceglie il bus appropriato per trasmettere a B secondo il verso di propagazione del collegamento.
- Attende un pacchetto P=0 sul bus di prenotazione, e immediatamente invia un pacchetto P=1.
- Pone DROP = ADD/DROP. Quando DROP scende a 0 utilizza il primo slot S=0 per trasmettere i dati.



Un problema è che i nodi più vicini all'inizio del bus hanno una maggior priorità. È possibile quindi porre un limite al num di accessi consecutivi. Col sistema di *pre-allocazione* degli slot è possibile fornire servizio isocrono.

FDDI

Pensato per una rete in ***fibra ottica*** (solitamente MAN) a 100 Mbps di estensione di qualche centinaio di chilometri. La topologia è a *doppio anello*: uno dei due rami è inattivo ed entra in funzione in caso di guasti o per aumentare le prestazioni.
Il livello MAC utilizza una metodologia di accesso ordinato e supporta trasferimento *asincrono*, *sincrono* e *isocrono* (FDDI-II). Nel caso di trasferimento asincrono lo standard prevede 8 livelli di priorità.
I frame sono di due tipi: ***token*** utilizzati per la gestione dell'accesso, e ***data*** per trasmettere le informazioni.
Il controllo di integrità è effettuato E2E.

Si usa un sistema di contatori per mantenere la sincronizzazione:

- *Token Target Rotation Time* (***TTRT***): valore condiviso, è calcolato sommando i tempi di trasferimento sincrono (α_i) con i tempi di trasferimento del token (d_i), indica la durata di un ciclo completo di trasmissioni sincrone.
- *Token Rotation Time* (***TRT***): Calcolato dai singoli nodi, indica il tempo effettivo tra due ricezioni del token. Può essere diverso tra i vari nodi, ma dovrebbe coincidere idealmente con il TTRT.
- *Token Holding Time* (***THT***): può assumere due valori:
  - Se la rete funziona peggio di quanto previsto (TRT > TTRT) esso vale → α_i
  - Se la rete funziona meglio di quanto previsto (TTRT > TRT) esso vale → TTRT-TRT

  Nel primo caso il nodo trasferirà solo dati *sincroni*, nel secondo potrà inviare anche dati *asincroni*.

La *banda* garantita ad un nodo è $B_i = \frac{\alpha_i}{TTRT} \cdot B$. L'*efficienza* della rete invece si misura con $\eta = \frac{TTRT - \sum d_i}{TTRT}$

5- ACCESSO MULTIPLO

L'accesso condiviso al mezzo fisico può portare a collisioni se non è preventivamente coordinato tra gli utenti. Per risolvere tali collisioni si introducono *tecniche ad accesso ordinato* e *casuale*:

- **Tecniche ad accesso ordinato**: l'accesso al canale viene definito in accordo con una procedura prefissata.
  1. *FDM/TDM*: in reti LAN si dimostrano inefficienti, poichè il singolo utente utilizza la propria risorsa di accesso (banda di frequenza o tempo di canale) senza continuità, avendo così uno spreco di capacità di accesso.
  2. *CDMA*: ogni utente trasmette la propria informazione occupando tutta la banda del canale condiviso; il ricevitore estrarrà dal segnale ricevuto solo la componente di interesse.
  3. *OFDMA*: si suddivide la banda di canale in sottobande, ciascuna centrata su una frequenza ortogonale a tutte le altre.
  4. *Tecniche Polling*: metodologia di "interrogazioni". Modalità:
     - *Roll-Call*: modalità centralizzata; il master gestisce le fasi di autorizzazione e rilascio del canale da parte dei nodi secondari (client);
     - *Hub-Polling*: modalità di cooperazione tra nodi (client).

     Indipendentemente dalle due modalità sopra descritte, una volta che un nodo client ha ricevuto l'autorizzazione all'accesso al canale, questa viene gestita in modo:
     - *gated*: si consente l'accesso al canale per un tempo massimo definito;
     - *esaustivo*: non viene introdotta alcuna limitazione al tempo di accesso al canale;
  5. *Token Passing*: implementazione distribuita; si basa sul concetto di "*passa parola*" del token (messaggio), il quale autorizza il nodo ad accedere in modo esclusivo al canale condiviso.

- **Tecniche ad accesso casuale**: non prevedono forme di coordinamento tra i nodi che condividono uno stesso canale e non esiste gerarchia tra nodi. Se si verifica una collisione si deve: *riconoscere* & *risolvere* tale collisione. Le tecniche utilizzate per effettuare queste due operazioni sono:
  1. *Aloha*: trasmissione dati a pacchetti (blocchi di bit di dimensioni fissate) e non vi è nessun controllo dello stato del canale (libero/occupato) nel momento del tentativo di accesso di un nodo.
     - *Aloha puro*: non prevede in maniera assoluta nessuna forma di coordinamento. Si ha condivisione del canale con un nodo centrale che è in grado di interpretare la informazione ricevuta ed inviare un messaggio di riscontro in modalità broadcast: il nodo interessato all'esito del tentativo di accesso si mette in ascolto sul canale broadcast (distinto da quello condiviso); la mancata ricezione del messaggio di riscontro implica un fallimento, e si attiva la modalità di risoluzione: si effettua il nuovo accesso con un ritardo casuale, scelto entro il tempo di back-off con probabilità uniforme.
     - *Aloha slotted*: si ha un minimo di coordinamento tra nodi (sincronizzazione); il tempo è diviso in slot di durata uguale al tempo di trasmissione del pacchetto: i nodi possono tentare l'accesso solo in corrispondenza degli istanti di inizio slot. Se si verifica collisione, si risolve come nell'Aloha puro (+ sincronizzazione).
  2. *CSMA*: ogni nodo ascolta il canale prima di effettuare il tentativo di accesso ("*ascolta prima di parlare*"): se durante la fase di sensing il nodo rivela la presenza del segnale portante, non accede e interpreta questo evento come *collisione virtuale*. Le collisioni non sono evitate totalmente a causa dei ritardi di propagazione del segnale; dunque vengono risolte come nell'Aloha.

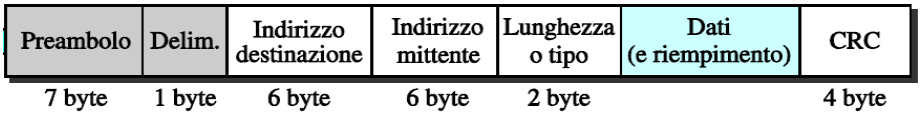
     Varianti della tecnica con rilevamento (sensing) della portante:
     - *CSMA 1-persistent*: un solo nodo è perennemente in ascolto sul canale (e quando ritiene opportuno effettua l'accesso): metodologia efficace solo con numero limitato di nodi.
     - *CSMA non-persistent*: si risolve il problema precedente ma si introduce un ritardo (spesso inutile) per ogni accesso successivo. Metodo efficiente in reti dense.
     - *CSMA p-persistent*: si usano i vantaggi delle due metodologie precedenti: si accede al canale su base statistica con prob. p , si ascolta con prob. $q = 1 - p$.
     - *CSMA/CD*: i nodi controllano il canale anche in fase di accesso al fine di rilevare una collisione ed interrompere il tentativo di accesso in corso ("*ascolta prima di parlare e mentre parli*").

6- RETE ETHERNET

La rete Ethernet può avere una topologia a bus o a stella e può funzionare sia su cavo coassiale, doppino telefonico o fibra ottica. Si è sviluppata poiché offre la possibilità di trasferire flussi informativi con rate elevato, ed ha una migliore flessibilità di gestione rispetto ad altre alternative (es. FDDI).

Livello Mac : gestisce l'accesso al canale: la tecnica adoperata è la **CSMA/CD** con modalità *1-persistent*. Tutte le tecnologie Ethernet forniscono (a livello rete) un servizio *connection-less* (quindi "no fasi di handshake"); inoltre se il frame ricevuto non supera il controllo con CRC viene scartato.

Formato frame:

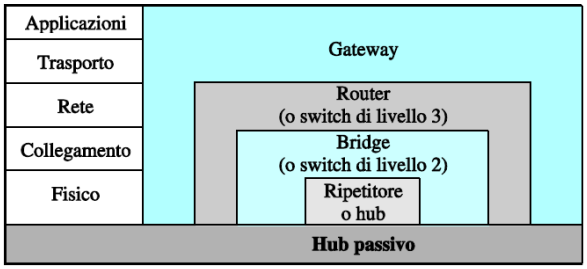


In tutte le generazioni di Ethernet, il formato del frame è identico.
E' imposta una lunghezza minima del frame (64 byte) e lunghezza massima (1518 byte). Per quest'ultima le motivazioni sono: migliore gestione dei buffer ; evitare la monopolizzazione del canale.

Livello Fisico: cambia in relazione alla generazione Ethernet considerata:

- *Ethernet Standard*: velocità di trasmissione = 10 Mbps . Implementazioni:
  - *10Base5* : cavo coassiale grosso, topologia a bus (lunghezza max 500m);
  - *10Base2*: cavo coassiale sottile e flessibile, topologia a bus (lunghezza max 185m);
  - *10Base-T*: doppino telefonico, topologia a stella (lunghezza max 100m);
  - *10Base-F*: fibra ottica, topologia a stella.
- *Ethernet veloce*: velocità di trasmissione = 100 Mbps , e retrocompatibile.
- *Ethernet gigabit*: velocità di trasmissione = 1 Gbps, e retrocompatibile.
- *Ethernet 10-gigabit*: velocità di trasmissione = 10 Gbps, e retrocompatibile.

Dispositivi di connessione: per far comunicare tra loro LAN distinte o per connettere una LAN con la rete globale, vi sono diversi dispositivi:



- **Hub Passivi**: operano sotto il livello fisico; sono connettori che permettono la continuità del segnale tra cavi di rete distinti;
- **Hub Attivi** (**Ripetitori**): connette a livello fisico due segmenti di una sottorete; rigenera il segnale. Quando più sezioni Ethernet sono connesse da un hub si comportano come LAN: si usa CSMA/CD.
- **Bridge**: controlla indirizzi MAC e prende decisioni in base a indirizzo sorgente e destinazione tramite la "*tabella del bridge*" (tab. dinamica con apprendimento) *filtrando* e *inoltrando* i pacchetti.
- **Switch livello 2**: possono operare:
  - *store-and-forward* : (come i bridge)che evita di inoltrare frame difettosi ma è lenta;
  - *cut-through*: (come i router) veloce ma può provocare la perdita di frame.

