



INTRODUZIONE ALLE RETI DI CALCOLATORI

a cura di

Giacomo PISCITELLI

Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari

Bari dicembre 2000



INDICE

1	TRASMISSIONE DELL'INFORMAZIONE.	2
1.1	CONCETTO DI INFORMAZIONE.	2
1.2	TRASMISSIONE UNICAST, BROADCAST, MULTICAST.	3
1.3	TRASMISSIONE SIMPLEX, HALF-DUPLEX, FULL-DUPLEX.	3
1.4	TRASMISSIONE ORIENTATA ALLA CONNESSIONE E PRIVA DI CONNESSIONE.	3
1.5	TRASMISSIONE AFFIDABILE E NON-AFFIDABILE.	4
2	CONCETTO DI RETE.	5
2.1	DEFINIZIONI.	5
2.2	CLASSIFICAZIONE DELLE RETI PER SCALA.	7
2.3	TOPOLOGIE DELLE RETI.	10
3	TECNICHE DI MULTIPLAZIONE (MULTIPLEXING).	11
3.1	FREQUENCY DIVISION MULTIPLEXING (FDM).	11
3.2	TIME DIVISION MULTIPLEXING (TDM).	11
4	TECNICHE DI COMMUTAZIONE (SWITCHING).	14
4.1	COMMUTAZIONE DI CIRCUITO.	14
4.2	COMMUTAZIONE DI MESSAGGIO.	14
4.3	COMMUTAZIONE DI PACCHETTO.	15
5	ARCHITETTURE DI RETE.	19
5.1	CONCETTO DI ARCHITETTURA.	19
5.2	NECESSITÀ DI UN MODELLO DI RIFERIMENTO DEI PROTOCOLLI.	22
5.3	LIVELLI OSI.	22
6	APPENDICE 1	34
6.1	IL LIVELLO 1: FISICO	34
6.2	I MEZZI TRASMISSIVI	35

Questa introduzione alle reti di calcolatori è stata ricavata dall'Appendice alla Tesi di Laurea del dott. Ing. Paolo MARRA dal titolo "Reti multiservizio: l'applicazione voce su IP"

1 Trasmissione dell'Informazione.

1.1 Concetto di Informazione.

– *Messaggio.*

Il dialogo tra due qualsiasi entità (uomini o macchine) avviene attraverso lo scambio di *messaggi* servendosi di un linguaggio opportuno, generalmente ben definito nell'aspetto sintattico, semantico e procedurale.

– *Informazione.*

Con il termine *informazione* indichiamo, in questa tesi, il “*contenuto*” di un messaggio ovvero ciò che “dà senso” al dialogo, indipendentemente dalla “forma” assunta dal messaggio: testo, dati, immagini, audio e video sono tutti esempi di “contenitori” di informazione.

– *Sorgenti informative.*

Le sorgenti informative sono i “*generatori di informazione*”, cioè quelle entità che “producono” informazione emettendo dei messaggi. Un telefono, un fax, una telecamera, uno strumento musicale sono tutti esempi di sorgenti informative ⁽¹⁾.

– *Considerazioni.*

Nonostante la pluralità delle forme in cui l'informazione può “materializzarsi”, è possibile, in generale, caratterizzare il modo in cui essa viene trasmessa individuando alcuni aspetti significativi. Tali “elementi distintivi” (descritti nei paragrafi successivi) esulano, oltretutto, dal particolare livello d'astrazione in cui ci si può porre nell'osservare l'intero sistema di comunicazione.

⁽¹⁾ Sono sorgenti informative anche oggetti più complessi come ad esempio un sistema per il telerilevamento o per la diffusione di video musicali o film.

1.2 Trasmissione unicast, broadcast, multicast.

In base al numero di destinatari del messaggio, la trasmissione può essere:

- *unicast*: il messaggio è rivolto ad un solo destinatario;
- *broadcast*: il messaggio è indirizzato a tutti i possibili destinatari;
- *multicast*: il messaggio è indirizzato ad uno specifico gruppo di destinatari.

Una conversazione telefonica o il trasferimento dati da un server ad un calcolatore terminale sono esempi di trasmissione unicast; esempi di trasmissione broadcast o multicast li ritroviamo in applicazioni tipo la diffusione audio/video o la teleconferenza.

1.3 Trasmissione simplex, half-duplex, full-duplex.

Un dialogo può svolgersi in modo:

- *simplex*: unidirezionale;
- *half-duplex*: bidirezionale, alterno;
- *full-duplex*: bidirezionale, contemporaneo.

1.4 Trasmissione orientata alla connessione e priva di connessione.

Un dialogo tra due o più entità può, in alcuni casi, richiedere delle particolari procedure per stabilire e terminare la conversazione; distinguiamo allora fra:

– Trasmissione orientata alla connessione (*connection-oriented*).

La conversazione avviene in 3 fasi: il mittente stabilisce una *connessione* con il ricevente, quindi la utilizza per la trasmissione del messaggio ed infine la rilascia in modo esplicito. Le unità informative vengono sempre ricevute nello *stesso ordine* in cui sono state trasmesse. Ciò è quanto accade per esempio in una conversazione telefonica.

– Trasmissione priva di connessione (*connection-less*).

In questo caso la trasmissione del messaggio avviene senza alcun coordinamento o pianificazione; conseguentemente, ogni messaggio deve contenere “l’indirizzo” completo del destinatario affinché possa essere recapitato in modo corretto. L’ordine con cui i messaggi vengono ricevuti generalmente *non* corrisponde all’ordine con cui sono stati spediti.

Un buon esempio di trasmissione connection-less è la spedizione di una lettera o di un telegramma.

1.5 Trasmissione affidabile e non-affidabile.

La trasmissione può avvenire in modo:

– *Affidabile.*

Le unità informative non vengono mai perse, duplicate o modificate dal sistema di trasmissione; in pratica, il messaggio ricevuto è sempre esattamente corrispondente nel contenuto al messaggio trasmesso. Una trasmissione affidabile è appropriata ad esempio nel trasferimento dei file (archivi o programmi).

– *Non-affidabile.*

Il messaggio può essere perso, duplicato o modificato (corrotto). Ad esempio, nella trasmissione della voce, la comprensibilità del parlato (e quindi il “contenuto” del messaggio vocale) non subisce, entro certi limiti, particolari alterazioni anche in presenza di rumore, distorsioni di ampiezza e fase, eco ed altre non-idealità.

2 Concetto di Rete.

2.1 Definizioni.

– Rete (network).

Una rete è un mezzo per trasferire *informazione* tra due o più dispositivi terminali ad essa collegati; può essere di 2 tipi: punto-punto e broadcast.

– Rete punto-punto (point-to-point network).

È composta da linee (*link*) e nodi (*switch o router*). Il collegamento tra due utenti della rete, richiede la definizione di un percorso (*path o route*), generalmente composto da più linee interconnesse. Ogni linea è ad uso *esclusivo* dei nodi che la terminano (Figura A.1).

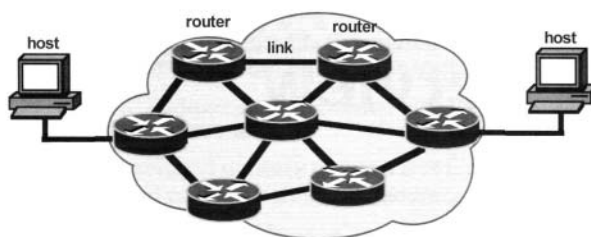


Figura A.1: Esempio di rete *punto-punto*.

– Rete broadcast (broadcast network).

Un *unico* canale di comunicazione (detto ancora linea o *link*) è *condiviso* da tutti gli utenti della rete; è necessario pertanto stabilire un meccanismo di contesa (ad accesso casuale oppure ordinato) per evitare che due o più utenti della rete tentino di trasmettere contemporaneamente, generando una “*collisione*” (Figura A.2).

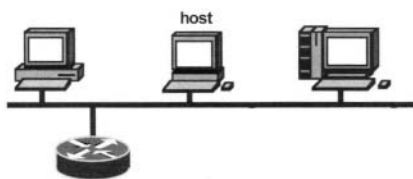


Figura A.2: Esempio di rete *broadcast*.

Poiché la linea broadcast collega tutti gli utenti, *non* occorre stabilire un percorso prima di effettuare una trasmissione: ogni messaggio inviato sulla rete è sempre “ascoltato” da tutti.

Rete punto-punto	Rete broadcast
è composta da linee e nodi; le linee sono ad uso esclusivo dei nodi che le terminano;	unico canale di comunicazione condiviso;
richiede un protocollo di routing.	richiede un protocollo di accesso multiplo.

Tabella A.1: Rete *punto-punto* e Rete *broadcast*.

– *Dispositivo terminale (end-system).*

Detto anche *host*, è un dispositivo connesso alla rete la cui funzione è quella di eseguire un'*applicazione* oppure un *servizio* utile all'utente.

Un end-system può essere un semplice telefono oppure un personal computer, una stazione di videoconferenza, un apparecchio TV, un centro informazioni elettronico o altro ancora.



Figura A.3: Esempi di *end-system*.

– *Nodo (switch o router).*

L'obiettivo della rete di permettere a due o più end-system di scambiarsi informazione, può essere risolto, piuttosto semplicemente, con l'uso di un mezzo *broadcast*, come ad esempio un bus condiviso (*Rete broadcast*). Questa soluzione è ovviamente applicabile se la rete *non* è di grandi dimensioni: quando la distanza tra gli utenti comincia ad essere dell'ordine del chilometro o delle decine di chilometri, è indispensabile affidarsi a dei collegamenti *punto-punto*.

Con N utenti, possiamo pensare ad una *mesh completa*, cioè ad $N(N-1)/2$ linee che collegano direttamente tutti a tutti, ma, se N è grande, è certamente meno dispendioso connettere gli utenti ad una *rete commutata* (N linee), composta da linee e nodi (Figura A.4).

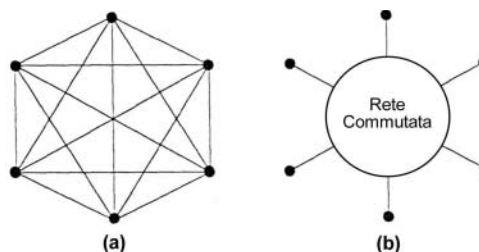


Figura A.4: (a) mesh; (b) rete commutata.

Un *nodo*, quindi, è un *dispositivo di commutazione*. Un nodo è più propriamente detto *switch* quando è chiamato a svolgere, su ogni unità informativa che lo attraversa, solamente un'operazione di commutazione (prestabilita); quando invece il percorso delle unità informative viene determinato mentre la trasmissione è in corso, la denominazione più corretta è quella di *router*.

Switch	Router
Commuta ogni unità informativa sulla prestabilita linea d'uscita.	Decide su quale linea d'uscita inoltrare l'unità informativa, quindi la commuta.

