INTRODUZIONE ALLE RETI DI COMPUTER

DEFINIZIONE DI "RETE DI COMPUTER"

L'utilizzo contemporaneo della tecnologia dei computer e della tecnologia delle telecomunicazioni ha dunque permesso la nascita delle **reti informatiche**, usate sia all'interno delle singole organizzazioni sia tra consorzi di organizzazioni sia tra singoli individui.

Che cos'è allora una *rete di computer*? Una semplice definizione è la seguente: ***una rete di computer è un insieme di computer collegati tra di loro.***

Nelle figura seguente è mostrato un semplice esempio di rete di computer:

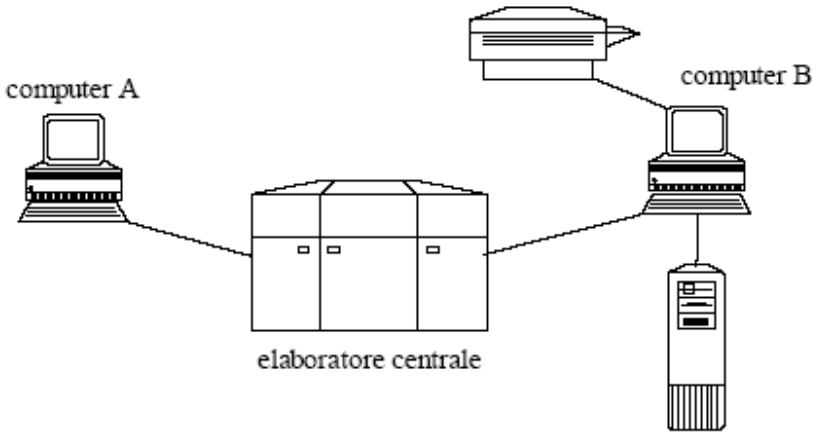


Figura 1 - Semplice esempio di rete di computer

I computer collegati alla rete possono essere i più vari, di marche diverse tra loro e con diverse capacità elaborative (dal *PC* al *mainframe*); ciascun computer ha delle proprie risorse (tipicamente *periferiche di input e di output*, *dischi rigidi* e così via) e a ciascun computer possono essere collegate una o più *stazioni d'utente*, altrimenti dette **terminali**. Con riferimento alla figura 1, si osserva, ad esempio, quanto segue:

- in primo luogo, abbiamo un elaboratore centrale al quale, come si vedrà nei dettagli più avanti, è nella maggior parte dei casi affidata la gestione della rete stessa; tale elaboratore, oltre a svolgere funzioni di controllo e gestione, può possedere delle risorse proprie;
- a tali risorse attingono, nel caso considerato, due diversi computer: la differenza, tra i due, è che il computer B, al contrario del computer A, dispone di due risorse in più, cioè una *stampante* e un ulteriore *disco rigido*;
- tuttavia, il fatto che tali due risorse siano a loro volta collegate alla rete (il disco rigido direttamente, mentre la stampante tramite il computer B) fa' si che anche il computer A ne possa usufruire, previa opportuna richiesta.

Anche le **linee di interconnessione**, che hanno il compito di trasmettere i dati tra computer e terminali oppure tra computer e computer, possono essere di svariati tipi: per esempio, in caso di lunghe distanze, la linea di interconnessione tradizionale è la **linea telefonica**, grazie anche e soprattutto alla sua diffusione capillare. Stanno diffondendosi adesso anche le connessioni su **fibra ottica** e ci sono dei particolari *standard di trasmissione* (come ad esempio lo *standard ATM* ideato dalle principali compagnie telefoniche in risposta allo *standard TCP/IP* della rete *Internet*) specificamente progettati per tali mezzi trasmissivi.

Vantaggi delle reti informatiche

Le *reti informatiche* portano diversi vantaggi agli utenti, grandi e piccoli, collegati. Li possiamo velocemente elencare come segue:

- le moderne organizzazioni sono spesso caratterizzate da una distribuzione di uffici su un territorio molto vasto (basti pensare alle grandi organizzazioni nazionali o addirittura mondiali); i computer ed i terminali ubicati in un determinato luogo devono poter scambiare *dati* e *programmi* con quelli che si trovano in luoghi diversi; usando, a questo scopo, una rete informatica, si ha un aggiornamento quotidiano e costante dell'insieme delle informazioni aziendali;
- il collegamento tra computer permette inoltre una migliore condivisione delle risorse aziendali: per esempio, gli utenti di un dato computer, normalmente dedicato ad una applicazione specifica, potrebbero trovarsi nella necessità di accedere a risorse di un altro computer; oppure, una situazione di carico di lavoro eccessivo su un sistema può essere risolta inviando parte del lavoro ad un altro sistema della rete;
- la rete permette inoltre di risolvere anomalie o guasti: se un sistema A è fuori uso, le sue mansioni possono essere svolte da un altro sistema B senza incidere eccessivamente sulle normali operazioni aziendali (si dice, in questo caso, che il sistema B svolge "*funzioni di back-up*");
- si possono infine trovare vantaggi anche in termini organizzativi: un operatore che viaggia, può essere dotato di un terminale o di un sistema terminale trasportabile (tipicamente un "*computer portatile*") che gli consente di svolgere le sue mansioni ovunque ci sia un collegamento in rete alla propria azienda.

I vantaggi di una rete informatica non riguardano solo le organizzazioni, ma anche i singoli individui:

- accesso ad informazioni remote, ad es.:
- accesso a servizi bancari;
- acquisti da casa;
- navigazione sul World Wide Web;
- comunicazioni fra persone:
- posta elettronica;
- videoconferenza;
- gruppi di discussione;
- divertimento:
- video on demand (selezione e ricezione via rete di un qualunque spettacolo tratto da un catalogo);
- giochi interattivi (contro macchine o avversari umani).

UN SEMPLICE SCHEMA DI RETE DI COMPUTER

Consideriamo una semplicissima rete, costituita da 2 computer collegati tra loro da una linea trasmissiva:

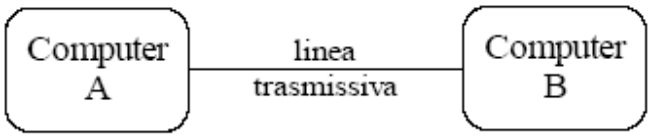


Figura 2 - Struttura schematica di una rete di due computer

Rientra in questo semplice schema anche il collegamento tra un computer ed un terminale (cioè una stazione utente), dato che gran parte delle stazioni terminali è attualmente costituita da veri e propri computer programmabili e quindi dotati del proprio sistema operativo e del proprio software applicativo.