  Lo switch è trasparente ai nodi in quanto questi specificano solo l'indirizzo del nodo destinazione.
  La funzionalità base è il "*filtraggio*": se va a buon fine si attiva la fase di inoltro.
  Una funzionalità importante degli switch è quella dell'autoapprendimento sulla tabella di inoltro.
- **Router**: processa i pacchetti ricevuti dal livello collegamento e li memorizza in un buffer, e dopo averli ulteriormente elaborati li inoltra in base alle "*tabelle di routing*". Un router opera in:
  - *store-and-forward*: memorizza il pacchetto in un buffer prima di inoltrarlo;
  - *cut-through*: l'inoltro dei pacchetti può avvenire anche senza la loro completa ricezione.
- **Gateway**: sinonimo di router, ma lavora a livelli superiori a quello di rete, e trasporta i pacchetti all'esterno di una rete locale.

7- RETI WIRELESS

Le reti wireless sono reti i cui terminali sono collegati tra loro attraverso un canale radio.
Le prestazioni di una rete wireless si esaminano con un'*analisi* ***SWOT***:

Vantaggi	Svantaggi	Opportunità	Rischi
connettività ubiqua	banda di accesso minore rispetto a reti cablate	applicazioni innovative	sicurezza limitata
costi minori	canale condiviso	accesso mobile	riservatezza
tecnologia matura	sensibilità interferenze	convergenza	

➢ IEEE 802.11

Questo standard è conosciuto come "**WiFi**" e si occupa solo delle specifiche del livello fisico e livello MAC.
Una rete IEEE 802.11 è formata da una cella elementare detta BSS che contiene uno o più terminali e una Base Station (BS) che coordina la rete: tale BS nelle WLAN è chiamata "*Access Point*" (AP).
Lo standard ammette *reti centralizzate* e reti "*ad hoc*": quest'ultima non prevede l'uso dell'AP.

Livello MAC

Il livello MAC nello standard IEEE 802.11 è comune a tutte le alternative disponibili per il livello fisico.
La tecnica di accesso al canale condiviso è basata sulla **CSMA/CA** : tecnica introdotta nelle reti wireless perché in tali reti non è sempre possibile rilevare in maniera affidabile una collisione. Tale tecnica si basa sulla rilevazione di portante con l'aggiunta di una funzionalità finalizzata a prevenire le collisioni.
L'accesso al canale prevede che vi sia un ritardo (detto **IFS**) per l'invio di un frame (anche se il canale è libero) affinché si riduca l'intervallo di vulnerabilità della tecnica di accesso. Tale IFS è a sua volta diviso in:

- *SIFS*: ogni terminale deve attendere almeno un tempo pari a SIFS prima di accedere al canale;
- *AIFS*(1): dà priorità di accesso a traffici sensibili ai ritardi (es: traffico voce);
- *DIFS*: tempo di attesa di un terminale una volta trovato il canale libero prima di tentare la trasmissione di un pacchetto;
- *AIFS*(4): usato per traffico a priorità più bassa, detto "traffico di background";
- *EIFS*: tempo di attesa prima che un terminale notifichi agli altri la recezione di un frame difettoso;

L'ordine degli intervalli stabilisce una ***priorità*** nell'effettuare l'accesso al canale.

La procedura di accesso al canale può avvenire:

- *in modo casuale* (**DCF**): se il canale risulta libero, il terminale fa partire il contatore a decremento inizializzato a DIFS; terminato questo tempo se il canale risulta libero, il terminale trasmette e rimane in attesa del riscontro ACK (per un tempo pari a SIFS), il quale se non viene ricevuto si assume che si sia verificata una collisione virtuale e si entra nello stato di contesa (nel quale si entra anche nel caso che il canale viene trovato occupato trascorso il DIFS).
  Entrati nella modalità contesa, si imposta il "*backoff timer*" ad un valore scelto con prob. uniforme nella "***finestra di contesa***" (CW), la quale incrementa secondo la formula $CW(n) = \min\{2CW(n-1), CW_{Max}\}$.

  La modalità CSMA/CA presenta una criticità operativa nota come "*problema del terminale nascosto*", il quale può essere risolto con una procedura di handshake tra terminale mittente e destinatario con l'invio di messaggi ***RTS*** e ***CTS*** : l'invio del CTS viene interpretato dai restanti terminali come una prenotazione del terminale mittente dell'RTS.
  Tale procedura di handshake introduce però il cosiddetto "*problema del terminale esposto*", che può essere risolto tramite l'impostazione (in ogni terminale) di un ulteriore contatore a decremento detto ***NAV***:
  appena un terminale invia un messaggio RTS o CTS, tutti i nodi vicini leggono il campo "*Duration ID*" del frame MAC e impostano il NAV a tale valore,cosicché tali terminali non tenteranno l'accesso finché NAV≠0.
  Può nascere il cosiddetto "*problema monopolizzazione dell'accesso*" a causa della priorità assegnata a quei terminali che conquistano il canale: per evitarlo, si assegna ad ogni terminale un valore max del tempo di trasmissione detto ***TXOP***.
- *in modo ordinato* (**PCF**): tecnica usata nelle reti con AP, basata sul paradigma *polling-response*: a turno l'AP interroga i singoli terminali per permettere loro di trasmettere, senza contesa, i propri pacchetti.

La modalità PCF ha priorità rispetto alla fase DCF: questo può portare al problema che i nodi che utilizzano solo la tecnica DCF potrebbero non riuscire mai ad accedere al canale. Questo si risolve dedicando un tempo massimo per la fase PCF : i terminali che usano la tecnica DCF impostano il loro NAV a tale valore.
Il meccanismo RTS/CTS risolve il problema del terminale nascosto, ma raramente viene applicato a poiché:

- non è efficace con i frame corti;
- non è utile nel caso di reti con AP (reti infrastrutturali);
- rallenta il trasferimento a causa della fase di set-up;
- non risolve il problema del terminale esposto;
- i terminali nascosti sono in genere rari.

Poiché nelle reti wireless abbiamo solo una notifica di "errata recezione" si sono introdotti due metodi per risolvere realmente il problema:

a. *ridurre la velocità* di trasmissione dei frame;
b. *frammentazione* dei frame.

Inoltre, poiché il *consumo energetico* è un prob. importante nelle reti wireless, si ha il meccanismo del "**beacon frame**": i nodi diventano attivi ad ogni ricezione di un messaggio di beacon (altrimenti in stand-by).
Infine, si deve saper gestire il QoS offerto agli utenti: questo si realizza tramite una distribuzione temporale delle diverse fasi di accesso, ed associando intervalli di attesa minori alle procedure con livello di priorità alto.

➢ IEEE 802.16

Tale standard è noto come **WiMAX** : nasce per poter usufruire di velocità di accesso superiori a quelle di reti IEEE 802.11 e su distanze maggiori.
Rispetto a IEEE 802.11, WiMAX consente di avere:

- gestione della qualità di servizio (*QoS*);
- confidenzialità delle informazioni scambiate;

L'accesso al canale è regolato secondo la tecnica **OFDMA**.
WiMAX prevede una stazione principale detta *Base Station* che si collega con i propri utenti (*Subscriber Station*).
Le modalità di gestione dell'accesso sono:

- *TDD*: (tecnica più usata) separate temporalmente le due fasi di trasmissione dalla BS verso le SS (*downlink*) e viceversa (*uplink*), separati da un intervallo di guardia per la commutazione;
- *FDD*: vengono assegnate bande distinte alle comunicazioni uplink e downlink (separati in frequenza).

Livello Fisico

WiMAX prevede la possibilità di trasmettere l'informazione con tre tecniche (non descritte) con lo scopo di aumentare la velocità di trasferimento dei flussi di info: occorre però mantenere l'integrità dei dati trasmessi.

Gestione della QoS

WiMAX prevede quattro ***classi di servizio***:

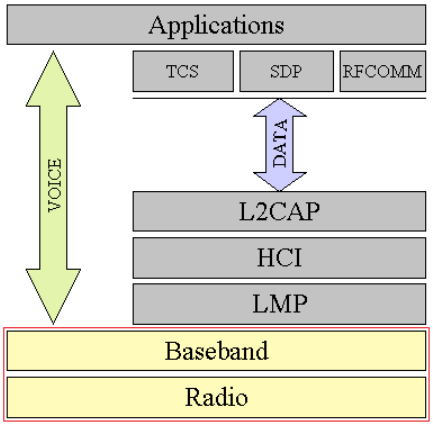
1. *Bit-rate costante*: risorse di accesso riservate per uso esclusivo ad intervallo di tempo regolari;
2. *Bit-rate variabile - tempo reale*: la BS interrompe regolarmente le SS interessate a tale servizio;
3. *Bit-rate variabile - non tempo reale*: la BS interroga regolarmente le SS interessate dal servizio;
4. *Best-effort*: la BS non interroga le SS, le quali si contendono automaticamente le risorse disponibili;

IEEE 802.15.1

Tale standard fa parte delle WPAN ed è conosciuto come **Bluetooth**.
Bluetooth prevede solo reti "*ad-hoc*": una rete Bluetooth viene chiamata ***Piconet*** e può avere fino a otto nodi attivi, di cui uno è il Master e gli altri Slave, che non comunicano tra loro; più Piconet formano una ***Scatternet***.

Architettura Protocollare Bluetooth

- *Livello Radio*: Bluetooth usa una bassa potenza di trasmissione, il data rate previsto è ≈ 1Mbps e la banda utilizzata è la ISM alla frequenza 2.4 GHz. Si suddivide la banda in 79 canali ciascuno di 1 MHz: per ridurre le interferenze si usa la tecnica **FHSS** (tempo banda di accesso lordo 625μs , netto 366μs);
- *Livello Baseband*: ha le funzionalità del livello MAC cioè per l'accesso al canale, e si usa la tecnica *polling*; mentre la trasmissione dati si fa con *TDD* con slot di durata 625μs e modalità half-duplex.
  Si prevede due tipologie di canali tra Master e Slave:
  - *SCO*: per collegamenti connection-oriented: minimizzare ritardo a discapito della qualità;
  - *ACL*: salvaguardare integrità dei dati piuttosto che il ritardo temporale.



Modalità operative

Un dispositivo Bluetooth può trovarsi nello stato di:

- *connessione*: se è connesso ad un Master ed è coinvolto nello scambio dati. I sottostati sono:
  - *active mode*: lo Slave partecipa attivamente allo scambio dati della piconet (indirizzo di 3 bit);
  - *hold mode:* stato concesso allo Slave per il risparmio energetico (non può ricevere messaggi);
  - *sniff mode*: per risparmio energetico; Master e Slave negoziano "*sniff interval*" e "*sniff offset*";
  - *park mode*: Slave non più attivo nella Piconet (indirizzo a 8 bit) ma ascolta il canale a intervalli.
- *stand-by*: lo Slave non è connesso a nessuna Piconet, oppure non è coinvolto nello scambio dati.