Tabella A.2: Elementi di commutazione.

– *Rete di Calcolatori.*

Una rete di calcolatori è una collezione di calcolatori autonomi collegati (detti *host*).

Due calcolatori sono *collegati* se in grado di scambiarsi informazioni. Richiedendo che i calcolatori siano *autonomi*, intendiamo escludere dalla definizione tutti sistemi gerarchici, nei quali cioè esiste una chiara relazione master/slave. Ad esempio, se un calcolatore può forzatamente accedere, bloccare o controllare un'altra unità, i calcolatori non sono autonomi: in una rete composta da unità autonome, ogni host può *liberamente* attivare un collegamento per effettuare una trasmissione.

– *Sistema Distribuito.*

Un sistema distribuito è un sistema S/w costruito per una rete di calcolatori. Esso rende “*trasparente*” all'utente (cioè non-visibile) l'esistenza di molteplici calcolatori autonomi ma cooperanti. Ciò permette di elaborare e/o archiviare l'informazione in qualsiasi punto essa venga prodotta, ma nello stesso tempo di renderla automaticamente disponibile ovunque nella rete.

2.2 Classificazione delle reti per scala.

E' un importante metro di classificazione perché, come si è visto, su scale diverse sono generalmente utilizzate tecniche diverse. Come illustrato in Figura A.5, è possibile distinguere tra:

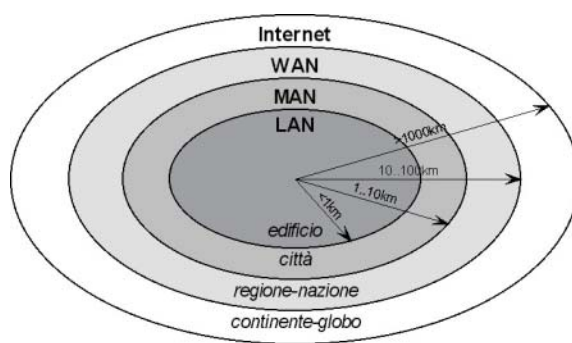


Figura A.5: Classificazione delle reti per scala.

– *Reti Locali o LAN (Local Area Network).*

La distanza tra gli utenti terminali varia tra pochi metri a circa 1 km; gli end-system possono trovarsi ad esempio nella stessa stanza, nello stesso edificio o nello stesso campus universitario. Essendo la dimensione limitata, è noto a priori il tempo di trasmissione nel caso peggiore; in genere adottano tecnologia trasmissiva broadcast. Velocità di trasmissione tipiche sono da 10 a 100 Mbps (megabit al secondo, cioè milioni di bit al secondo), con basso ritardo di propagazione del segnale da un capo all'altro del canale (qualche decina di microsecondi) e basso tasso di errore. Le LAN hanno in genere una topologia ben definita, bus o ring.

Esempi ben noti di LAN per la trasmissione dati sono *Ethernet* e *Token Ring*; anche una rete composta da un PABX (*Private Automatic Branch eXchange*) che connette tra loro dei telefoni costituisce una rete locale, anzi un PABX numerico potrebbe essere impiegato anche per la trasmissione dati a commutazione di circuito.

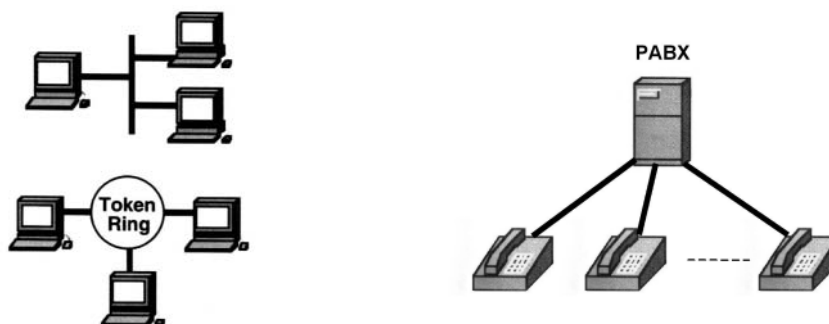


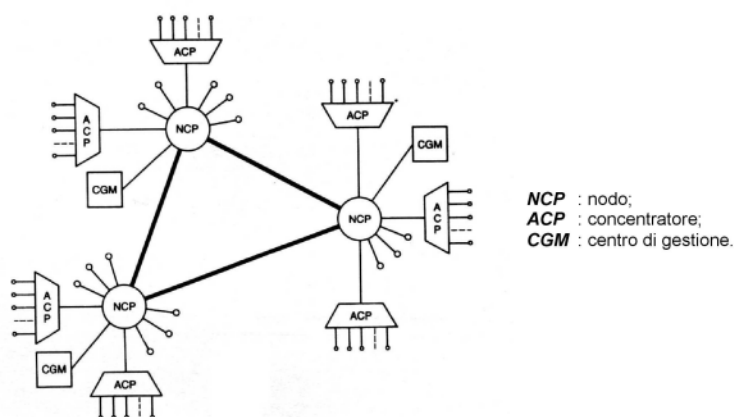
Figura A.6: Esempi di LAN.

– *Reti Metropolitane o MAN (Metropolitan Area Network).*

La distanza tra gli end-system, posti ad esempio in una stessa città o quartiere, è dell'ordine di 10 km. Le reti metropolitane sono generalmente pubbliche. Esistono due standard: IEEE 802.6 o DQDB (Distributed Queue Dual Bus) e FDDI.

– *Reti Geografiche o WAN (Wide Area Network).*

La distanza varia tra i 100÷1000 km, con una rete che copre ad esempio un'intera nazione o continente. Esempi di reti geografiche sono: la rete telefonica pubblica (PSTN), le reti X.25 (vedi Figura A.7) e Frame-Relay, progettate per la trasmissione dati, le reti ISDN e ATM, concepite invece per offrire servizi multipli.



**Figura A.7: Struttura di una rete WAN
(l'esempio si riferisce alla rete X.25 italiana ITAPAC).**

Una WAN è tipicamente costituita di due componenti distinte:

- un insieme di elaboratori (*host* oppure *end system*);
- una *communication subnet* (o *subnet*), che connette gli *end system* fra loro. Il suo compito è trasportare messaggi da un *end system* all'altro.

Di norma la *subnet* consiste, a sua volta, di due componenti.

Le linee di trasmissione (dette anche circuiti o canali) e gli elementi di commutazione (*switching element*) tipicamente chiamati *router*; questi ultimi sono elaboratori specializzati utilizzati per connettere fra loro due o più linee di trasmissione. Quando arrivano dati su una linea, l'elemento di commutazione deve scegliere una linea in uscita sul quale instradarli.

Una tipica WAN è utilizzata per connettere più LAN fra loro mediante coppie di *router*. Ogni *router*, in generale, deve:

- ricevere un pacchetto da una linea in ingresso;

- memorizzarlo per intero in un buffer interno;
 - appena la necessaria linea in uscita è libera, instradare il pacchetto su essa.
- *Internet.*
- E' un'*interconnessione* di due o più *reti*, con distanze coperte anche dell'ordine di 10000 *km*. *Internet* ne è un esempio. Essa è una federazione di reti (una inter-net, per l'appunto), che condividono un insieme di protocolli (o protocol suite), noti come TCP/IP.

2.3 Topologie delle reti.

Varie topologie sono possibili (Figura A.8):

- bus,
- stella,
- anello,
- albero,
- mesh completa,
- mesh irregolare.

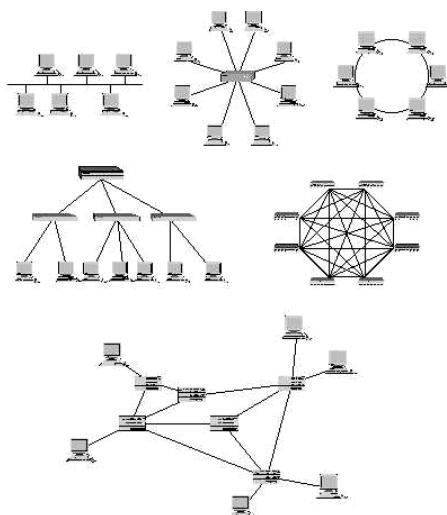


Figura A.8: Topologie delle reti.

Le topologie *regolari* sono tipiche in ambito LAN, le *irregolari* su scala WAN.

3 Tecniche di Multiplazione (Multiplexing).

Nelle reti per la trasmissione dell'informazione il mezzo trasmissivo viene generalmente suddiviso in più *canali* (suddivisione logica) in modo tale da ospitare, senza creare conflitti, flussi provenienti da più sorgenti indipendenti (*multiplexing*).

Le tecniche di multiplazione più frequentemente utilizzate sono FDM e TDM.

3.1 Frequency Division Multiplexing (FDM).

La banda a disposizione è suddivisa in *sottobande* (*canali*) separate da una *banda di guardia* necessaria per la loro separazione mediante dei filtri passa-banda (Figura A.9).

La tecnica *non* ottimizza l'uso della banda a causa delle indispensabili bande di guardia e *non* produce canali con uguali prestazioni (generalmente i canali collocati sugli estremi della banda ricevono un servizio peggiore).

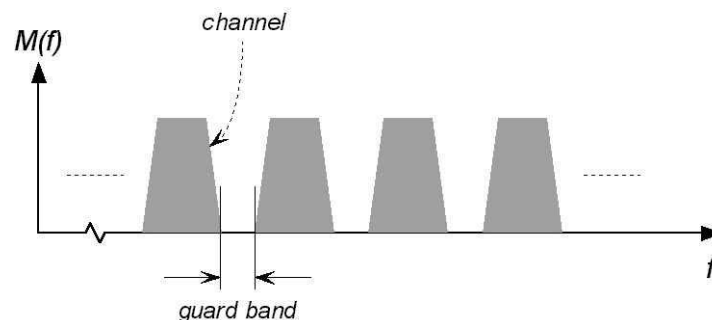


Figura A.9: Multiplazione a divisione di frequenza.

3.2 Time Division Multiplexing (TDM).

L'intera banda viene assegnata, a turno, ai vari canali per un tempo *finito*.

Il tempo può essere suddiviso in intervalli di lunghezza fissa (*time-slot*) o variabile; eventualmente più intervallini possono formare una trama o *frame*.

Distinguiamo gli schemi seguenti (vedi Figura A.10):

<i>unslotted:</i>	nessuna preliminare suddivisione del tempo;
<i>(unframed) slotted:</i>	preliminare suddivisione del tempo in intervalli di lunghezza fissa (slot);
<i>framed and slotted:</i>	gli slot sono raggruppati in frame; slot omologhi in frame distinti sono associati in modo logico tra loro (appartengono cioè allo stesso canale);
<i>framed hybrid slotted-unslotted:</i>	abbiamo una suddivisione del tempo in frame ed ogni frame contiene una sezione slotted ed una unslotted.