Descriviamo allora nei dettagli quale può essere il funzionamento di una rete del genere,

procedendo anche a perfezionare quello schema per il momento assolutamente generico.
Il computer A avrà del **software applicativo** (che brevemente si indica con **AP**) che deve erogare servizi all'utente, nel senso che si tratta di quell'insieme di programmi (**applicazioni**) che consentono al computer di rispondere alle esigenze dell'utente; quest'ultimo ha la possibilità di inoltrare le proprie richieste al computer mediante opportuni *strumenti di immissione*, quali una tastiera, un lettore di tessere e così via.
Supponiamo allora che una applicazione AP_{A1} del computer A chieda di accedere, per rispondere alla richiesta del proprio utente locale, alle risorse di una applicazione AP_{B1} presente nel computer B. Supponiamo inoltre che, contemporaneamente, una applicazione AP_{B2} del computer B richieda di accedere ad una applicazione AP_{A2} del computer A, per esempio al fine di leggere un certo file.
Abbiamo dunque una situazione in cui, contemporaneamente, ciascun computer chiede di accedere a parte delle risorse dell'altro computer.
Il fatto che esista una sola linea di connessione tra i due computer comporta allora che questa unica *linea fisica* debba essere impiegata per due *interazioni logiche* diverse: nel nostro esempio, abbiamo l'interazione tra AP_{A1} e AP_{B1} e l'interazione tra AP_{B2} e AP_{A2}. Perché questo sia possibile, è necessario che sui due sistemi sia presente uno **strato di software** capace sia di indirizzare, in partenza, i vari messaggi dell'applicazione interessata sia anche di smistare i messaggi in arrivo. Il software del computer B deve da un lato rispondere alla richiesta del computer A e, dall'altro, inviare al computer A la richiesta di accedere alla applicazione AP_{A2}. Stesso discorso ovviamente per il computer A.
Questo *software* ha anche altri compiti: per esempio, esso deve rendere facile la programmazione della gestione delle richieste di trasmissione, amministrando in proprio tutte le complesse funzioni trasmissive; inoltre, esso deve anche provvedere ad inviare effettivamente i dati nella rete.
Premesso questo, diamo una prima importante definizione che spesso ricorrerà in seguito:
prende il nome di ***Data Terminal Equipment*** *(brevemente DTE) il complesso costituito dal sistema, dal terminale (che può accompagnare o sostituire il sistema) e dalle relative risorse (applicazioni, strumenti di INput e di OUTput) collegati in rete per la trasmissione dei dati.*
Il DTE può essere dunque un mainframe, un semplice PC o anche semplicemente un terminale.
Scopo della rete è l'interconnessione dei vari DTE per la condivisione delle risorse, lo scambio di dati e la cooperazione tra i processi applicativi
Uno schema più completo di rete tra due computer può essere il seguente:

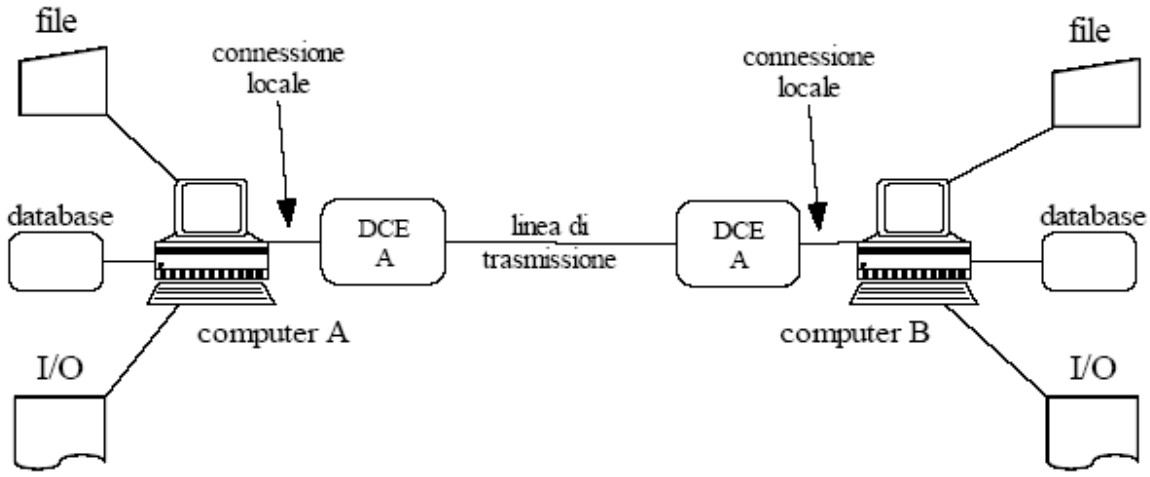


Figura 3 - Struttura dettagliata di una rete di due computer collegati mediante una linea trasmissiva

Il computer A e tutte le risorse (file - database - I/O) ad esso connesse costituisce il DTE A, mentre il computer B, con le proprie risorse, costituisce il DTE B. Come evidenziato dalla figura, ciascun DTE è collegato alla **linea di trasmissione** mediante un apposito dispositivo, che prende il nome di **Data Circuit-Terminating Equipment** (brevemente DCE): quando la linea di trasmissione è la normale linea telefonica, il DCE è un normale **modem**.
Nello schema appena tracciato si evidenziano sia *connessioni logiche* sia *connessioni fisiche*:

- il termine "logico" significa, in questo contesto, che i DTE non sono coinvolti con gli aspetti fisici della trasmissione: l'applicazione A1 ha solo bisogno di inviare una richiesta di READ corredata da un qualcosa (il cosiddetto *identificatore*) che consenta di individuare, nel computer B, i dati richiesti; ovviamente, la controparte B1 deve essere in grado di interpretare correttamente la richiesta di READ in modo da preparare la risposta; questi sono appunto gli *aspetti logici* della connessione;
- l'effettivo scambio di dati avviene poi sfruttando il *collegamento fisico*, costituito dalla linea di connessione tra i due DCE, dai due DCE stessi e dalle due linee che collegano ciascun DCE col proprio computer.