RFID

RFID consente ad oggetti di uso comune di poter essere parte di una rete di comunicazione (riconoscimento di oggetti o recupero dati in modo automatico), spesso senza alimentazione (alimentazione fornita dal lettore).
Un sistema RFID è composto da:

- ***Tag***: costituiti da: trasponder, antenna, batteria;
- ***Lettore***: parte attiva del sistema alimentati da sorgente propria; interrogano i tag e recuperano info.

Se un lettore riceve più risposte da tag vicini si può avere una collisione, che viene risolta con la tecnica ***Aloha***.

8- RETI DI SENSORI

Le reti di sensori possono utilizzare supporto trasmissivo cablato o wireless, quest'ultima conosciuta come **WSN**: reti autonomiche di sensori, la cui idea base è quella di combinare la misura di una o più grandezze fisiche con la trasmissione dei dati acquisiti verso il ***sink***, dove verranno elaborate.
Un sensore implementa due funzioni base:

- *sensing*: acquisire dall'ambiente le misure delle grandezze fisiche di interesse;
- *comunicazione*: trasferire i dati raccolti verso il sink. Modalità:
  - *push*: si attiva il sensore in corrispondenza di precisi istanti temporali;
  - *pull*: necessita di una sollecitazione esterna per l'attivazione delle funzionalità del sensore.

Elementi di una WSN

- ***sensori***: eseguono le misure (*sensing*) di grandezze fisiche, e trasferiscono i dati ai sink. Struttura:
  - *unità di controllo* (processore);
  - *trasmettitore/ricevitore* (interfaccia radio);
  - *batteria*;
  - *trasduttori*.
- ***sink***: provvedono alla raccolta e all'elaborazione di dati provenienti dai sensori;
- ***attuatori***: interpretano ed eseguono comandi conseguenti all'elaborazione dei dati da parte del sink;
- ***processori***: ulteriori nodi con capacità elaborative per ridurre la ridondanza dei dati da trasferire al sink.

Accesso Multiplo

Le metodologie utilizzate sono di tipo a contesa (ispirate a CSMA/CA), le quali vogliono prevenire le collisioni.

1. *MACA*: a differenza di CSMA/CA non prevede la rivelazione della portante. Ogni volta che avviene una collisione, si raddoppia la finestra di Back-off (intervallo di tempo entro cui si programma il tentativo di accesso successivo) fino ad un valore max; viceversa la finestra viene riportata al valore minimo.
   Questo meccanismo però non garantisce equità nella condivisione dell'accesso. Soluzione adottata:
   - si inserisce nell'header dei pacchetti il valore della finestra di back-off del nodo che ha avuto successo nell'accesso, in modo che gli altri nodi sincronizzano a tale valore le loro finestre;
2. *MACAW*: protocollo nato come soluzione al problema di equità del MACA: si ha il meccanismo ***MILD***;
3. *S-MAC*: protocollo di accesso multiplo nato dal requisito di cercare di preservare l'energia di un nodo in una WSN: si mette in stand-by un nodo quando questo non è partecipe all'attività della rete.
   Rimane il problema del coordinamento tra nodi vicini, che si devono trovare attivi quando si necessita della loro cooperazione per trasferire un'informazione.
   Ulteriore meccanismo di risparmio energetico:
   - *Controllo potenza di segnale*: si sceglie il livello di potenza per trasmettere dati basandosi sulla stima della potenza del mex di risposta CLR del nodo destinatario

➢ IEEE 802.15.4

IEEE 802.15.4 specifica livello fisico e livello MAC per reti **LR-WPAN** : reti wireless personali a basso data rate (250 kbit/s per circa 10 m). Tale standard è rivolto alla comunicazione tra i dispositivi classificati in:

- ***RFD***: semplici, con limitate capacità di calcolo; comunicano solo con dispositivi FFD (non tra loro);
- ***FFD***: svolgono la funzione di coordinatore; comunicano con qualsiasi dispositivo in visibilità radio.

Per quanto riguarda la comunicazione a livello fisico e MAC (sempre di tipo *link-to-link*), le modalità sono:

- *invio dati da un nodo al coordinatore*;
- *invio dati dal coordinatore al nodo*;
- *comunicazione bidirezionale tra due nodi* (reti peer-to-peer), usato solo con dispositivi FFD.

Il livello MAC fornisce due tipi servizi:

- *MAC Data Service*: permette la trasmissione e la recezione dei pacchetti MPDU verso il livello fisico;
- *MAC management service*: che fornisce servizi quali accesso al canale, tempi divisi in slot temporali, invio di pacchetti ACK, associazione/dissociazione ad una WPAN, servizi legati alla sicurezza.

Lo standard 802.15.4 prevede la condivisone del canale su base accesso casuale secondo la CSMA/CA. Modalità:

- *Beaconless CSMA/CA*: il nodo ascolta il canale: se è libero accede; altrimenti attende (periodo aleatorio);
- *Beacon-enabled CSMA/CA*: ripartizione temporale gestita dal coordinatore della WPAN: si usano pacchetti di sincronismo (***Beacon Frame***) con lo scopo di suddividere l'asse temporale in superframe.

➢ IEEE 802.15.3

Standard nato per le reti **HR-WPAN**: reti wireless ad alto data rate (da 11 a 55 Mbps per distanze > 70m).
Si introduce il concetto di QoS ed è adatto a trasferimento di dati multimediali.
Tale standard ha molti punti in comune con Bluetooth, ma definisce modalità per coesistere con gli standard già esistenti. La rete realizzata da dispositivi IEEE 802.15.3 viene definita piconet, ma a differenza di Bluetooth, i dispositivi possono comunicare tra di loro (D2D); si possono inoltre realizzare reti estese (più piconet insieme).
La sincronizzazione della comunicazione tra dispositivi avviene mediante la definizione di un *superframe*.

➢ 6LoWPAN

Protocollo progettato per adeguare IPv6 al livello fisico e MAC dello standard IEEE 802.15.4 (LR-WPAN): il problema della trasmissione di un pacchetto IPv6 in una rete LR-WPAN è la sua dimensione (dimensione minima IPv6 = 1280 byte ; dimensione massima IEEE 802.15.4 = 127 byte). Il protocollo 6LoWPAN definisce:

- *compressione dell'Header*: i campi dell'header IPv6 vengono compressi o eliminati quando *l'adaption-layer* (livello di adattamento introdotto tra il livello rete e collegamento) è in grado di ricavarli direttamente dalle informazioni del livello MAC o li conosce a priori;
- *frammentazione*: pacchetti IPv6 frammentati in più frame di livello MAC per requisiti MTU IPv6 minimi.

Da notare infine che l'header 6LoWPAN presenta una sequenza analoga all'header IPv6.

➢ Data Centric Forwarding

La modalità **DC** nasce come metodologia di inoltro dei dati in una WSN alternativa alla tecnica Address-based.
Tecniche di inoltro delle informazioni usate in una WSN:

- ***Flooding***: ogni nodo di una WSN che riceve o genera un messaggio lo inoltra a tutti i suoi vicini : vi sono problemi di congestione e consumo potenza, parzialmente risolti con il TTL nell'header del messaggio;
- ***Gossiping***: l'inoltro del messaggio non a tutti i vicini ma solo ad uno, in base ad una decisione statistica;
- ***Direct Diffusion***: si identifica i dati generati da un sensore mediante una coppia attributo-valore. Fasi:
  - *Interest e Gradient*: un elemento della WSN interessato a certe informazioni inoltra la richiesta di interesse (interest) in modalità flooding: ad ogni possibile rotta dal nodo proprietario dell'info al richiedente viene associata una misura (gradient) di qualità; si sceglie poi la rotta migliore;
  - *Data Propagation*: ricevuta la dichiarazione di interesse, l'info viene inoltrata sulla rotta scelta;
  - *Reinforcement*: attivata la procedura di raccolta delle info, il nodo richiedente (sink) può rinforzare la sua richiesta aggiungendo ulteriori specifiche.

  Tale tecnica è di tipo *query-driven*: non si adatta ad applicazioni che richiedono un continuo invio di dati.
- ***SPIN***: protocollo basato sulla diffusione nella WSN dei metadati (descrizione dei dati). Tipi di messaggi:
  - *ADV*: nodo che pubblicizza ai vicini il metadato;
  - *REQ*: il vicino è interessato al metadato e ne fa richiesta al nodo sorgente;
  - *DATA*: si invia realmente il metadato.

  Questo processo si ripete nella rete per i nuovi proprietari del metadato: non si garantisce però il trasferimento dell'informazione a tutti i nodi della rete (i nodi non interessati bloccano i dati).

➢ In-Network Processing

Spesso in una WSN non è efficiente trasmettere direttamente le informazioni raccolte verso uno stesso sink.
Metodologie impiegate per ridurre la ridondanza, contenere la congestione e ridurre il consumo di potenza:

- ***Data Aggregation***: combinazione di informazioni generati da sensori diversi in un singolo messaggio;
- ***Data Fusion***: processo di elaborazione delle info acquisite individualmente da sensori diversi che permettono di rendere disponibili info non rilevate direttamente (aggregazione semantica);
- ***Clustering***: partizionare un insieme di oggetti secondo un criterio: in una WSN i metodi di clustering sono efficaci per migliorare l'inoltro dei messaggi dalla rete verso l'esterno e viceversa. Tecniche:
  - *LEACH*: distribuisce il dispendio di potenza in maniera equa tra tutti i nodi della WSN: si sceglie in maniera casuale alcuni nodi e si eleggono clusterhead (temporanei), i quali gestiscono le comunicazioni con i nodi del proprio cluster. A questo scopo si ha l'algoritmo di LEACH.
  - *HEED*: evoluzione della LEACH; si determina il numero di clusterhead effettivamente necessari. La differenza rispetto a LEACH è l'utilizzo dell'informazione sull'energia residua nei nodi.