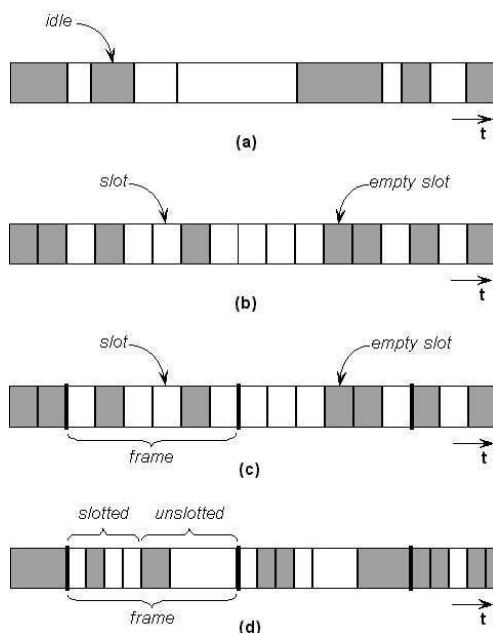


Figura A.10:
(a) TDM unslotted,
(b) slotted,
(c) framed and slotted,
(d) framed hybrid slotted-unslotted.

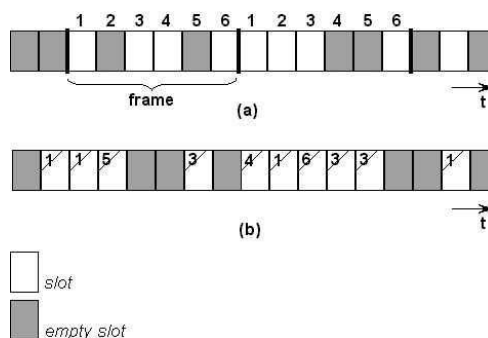


Figura A.11:
(a) S-TDM nel caso framed-and-slotted,
(b) A-TDM nel caso slotted.

Inoltre, ogni intervallino temporale può essere associato ad una particolare sorgente in modo statico o sincrono (*associazione implicita*) oppure dinamico o asincrono (*associazione esplicita*). Precisamente (vedi Figura A.11):

- *TDM sincrono o S-TDM (solo nei casi framed-and-slotted e hybrid).*

l'associazione è *statica* ed *implicita*. E' applicabile solo nel caso framed and slotted o ibrido nella sezione slotted. L'associazione è stabilita dalla *posizione* dello slot nel frame. Tende a sprecare banda se la sorgente non emette in modo *continuativo* (sorgente bursty).

- *TDM asincrono o A-TDM (sempre applicabile).*

l'associazione è *dinamica* ed *esplicita*. E' sempre applicabile ed è stabilita da un'*etichetta* memorizzata nel blocco di dati (intestazione). Consentendo un utilizzo della linea su domanda, tende a massimizzarne l'utilizzo.

Dalle precedenti classificazioni possiamo derivare i modelli di multiplatori TDM elencati nella Tabella A.3.

	unslotted	slotted	framed and slotted	framed hybrid
<i>TDM sincrono</i>	-	-	TDM	TDM Ibridi (*)
<i>TDM asincrono</i>	TDM statistici (STDM)	Cell Relay	-	TDM Ibridi (*)

(*) il funzionamento è sincrono nella sezione slotted, asincrono in quella unslotted.

Tabella A.3: Multiplatori TDM.

4 Tecniche di Commutazione (Switching).

Le tecniche di commutazione sono di 3 tipi: di *circuito*, di *messaggio* e di *pacchetto*; in particolare, nella commutazione di pacchetto è possibile distinguere tra 2 differenti approcci, a *datagramma* e a *circuito virtuale*.

4.1 Commutazione di circuito.

In una rete a commutazione di circuito ad ogni connessione (effettuata su domanda o in modo semi-permanente) vengono assegnate *stabilmente* delle risorse di rete (un canale, rappresentato ad esempio da uno slot TDM), in modo tale da creare un *circuito* che connette i due utenti terminali (Figura A.12).

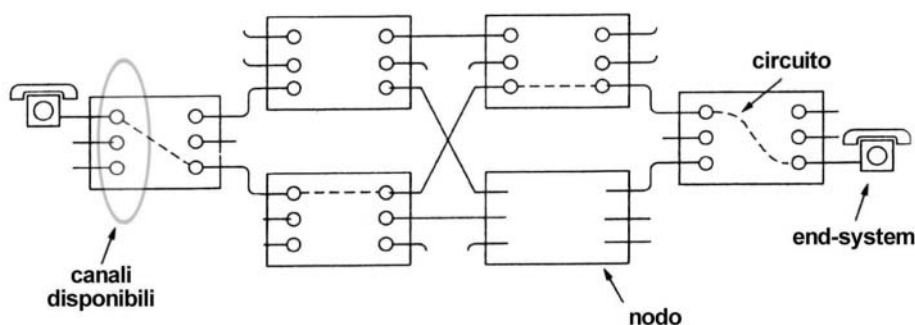


Figura A.12: Commutazione di Circuito.

Progettata per la trasmissione della *voce*, questa tecnica ha subito negli anni un'evoluzione segnata principalmente dall'impiego di tecniche numeriche anziché analogiche per la trasmissione e la commutazione stessa. Ciò ha permesso il suo utilizzo per uno scopo completamente diverso: la trasmissione dei *dati*.

4.2 Commutazione di messaggio.

La trasmissione *non* richiede una preliminare fase di set up: non appena un utente ha un messaggio da spedire lo invia al nodo più vicino. Presso il nodo il messaggio viene processato con una tecnica detta "*store-and-forward*": viene prima completamente memorizzato (*store*), poi ispezionato per individuare la sua destinazione ed infine inoltrato (*forward*) al nodo successivo non appena la linea d'uscita necessaria si rende disponibile.

In Figura A.13 sono illustrate le tecniche della commutazione di circuito e della commutazione di messaggio a confronto, supponendo che siano interessati 3 nodi della rete (A, B e C). Si può osservare che, nella commutazione di messaggio, la trasmissione inizia immediatamente, senza alcuna predisposizione dell'interconnessione A→C; ciò comporta una riduzione del tempo complessivo d'impegno del collegamento nel caso di messaggi *brevi* ⁽²⁾.

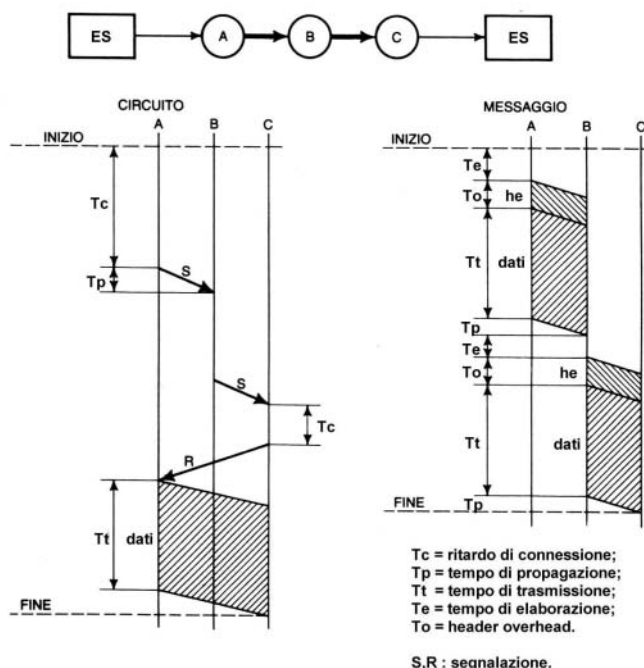


Figura A.13: Commutazione di *Circuito* vs Commutazione di *Messaggio*.

Osserviamo che, poiché non c'è limite alla *dimensione* del messaggio, una moderna implementazione di questa tecnica di commutazione richiederebbe delle memorie di massa presso ogni nodo della rete; inoltre, un grosso messaggio potrebbe impegnare una linea tra due nodi anche per minuti, bloccando il traffico alle sue spalle. Per questi motivi, nelle moderne reti di telecomunicazione, la commutazione di messaggio *non* è vantaggiosa: la tecnica store-and-forward è efficiente solo se i messaggi vengono frammentati in piccole unità informative.

4.3 Commutazione di pacchetto.

E' un'evoluzione della commutazione di messaggio, essendo anch'essa basata sulla tecnica store-and-forward; la differenza sostanziale è nella dimensione delle unità informative che

⁽²⁾ Grazie a questa proprietà, le reti a commutazione di messaggio hanno trovato una tipica applicazione nel *servizio*

circolano nella rete: un messaggio viene infatti suddiviso (prima della trasmissione) in piccole unità informative (ad esempio 1500 byte) dette pacchetti (*packets*).

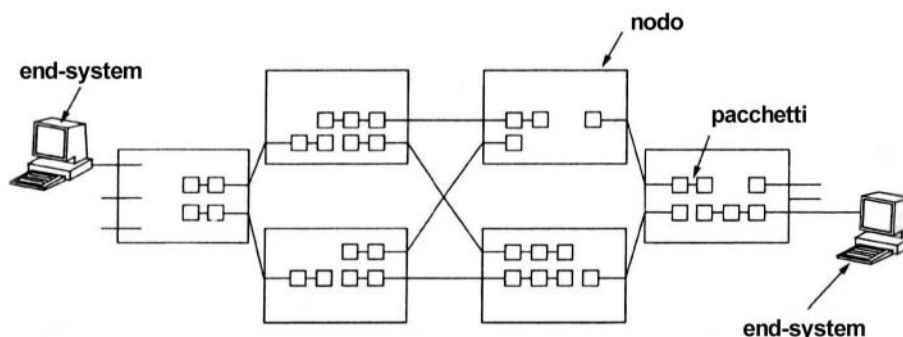


Figura A.14: Commutazione di Pacchetto.

Ogni pacchetto contiene un'intestazione (*header*), che specifica l'indirizzo del destinatario più informazioni di controllo, seguita dai dati utili (carico o *payload*).

Il funzionamento è simile a quello della commutazione di messaggio: appena un end-system è pronto per la trasmissione, invia il primo pacchetto al nodo più vicino; quest'ultimo memorizza il pacchetto temporaneamente (*store*), decide su quale linea d'uscita inoltrarlo ed infine lo incoda su quella linea. Non appena la linea è libera il pacchetto viene trasmesso (*forward*).

La suddivisione in pacchetti permette di ridurre il tempo di transito del messaggio attraverso la rete. Ciò è messo in evidenza nella Figura A.15 in cui si è supposto di trasmettere, attraverso n nodi, un messaggio divisibile in p pacchetti ⁽³⁾.