Le *interfacce comunicative* dei due DTE, ossia organi e programmi responsabili, rispettivamente, degli aspetti fisici e logici della trasmissione, dialogano tra loro mediante l'uso di *protocolli*: *un **protocollo** è una serie di norme, convenzioni e tecniche per lo scambio di dati, di comandi e di informazioni di controllo tra due DTE.*
Come vedremo meglio in seguito, con l'introduzione del *modello ISO/OSI*, esistono molti **livelli** di protocolli: si va dal livello più basso, che regola semplicemente il modo di trasmettere i segnali binari sulla linea, al livello più alto, che invece indica come interpretare dati e comandi *a livello applicativo*, passando per una serie variabile di ulteriori livelli. Al giorno d'oggi, molte organizzazioni desiderano usare interfacce e protocolli comuni e standardizzati, al fine di avere la maggiore capacità di interconnessione possibile.

RETI "PUNTO-A-PUNTO", "MULTIPUNTO" E "BROADCAST"
Partiamo da una semplice definizione: *un circuito fisico è detto **punto-a-punto** quando collega due soli DTE.*
La figura seguente mostra un esempio di circuito punto-a-punto:

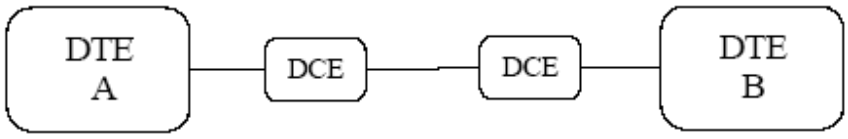


Figura 4 - Circuito fisico punto-a-punto

Il collegamento punto-a-punto è spesso utilizzato nella connessione tra due computer oppure in quella tra un computer ed un terminale. I principali vantaggi di questa configurazione sono i seguenti:

- semplicità di gestione: quello che viene trasmesso da un DTE è sempre diretto all'altro;
- tempi di attesa nulli: il DTE che deve trasmettere trova sempre il circuito disponibile, per cui può trasmettere ogni volta che ne ha bisogno.

Ci sono però anche degli svantaggi, legati essenzialmente alla linea di collegamento:

- in primo luogo, il costo della linea, specie se essa corre su una distanza notevole, può diventare elevato;
- inoltre, una organizzazione che volesse collegare, al proprio mainframe, 10.000 terminali con questa tecnica, dovrebbe provvedere a installare 10.000 linee di collegamento

Al fine di ridurre i costi complessivi della linea, si può invece pensare alla configurazione *multipunto*: `un circuito fisico` ***multipunto*** `consiste nel mettere più di due DTE sulla stessa linea.`

La figura seguente mostra una configurazione multipunto con un numero imprecisato di DTE:

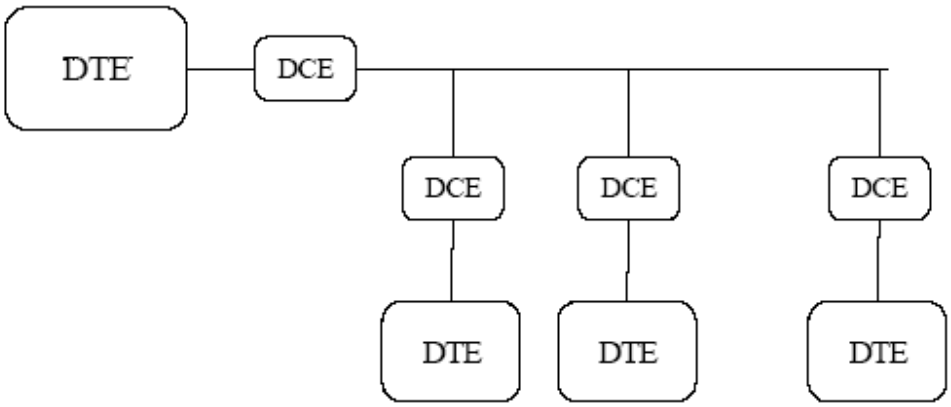


Figura 5 - Circuito fisico multipunto

La configurazione prevede dunque un *DTE principale*, le cui funzioni saranno chiare tra un attimo, collegato, tramite il proprio DCE e tramite una sola linea di comunicazione, ad un numero N di altri *DTE secondari*, ciascuno dotato del proprio DCE.

`Il problema principale della configurazione multipunto è che può nascere una` ***contesa***`, ossia una situazione in cui più di un DTE ha bisogno di usare la linea per trasmettere il proprio messaggio.`

Questo problema nasce dal fatto che la linea di trasmissione è in grado di trasmettere solo un messaggio alla volta in ciascun senso di direzione: ciò significa che, al massimo, ci può essere un messaggio in corso di trasmissione in un senso e un altro messaggio in corso di trasmissione nel senso opposto. Ciò comporta che un DTE che voglia trasmettere, possa trovare la linea già occupata

e debba perciò attendere che essa si liberi. Dal punto di vista dell'utente, questo significa tempi di trasmissione superiori rispetto alla configurazione punto-a-punto, visto che, in quel caso, il canale di trasmissione non può mai risultare occupato. Possiamo esprimerci dicendo che *il* ***tempo medio di attesa****, per il generico utente della rete, è nullo nella configurazione punto-a-punto, mentre non è nullo in quella multipunto.*

La gestione di una rete con la configurazione multipunto è dunque piuttosto complessa. E' necessaria la presenza di "qualcuno" che regoli la conversazione sul circuito fisico, ossia che stabilisca, sulla base di precise regole, quale stazione possa trasmettere in un determinato momento. Questo "qualcuno" è ovviamente uno dei DTE connessi alla rete e prende perciò il nome di **master**: come si nota nella figura 5, esso è normalmente situato ad un estremo della linea e costituito da un computer. Gli altri DTE collegati sono detti invece **slave** e possono comunicare solo dietro autorizzazione del master.

Il master deve dunque svolgere un lavoro ulteriore rispetto ai normali compiti applicativi e puramente trasmissivi: esso deve dedicare risorse per gestire in modo opportuno l'assegnazione del diritto a trasmettere sulla linea.