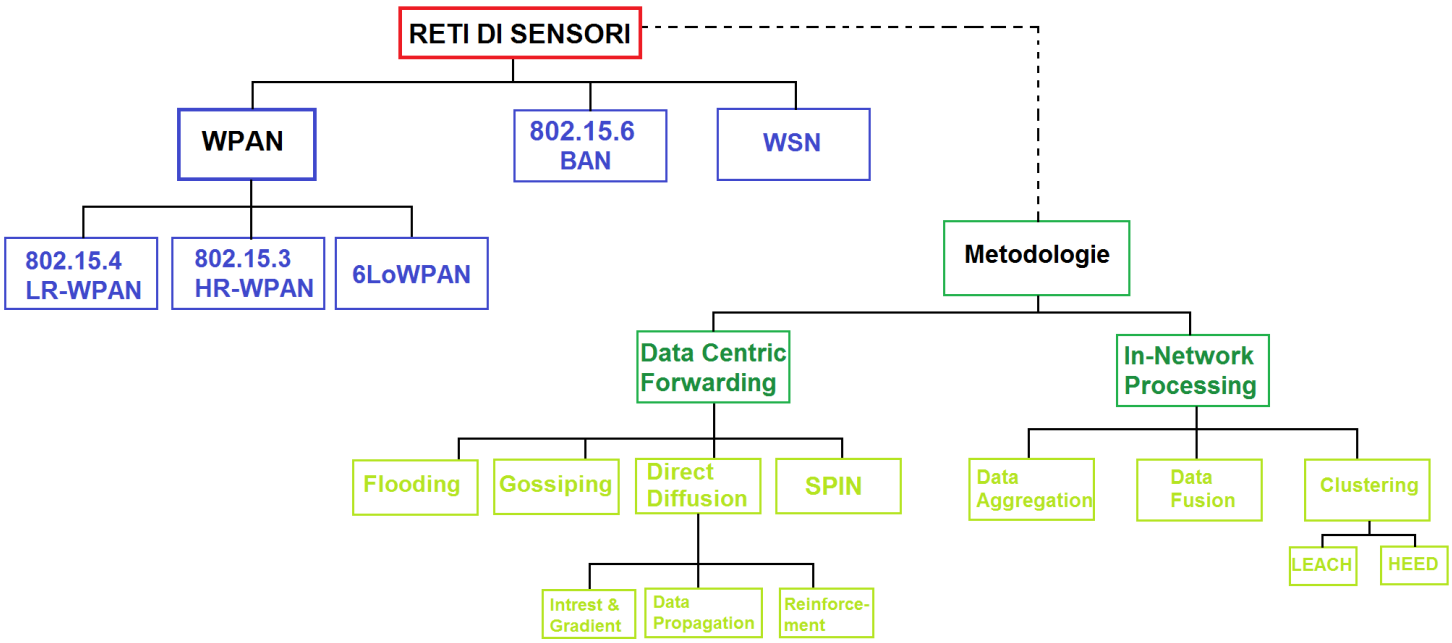
IEEE 802.15.6 - Body Area Network (BAN)

Le **BAN** sono reti formate da dispositivi indossabili, il cui raggio di copertura è circa 1 m. Si introduce il QoS.
La comunicazione tra nodi e hub è bidirezionale, ed avviene dividendo l'asse temporale in superframe (uguali), i quali possono essere divisi a loro volta in slot temporali (uguali). Un hub può operare in uno dei tre modi:

- *superframe con beacon*;
- *superframe senza beacon*: necessita della temporizzazione perché non vi è il beacon;
- *senza superframe senza beacon*.

La tecnica di accesso al canale può essere di tipo:

- *CSMA/CA*;
- *Aloha Slotted*: usata solo se l'hub decide di adottare la tecnica del superframe con beacon.



9- RETE ISDN

Raccomandazione X.25

X.25 è un protocollo che si riferisce all'interfaccia utente-rete: comprende 3 livelli (**Fisico**, **Collegamento** e **Pacchetto**) corrispondenti ai primi 3 livelli OSI. È stata pensata per reti che utilizzano mezzi trasmissivi di scarsa qualità, quindi vi è molto controllo di integrità a livello Collegamento. I servizi previsti da X.25 sono:

- *Chiamata Virtuale* : fase di set-up per ogni richiesta di collegamento;
- *Chiamata Virtuale Permanente*: cammino virtuale preventivamente definito (no fase di set-up).

La rete è composta da **DTE** (*Data Terminal Equipment*) cioè gli apparati utente, e da **DCE** (*Data Circuit Equipment*) cioè i nodi della rete a cui il DTE è connesso. *X.25 definisce appunto l'interfaccia tra DTE e DCE*.
In particolare:

- Il livello Pacchetto identifica un circuito con un'etichetta e il singolo pacchetto con un numero di sequenza;
- Il livello Collegamento provvede a creare la trama *LAPB* e la delimita da due campi Flag (in cima ed in fondo) per realizzare la funzionalità di framing (usa il meccanismo *bit stuffing*).

L'operazione di multiplexing realizzata dai DCE consiste nell'associare alla coppia (etichetta1, ingresso) la coppia (etichetta2, uscita).

Frame Relay

Evoluzione di X.25, implementa solo i primi 2 livelli OSI. Con un minor costo permette velocità di accesso più elevate (da 1,544 a 44,376 Mbps) inoltrando i flussi informativi su un circuito virtuale, ma fornisce un servizio non affidabile (i frame corrotti vengono scartati).
Una caratteristica importante è la sua *dinamicità*: consente un'allocazione di capacità di accesso variabile e scalabile in relazione alle esigenze dell'utente (on-demand).
Come X.25, fornisce un servizio di ***Circuito Virtuale Permanente*** o ***Temporaneo***. In modo simile a X.25 identifica ogni *circuito virtuale* con un'etichetta (**DLCI**), e la commutazione avviene associando alla coppia (DLCI1, ingresso) la coppia (DLCI2, uscita).

ISDN

Le reti ISDN nascono con il proposito di supportare, tramite un'unica interfaccia, servizi diversi (dati, voce…) utilizzando le infrastrutture già esistenti. Le modalità sono ***commutazione di circuito***, ***pacchetto*** o ***dedicata***. Ha un'architettura a livelli, dunque permette per esempio di utilizzare X.25 come protocollo di rete.
ISDN prevede una rete di segnalazione basata sul protocollo **SS7**.
Dal punto di vista fisico, ISDN è basata sul doppino telefonico, mentre dal punto di vista logico i trasporto delle info è strutturato sulla base dei seguenti tipi di canale:

- **B** (64 Kbps): canale di base di un utente, supporta tutti i servizi. Le modalità di connessioni possibili sono *commutazione di circuito* (con segnalazione su canale D separato), *commutazione di pacchetto* (usando X.25), a *trama* (usando Frame Relay) o *dedicata*.
- **D** (16-64 Kbps): trasferiscono le info di segnalazione per supportare i canali B o dati a basso bitrate.
- **H** (384-1920 Kbps): usati per trasferire flussi aggregati in TDM, e la trasmissione veloce di dati.

Sono stati definiti due tipi di accesso: **base** (2B+ D) e **primario** (30B+D o canali H).

L'accesso ad una rete ISDN di un dispositivo standard ISDN (**TE1**) avviene con un'interfaccia diretta ad un dispositivo che opera fino a livello rete (**NT2**) tramite un punto di riferimento **S**. Per collegare invece un dispositivo non standard occorre inserire un adattatore (**TA**) collegato al terminale (**TE2**) tramite un punto di riferimento **R**. L'NT2 è collegato tramite un punto di riferimento **T** ad un dispositivo che opera a livello collegamento (**NT1**). Quest'ultimo comunica con il resto della rete tramite un punto di riferimento **U**.

L'architettura protocollare di ISDN specifica i primi 3 livelli OSI, mentre gli altri sono di pertinenza dell'utente, implementati E2E.
Il *livello fisico* è comune per i canali B e D, perché si può utilizzare lo stesso mezzo fisico per trasmettere dati anche di tipo diverso.
Il *livello 2* del canale D , basato sul protocollo LAPD, è indipendente dalle applicazioni di livello 3 (*segnalazione di controllo, commutazione pacchetto X.25*); le applicazioni supportate dal canale B, basato su LAPB, sono: *commutazione di pacchetto, comm. di circuito* o *comm. di circuito semipermanente*.

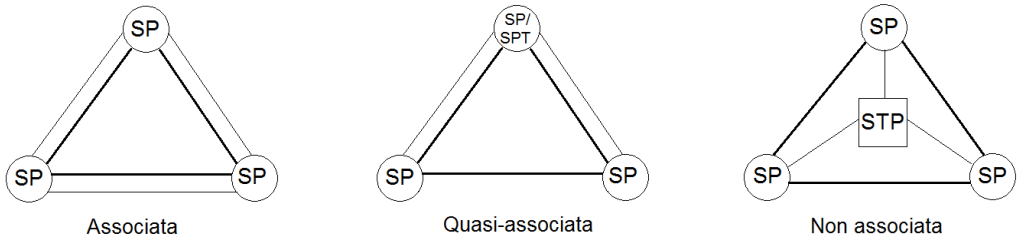
10- RETI DI SEGNALAZIONE & SS7

Una **rete di segnalazione** è un sistema che utilizza dei messaggi (***segnalazioni***) per supportare l'attivazione e il funzionamento di servizi (fonici) a *commutazione di circuito*. La segnalazione può essere:

- ***Utente***: attiva la connessione tra l'utente e la rete vista come mezzo;
- ***Inter-Nodo***: gestisce le comunicazioni tra nodi interni della rete per permettere il trasf. dei dati. Questa tipologia può essere realizzata in due modi:
  - *Associata al canale*: dati e segnalazione sono trasmessi sullo stesso canale fisico. Metodi:
    - ° *In banda*: la segnalazione è trasmessa insieme (o al posto) dell'informazione;
    - ° *Fuori banda*: la segnalazione utilizza un canale logico diverso dall'informazione: sono flussi indipendenti.
  - *A canale comune*: I canali per dati e segnalazioni sono distinti fisicamente.

Quest'ultima metodologia è il riferimento per l'implementazione dei sistemi di segnalazione.
È realizzata dunque un'infrastruttura apposita, esclusiva per la trasmissione della segnalazione: essa è la cosiddetta rete di segnalazione, la quale utilizza la commutazione di circuito, ed è formata da *Signal Point* -SP (aggregatore di segnali) e da *Signal Trasfer Point* - STP (router).
Tale rete di segnalazione può essere:

- *Associata*: usa solo SP; la rete è parallela a quella dati, costa meno per quanto riguarda gli apparati ma è vincolata dalla topologia di rete di base.
- *Quasi associata*: utilizza dispositivi integrati SP/STP;
- *Non associata*: utilizza sia SP che STP, ha un costo maggiore ma permette di utilizzare meno collegamenti, dunque è indipendente dalla rete di base.