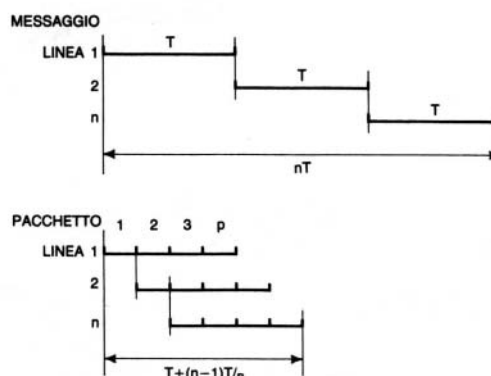


Figura A.15: Commutazione di Messaggio vs Commutazione di Pacchetto.

telegrammi.

⁽³⁾ Per semplicità si sono considerati trascurabili i tempi di propagazione, di elaborazione e di attesa nelle code.

Abbiamo due sottocasi (vedi Figura A.16 e Figura A.17):

– *Approccio a Datagramma.*

Ogni pacchetto, detto datagramma (*datagram*), è instradato indipendentemente dagli altri pacchetti che compongono lo stesso messaggio. In generale quindi, anche se diretti alla stessa destinazione, pacchetti diversi seguiranno *percorsi diversi*, giungendo a destinazione *fuori ordine*. Ogni nodo decide il percorso ottimale per ogni datagram che lo attraversa, basandosi sulla propria conoscenza dello stato (mutevole) della rete (*routing per pacchetto*).

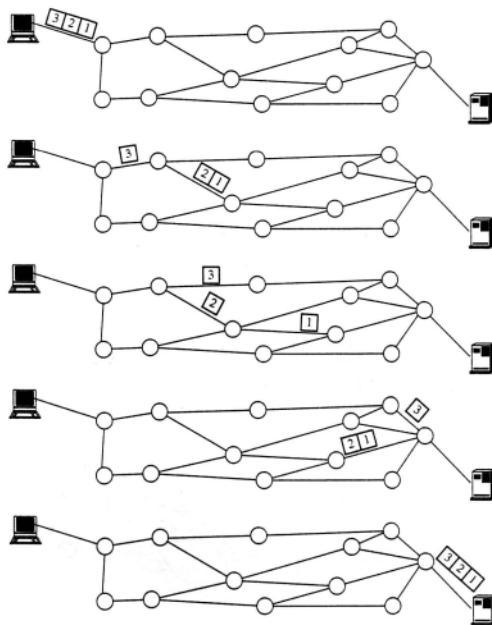


Figura A.16: Approccio a Datagramma.

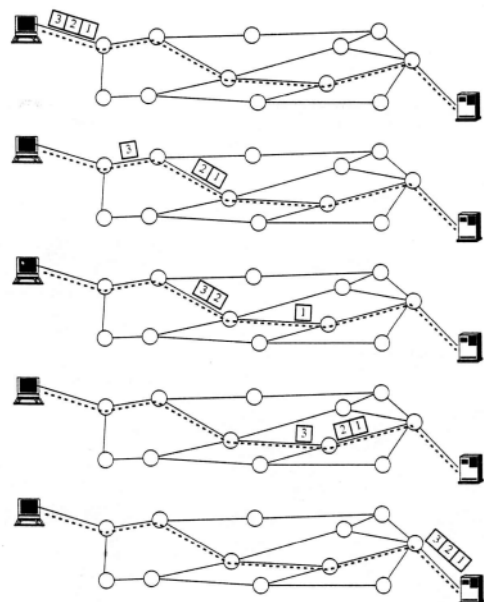


Figura A.17: Approccio a Circuito Virtuale.

– *Approccio a Circuito Virtuale.*

Tutti i pacchetti di uno stesso messaggio vengono instradati allo stesso modo; il percorso, detto circuito virtuale (*virtual circuit*) in analogia con i circuiti della commutazione di circuito, viene deciso, una volta per tutte, *prima* che i pacchetti vengano spediti, utilizzando dei pacchetti di segnalazione. Con questa tecnica, ogni nodo non dovrà effettuare alcuna operazione di routing, eccetto nella fase di set up del circuito virtuale (*routing per sessione*); inoltre, l'intestazione dei pacchetti che trasportano carico utile, anziché contenere un indirizzo completo, dovrà specificare solamente il *VC identifier*, necessario presso i nodi per commutare i pacchetti sulla *predefinita* linea d'uscita.

– *Datagram vs Virtual Circuit.*

Le differenze tra datagram e circuiti virtuali sono riassunte nella Tabella A.4.

	Datagram	Virtual Circuit
<i>fase di set up</i>	no	sì
<i>ogni pacchetto segue lo stesso percorso</i>	no	sì
<i>i pacchetti vengono ricevuti in sequenza</i>	no	sì
<i>indirizzamento</i>	completo	identificatore
<i>tempo di transito</i>	elevato	basso
<i>effetto di un guasto</i>	pochi pacchetti persi	tutti i VC coinvolti vengono terminati
<i>controllo delle congestioni</i>	possibile ma complesso	semplice se è possibile prenotare le risorse

Tabella A.4: Datagram vs Virtual Circuit.

▪ Caratteristiche dei *Datagram*:

1. *non* occorre una fase preliminare di *set up*;
2. la rete può reagire efficacemente nel caso di nodi o linee malfunzionanti o congestionate, reindirizzando i pacchetti altrove (elevata *affidabilità*);
3. occorre *riordinare* i pacchetti presso il destinatario.

▪ Caratteristiche dei *Circuiti Virtuali*:

1. i pacchetti vengono ricevuti *in sequenza*;
2. la trasmissione è molto *rapida*: ogni nodo non deve ricalcolare il percorso per ogni pacchetto ma solamente commutarlo;
3. se un nodo o una linea cessa di funzionare, tutti i VC che usano quel nodo o quella linea vengono necessariamente *terminati*.

5 Architetture di Rete.

5.1 Concetto di Architettura.

– *Il problema della comunicazione tra due end-system.*

Per ridurre la *complessità* della progettazione è opportuno suddividere il problema della trasmissione tra due o più utenti terminali in un insieme di sottoproblemi ragionevolmente autonomi e, come tali, più semplici da gestire. Ogni sottoproblema è risolto da un *modulo* (S/w o H/w); i moduli sono organizzati in una struttura *stratificata* per individuare diversi livelli d'astrazione, cioè vari modi di “vedere” la rete.

– *Strati o livelli (layers).*

Ogni *layer* ha il compito di offrire precisi *servizi* al livello superiore, nascondendo i dettagli della loro implementazione: il livello n è quindi un “fornitore di servizi” per il livello $n+1$ ed un “utilizzatore di servizi” per il livello $n-1$.

Un livello soddisfa i seguenti requisiti:

1. svolge una ben determinata *funzione*;
2. è creato per realizzare un nuovo livello di *astrazione*;
3. i *limiti* dei livelli sono scelti in modo tale da minimizzare il flusso informativo alle interfacce;
4. l'implementazione di un livello può essere *sostituita* senza innescare modifiche nei livelli superiori o inferiori;
5. il *numero* di livelli è abbastanza grande da permettere a funzioni distinte di non essere forzatamente inserite nello stesso livello e, contemporaneamente, abbastanza piccolo da non appesantire l'intera architettura.

– *Interfacce.*

Due livelli adiacenti sono separati da un'*interfaccia*: essa definisce quali servizi il livello n rende disponibili al livello $n+1$; ogni servizio ha un indirizzo detto *SAP* (*Service Access Point*) e viene attivato usando dei comandi detti *primitive*.

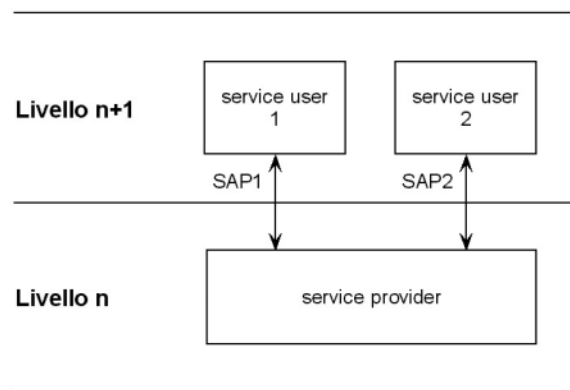


Figura A.18: Interfaccia tra due livelli.

La comunicazione tra due end-system *A* e *B* avviene facendo comunicare i layers corrispondenti sui due sistemi (chiamati entità pari o “*peers*”), anche se, come si vedrà, tale dialogo è generalmente indiretto.

– *Protocolli.*

La comunicazione tra entità pari del livello *n* avviene per mezzo dello scambio di blocchi di dati (detti *Protocol Data Unit* o *PDU*), formattati secondo delle convenzioni dette “*protocollo del livello n*”. Una PDU si compone di un’intestazione o *header*, eventualmente di una coda o *trailer*, contenenti le informazioni di controllo scambiate dalle entità pari, e di un campo dati (carico o *payload*) contenente la PDU del livello *n+1*.

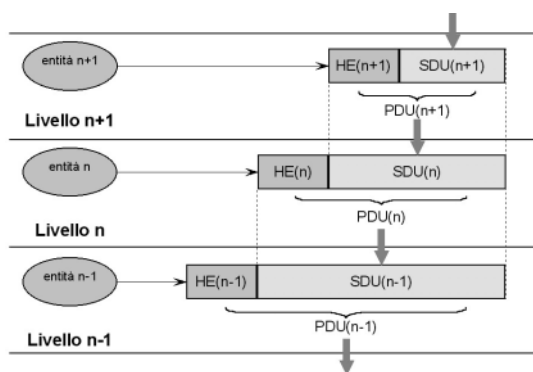


Figura A.19: Trasmissione.

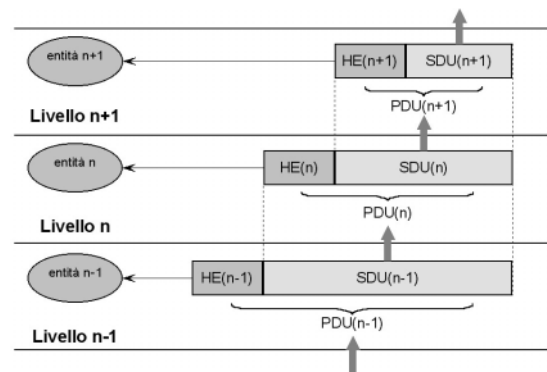


Figura A.20: Ricezione.

Quest’ultima, nel contesto del livello *n*, è detta più propriamente *SDU* (*Service Data Unit*) del livello *n*.

Le convenzioni del protocollo riguardano i seguenti aspetti:

1. *sintattico* (la struttura dei dati);
2. *semantico* (il significato dei dati);
3. *procedurale* (la sequenza di eventi).

– *Architettura di Rete.*

Un'architettura di rete è definita specificando i *livelli*, le *interfacce* ed i *protocolli*.

– *Comunicazione tra pari.*

Consideriamo un'applicazione presente sull'end-system *A* che desidera comunicare con un'applicazione presente sull'end-system *B*; supponiamo che tra i due end-system esista una connessione fisica diretta (Figura A.21).