I principali limiti della configurazione multipunto sono i seguenti:

- limiti tecnici: ogni "*derivazione intermedia*", ossia ogni DTE che viene inserito nella linea, comporta un degrado delle caratteristiche elettriche del segnale trasmesso: infatti, quanto più lungo è il percorso che il segnale deve percorrere, tanto maggiori sono i disturbi (e quindi le distorsioni) e le attenuazioni cui è soggetto; ecco perché esistono dei limiti normativi al numero dei DTE collegabili in multipunto;
- limiti funzionali: dato che esiste una logica di scelta, rappresentata da un preciso protocollo, è possibile collegare, sulla linea multipunto, solo terminali che adottino lo stesso protocollo;
- limiti applicativi: al crescere del numero di terminali collegati, cresce il traffico sulla linea e quindi, mediamente, cresce anche il *tempo di attesa*; questo è un altro motivo che obbliga a limitare il numero di terminali, in funzione del carico globale trasmesso e dei tempi di risposta tipici delle applicazioni utilizzate.

Un altro aspetto negativo della configurazione multipunto è che, se si dovesse guastare il DTE master, ciò comporterebbe automaticamente un blocco dell'intera rete.

All'opposto delle reti multipunto e punto-a-punto si collocano le cosiddetta **reti broadcast:** queste sono dotate di un unico canale di comunicazione che è condiviso da tutti gli elaboratori. Brevi messaggi (spesso chiamati ***pacchetti***) inviati da un elaboratore sono ricevuti da tutti gli altri elaboratori. Un indirizzo all'interno del pacchetto specifica il destinatario.

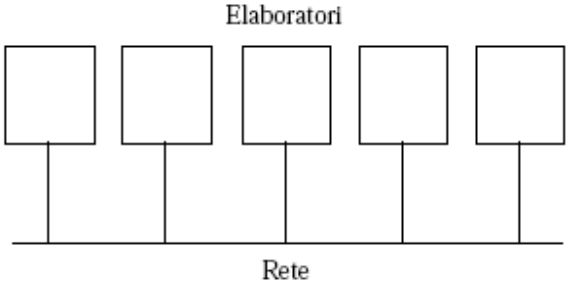


Figura 6 - Schema logico di una rete broadcast

Quando un elaboratore riceve un pacchetto, esamina l'indirizzo di destinazione; se questo coincide col proprio indirizzo, il pacchetto viene elaborato, altrimenti viene ignorato.

Le reti broadcast, in genere, consentono anche di inviare un pacchetto a tutti gli elaboratori,

usando un opportuno indirizzo. Si parla in questo caso di **broadcasting** (si pensi alla diffusione radio-televisiva). In tal caso tutti prendono in considerazione il pacchetto.
Un' altra possibilità è inviare il pacchetto ad un sottoinsieme degli elaboratori: si parla in questo caso di **multicasting** e succede che solo gli elaboratori del suddetto sottoinsieme prendono in considerazione il pacchetto, che invece viene ignorato dagli altri. In ciascun pacchetto è presente un bit che indica che si tratta di una trasmissione in multicasting, mentre i rimanenti bit contengono l'indirizzo del gruppo destinatario ed ovviamente i dati. In particolare, il bit che indica o meno il multicasting appartiene allo stesso campo contenente l'indirizzo: se N sono i bit di tale campo, quindi, solo N-1 sono riservati all'indirizzo vero e proprio.

TOPOLOGIE DI RETE

Partiamo ancora una volta da una definizione: *prende il nome di* ***topologia di rete*** *la configurazione geometrica dei collegamenti tra i vari componenti della rete.*
Esistono vari tipi di topologie, la scelta dei quali è legata al conseguimento di alcuni obiettivi fondamentali:

- in primo luogo, è necessario assicurare la massima affidabilità complessiva della rete, rispettando, ovviamente, alcuni vincoli economici; *affidabilità della rete* significa diverse cose: ad esempio, significa trovare delle possibili strade alternative tra due DTE quando la strada normalmente percorsa (che può essere per esempio quella più breve) viene interrotta a causa del malfunzionamento di qualche componente intermedio (linea, DSE o altro) o a causa di un intervento di manutenzione della stessa; significa anche buona qualità della trasmissione, ossia numero di errori più basso possibile e la presenza di strumenti e procedure per risolvere le situazioni di errore. L'affidabilità della rete è spesso tenuta sotto controllo da strumenti (software e sistemi) che si dice svolgono funzioni di **Network Management**, ossia appunto *gestione della rete*;
- in secondo luogo, è necessario consentire un alto rendimento complessivo della rete, intendendo con questo, tra le altre cose, tempi di risposta sufficientemente brevi. Il **rendimento complessivo** della rete si può misurare in *transazioni elaborate nell'unità di tempo*. Esso dipende da una serie di fattori:

∗ numero e tipo di sistemi collegati;
∗ capacità di parallelismo dei sistemi, ossia capacità di elaborare, nello stesso tempo, più di una transazione;
∗ portata della linea di trasmissione o delle linee di trasmissione;
∗ numero di linee di trasmissione;
∗ capacità di parallelismo di trasmissione in rete.

In particolare, è importante il cosiddetto **tempo di risposta**, ossia l'intervallo di tempo che intercorre tra l'istante in cui una data applicazione fa richiesta di dati e l'istante in cui tali dati arrivano effettivamente all'applicazione. Questo tempo di risposta è somma di una serie di tempi:
∗ *tempo di input* (tempo necessario perché l'applicazione generi la richiesta e la invii sulla linea)
∗ *tempo di trasmissione in un senso* (tempo necessario perché la richiesta giunga al destinatario)
∗ *tempo di elaborazione* (tempo richiesto dal destinatario per rendere disponibili i dati richiesti e inviarli sulla linea)
∗ *tempo di trasmissione in senso opposto* (tempo necessario perché i dati giungano alla stazione che ne ha fatto richiesta)

∗ *tempo di output* (tempo necessario perché i dati siano effettivamente a disposizione dell'applicazione cui necessitano).

Questo tempo di risposta dipende dai seguenti fattori:

∗ caratteristiche dell'applicazione che richiede i servizi della rete;
∗ tipo di terminale;
∗ portata e carico delle linee utilizzate;
∗ numero di componenti di rete attraversati.