SS7

È un sistema di segnalazione moderno, basato su una *tecnologia numerica*.
La rete che veicola i messaggi di segnalazione è una rete a *commutazione di pacchetto*, in particolare basata su tecnologia a *datagramma*.
SS7 ha derivato il principio dell'architettura a strati (tipo ISO/OSI) , e dunque utilizza un'architettura protocollare composta da 4 livelli:

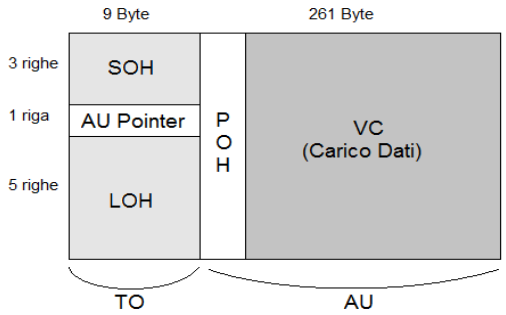
- I primi 3 livelli (*MTP L1, L2* e *L3*) corrispondono ai primi 3 livelli OSI e costituiscono nel loro insieme la parte di trasferimento di messaggi;
- il quarto (implementato E2E) comprende le funzionalità del resto della pila OSI (*User Part*).

11- RETE SDH

SDH è una metodologia per il trasporto dell'informazione in *forma digitale* in reti WAN ad alta velocità; difatti il mezzo trasmissivo è la *fibra ottica*.
Una rete SDH è basata sulla tecnica TDM sincrona, che richiede la sincronizzazione di tutti gli apparati della rete (si inviano trame anche se non trasportano informazione).

A causa delle richieste a larga banda, si è dovuto sostituire la tecnologia PDH con la cosiddetta SDH. Vantaggi:

- *multiplazione sincrona*: si possono aggregare flussi a rate più basso con flussi a velocità più elevate;
- *retro-compatibilità con PDH*;
- topologia di *rete ad anello*;
- *ambiente di tipo aperto* : cooperazione apparati di rete di costruttori diversi;
- *standard consolidato*;
- *uniformità* della gerarchia a livello mondiale (cosa che non valeva per il PDH).

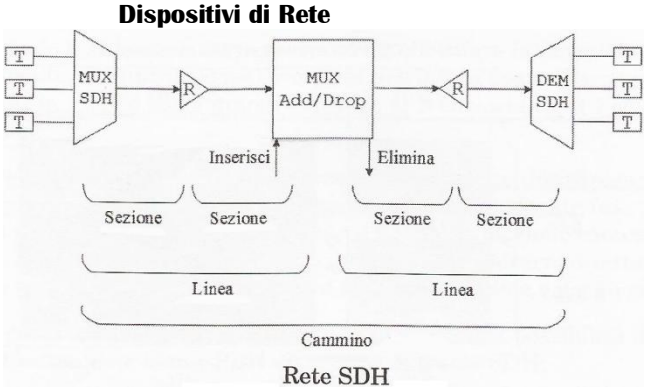


L'elemento caratterizzante SDH è una speciale *struttura a trama* detta ***STM-1*** (periodo di ripetizione 125μs) che con l'aggiunta dell'overhead permette (oltre all'estrazione diretta di un singolo traffico tributario senza la demultiplazione) il trasferimento di informazioni essenziali per la corretta gestione della rete e per la sua auto-protezione a fronte di guasti : si raggiunge così elevati livelli di qualità di servizio.
La trama STM-1 è rappresentata come matrice di byte: 9×270 , per un totale di 2430 byte, trasmessi a velocità 155.52 Mbps.

La trama SDH viene trasmessa sequenzialmente per righe. Essa è suddivisa in due parti fondamentali:

- *Transport Overhead* (TO): dedicata alle informazioni di servizio generali; a sua volta suddivisa in:
  - *SOH*: informazioni di servizio relative alla trama nel suo complesso;
  - *AU Pointer*: specifica la posizione dei dati nel frame;
  - *LOH*: dedicata a funzioni di recupero di errori.
- *Adiminstrative Unit* (AU): dedicata al trasporto di dati e segnalazione di cammino; a sua volta divisa in:
  - *VC*: trasporto dei singoli flussi tributari;
  - *POH*: informazioni di servizio aggiuntive.

Dispositivi di Rete

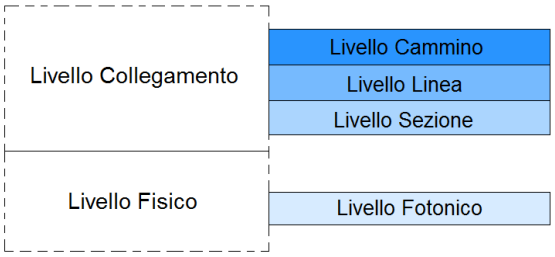


Gli elementi della Rete SDH sono:

- *MUX/DEM SDH*: punto di accesso alla rete SDH da parte dei terminali di utente: ricevono/trasmettono il segnale e lo inseriscono in una trama SDH (MUX); a lato destinazione si estraggono i vari flussi tributari (DEM);
- *Ripetitore*: rigenera il segnale per lunghe distanze;
- *MUX Add/Drop*: inserire ed estrarre flussi tributari a bit rate inferiore rispetto al bit rate della trama ricevuta;
- *Terminali*: utenti con info da trasmettere/ricevere.

Architettura a strati

L'architettura di una rete SDH è a strati (come ISO/OSI), in cui vi sono quattro livelli:



1. ***Livello Fotonico***: trasmissione delle trame nel canale ottico di collegamento;
2. ***Livello Sezione***: garantisce l'integrità e la gestione (*framing*) del trasporto del segnale in una sezione;
3. ***Livello Linea***: gestione del trasferimento delle trame SDH su base linea;
4. ***Livello Cammino***: garantisce il corretto trasferimento di un flusso informativo da terminale sorg. a dest.

In SDH, al contrario di ISO/OSI, non vi è l'incapsulamento successivo: info già presenti nella struttura della trama.

12- RETE ATM

La funzionalità primaria di **ATM** è quella di gestire il *trasporto ad alto data rate* (tramite *fibra ottica*) di diverse tipologie di traffico (voce, dati, multimediale); oltre a prevedere l'uso di commutazione veloce di pacchetto.

ATM nasce dopo l'esigenza di definire un'unica modalità di trasporto dell'informazione che potesse adattarsi al trasferimento di traffici diversi (file, e-mail, voce, video) e di essere in grado di rispettare requisiti di QoS diversi. La struttura ATM non trova un riscontro preciso nella pila ISO/OSI.

ATM incapsula i dati in un'unità fondamentale di lunghezza fissa, detta ***cella***: 53 byte, di cui 5 di header.
Ciò è stato fatto perché si facilita l'interpretazione dei messaggi stessi ai dispositivi di rete, aumentando la velocità di commutazione: il circuito virtuale viene definito "***canale virtuale***", che non risulta mai inattivo.
La tecnica di accesso al canale è di tipo TDMA sincrono, e non vi è un'assegnazione rigida delle risorse.

ATM identifica il traffico secondo le seguenti classi:

- *CBR*: la rete ATM garantisce un data rate costante al collegamento per tutta la sua durata;
- *VBR-Real Time*: traffico di tipo interattivo che richiede il rispetto di precisi vincoli sul ritardo di trasporto;
- *VBR-Non Real Time*: traffico di tipo interattivo, ma non si ha garanzia riguardo il ritardo di trasferimento;
- *ABR*: non richiede né un trasferimento in tempo reale né un data rate costante: adattamento dinamico;
- *UBR*: simile ad ABR ma non ci sono garanzie: classico servizio "Best effort".

NB: ATM non prevede il controllo dell'integrità delle info, poichè la fibra ottica è in pratica immune da errori.

Cella ATM

Se il flusso di utente è > 48byte, si attiva la suddivisione del flusso stesso (*frammentazione*) in più celle. Formati:

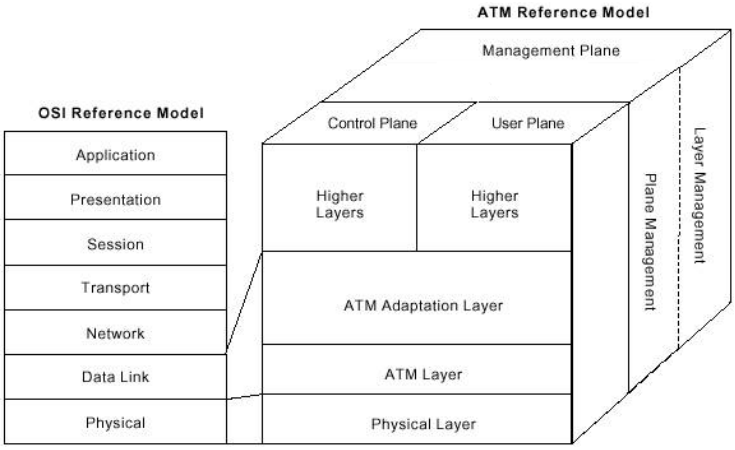
- ***UNI***: riferito ai messaggi scambiati tra i terminali di utente e i nodi della rete;
- ***NNI***: utilizzato per lo scambio di messaggi tra due nodi interni alla rete ATM.

A parte piccole differenze nei campi, quelli principali sono:

- *controllo*: (presente solo nella cella UNI) controllo del flusso;
- *VPI*: identifica il percorso virtuale di uno specifico flusso;
- *VCI*: identifica il canale virtuale associato;
- *Tipo*: identifica la tipologia di dati;
- *CLP*: (1 bit) priorità di eliminazione;
- *CRC*: rileva e corregge gli errori.

Architettura protocollare ATM

Lo standard ATM prevede una struttura protocollare con tre piani distinti:



1. *Piano d'utente*: responsabile dei servizi richiesti dall'utente;

2. *Piano di controllo*: responsabile del controllo e della segnalazione nella rete ATM;

3. *Piano di gestione*: sovraintende ad una corretta cooperazione tra i piani utente e controllo.

I piano di utente e il piano di controllo sono strutturati in livelli:

- *livello fisico*: comune ad entrambi i piani; è il responsabile del trasferimento delle info dalla sorgente alla destinazione. Il livello fisico ATM a sua volta è diviso in:
  - *PM*: interfaccia con il messo fisico vero e proprio;
  - *TC*: responsabile di trasformare flusso di celle in flusso di bit (e viceversa).