L'applicazione invia l'informazione da trasmettere al livello *N* dell'architettura: poiché il livello *N* non possiede una connessione con il suo pari, aggiunge un header contenente ciò che ha da dirgli e passa tutta la *PDU(N)* al livello *N-1*. La discesa dei layer prosegue, con i dati che crescono complessivamente di dimensione, finché non viene raggiunto il *livello fisico* (layer *1*), in corrispondenza del quale la comunicazione può realmente aver luogo (Figura A.19).

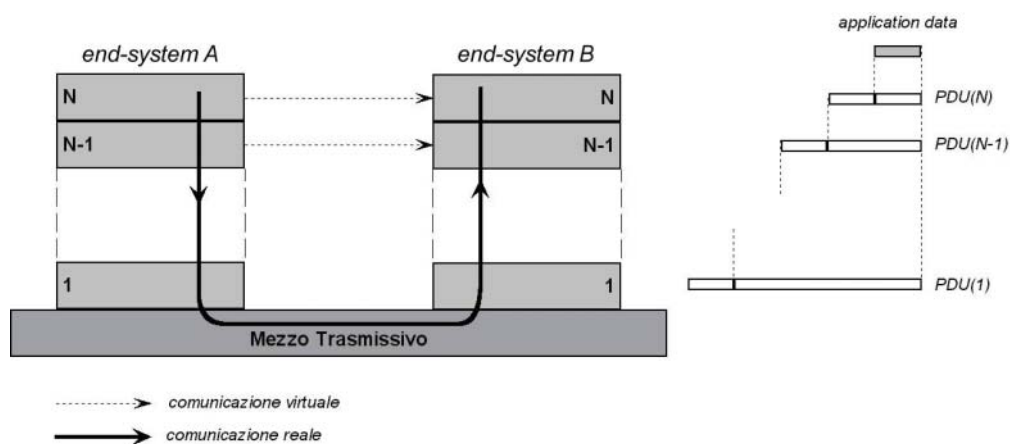


Figura A.21: Comunicazione tra entità pari.

Ricevuta la *PDU(1)*, il livello *1* dell'end-system *B* estrae il suo header ed invia l'*SDU(1)* al livello 2. Quest'ultimo preleva il suo header, contenente il messaggio proveniente dal suo pari, ed inoltra il carico al livello superiore e così via (Figura A.20). I dati risalgono così i vari

strati ed infine vengono consegnati all'applicazione destinataria; nel contempo ogni livello ha potuto dialogare con il suo pari.

In conclusione, i processi di un dato livello n , ad eccezione di quelli appartenenti al livello fisico (i quali dialogano direttamente attraverso il mezzo trasmissivo), comunicano utilizzando un servizio del livello $n-1$: ogni livello dell'architettura utilizza le prestazioni del livello inferiore per fornire al livello superiore un servizio più "pregiato" ⁽⁴⁾.

5.2 Necessità di un modello di riferimento dei protocolli.

Il problema del trasporto dell'informazione è alquanto complesso: nel 1980 la *International Organization for Standardization* (ISO) comprese la necessità di definire un modello generale di rete che favorisse i costruttori (*vendors*) nella realizzazione di dispositivi di rete compatibili. Nacque così il modello OSI (*Open System Interconnection*) che rapidamente divenne un modello di riferimento universalmente riconosciuto.

Oggi esistono numerosi modelli architetturali, spesso proprietari, ma tutti possono essere ricondotti in modo più o meno diretto al modello OSI: tale confronto è spesso utilmente proposto proprio per favorire una rapida comprensione delle nuove (e vecchie) architetture.

5.3 Livelli OSI.

– Caratteristiche generali.

Il modello OSI è composto da 7 livelli, ciascuno chiamato a svolgere un insieme ben definito di funzioni ⁽⁵⁾.

Per meglio comprendere la struttura del modello OSI, cerchiamo di individuare alcuni elementi essenziali per il funzionamento di una rete di telecomunicazioni.

Certamente è essenziale l'implementazione di un *servizio di trasporto* che, indipendentemente dal mezzo trasmissivo utilizzato, possa trasferire informazione tra due o più utenti della rete, possibilmente senza errori e con le necessarie prestazioni.

⁽⁴⁾ Si dice, pertanto, che un'architettura a strati funziona secondo il principio del "valore aggiunto": la sua descrizione definisce, per ogni livello n , i servizi resi disponibili dal livello inferiore $n-1$ e quelli offerti al livello superiore $n+1$, vale a dire il "valore aggiunto" dal livello n .

⁽⁵⁾ Il modello di riferimento OSI definisce solamente le *funzioni* di ciascun livello: poiché non vi è alcun accenno né alle interfacce, né ai protocolli, il modello OSI *non* è un'architettura.

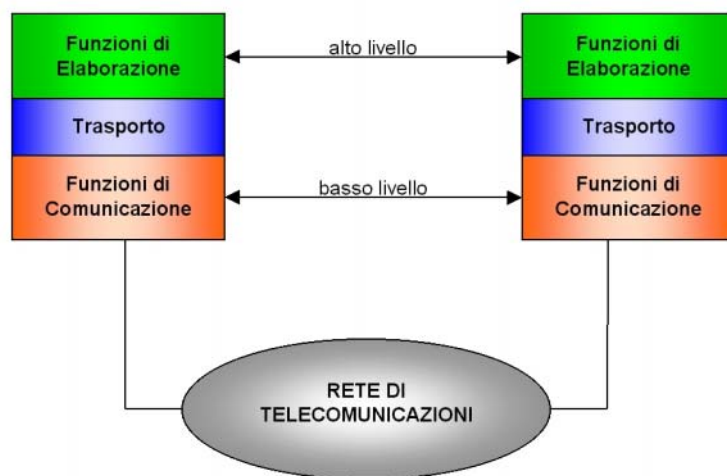


Figura A.22: Funzioni OSI.

Tale servizio è collocato, in OSI, al *livello 4*. La sua indipendenza dalle tecnologie impiegate nella rete è ottenuta mediante opportuni *protocolli di basso livello*, mentre la cooperazione tra i processi applicativi, poiché oltrepassa il campo delle comunicazioni per investire quello dell'elaborazione, richiede un insieme di *protocolli di alto livello* (vedi Figura A.22).

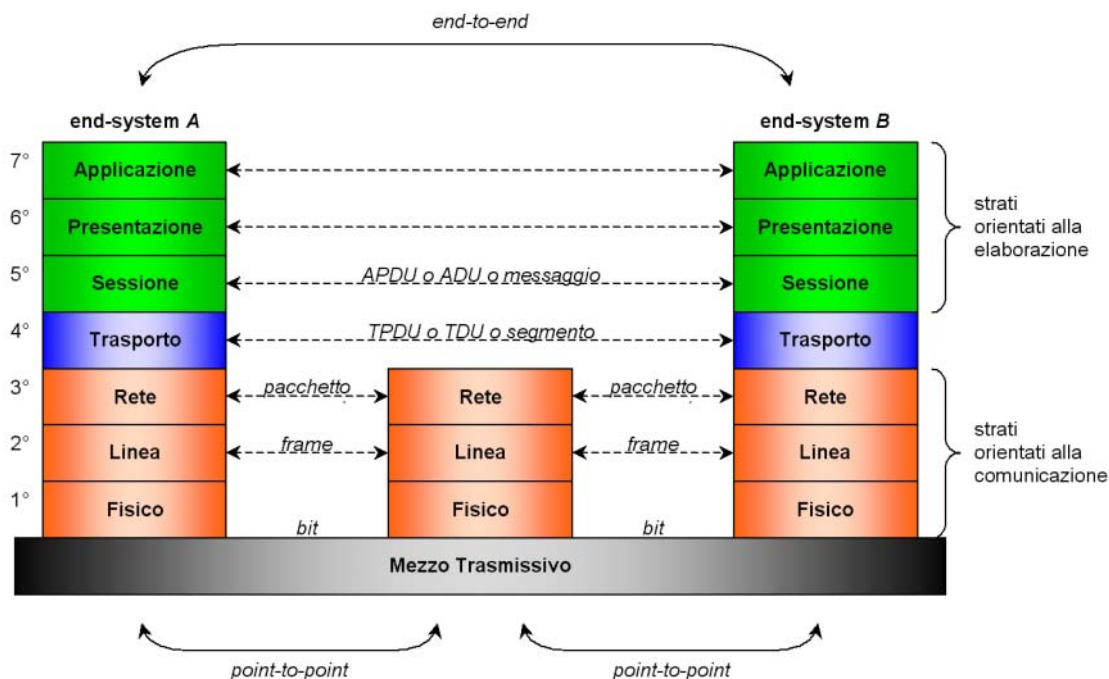


Figura A.23: Livelli OSI.

Per chiarire la differenza tra protocolli di basso ed alto livello, un esempio molto semplice è quello del *telefono*, dove nei primi possono essere comprese tutte le operazioni necessarie per

l'attivazione del collegamento tra i due interlocutori, e nei secondi le procedure per effettuare la conversazione vera e propria (riconoscimento reciproco, argomento noto ad entrambi, impiego della stessa lingua).

In OSI (Figura A.23), i protocolli di basso livello comprendono i primi 3 strati: il livello *fisico*, il livello di *linea* ed il livello di *rete*, con funzioni orientate alla comunicazione; mentre i protocolli di alto livello includono gli ultimi 3 strati (*sessione*, *presentazione* ed *applicazione*), con funzioni orientate verso l'elaborazione; al centro, si colloca il livello di *trasporto*.

5.3.1 L1: livello fisico (physical).

Definisce le specifiche elettriche, meccaniche e procedurali necessarie all'attivazione, gestione e disattivazione del *mezzo trasmissivo* posto sotto il livello fisico. Queste specifiche riguardano connettori, trasmettitori, ricevitori, segnali, forme d'onda, livelli di tensione, temporizzazioni, bit-rate, distanze, ecc.

Al livello fisico l'informazione si presenta sotto forma di *flusso di bit*: non è possibile perciò riconoscere il significato dei bit né vedere la loro suddivisione in blocchi di dati.

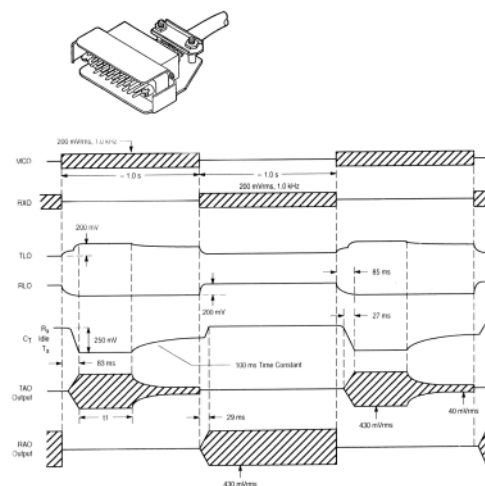


Figura A.24: Il livello fisico.