- Infine, l'ultimo obiettivo da perseguire in una rete è quello di minimizzare i costi di rete, facendo in modo, per esempio, che il numero complessivo delle linee sia minimo (il che si può ottenere facendo ricorso a *collegamenti commutati* nel caso di terminali con basso carico trasmissivo e a *collegamenti permanenti* solo per le locazioni che interscambiano un alto volume di dati.

Sulla base di questi obiettivi la topologia della rete che si intende realizzare va scelta tra quelle elencate di seguito, che sono le più comuni. Anticipiamo che le topologie di rete saranno dettagliatamente esaminate in seguito, nel capitolo sulle LAN (*Local Area Network*).

Rete gerarchica (o ad albero)

Questo tipo di configurazione è quella più comune e può essere rappresentata graficamente nel modo seguente:

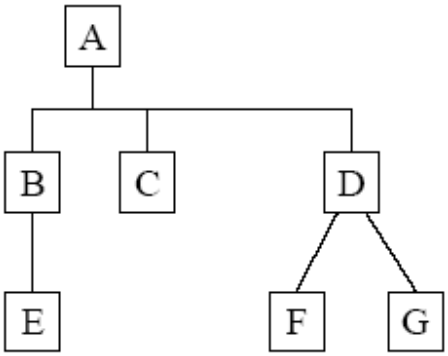


Figura 9 - Topologia di rete ad albero

Il traffico di dati va dai sistemi (o dai terminali) dei livelli più bassi verso i sistemi intermedi o verso il sistema del livello più alto. Quest'ultimo è in genere il sistema più potente dell'intera struttura, visto che deve provvedere alle richieste di tutta la rete. Spesso, esso è responsabile della gestione completa della rete, ma è anche possibile che ci sia un cooperazione, per la gestione ed il controllo della rete, tra il sistema principale e alcuni o tutti i sistemi del livello immediatamente inferiore: per esempio, a tali sistemi di livello inferiore possono essere affidati compiti gestionali specifici oppure limitati ad una specifica sottorete.

Per quanto riguarda le applicazioni residenti nei vari sistemi, ce ne sono alcune che interessano la generalità o quasi degli utenti nel sistema di livello più alto (nel senso che sono accessibili solo da questi), mentre altre applicazioni sono interesse sempre più locale man mano che si scende nella gerarchia.

La topologia a albero presenta fondamentalmente i seguenti inconvenienti:

- il sistema principale, se è sovraccarico di lavoro, può diventare un collo di bottiglia per l'intera rete, il che comporta un rallentamento dei servizi per tutti gli utenti;
- la caduta del sistema principale rende inoltre inutilizzabile l'intera rete.

A quest'ultimo inconveniente si può però ovviare adottando un **sistema di back-up**: bisogna cioè mettere in grado uno o più altri sistemi della rete di svolgere le stesse funzioni del sistema principale nel momento in cui questo dovesse venire a mancare.

Rete a stella

La configurazione a stella è simile a quella ad albero, con la fondamentale differenza che non c'è alcuna distribuzione funzionale, ossia non ci sono livelli diversi: in altre parole, tutte le funzioni riguardanti gli utenti periferici sono realizzate nel nodo centrale.
Lo schema è dunque il seguente:

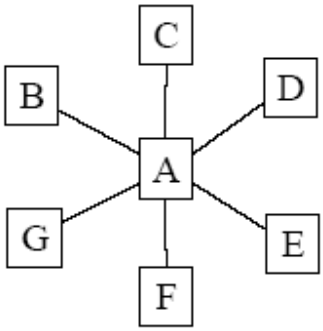


Figura 10 - Topologia di rete a stella

Questo topologia presenta, accentuati, gli stessi pregi e difetti della struttura ad albero.

Rete a dorsale

Questa configurazione è diventata popolare in quanto è adottata dalle reti locali di tipo **Ethernet**, delle quali si parlerà in seguito. La caratteristica è che c'è un unico cavo che collega tutte le stazioni, come nello schema seguente:

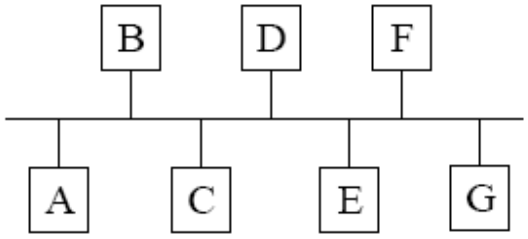


Figura 11 - Topologia di rete a dorsale

La trasmissione di una stazione viene ricevuta da tutte le altre. In qualche modo, è l'analogo del bus che viene usato nelle architetture dei moderni calcolatori: il *bus* è l'insieme di cavi elettrici che mettono in comunicazione tutti i dispositivi (CPU, memoria, periferiche) da cui il calcolatore è costituito.
Il vantaggio fondamentale della *configurazione a dorsale* è nel software per l'accesso, il quale, nel caso di rete locale, è davvero molto semplice.
I principali inconvenienti sono invece i seguenti:

- i potenziali problemi di prestazioni dovuti al fatto che unico cavo serve tutte le stazioni;
- una eventuale interruzione del cavo mette fuori uso l'intera rete;
- la mancanza di punti di concentrazione rende difficoltosa l'individuazione di eventuali punti di malfunzionamento.

Topologia ad anello (ring)
Questa configurazione è stata resa popolare dalle **LAN** di tipo *Token-Ring*. Essa è schematizzata nella figura seguente:

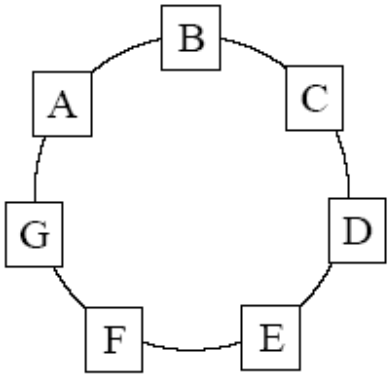


Figura 12 - Topologia di rete ad anello

La trasmissione è in questo caso unidirezionale (i dati viaggiano cioè solo in un senso), ma, essendo l'anello un circuito chiuso su se stesso, è possibile inviare un messaggio da qualsiasi stazione verso qualsiasi altra.
Un importante pregio di questa topologia è che apre ottime prospettive per l'utilizzo della **fibra ottica**.