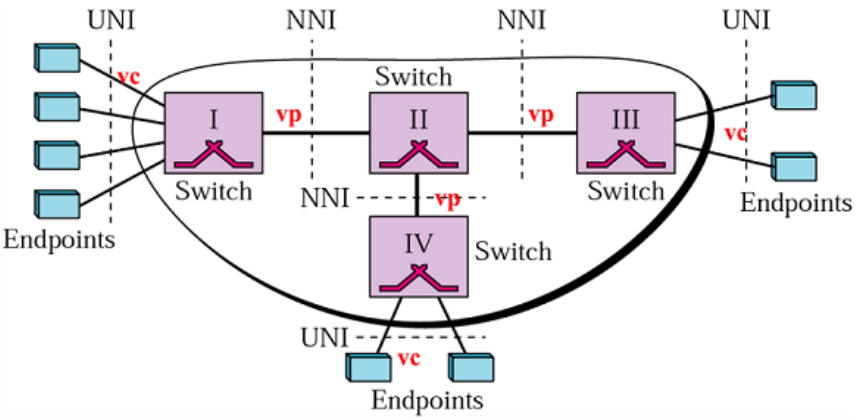
- *livello ATM*: comune ad entrambi i piani; indipendente dal livello fisico ed è responsabile della consegna delle celle tra mittente e destinatario: si basa sul principio "*Core & Edge*", in cui le funzionalità superiori al livello ATM risiedono solo nel terminale sorgente e destinazione.
  Il compito di tale livello è quello di instradare le celle nella rete lungo il percorso verso la destinazione: l'instradamento avviene leggendo i valori dei campi ***VPI*** e ***VCI***, tramite la *routing table* (modalità simile a quella di X.25 e Frame Relay).
  Poiché non vi è un servizio affidabile su base link-to-link, si introduce un controllo su base E2E.
- *Livello AAL*: rende compatibile il flusso informativo con il formato delle celle ATM (le funzionalità di AAL sono implementate nel nodo mittente e destinazione): in pratica lo strato AAL, in trasmissione, converte l'informazione ricevuta in segmenti di 48byte; in ricezione converte il payload nel formato utilizzabile dai livelli superiori. La suddivisione e la ricostruzione avviene con modalità diverse in base al servizio del livello superiore: ogni classe di servizio ATM ha il suo protocollo. In particolare questi sono:
  - *AAL 1*: bit rate costante;
  - *AAL2*: bit rate variabile;
  - *AAL 3/4*: usato per servizi non in tempo reale, orientati o non alla connessione;
  - *AAL 5*: best effort.

  AAL è diviso in due sottolivelli:
  - *SAR*: responsabile della segmentazione e del riassemblaggio;
  - *CS*: controlla che in fase di segmentazione e riassemblaggio non vi siano errori.

Architettura ATM

ATM prevede l'uso di due sole interfacce:

1. ***UNI***: definisce le procedure con cui un terminale finale ATM e un nodo della rete "parlano" tra loro;
2. ***NNI***: utilizzata per far comunicare tra loro i commutatori della rete ATM.



ATM prevede il trasporto delle informazioni in modalità *connection-oriented* (circuito virtuale): tra due utenti (mittente-destinazione) viene creato un collegamento virtuale seguito da tutte le celle del flusso informativo.
La connessione fisica è identificata con:

- ***TP***: canale di trasmissione;
- ***VP***: cammini virtuali che sommati formano la capacità di trasporto del TP;
- ***VC***: collegamenti virtuali che sommati formano la banda allocata ai VP.

Le connessioni virtuali tra coppie sorgente/destinazione sono identificate in maniera univoca tramite il ***VPI*** (identificatore di cammino virtuale) che è principale, e il ***VCI*** (identificatore di circuito virtuale) che è secondario.
I valori *VPI/VCI* sono locali e possono essere ridefiniti dai nodi della rete ATM per esigenze di switching (label swapping).

13- ALGORITMI DI ROUTING

Il compito di instradare un pacchetto dalla sorgente alla destinazione spetta ai ***protocolli di routing***, i quali si trovano a livello IP per reti TCP/IP. Un **algoritmo di routing** (o *instradamento*) è una funzionalità software che, in base ad uno specifico criterio, decide verso quale interfaccia di uscita andranno inoltrati i pacchetti: per raggiungere tale obiettivo può esser necessario avere una conoscenza di tutta la rete (pre-acquisita o aggiornata) per individuare i percorsi E2E migliori.

Il **router** è il dispositivo che attua gli algoritmi di instradamento: quest'ultimo può essere attuato mediante l'ausilio di ***tabelle di routing***, aggiornate periodicamente o acquisite in modalità reattiva.
Criteri per confrontare algoritmi di routing:

- *semplicità*: l'algoritmo deve essere facilmente implementabile (poche risorse computazionali);
- *robustezza*: l'algoritmo deve funzionare per qualsiasi topologia della rete e condizioni di traffico, adeguandosi ai cambiamenti imprevisti;
- *stabilità*: la convergenza sulla soluzione ottima di equilibrio deve essere più veloce possibile;
- *ottimalità*: l'algoritmo deve scegliere il percorso ottimo secondo determinate metriche e criteri.

Il routing può essere: ***diretto*** (mitt. e dest. in una stessa rete) o ***indiretto*** (mitt. e dest. in reti diverse).
Oltre alla funzionalità di routing, il router ha anche quella di *forwarding*; in particolare:

- ***routing***: si riferisce ad una operazione che coinvolge tutti i router di una rete che concorrono ad individuare percorsi ottimi tra coppie sorgente e destinazione, prendendo dunque delle decisioni;
- ***forwarding*** (inoltro): si riferisce al trasferimento di un pacchetto da un'interfaccia in ingresso del router all'interfaccia di uscita desiderata, senza prendere decisioni ma solo ricercando l'uscita in base alle info.

Gli algoritmi di routing possono essere classificati in base a certi criteri. In particolare:

- ***statici***: basati su decisioni predefinite e non attuali; non sono dunque reattivi.
- ***dinamici***: prevedono procedure di aggiornamento periodico delle decisioni, oppure in base a reazioni.

- ***con tabella***: permettono di associare linee d'uscita disponibili alle specifiche richieste di instradamento;
- ***senza tabella***: non prevede nessun abbinamento tra linee di ingresso e uscita: algoritmi di tipo reattivo;
- ***gerarchici***: riferiti a reti estese nel quale non è praticamente possibile eseguire un instradamento diretto, e dunque si partiziona gli utenti in sottoreti secondo uno schema gerarchico (ciascuna sottorete può attuare un instradamento con metodologie proprie e indipendenti).

- ***centralizzati***: prevedono esclusivamente tabelle e un'unità di elaborazione centralizzata;
- ***distribuiti***: l'algoritmo viene eseguito in forma distribuita (cooperativa); si prevede l'uso di tabelle;
- ***isolati***: realizzazione senza tabella; l'algoritmo viene eseguito locale (stand-alone).

Algoritmi senza tabella

Questi operano in modalità *reattiva*: non hanno disponibili le info necessarie per definire la politica di instradamento dei flussi. Tale politica viene dunque stabilita su richiesta (*on-demand*). Tecniche:

- *Random*: tecnica semplice, in quanto prevede l'inoltro del pacchetto su uno dei collegamenti disponibili (eccetto quello da cui il pacchetto è arrivato) selezionandolo secondo la distribuzione uniforme. Lo svantaggio è che non si ha garanzia riguardo un utilizzo ottimo delle risorse di rete.
- *Flooding*: appena arriva un pacchetto si replica su tutte le interfacce del router (eccetto quella su cui è arrivato il pacchetto). Tale tecnica è molto robusta, non richiede decisioni e rende sicura la consegna del pacchetto, ma ha lo svantaggio di copiare tante volte uno stesso pacchetto che si trova nella rete.
- *Source routing*: algoritmo "source routing": il nodo sorgente specifica il percorso del pacchetto nella rete fino al nodo di destinazione (info presente nell'header del pacchetto). Modalità:
  - *path server* (centralizzato): il percorso sorgente-destinazione è inviato al nodo sorgente da un server centrale; l'affidabilità è legata all'affidabilità del server centrale stesso.
  - *path discovery* (distribuito): tutti i nodi cooperano a definire il percorso da seguire (pacchetto esploratore): si memorizza l'indirizzo di ogni nodo nell'header del pacchetto.

Algoritmi con tabella

Tali algoritmi possono essere di tipo *centralizzato* o *distribuito*. A seconda che aggiornino le informazioni nelle tabelle oppure le lascino inalterate una volta definite, si parla di algoritmi ***dinamici*** o ***statici***.
Il principio di ottimalità consiste nell'individuare il percorso a **metrica** (*costo*) minima che colleghi la sorg. alla dest. ; si noti che la metrica può avere diverse definizioni: ritardo di propagazione, banda, affidabilità. Algoritmi:

- **Distance Vector**: *algoritmo con tabella, dinamico e distribuito*.
  La *tabella* di ogni nodo (router) contiene le info aggiornate periodicamente relative alla distanza *con ogni possibile destinazione* ed il Next-Hop router a cui inoltrare i pacchetti. La configurazione ottima della tabella di routing viene determinata tramite l'algoritmo di ***Bellman-Ford***:
  " *Trovare percorsi a costo minore a partire da un nodo sorgente, selezionandoli progressivamente* ".
  Per implementare tale algoritmo ogni router dispone della tabella di routing e di una struttura dati detta ***distance vector*** per ogni collegamento: essa contiene info ricavate dalla tabella di routing del router collegato tramite quel link. Il calcolo delle tabelle avviene dunque per fusione di tutti i distance vector.
  Tale DV si può pensarlo come formato da coppie: *indirizzo nodo destinazione, costo collegamento*.
  Criticità: complessità computazionale elevata & lenta convergenza ad un instradamento stabile; per questo motivo Bellman-Ford è poco adatto a contesti con elevato livello di dinamicità.

  Ulteriore criticità : ***conteggio all'infinito***; la lenta convergenza causa un ritardo nella percezione della rete di un guasto di un collegamento o di un nodo (es. con tre nodi A,B,C). Soluzioni:
  - *infinito finito*: quando il costo di un link supera un valore max di riferimento, il link stesso non è in pratica possibile;
  - *split horizon*: prevede l'invio di aggiornamenti solo per i cammini verso nodi non connessi direttamente con il nodo di destinazione degli aggiornamenti: si risolve il problema della instabilità, ma nel DV quando una info non viene aggiornata per troppo tempo viene eliminata.
  - *poisoned reverse*: risolve il problema della riga sopra, assegnando un valore elevato al costo dei collegamenti che coinvolgono il router di destinazione del DV: non si aggiornano le scelte dei collegamenti relativi, ed inoltre si rinfresca le info della tabella evitando l'eliminazione.