Particolare importanza riveste invece il problema della corretta *individuazione del livello logico* dei bit ricevuti; è fondamentale “sondare” ciascun bit in un istante scelto in modo tale da cadere con la massima probabilità nella zona centrale del “tempo di cifra”. Ciò può essere attuato in *due* modi:

– *Trasmissione Asincrona.*

La linea è normalmente in uno stato di riposo (*idle*): prima di inviare il 1° bit, il trasmettitore deve portare la linea in condizioni di lavoro (impulso di *start*). Il ricevitore, quando rileva tale cambiamento, attiva un clock interno, isofrequenziale con quello di trasmissione. Conclusa la trasmissione dell'ultimo bit, la linea viene riportata nello stato di attesa (impulso di *stop*). Pertanto, è sufficiente che la sincronizzazione del clock, presso il destinatario, venga

mantenuta *solamente* per un intervallo di tempo *limitato*, quello compreso tra gli impulsi di start e stop.

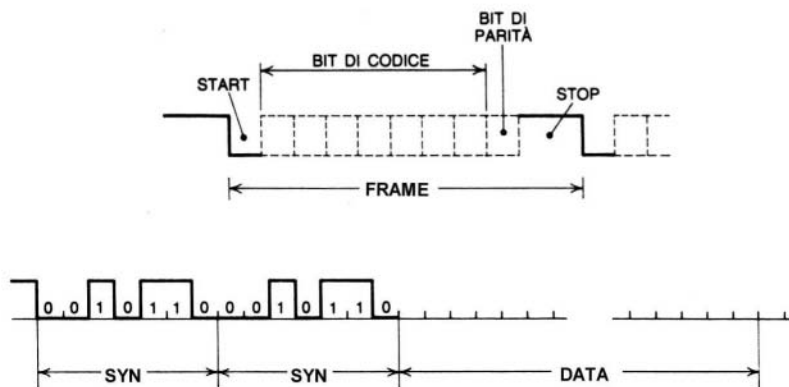


Figura A.25: Trasmissione Asincrona e Sincrona.

– *Trasmissione Sincrona.*

Non esistono impulsi di sincronizzazione: i bit sono trasmessi in modo continuo, senza alcuna interruzione. Presso il ricevitore, la temporizzazione necessaria viene estratta dallo stesso segnale ricevuto, grazie ad un circuito ausiliario.

Se il mezzo trasmissivo ha una banda utile incompatibile con quella del segnale che si desidera trasmettere, è necessario impiegare dei ricetrasmittitori opportuni (*tranceiver*) che facciano uso di forme d'onda con migliori caratteristiche spettrali (favorendo, possibilmente, l'estrazione della temporizzazione) oppure, se è necessario “spostare” nettamente la collocazione spettrale del segnale, dei veri e propri modulatori/demodulatori (*modem*).

Maggiori dettagli sul livello fisico sono riportati in Appendice 1

5.3.2 L2: livello di linea (data-link).

Il livello fisico accetta un flusso di bit ad un estremo della linea e lo consegna all'altro estremo senza garanzie: il numero di bit consegnati può essere maggiore, minore o uguale del numero di bit trasmessi ed alcuni bit possono assumere valori errati. Il livello di linea ha il compito di presentare al livello di rete un servizio di comunicazione point-to-point che trasmette frame, anziché flussi non specificati di bit, ed in grado di rilevare (e qualche volta anche correggere) eventuali errori di trasmissione (Figura A.26).

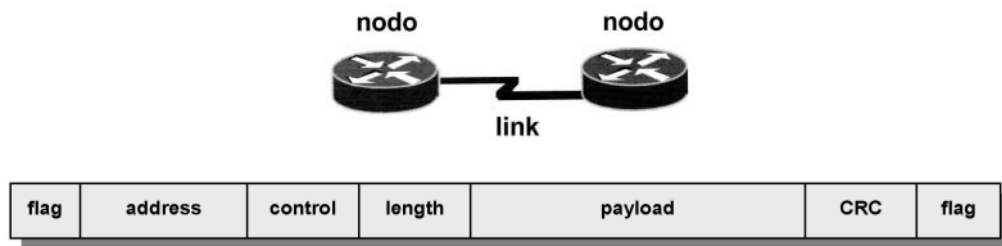


Figura A.26: Visibilità a livello di linea.

Un tipico protocollo di linea è HDLC (High-level Data Link Control) dal quale sono stati derivati molti protocolli “specializzati” come, ad esempio, LAP (Link Access Procedure), LAPB (Link Access Procedure, Balanced), IEEE 802.2, Ethernet (802.3) nelle sue varie realizzazioni, Token Ring (802.5) e Token Bus (802.4), PPP (Point-to-Point Protocol).

Distinguiamo le funzioni del livello di linea in base al tipo di rete:

– *Rete punto-punto.*

Le funzioni fondamentali sono le seguenti:

- Impacchettamento dei bit (*framing*).

I limiti dei frame vengono riconosciuti o creati tramite delle particolari sequenze di bit (*flag byte*) inserite in testa ed in coda al frame. Se la stessa sequenza occasionalmente compare nei dati viene modificata in modo noto con un procedimento detto *bit stuffing*. Eventualmente, può essere aggiunto un campo che contiene la dimensione del frame (ad esempio in byte) in modo da eseguire un controllo incrociato.

- Rilevazione degli errori (*error checking*).

Utilizzando un campo di *checksum* il destinatario può verificare se il frame è esente da errori; normalmente la checksum contiene un codice a ridondanza ciclica (*Cyclic Redundancy Check* o *CRC*). I provvedimenti presi in caso di errore dipendono dal tipo di architettura: nelle architetture tradizionali viene chiesta la *ritrasmissione* del frame (*Automatic Repeat Request* o *ARQ*), in quelle *fast-packet* un frame errato viene semplicemente *scartato*, rimandando al livello di trasporto (end-to-end) la gestione delle ritrasmissioni (se è necessario un servizio affidabile).

- Controllo del flusso (*flow control*).

Consente all’entità ricevente di regolare il flusso di frame trasmesso dalla entità sorgente. Al livello 2 questo controllo viene esercitato su ogni *linea*. La regolazione evita che una

sorgente troppo veloce inondi di frame un ricevitore lento, causando l'*overflow* dei buffer e la conseguente perdita dei frame in eccesso. Spesso il controllo del flusso è integrato nel meccanismo di controllo degli errori (se previsto).

– **Rete broadcast.**

Il mezzo trasmissivo è condiviso da tutti gli utenti della rete: per disciplinare gli accessi, è necessario implementare un meccanismo di contesa (ad accesso ordinato o casuale).

Il livello data-link è in tal caso suddiviso in 2 sottolivelli (vedi Figura A.27):

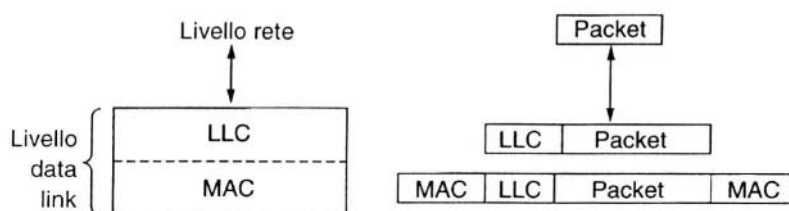


Figura A.27: Sottolivelli data-link.

- LLC (Logical Link Control).
Svolge le funzioni a, b, c come nelle reti punto-punto.
- MAC (Medium Access Control).
Implementa il protocollo di accesso multiplo (Aloha, CSMA/CD, ecc.).

5.3.3 L3: livello di rete (network).

E' un livello complesso che si occupa di trasmettere *pacchetti* dalla sorgente alla destinazione (cioè tra end-points): questa funzione è chiaramente diversa da quella del livello data-link che ha il compito ben più modesto di trasportare *frame* da un estremo all'altro di una *linea* (vedi Figura A.28). Generalmente l'interfaccia rete/trasporto rappresenta il *confine della rete*: l'utente ha cioè il controllo sui protocolli e sulle interfacce da utilizzare dal livello di trasporto in sù.

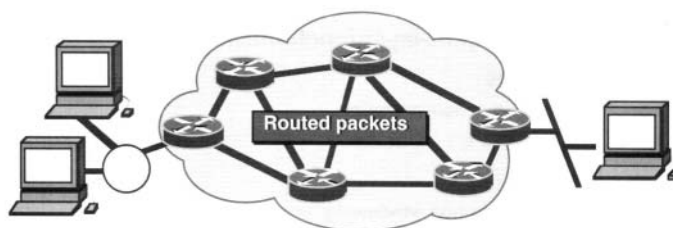


Figura A.28: Visibilità a livello di rete.

Distinguiamo nuovamente tra:

– *Rete punto-punto.*

Le funzioni di livello 3 sono:

- Instradamento (*routing*).

Per permettere a due o più utenti di comunicare (unicast, multicast o broadcast), il livello di rete deve conoscere la *topologia* della rete (in modo totale o parziale) e scegliere percorsi appropriati per i pacchetti implementando un opportuno *algoritmo di routing*. E' al livello di rete che si pone l'importante distinzione tra:

⇒ Approccio a Datagram:
i pacchetti di uno stesso messaggio seguono percorsi generalmente diversi (routing per pacchetto);

⇒ Approccio a Circuito Virtuale:
tutti i pacchetti di uno stesso messaggio seguono lo stesso pre-definito percorso (routing per sessione).

Il percorso ottimale tra due end-points dipende da moltissimi fattori, ad esempio:

- numero di linee e nodi coinvolti,
- lunghezza delle linee,
- bilanciamento del carico,
- presenza di congestioni,
- costo delle linee,
- ecc.

Nei casi di comunicazione *multicast* o *broadcast* il livello di rete deve evitare, fin quando è possibile, la duplicazione dei pacchetti provenienti dalla stessa sorgente ma diretti a più destinazioni, cercando di “clonare” i pacchetti solo in corrispondenza delle diramazioni del percorso (Figura A.29).

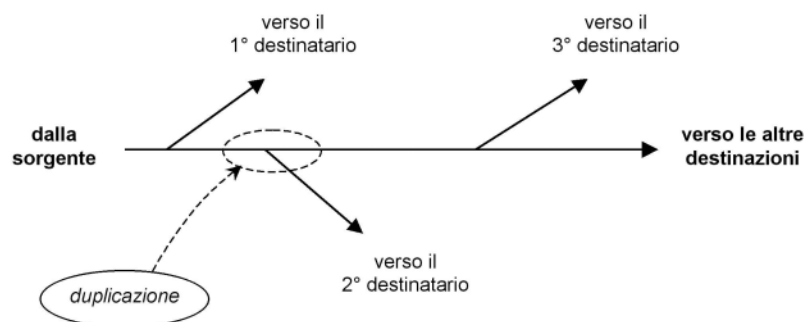


Figura A.29: Esempio di distribuzione *multicast*.