Topologia a maglia
Quest'ultima topologia consiste nel collegare le varie stazioni con diversi circuiti, ad esempio come indicato nella figura seguente:

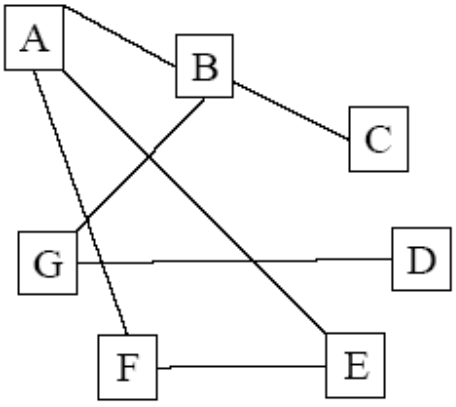


Figura 13 - Topologia di rete a maglia

Una topologia di questo tipo assicura buone prestazioni in quanto il traffico viene ripartito sui vari percorsi. Inoltre, essa conferisce una elevata affidabilità all'intera struttura, proprio grazie alla presenza di *percorsi multipli*.
Allo stesso tempo, però, i costi dei collegamenti possono anche essere elevati ed inoltre la gestione della struttura è chiaramente più complessa rispetto agli altri casi esaminati.

VELOCITÀ DI TRASMISSIONE

Quando avviene uno scambio di dati tra due computer, tali dati viaggiano sulla linea in forma di valori binari, ossia di sequenze di valori (i cosiddetti **bit**, che sta per *Binary digIT*, ossia "cifra binaria") 0 ed 1. Per esempio, la sequenza 11000001 rappresenta la lettera "A" secondo il **codice EBCDIC** adottato dalla IBM.

I bit vengono ovviamente trasmessi uno alla volta: da un punto di vista fisico, è possibile trasmettere una cifra binaria, su una linea telefonica, semplicemente alternando, tra due valori (che rappresentano i valori 0 ed 1), il voltaggio della corrente passante oppure interrompendo o meno la segnalazione o la direzione del flusso di corrente.

Ad ogni modo, a prescindere dal tipo di segnalazione binaria usata, una linea è caratterizzata dalla sua *portata*: si definisce **portata** di una linea il numero di bit al secondo (brevemente **bps**) che è possibile immettere su di essa.

Per esempio, supponiamo di avere un terminale capace di trasmettere a 4800 bps sulla linea cui è collegato: ciò significa che il terminale può inviare 4800 bit al secondo sulla linea. Allora, se il messaggio da trasmettere è lungo complessivamente 9600 bit, è chiaro che saranno necessari 9600/4800=2 secondi perché tale messaggio venga trasmesso.

E' chiaro che maggiore è la portata trasmissiva di una linea, più veloce è la trasmissione dei vari messaggi. Questo è il motivo per cui spesso si confondono, impropriamente, i termini "portata" della linea con la "velocità" della linea stessa.

I valori di portata per le normali linee telefoniche sono diversi a seconda che la linea sia commutata o dedicata o che si tratti di una linea di speciale qualità:

- le normali linee commutate vanno da un minimo di 600bps ad un massimo che attualmente è di 57600 bps;
- le linee dedicate raggiungono valori di 64000 bps e superiori;
- le linee espressamente progettate per la trasmissione digitale (tipicamente le *fibre ottiche*) sono in grado di arrivare anche a 2M bps e sono previsti ulteriori incrementi.

Tuttavia, anche il valore di 2M bps è molto basso se confrontato con quello delle comuni **LAN** (reti locali), senza poi considerare, ovviamente, le connessioni dirette tra un sistema e i suoi dispositivi.

I motivi per cui ci sono questi differenti valori di portata (o velocità) sono molteplici: sicuramente, le linee telefoniche, essendo progettate per la voce, presentano un tasso di disturbo che normalmente non è dannoso per la trasmissione vocale, mentre è causa di errori molto frequenti nella trasmissione binaria. E' per questo che vengono progettate **linee di qualità speciale**, adatte per la trasmissione digitale.

Ad ogni modo, si tenga presente che i segnali si degradano sempre, mentre si propagano nei mezzi trasmissivi, rispetto al segnale originario. Questo degrado, se supera un certo valore, rende il segnale originario irriconoscibile e porta quindi ad errori di trasmissione. I motivi fisici del degrado sono molteplici: citiamo la distanza della comunicazione, la velocità trasmissiva ed il tipo di conduttore usato.

Ecco dunque che *è consigliabile limitare, specialmente sulle lunghe distanze, la velocità trasmissiva.*

Grazie alla introduzione della fibra ottica al posto del normale cavo conduttore in rame, si è ottenuta una riduzione del degrado del segnale e si possono perciò raggiungere maggiori velocità di trasmissione.

TRASMISSIONE DIGITALE E MODULAZIONE

All'inizio del capitolo abbiamo detto che una rete di computer è costituita da due o più DTE collegati tra loro al fine di scambiarsi informazioni e condividere risorse: *la comunicazione*

tra due DTE avviene dunque scambiando dati digitali.
Il **flusso trasmissivo digitale** è continuo e ripetitivo così come quello analogico. Tuttavia, presenta una differenza fondamentale rispetto a quest'ultimo: mentre un **segnale analogico** può assumere tutti i possibili valori entro un intervallo prestabilito, un segnale digitale assume solo 2 valori discreti corrispondenti ai valori binari da trasmettere.
Questi valori discreti si ottengono facendo variare, nel modo quanto più brusco possibile, il valore del segnale da un livello all'altro. Uno schema ideale di segnale digitale è quello della figura seguente:

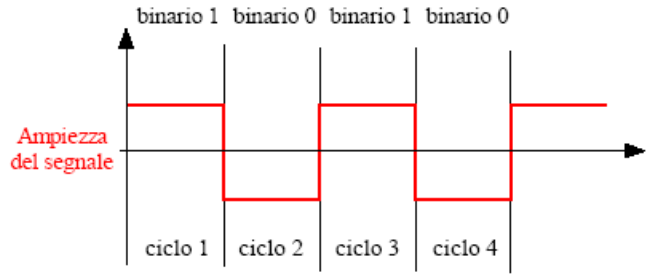


Figura 18 - Esempio di segnale digitale ideale (onda quadra)