- **Link state**: *algoritmo con tabella, dinamico e distribuito*.
  Tale algoritmo prevede che ogni nodo acquisisce la conoscenza globale della rete partendo dalla conoscenza (metrica o costo) dello *stato dei collegamenti verso i nodi vicini*: acquisite tali info, il nodo proprietario le condivide con i suoi vicini inviando dei pacchetti detti **LSP** in modalità *flooding*.
  Ogni nodo può così individuare i cammini migliori verso tutte le possibili destinazioni: si assume quindi che ogni nodo (router) disponga della conoscenza (mappa) completa della rete.
  L'algoritmo per la creazione della tabella ottima di routing è quello di ***Dijkstra***:
  " *Trovare percorsi migliori da un nodo sorgente verso tutti gli altri nodi, definendoli per costi crescenti* ".
  La mappa completa della rete viene acquisita in maniera cooperativa tramite lo scambio di LSP, trasmessi in modalità *flooding selettiva*: seleziona i collegamenti sui quali ripetere un pacchetto in base alla stima della direzione di arrivo.
  Il link state dunque può gestire reti di grandi dimensioni, converge rapidamente e difficilmente genera cicli; lo svantaggio risiede nella complessità di elaborazione che i router devono possedere.

Confronto tra i due algoritmi:

- nel DV ogni nodo comunica solo con i vicini per scambiarsi informazioni riguardo al costo del percorso tra di loro; nel link state ogni nodo comunica con tutti i nodi della rete per acquisire una visione globale;
- nel DV tutti i router cooperano attivamente per definire tabelle ottime di routing; nel link state i router interagiscono tra loro solo per diffondere nella rete le info ai router vicini necessarie per aggiornare lo *stato* dei collegamenti: ogni router provvederà in modalità autonoma a ridefinire la propria tabella di routing secondo l'algoritmo di Dijkstra (complessità $N \cdot log(N)$);
- in caso di variazione di costo per un collegamento, con link state tutta la rete viene inondata indipendentemente dall'effettiva utilità; nel DV il cambio di costo viene notificato a tutti i nodi della rete solo se esso introduce un cambio di percorso a minor costo per uno dei nodi collegati a quel link;
- link state è più robusto rispetto a DV, poichè in quest'ultimo ogni calcolo sbagliato dei costi viene passato ai nodi adiacenti, favorendo il propagarsi dell'errore nell'intera rete.

Esempi pratici di algoritmi:

- **RIP** : esempio pratico di protocollo basato sul *distance-vector* (DV).
  Esso utilizza il numero di salti tra sorgente e destinazione come metrica di costo (ogni salto costa 1).
  Tale protocollo evita l'instabilità dei metodi DV impostando come valore di irraggiungibilità di un nodo il valore 15 (questa scelta limita l'uso di RIP in reti con distanza < 15).
  In tale protocollo, i router vicini si scambiano le info di aggiornamento (*RIP advertisement*) delle tabelle ogni 30 secondi: trascorso un tempo max di 180 s, tale router viene considerato irraggiungibile.
  A livello applicazione il RIP è implementato da un processo ***demone***, sempre in esecuzione, che offre servizi di routing. Le caratteristiche principali sono:
  - I RIP advertisement hanno una struttura semplice e vengono scambiati solo con i router vicini;
  - La convergenza è relativamente veloce grazie ai router che non sono "troppo distanti". Non viene evitata l'instabilità dovuta al problema del conteggio all'infinito che tuttavia può essere contenuto utilizzando i metodi *split-horizon* e *poissoned reverse*;
  - La robustezza è garantita dalla modalità di aggiornamento prevista che consente di diffondere nella rete ogni variazione di contesto che genera cambiamenti nelle decisioni di instradamento.

- **OSPF**: esempio pratico di protocollo basato sul *link state*.
  Tale protocollo è "*aperto*", cioè le sue specifiche sono pubbliche. Esso è un'evoluzione del protocollo RIP, ed utilizza la modalità *flooding* per diffondere info relative allo stato dei collegamenti .
  Si basa sull'algoritmo di Dijsktra, in cui i costi possono essere diversi tra loro; infatti OSPF non si occupa della metrica utilizzata ma definisce una procedura ad essa trasparente.
  Infine, i pacchetti di aggiornamento (LSP) vengono trasportati direttamente da IP come se fossero provenienti dal livello TCP. Le caratteristiche principali sono:
  - Lo scambio di informazioni tra i router OSPF può essere autenticato;
  - Quando sono presenti percorsi con lo stesso costo OSPF consente di utilizzarli indistintamente senza necessità di usarne uno soltanto (bilanciamento del carico);
  - Permette di supportare in forma unificata instradamento unicast, multicast e broadcast;
  - La tabella di routing viene creata autonomamente da ogni router.

Vediamo adesso l'architettura di un ***router Link State*** (in pratica): ogni volta che il *receive processor* ha un pacchetto in ingresso, questo ne verifica il tipo. Si possono avere queste alternative:

- *pacchetto dati di transito*: il receive processor lo trasferisce al *forwarding processor* che ne determina l'instradamento;
- *pacchetto dati destinato al router*: il pacchetto viene passato ai livelli superiori;
- *pacchetto di Hello*: sono pacchetti per segnalare la presenza di un nodo vicino. Il router ricevente verifica se già conosce il nodo vicino; se così non fosse lo inserisce nella lista dei suoi vicini e notifica i nodi adiacenti con un messaggio LSP.

Algoritmi gerarchici

Poiché le tabelle di routing crescono proporzionalmente alle dimensioni della rete, non si può assumere che qualsiasi algoritmo possa gestire qualunque tipo di rete. Per queste ragioni, in reti complesse, il routing viene organizzato in modo gerarchico: si partiziona la rete in regioni tra loro interconnesse secondo uno schema gerarchico. Ogni rete è considerata autonoma: si parla appunto di **Autonomous System** (**AS**).
Tutti i router appartenenti ad un AS funzioneranno con lo stesso protocollo di routing.

Routing su base etichetta: MPLS

Il **MPLS** fu sviluppato al fine di migliorare la velocità di commutazione dei classici router IP: si usa la cosiddetta "*commutazione su base etichetta* " (analoga a X.25, Frame-Relay , ATM) di lunghezza fissa, ovvero MPLS aggiunge un'etichetta di 32 bit tra l'header IP e l'header di livello 2. I quattro campi di tale etichetta sono:

- *label*: (20 bit) è il valore dell'etichetta, il quale viene deciso autonomamente dai commutatori;
- *Exp*: (3 bit) indica la classe di servizio;
- *S* : (1 bit) se è 1 indica che l'etichetta è l'ultima dello stack, se è 0 ne esistono altre successive;
- *TTL* : (4 bit) si copia il valore del TTL dell'header IPv4 in questo campo, l'ultimo router router del mondo cambierà il valore dell'header IP con quello presente nell'etichetta.

NB: MPLS può instradare qualsiasi protocollo di livello rete: da qui il nome "***multiprotocollo***".
I router che operano secondo MPLS vengono chiamati ***LSR*** e un flusso con una prestabilita etichetta viene definito ***FEC*** : per ogni FEC viene stabilito un percorso all'interno della rete che opera con MPLS.
Si noti che le "*label*" hanno *valore locale* all'interno della rete, ogni LSR deve sapere i valori associati dai vicini e deve conoscere come tali vicini hanno associato le label alle FEC (*label binding*).
MPLS permette di effettuare anche il **traffic engineering**: a differenza di IP che opera su base pacchetto e non prevede cammini multipli, MPLS opera su base flusso e quindi si possono usare percorsi diversi per lo stesso flusso (ogni percorso può essere soggetto a QoS diversa).

Routing Broadcast e Multicast

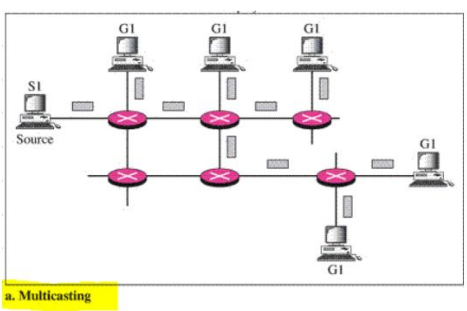
In generale vi sono tre modalità di instradamento di un pacchetto, ovvero:

- ***Unicast***: modalità di instradamento tra coppie isolate sorgente-destinazione;

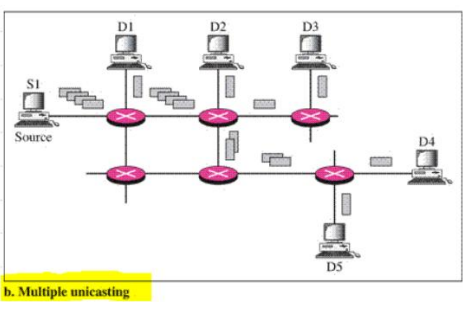
- ***Broadcast***: spedizione pacchetto del tipo "*uno a tutti*": non viene usata in reti estese (congestione).
  ° Un metodo semplice per attuare il broadcast è quello di prevedere tanti collegamenti unicast quanti sono i dispositivi della rete: ovviamente non è una buona soluzione, causa elevato num di collegamenti.
  ° *Multidestinaton routing*: soluzione migliore, in quanto la sorgente invia un solo pacchetto che nel suo campo indirizzo inserisce la lista di tutte le destinazioni; il problema risiede nella conoscenza di tutti gli indirizzi dei nodi della lista.
  ◦ *Flooding*: altra buona tecnica di comunicazione broadcast, ma presenta gli svantaggi già citati.
  ° Tecnica *RPF* (instradamento sul percorso inverso): prevede l'inoltro di un pacchetto broadcast da parte del router su tutte le sue interfacce (tranne quella di ricezione) solo se esso è arrivato seguendo il cammino più breve tra router stesso e nodo sorgente: se ciò non è verificato, il pacchetto è scartato.
  ° *Spanning* tree: il nodo sorgente è la radice e ogni volta che un pacchetto (per il broadcast) arriva ad un nodo, questo viene replicato su tutti i rami di uscita ad esso riferiti; il problema risiede nella conoscenza dell'albero ricoprente da parte di ogni nodo.