- Controllo delle congestioni (*congestion control*).

Quando nella rete (o in una sua parte) sono presenti troppi pacchetti, le prestazioni degradano vistosamente: questo fenomeno è detto *congestione*. In presenza di congestioni le code nei nodi si allungano e cominciano a perdere pacchetti, allungando i tempi di trasporto. Le sorgenti tendono ad alimentare la congestioni una volta innescate: i pacchetti che non giungono a destinazione entro un certo tempo vengono generalmente ritrasmessi, anche più di una volta.

Il livello di rete ha il compito di evitare concentrazioni di traffico distribuendo il carico sulla rete in modo uniforme (*load-balancing*), di attivare in caso di overload eventuali linee di riserva (*backup*) ed infine, qualora le precedenti misure non siano sufficienti, ridurre i flussi d'ingresso ai confini della rete e/o negare nuove richieste di servizio.

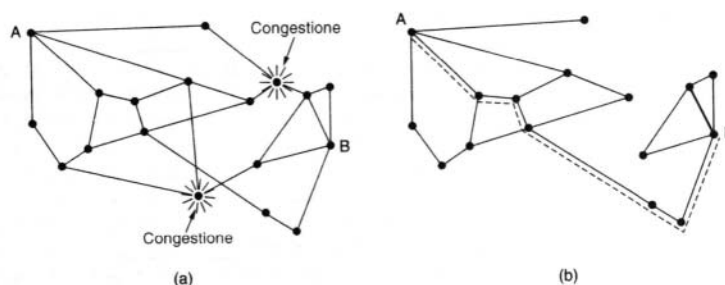


Figura A.30: (a) una rete congestionata; (b) rete ridisegnata che evita le congestioni.

– Rete broadcast.

Il livello di rete è assente.

5.3.4 L4: livello di trasporto (transport).

Il livello di trasporto costituisce la “*frontiera*” tra le funzioni di comunicazione e quelle di elaborazione. Esso dispone di mezzi per l’apertura, il mantenimento e la chiusura di *connessioni di trasporto* per la trasmissione di dati full-duplex tra interlocutori *end-to-end*; tutto ciò mascherando qualsiasi problema relativo alla trasmissione e garantendo, quindi, l’indipendenza delle funzioni di elaborazione dalle caratteristiche della rete utilizzata. Conseguentemente, a partire dal livello di trasporto, la rete *non* è più visibile (Figura A.31).

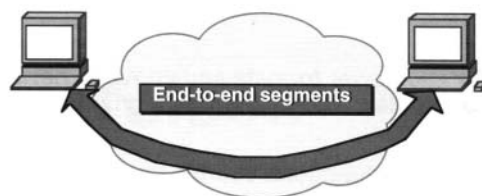


Figura A.31: Visibilità a livello di trasporto.

Funzioni principali:

- Fornitore di servizi di trasporto.

L'entità di trasporto mette a disposizione delle applicazioni dei servizi di comunicazione *connection-oriented* oppure *connection-less*, *affidabili* o *non-affidabili* ed eventualmente con una specifica *QoS*. Nel caso *connection-oriented* gestisce le operazioni di creazione (*set up*) e rilascio ordinato (*tearing down*) delle connessioni.

- Frammentazione dei messaggi (*fragmentation*).

Lunghi messaggi provenienti dalle applicazioni vengono suddivisi in segmenti di dimensione opportuna. Eventuali ritrasmissioni interessano così solamente un segmento anziché l'intero messaggio. Come conseguenza della frammentazione, è necessario arricchire l'header con dei numeri di sequenza.

- Gestione degli errori (*error recovery*).

Il livello di rete fornisce generalmente un servizio non-affidabile.

Se un'applicazione richiede un servizio *affidabile*, il livello di trasporto ha il compito di assicurarsi che tutti i segmenti giungano correttamente a destinazione, eventualmente dopo ripetute *ritrasmissioni* (ARQ). La tecnica qui implementata differisce da quella possibilmente presente a livello di linea; gli scopi sono, infatti, diversi: il *livello 4* si occupa di trasmettere in modo affidabile interi messaggi su un percorso *end-to-end*, il *livello 2* di trasmettere senza errori piccoli frame su una linea *punto-punto*.

- Controllo del flusso (*flow control*).

Al livello 4 il controllo viene esercitato per evitare che l'end-system A (mittente) inondi di segmenti l'end-system B (destinatario). Il controllo del flusso è spesso integrato nel meccanismo ARQ (protocolli *sliding windows*). La Figura A.32 illustra la differenza tra regolazione del flusso per connessione e regolazione del flusso a livello di linea.

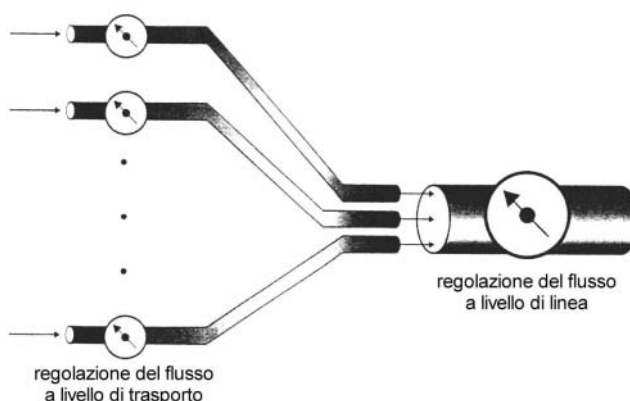


Figura A.32: Regolazione del flusso.

5.3.5 L5: livello di sessione (session).

E' il più basso tra i livelli di elaborazione; la sua funzione è quella di consentire lo scambio di messaggi tra *end-points* per mezzo di una connessione logica esente da interruzioni ed errori. Ogni connessione di sessione utilizza *una* connessione di trasporto, la quale non è necessariamente impegnata per tutta la durata della sessione: il livello di sessione, infatti, può autonomamente decidere di chiudere una connessione di trasporto e di riaprirne un'altra, qualora ciò fosse conveniente, senza produrre conseguenze sui livelli superiori. Se, ad esempio, si verifica un guasto mentre la trasmissione è in corso, la connessione di trasporto corrente viene *terminata* e ne viene *aperta* una seconda, riattivando così il trasferimento delle unità informative esattamente dal punto in cui era stato interrotto.

5.3.6 L6: livello di presentazione (presentation).

Consente la corretta *interpretazione* dei messaggi scambiati dalle *applicazioni*, indipendentemente dai formati, dai codici e da tutte le altre convenzioni impiegate in ogni end-system. Si può dire che lo strato di presentazione agisce come un *interprete*: la situazione è infatti analoga a quella di due persone di lingue diverse che possono liberamente dialogare grazie a due interpreti che traducono i loro discorsi in un linguaggio assunto come riferimento comune (la lingua inglese, per esempio).

5.3.7 L7: livello di applicazione (application).

E' il livello OSI più vicino all'utente; esso ha la caratteristica di fornire servizi non ad un livello OSI, ma ad un'applicazione che è *fuori* dalla copertura del modello. I livelli inferiori (L1..L6) forniscono un supporto di comunicazione tra 2 o più end-system affidabile, ma non effettuano nessun compito *utile* per l'utente: al livello 7 troviamo dei protocolli che svolgono funzioni utili alle applicazioni reali e che pertanto non ha senso *replicare* in ciascuna di esse (ciò giustifica l'introduzione di un ulteriore layer sopra il livello 6).

Lo scenario è piuttosto vasto; abbiamo protocolli opportuni per svolgere varie *funzioni*:

- *trasferimento di file (FTP),*
- *posta elettronica (SMTP),*
- *newsgroup (NNTP),*
- *collegamento di documenti ipertestuali (HTTP),*
- *traduzione di indirizzi mnemonici in indirizzi di rete (DNS),*
- *crittografia, autenticazione, firme digitali,*
- *gestione e manutenzione della rete (SNMP),*
- *codifica e decodifica di audio e video in tempo reale,*
- ecc.*

La Tabella A.5 riassume le funzioni principali dei vari livelli OSI.

livello	nome	funzioni
L1	<i>Fisico</i>	- attivazione/disattivazione del MT; - sincronizzazione e decisione; - eventuale modulazione/demodulazione.
L2	<i>Linea</i>	- framing; - controllo degli errori; - controllo del flusso (escluso fast-packet); - controllo dell'accesso al MT (solo reti broadcast).
L3	<i>Rete</i>	- routing; - controllo delle congestioni.
L4	<i>Trasporto</i>	- fornitore di servizi di trasporto; - frammentazione dei messaggi; - gestione degli errori; - controllo del flusso.
L5	<i>Sessione</i>	crea una connessione logica esente da interruzioni ed errori.
L6	<i>Presentazione</i>	permette la corretta interpretazione dei messaggi.
L7	<i>Applicazione</i>	fornitore di servizi utente.

Tabella A.5: Funzioni dei livelli OSI.



6 Appendice 1

6.1 Il livello 1: fisico

Lo studio del livello fisico richiede un preliminare studio della teoria dei segnali che esula dagli scopi di queste note. Ci limiteremo ad osservare alcuni aspetti principali in maniera elementare. L'informazione può essere trasmessa a distanza variando opportunamente una qualche caratteristica fisica del mezzo scelto per la trasmissione. Tale variazione si propaga, con una certa velocità, lungo il mezzo di trasmissione e dopo un certo tempo arriva all'altra estremità del mezzo, dove può venir rilevata. Ad esempio, se il mezzo è un cavo metallico, si può variare la tensione applicata ad un'estremità. Tale variazione di tensione verrà successivamente rilevata all'altra estremità. I mezzi trasmissivi sono sostanzialmente di tre tipi:

- *mezzi elettrici* (cavi): in essi il fenomeno fisico utilizzato è l'energia elettrica;
- *mezzi ottici* (LED, laser e fibre ottiche): in essi il fenomeno utilizzato è la luce;
- *mezzi wireless* (onde radio): il fenomeno fisico è l'onda elettromagnetica, una combinazione di campo elettrico e campo magnetico variabili, che si propaga nello spazio e che induce a distanza una corrente elettrica in un dispositivo ricevente (antenna).

In linea di principio, la trasmissione può avvenire con due modalità differenti:

- ☞ trasmissione di segnale analogico;
- ☞ trasmissione di segnale digitale.

Il primo può variare gradualmente in un intervallo costituito da un numero infinito di possibili valori; il secondo può variare solamente passando bruscamente da uno all'altro di un insieme molto piccolo di valori (da due a qualche decina). Si tenga presente però che il fenomeno fisico utilizzato non è digitale, ma analogico. Un segnale quindi non può passare istantaneamente da un valore ad un altro, ma impiegherà un certo tempo per effettuare la transizione. La conseguenza è che un mezzo fisico farà del suo meglio per trasportare un segnale digitale, ma non riuscirà a farlo arrivare esattamente com'è partito.