Si tratta, cioè, di un'onda quadra che oscilla tra due valori: il valore "alto" corrisponde al valore binario 1, mentre il valore "basso" corrisponde al valore binario 0. Ovviamente, si è usato l'aggettivo *ideale* in quanto le transizioni tra un livello all'altro, nella realtà, non sono mai così brusche, ma avvengono sempre con una certa pendenza non nulla: l'impossibilità di renderle così brusche deriva sia dai limiti fisici dei dispositivi deputati a generare tali segnali sia anche dagli arrotondamenti dovuti ai disturbi dei quali si parlava nel paragrafo precedente e dovuti alle caratteristiche non ideali dei mezzi di trasmissione.
Nel caso della trasmissione digitale su linee telefoniche, cioè su linee non espressamente progettato per il transito di segnali digitali, è necessario ricorrere ad un particolare DCE che sia in grado di adattare la sequenza dei segnali digitali alle caratteristiche della trasmissione analogica, ossia, in definitiva, alle caratteristiche della linea da utilizzare: questo particolare DCE è il già citato **modem**.
Il termine *modem* è la contrazione di "*MOdulatore/DEModulatore*" e questo indica quali siano le funzioni principali del modem: supponiamo di avere due DTE collegati tra loro attraverso una linea di trasmissione; ciascun DTE è collegato alla linea mediante il proprio modem, come nella figura seguente:

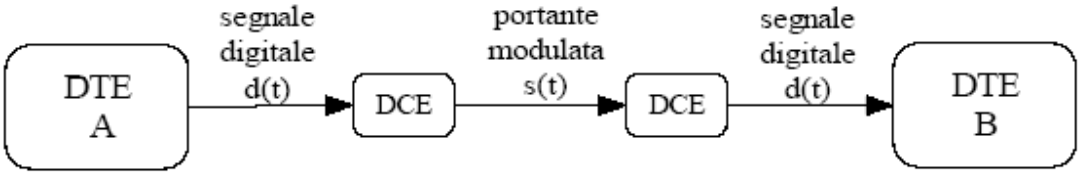


Figura 19 - Schema della trasmissione digitale con portante modulata

Supponiamo che il DTE A voglia inviare un messaggio al DTE B; si procede allora nel modo seguente:
- il DTE A invia il messaggio, sotto forma di segnale digitale, al proprio modem;
- il modem "*modula*" un segnale continuo sinusoidale, che prende il nome di **portante**, in modo che esso riproduca, istante per istante, le caratteristiche del segnale digitale emesso dal DTE A;
- la **portante modulata** generata dal modem A viene inviata sulla linea e giunge quindi al modem B;

- il modem B si comporta in modo inverso al modem A, nel senso che **demodula** il segnale, ossia ne tira fuori il segnale digitale emesso dal DTE A, e lo invia quindi al DTE B.

Abbiamo usato l'espressione *modulare una portante sinusoidale*. Vediamo di capire meglio di cosa si tratta. Con il termine **portante** noi indichiamo un segnale continuo che generalmente è di tipo sinusoidale, ossia un segnale del tipo

$$c(t) = A_C \cos(\omega t + \varphi)$$

Modulare questa portante in base al segnale digitale d(t) emesso dal DTE A significa variare una delle caratteristiche di questa portante (ampiezza A_C, frequenza $\omega/2\pi$ o fase φ) in modo proporzionale, in ciascun istante, al valore assunto, in quell'istante, da d(t). A seconda di quale caratteristica venga variata, noi abbiamo 3 possibili casi di modulazione:

- si parla di **`modulazione di ampiezza`** quando è l'ampiezza della portante che viene fatta variare in modo proporzionale al valore di d(t); il segnale modulato (o "portante modulata") generato dal modulatore è in questo caso del tipo

$$s(t) = d(t)A_C \cos(\omega t + \varphi)$$

In tal modo, se d(t) è un segnale che oscilla tra i valori +1 e -1, allora l'ampiezza di s(t) oscilla tra $+A_C$ e $- A_C$;

- si parla invece di **`modulazione di frequenza`** quando è la frequenza della portante che viene fatta variare in modo proporzionale al valore di d(t); se, per esempio, la frequenza della portante è di 2400 Hz (il che significa che l'onda portante compie 2400 oscillazioni al secondo), modulare secondo d(t) significa, per esempio, fare in modo che essa scenda al valore 1800Hz quando il valore di d(t) è 0 e salga al valore 3000 Hz quando il valore è 1; in tal modo, il modem che riceve la portante modulata s(t), estrae il valore 0 se, ad un passaggio, misura una frequenza di 1800 Hz, mentre estrae il valore 1 se la frequenza rilevata è di 3000 Hz;

- si parla infine di **`modulazione di fase`** quando è la fase della portante che viene fatta variare in modo proporzionale al valore di d(t).

Le tecniche di modulazione possono coinvolgere segnali modulanti sia di tipo analogico sia di tipo digitale, dove ricordiamo ancora che per **segnale analogico** intendiamo un segnale che può assumere qualsiasi valore all'interno di un certo intervallo, mentre per **segnale digitale** intendiamo sempre un segnale analogico, che però può assumere solo due diversi valori. Quando il segnale modulante è di tipo digitale (per cui siamo nell'ambito della **modulazione digitale**), mentre la modulazione d'ampiezza può essere fatta con soli due livelli di discontinuità, la modulazione di fase e la modulazione di frequenza permettono di scegliere più livelli di discontinuità. Senza addentrarci nella spiegazione di questa affermazione, limitiamoci a citare un esempio: nella modulazione di fase, è possibile fare in modo che la fase della portante modulata assuma 4 diversi valori: a 4 diversi valori è possibile associare 4 diverse combinazioni di 2 bit (precisamente 00, 01, 10 e 11), il che significa che viene aumentato il numero di bit che è possibile trasmettere con ogni variazione del segnale. Se i valori della fase fossero 16, ogni singola variazione del segnale in arrivo al modem ricevente corrisponderebbe a 4 bit emessi dal DTE sorgente. Questo consentirebbe di trasmettere in linea lo stesso numero di cicli al secondo, ma a ciascun ciclo sarebbero associati 4 bit, il che significa che la velocità di trasmissione (in bps) diventa 4 volte più grande.