- ***Multicast***: spedizione pacchetto del tipo "uno a molti". Vi sono due modalità attuative:
  ¬ ***Multicast base*** (multicast classico): è necessario che il gruppo di destinatari sia identificato in maniera univoca mediante un indirizzo comune. La sorgente invia un solo pacchetto nella rete con l'indirizzo del gruppo multicast. I router raggiunti dal pacchetto (o da una sua copia) provvedono a duplicarlo in tante copie quanti sono gli host componenti il gruppo multicast.
  ¬ ***Unicast multiplo***: non è necessario definire un indirizzo univoco per il gruppo: la sorgente invia nella rete un numero di copie di uno stesso pacchetto quante sono le dest. finali.
  Le criticità di questa tecnica sono: efficienza (nel multicast base su ogni link viene inoltrata una sola copia del pacchetto d'interesse) e ritardo (dovuto a più copie d'uno stesso pacchetto).

  Vi sono due modalità di gestione di un inoltro multicast (base), ovvero:
  ° ***Instradamento multicast con albero condiviso dal gruppo***: si basa sulla costruzione di un albero a costo minimo per connettere tutti i nodi del gruppo multicast. E' un approccio centralizzato: si individua un nodo come coordinatore del gruppo, ed ogni nodo che voglia inviare un flusso informativo al gruppo multicast lo indirizza in modo unicast al coordinatore che si fa carico di inoltrarlo a tutti i membri del gruppo. L'albero si forma su richiesta dei nodi.
  ° ***Instradamento multicast con albero basato sull'origine***: la differenza è che in questo caso l' albero ricoprente non è unico ma viene definito per ogni possibile sorgente verso i nodi che desiderano ricevere l'info (gruppo multicast). L'albero ricoprente viene definito tramite l'RPF. Viene inoltre introdotta la cosiddetta "potatura", attuata quando un router non ha collega-menti verso nessun nodo del gruppo multicast.



a. Multicasting



b. Multiple unicasting

14- CONTROLLO DELA CONGESTIONE

Il controllo della congestione ha come obiettivo quello di prevenire o limitare le perdite eccessive di prestazioni in termini di ritardo di trasferimento dei flussi informativi e nel throughput dei collegamenti stessi.
La congestione si manifesta quando il numero di pacchetti inviati è talmente elevato da saturare la capacità di trasporto dei collegamenti. Vediamo la congestione in una rete a commutazione di pacchetto.

L'effetto della congestione è un progressivo decadimento delle prestazioni della rete: la soluzione tipica adottata è quella di rallentare il tasso di inoltro dei pacchetti fino (se è necessario) ad interromperlo.
Vi sono due tipi di controllo della congestione:

- **Controllo Proattivo**: tali metodi hanno l'obiettivo di prevenire la congestione evitando che si manifesti.
  - *Queue management*: si dà la possibilità ai router di eliminare i pacchetti ritenuti in eccesso;
  - *Traffic Shaping*: controllano l'intensità e la frequenza di invio dei pacchetti in un collegamento.
    In particolare, due tecniche afferenti alla classe dei metodi Traffic Shaping sono:
    - ***Leaky Bucket*** (secchio bucato): garantisce un rate massimo.
      Questo principio utilizza un meccanismo per evitare che un afflusso eccessivo di pacchetti per un collegamento ne provochi la congestione. I pacchetti relativi al collegamento d'interesse non vengono trasmessi subito sono invece inseriti in un buffer (ordine FIFO), per poi essere prelevati ed inviati nel collegamento. Notiamo che la frequenza di inoltro nel collegamento rimane costante e congruente col limite prefissato per evitare la congestione.
    - ***Token Bucket***: garantisce un rate medio.
      Il metodo Leaky Bucket è penalizzante quando l'attività di accesso è intermittente. Di conseguenza si usa la tecnica Token Bucket che permette di conservare le autorizzazioni (token) all' inoltro di pacchetti non utilizzate che arrivano con una freq. fissa e costante. I token non usati vengono conservati in un buffer (*Bucket*): il num max di token che possono essere conservati come credito per accessi futuri è però limitato.
      I pacchetti arrivati a gruppi possono essere inoltrat sequenzialmente secondo il loro ordine.

- **Controllo Reattivo**: tali metodi hanno come obiettivo la risoluzione della congestione mediante determinate azioni non appena questa si verifica ed è stata rilevata. Il metodo più usato è:
  - ***Sliding Window***: controlla l'inoltro di pacchetti in una rete che fornisce un collegamento E2E affidabile (con riscontro della recezione). Ogni pacchetto è etichettato con un numero univoco di sequenza che ne permette l'identificazione senza ambiguità.
    Si definisce un parametro WL detto ampiezza della finestra: rappresenta il numero di pacchetti che possono essere trasmessi in sequenza (senza interruzione) nel collegamento. Tale valore è impostato in modo che prima della conclusione della trasmissione di tutti i pacchetti contenuti nella finestra, pervenga al nodo il messaggio di riscontro dell'avvenuta ricezione almeno del primo pacchetto. Quando questo accade, si scorre la finestra del nodo sorgente in avanti di 1 posizione e si procede alla trasmissione del pacchetto successivo; dopodiché si procede come sopra.
    Se dunque la rete presenta congestione, si attiva il meccanismo di risoluzione: si limita la frequenza di invio di nuovi pacchetti. La congestione viene riconosciuta perché il tempo di riscontro del primo pacchetto della finestra esce dall'ampiezza temporale della finestra stessa. Ciò provoca la chiusura della finestra: non si procede alla trasmiss. di un nuovo pacchetto finché non arriva il riscontro del capofila.
    Infine, il meccanismo a finestra non garantisce all'utente una frequenza minima di inoltro dei pacchetti.

Controllo della congestione in TCP

TCP impone ad ogni nodo che invia pacchetti nella rete un limite alla frequenza di inoltro determinato in relazione al livello di congestione nella rete: il controllo della congestione avviene con la sliding window, che può essere aumentata o diminuita (*self-clocking*) in relazione al traffico presente nella rete. Modalità:

- ***Slow Start***: poiché la sorgente non può conoscere il livello di congestione, inizia a trasmettere lentamente, ovvero aumenta di un'unità l'ampiezza della finestra per ogni pacchetto riscontrato entro il tempo di finestra (time-out); appena tale condizione è violata si dimezza la dimensione della finestra. Dopodiché si inizia di nuovo la fase di slow start. Si noti che la crescita della finestra è esponenziale.
- ***Congestion Avoidance***: uguale al metodo Slow Start, solo che incrementa linearmente la finestra.
- ***Fast Recovery***: consente di distinguere la situazione di una congestione lieve o critica; dunque sceglie se attuare il metodo Slow Start o Congestion Avoidance.

15- SICUREZZA NELLE RETI

La *sicurezza nelle reti* riguarda tutte le procedure che consentono un collegamento per lo scambio di info.
In particolare, si vuol essere certi che il collegamento è effettivamente quello richiesto e che non vi sia alcuna intrusione esterna.

Faremo riferimento a due persone (o dispositivi) detti Alice e Roberto che vogliono scambiarsi info segrete.
I principali requisiti per una comunicazione sicura sono:

- ***privacy*** : solo il mitt. e il dest. possono conoscere il contenuto del messaggio scambiato;
- ***integrità*** : evitare che il messaggio scambiato subisca alterazioni di contenuto;
- ***autenticazione*** : mitt. e dest. devono avere reciprocamente certezza della loro identità;
- ***sicurezza operativa e di apparati*** : applicazioni e apparati di rete con protezioni contro intrusioni.

La **crittografia** è una metodologia per mascherare i messaggi: il messaggio nella sua forma originale è detto *testo in chiaro*. L'operazione che trasforma il messaggio in un *testo cifrato* è detto **algoritmo di cifratura**.
Per attivare la procedura di *cifratura*, si deve conoscere un'informazione segreta detta **chiave**; in recezione (al contrario) si deve attivare la procedura di *decifratura*, la quale è anch'essa fatta sulla base di un'informazione segreta detta (anche in questo caso) *chiave*.
Quest'ultima "chiave" può essere un segreto condiviso (*chiave simmetrica*) oppure no (*chiave pubblica*).

La crittografia si distingue in:

- **Crittografia Simmetrica**: la chiave utilizzata dal mittente e dal destinatario è la stessa, ed è nota ad entrambi (viene detta "*chiave segreta*");
- **Crittografia Asimmetrica**: sono previste *due* diverse tipologie di chiavi per ogni utente:
  - *chiave pubblica*: informazione nota a tutti;
  - *chiave privata*: informazione segreta, nota solo al possessore della chiave stessa.

Crittografia a chiave simmetrica

Un primo esempio pratico di un algoritmo di cifratura si ha con il "*cifrario di Cesare*": si sostituisce ogni lettera dell'alfabeto con un'altra sfasata di k posizioni: il valore "k" costituisce la chiave segreta (o chiave simmetrica).

Crittografia a chiave pubblica (asimmetrica)

Supponiamo che Alice desideri inviare a Roberto un messaggio che deve rimanere segreto agli estranei.
Roberto (in questa tipologia di crittografia) non possiede un'unica chiave segreta (come nel caso dei sistemi a chiave simmetrica) ma due: una *pubblica* (visibile a chiunque ma "personale") e una *privata* (che solo lui conosce). Indichiamo con K_R^+ e K_R^- la chiave pubblica e privata di Roberto. Ovviamente Alice deve conoscere la chiave pubblica di Roberto K_R^+ , e codifica il suo testo in chiaro (detto m) utilizzando suddetta chiave K_R^+ e un algoritmo di cifratura dato, generando così il messaggio cifrato $K_R^+(m)$.
Quando Roberto riceve il messaggio cifrato, utilizza la sua chiave privata K_R^- e un algoritmo per decodificarlo; ovvero calcola $K_R^-\left(K_R^+(m)\right)$, ottenendo così m (testo in chiaro).

Algoritmo RSA: impiegato per definire la coppia di chiavi (pubblica e privata). La procedura è la seguente:

1. Roberto sceglie due numeri primi e molto grandi, detti p e q (più alti sono e più RSA è sicuro);
2. calcolare $n = p \cdot q$ e $z = (p-1)(q-1)$;
3. scegliere un numero $e \neq 1 < n$ primo con z , detto esponente pubblico;
4. trovare un numero d tc. $(ed) \bmod(z) = 1$;
5. la chiave pubblica di Roberto è $K_R^+ = (n\,, e)$, quella privata è $K_R^- = (n\,, d)$.

Il testo di un messaggio viene trasformato in una sequenza di numeri, associando ad ogni lettera il numero corrispondente alla sua posizione nell'alfabeto.
Consideriamo il caso di un numero (lettera) generico $m < n$, dove $n = p \cdot q$. La cifratura di Alice si basa sulla conoscenza della chiave pubblica di Roberto $K_R^+ = (n\,, e)$.
In particolare Alice calcola $c = m^e \bmod(n)$, mentre Roberto decifra calcolando $m = c^d \bmod(n)$.