Dalla analisi di Fourier sappiamo che un generico segnale variabile nel tempo è equivalente ad una somma di funzioni sinusoidali aventi ciascuna una propria ampiezza e frequenza. Si può quindi rappresentare un segnale di durata T attraverso il suo spettro di frequenze, ossia attraverso la sua scomposizione in sinusoidi.

Qualunque segnale è quindi caratterizzato da un intervallo di frequenze nel quale sono comprese le frequenze delle sinusoidi che lo descrivono. Esso va sotto il nome di banda di frequenza del segnale.

Anche i mezzi trasmissivi sono caratterizzati da un intervallo di frequenze, detta *banda passante*, che rappresenta l'intervallo di frequenze che il mezzo fisico è in grado di trasmettere senza alterarle oltre certi limiti. Le alterazioni principali sono la *attenuazione* e l'introduzione di *ritardo*, che di norma variano al variare delle frequenze trasmesse. In generale, i mezzi trasmissivi: attenuano i segnali in proporzione alla distanza percorsa e alla frequenza del segnale; propagano i segnali a velocità proporzionali alle loro frequenze. Una conseguenza è che, per qualunque mezzo trasmissivo, la banda passante si riduce all'aumentare della lunghezza del mezzo stesso.

Perché un segnale sia ricevuto come è stato trasmesso, è necessario che la banda passante sia uguale o più ampia della banda di frequenza del segnale stesso. Altrimenti, il segnale viene privato di alcune delle sue armoniche (tipicamente quelle di frequenza più elevata) e viene quindi distorto, cioè alterato. Se un numero sufficiente di armoniche arriva a destinazione, il segnale è comunque utilizzabile. Per maggiori informazioni si vedano i teoremi di Nyquist e di Shannon.

6.2 I mezzi trasmissivi

Analizziamo ora brevemente i mezzi trasmissivi tipicamente utilizzati nelle reti di calcolatori.

– Doppino telefonico

Consiste di una coppia di conduttori in rame intrecciati l'uno coll'altro in forma elicoidale. Ciò fa sì che si minimizzino le interferenze fra coppie adiacenti (due fili paralleli costituiscono un'antenna; se sono intrecciati no). Usato, storicamente, per le connessioni terminali del sistema telefonico (da casa alla centrale più vicina).

La larghezza di banda dipende dalla lunghezza, ma comunque si può trasmettere a diversi Mbps su distanze fino a qualche km. Esistono varie categorie di doppini, in particolare:

categoria 3: due fili isolati, leggermente attorcigliati. Quattro coppie contenute in una guaina di plastica. Comune nei cablaggi telefonici interni agli edifici (si possono avere quattro telefoni per stanza);

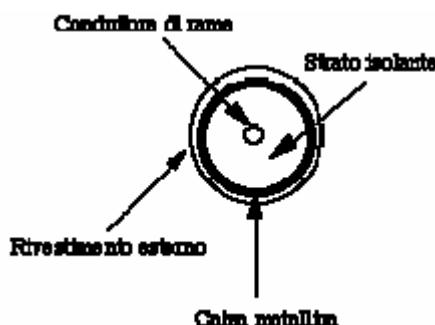
categoria 5: simile alla cat. 3, ma con un più fitto avvolgimento (più giri per centimetro) e con isolamento in teflon. Migliore qualità del segnale sulle lunghe distanze, adatto a collegamenti in alta velocità in ambito LAN (ad esempio per Ethernet a 100 Mbps, ATM a 34 Mbps).

Entrambi i tipi sono spesso chiamati UTP (Unshielded Twisted Pair), per distinguerli da un altro tipo, detto STP (Shielded Twisted Pair) che è schermato e quindi offre migliori prestazioni, ma è molto più ingombrante e, di fatto, non viene usato quasi più.

– Cavo coassiale

E' un altro comune mezzo di trasmissione; offre un miglior isolamento rispetto al doppino e quindi consente velocità di trasmissione maggiori su distanze superiori.

E' costituito da un conduttore centrale in rame circondato da uno strato isolante all'esterno del quale c'è una calza metallica.



In passato molto usato nel sistema telefonico per le tratte a lunga distanza, è ormai sostituito quasi ovunque dalla fibra ottica. Rimane in uso per la TV via cavo e in molte LAN. Tipi di cavo coassiale:

Baseband coaxial cable (50 ohm): il cavo baseband è usato per la trasmissione digitale, e consente velocità da 1 a 2 Gbps fino a circa 1 km. Per distanze superiori si devono interporre amplificatori. Per banda base intendiamo che l'intera banda passante è usata per una singola trasmissione, di tipo digitale.

Broadband coaxial cable (75 ohm): è usato per la trasmissione analogica. E' il cavo standard della TV. Offre una banda di 300 MHz e può estendersi fino a quasi 100 km. Per broadband si intende, nella telefonia qualunque trasmissione più ampia di 4 kHz; nella trasmissione dati si riferisce a un cavo su cui viaggia un segnale analogico che, con opportune tecniche di

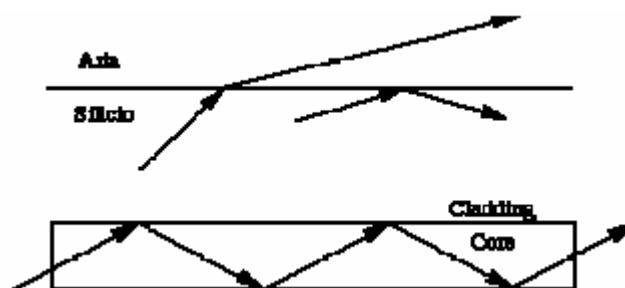
multiplazione, viene usato per effettuare contemporaneamente più trasmissioni distinte, separate in differenti bande di frequenza. La banda totale è suddivisa in canali di banda più piccola (ad es. 6 MHz per ciascun segnale TV) indipendenti gli uni dagli altri. Mentre un canale porta un segnale TV, un altro può portare una trasmissione dati (ovviamente con apparecchiature di conversione digitale/analogica e viceversa), tipicamente a 3 Mbps.

Tecnicamente, il cavo broadband è inferiore a baseband per la trasmissione digitale, ma ha il vantaggio di essere già in opera in grandi quantità.

– *Fibre ottiche*

Sono mezzi trasmissivi costituiti da un sottilissimo cilindro centrale in vetro, (*core*) circondato da uno strato esterno (*cladding*) di vetro avente un diverso indice di rifrazione e da una guaina protettiva. Sono quindi raggruppate insieme in una guaina esterna.

Le fibre ottiche sfruttano il principio della deviazione che un raggio di luce subisce quando attraversa il confine fra due materiali diversi (core e cladding nel caso delle fibre). La deviazione dipende dagli indici di rifrazione dei due materiali. Oltre un certo angolo, il raggio rimane intrappolato all'interno del materiale.



Deviazione del raggio luminoso

Tipi di fibre ottiche:

- *multimodali*: raggi diversi possono colpire la superficie con diversi angoli (detti mode), proseguendo quindi con diversi cammini. Il diametro del core è di 50 micron;
- *monomodali*: sono così sottili (il diametro del core è 8-10 micron) che si comportano come una guida d'onda: la luce avanza in modo rettilineo, senza rimbalzare. Sono più costose ma reggono distanze più lunghe (fino a 30 km).

Le fibre ottiche hanno prestazioni molto maggiori dei precedentemente analizzati mezzi trasmissivi: è facilmente raggiungibile una velocità di trasmissione di 50.000 Gbps (50 Tbps)

con un bassissimo tasso d'errore. La pratica attuale di usare velocità dell'ordine dei Gbps dipende dall'incapacità di convertire più velocemente segnali elettrici in luminosi. Infatti, nelle fibre ottiche, il mezzo fisico utilizzato è ovviamente la luce, e un impulso luminoso rappresenta un 1 mentre la sua assenza uno zero.

Le fibre ottiche sono costruite con un vetro speciale, molto trasparente, per cui offrono una bassissima attenuazione del segnale luminoso. L'attenuazione dipende anche dalla lunghezza d'onda della luce, per cui si usano comunemente tre particolari bande per la trasmissione (tutte nell'infrarosso vicino), larghe da 25.000 GHz a 30.000 GHz ciascuna (un'enormità). Un sistema di trasmissione ottica ha tre componenti. La sorgente luminosa che può essere un LED o un laser, essa converte un segnale elettrico in impulsi luminosi; il mezzo di trasmissione che è la fibra vera e propria; il fotodiodo ricevitore che converte gli impulsi luminosi in segnali elettrici. Il tipico tempo di risposta di un fotodiodo è 1 nsec., da cui il limite di 1 Gbps.

Ci sono due topologie comuni per le reti basate su fibre ottiche:

- *anello*: mediante la concatenazione di più spezzoni di fibre ottiche si crea un anello. Tutti collegamenti sono punto a punto. L'interfaccia può essere passiva (fa passare l'impulso luminoso nell'anello) o attiva (converte l'impulso in elettricità, lo amplifica e lo riconverte in luce);
- *stella passiva*: l'impulso, inviato da un trasmettitore, arriva in un cilindro di vetro al quale sono attaccate tutte le fibre ottiche; viene poi distribuito alle fibre ottiche uscenti. Si realizza così una rete broadcast.

Vantaggi e svantaggi delle fibre ottiche:

- ☞ leggerezza a parità di larghezza di banda (due fibre sono più capaci di 1.000 doppini, 100 kg/km contro 8.000 kg/km);
- ☞ totale insensibilità a disturbi elettromagnetici;
- ☞ difficile inserimento di intrusi per spiare il traffico;
- ☞ elevato costo delle giunzioni;
- ☞ comunicazione unidirezionale (necessarie due fibre per comunicazione duplex).

– *Trasmissioni senza filo (wireless)*

Le onde elettromagnetiche, create dal movimento degli elettroni, viaggiano nello spazio (anche vuoto) alla velocità della luce e possono indurre una corrente in un dispositivo



ricevente (antenna) anche molto distante. Le porzioni dello spettro elettromagnetico utilizzabili per la trasmissione dati includono: onde radio; microonde; raggi infrarossi; luce visibile; raggi ultravioletti.

In generale, almeno per le onde radio, l'allocazione delle frequenze dipende da un'autorità statale.

Man mano che si sale di frequenza si hanno comportamenti diversi. Le onde radio, di frequenza più bassa, passano attraverso gli edifici, percorrono lunghe distanze e vengono riflesse dalla ionosfera; a frequenze più elevate (lunghezza d'onda dell'ordine dei cm o mm) sono estremamente direzionali e vengono fermate dagli ostacoli, compresa la pioggia e in tutti i casi sono soggette a interferenze elettromagnetiche. La trasmissione (almeno per basse frequenze) è inerentemente di tipo broadcast.

Anche in questo ambito la velocità di trasmissione è funzione dell'ampiezza della banda utilizzata. Si trasmettono informazioni modulando l'ampiezza, la frequenza e/o la fase dell'onda.