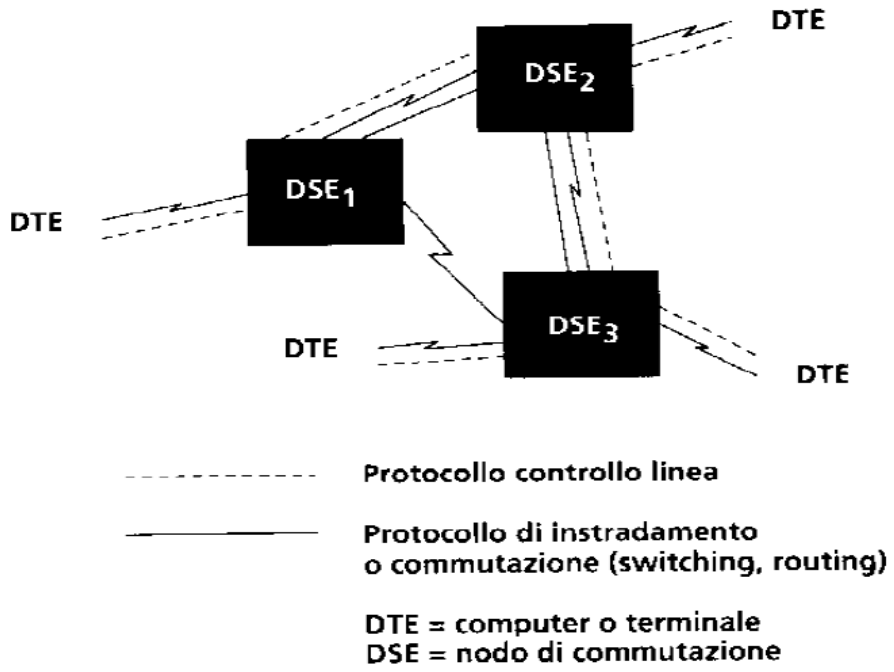
In generale, se n sono i livelli possibili (per la fase o per la frequenza), la quantità $\log_2 n$ rappresenta il numero di bit che è possibile trasmettere con un'unica variazione del segnale.

In effetti, le maggiori velocità che si ottengono attualmente su linee analogiche sono quasi sempre dovute alla perfezionata tecnologia dei modem in grado di trasmettere su livelli multipli.

Relazioni tra terminali e computer

PROTOCOLLI PER LA COMUNICAZIONE TERMINALE-COMPUTER

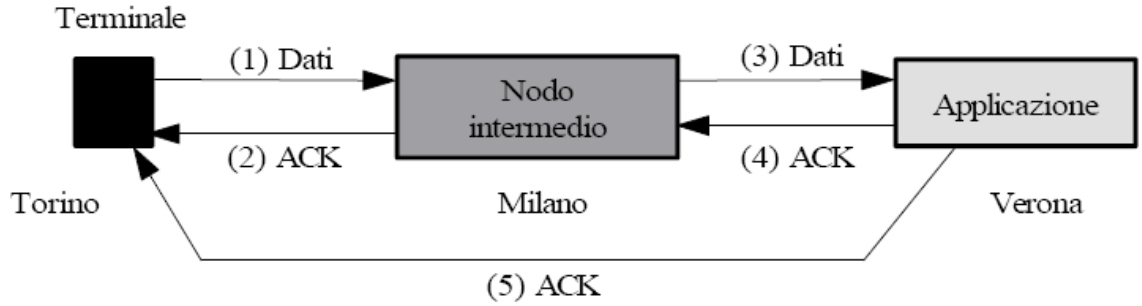
Richiamiamo che *un protocollo è una* serie di norme, convenzioni e tecniche per lo scambio di dati, di comandi e di informazioni di controllo tra due DTE.
Esistono molti livelli di protocolli: si va dal livello più basso, che regola il modo di trasmettere i segnali binari sulla linea (protocollo di connessione), al livello più alto, che invece indica come interpretare dati e comandi a livello applicativo, passando per una serie variabile di ulteriori livelli.
Nella interazione tra le stazioni di una rete vanno utilizzati vari tipi di protocolli. Consideriamo il caso di una trasmissione di dati tra due stazioni: la situazione più semplice è quella in cui le 2 stazioni si trovano agli estremi di una singola linea; in questo caso, è sufficiente un protocollo di linea per regolare il flusso tra le due stazioni; diverso è il caso in cui le 2 stazioni sono connesse mediante più linee oppure fanno parte di una *rete magliata* (come quella della figura seguente), nel qual caso potrebbero comunicare attraverso percorsi (o strade, o route, costituite da linee e nodi da attraversare) diversi:



*Rete magliata costituita da 4 computer (DTE) connessi tramite 3 nodi di commutazione (DSE, detti anche **switch**). Il protocollo che sovrintende al passaggio di dati sulla singola linea è un protocollo di linea, mentre quello che sovrintende al collegamento tra DTE sorgente e DTE destinazione è un protocollo di instradamento (o di commutazione).*

Una volta individuata la stazione (DTE) destinazione, bisogna stabilire quale strada usare per connetterla alla stazione (DTE) sorgente. Questa scelta compete al cosiddetto *protocollo di instradamento* (**routing protocol**) che quindi si aggiunge al *protocollo di linea* necessario al passaggio di dati su ciascuna linea. In altre parole, solo dopo la scelta del percorso interviene il protocollo di linea per la gestione dei singoli collegamenti. Tale protocollo viene usato tante volte quante sono le linee che costituiscono il percorso fissato.

C’è poi un ulteriore livello superiore di protocollo. Per illustrarlo, consideriamo la figura seguente, in cui è presente un terminale, situato fisicamente a Torino, che intende connettersi ad una applicazione situata fisicamente a Verona:



Conferme da applicazione (Verona) ad utente (Torino) mediante protocollo di transport. I numeri tra parentesi indicano la sequenza temporale dei messaggi: (1) dati dal terminale al nodo intermedio (2) conferma dal nodo intermedio all’utente (3) dati dal nodo intermedio all’applicazione (4) conferma dall’applicazione al nodo intermedio (5) conferma dell’applicazione all’utente

L’unica possibilità perché il terminale di Torino comunichi con l’applicazione di Verona è quella di passare attraverso il *nodo intermedio* situato a Milano. Non si pone dunque il problema della scelta del percorso, essendo presente 1 sola possibilità.
Il terminale di Torino invia un messaggio per il terminale di Verona e lo fa usando un protocollo di linea; tale protocollo, comunque sia stato pensato, prevede una risposta da parte della stazione ricevente sull’esito positivo o negativo della trasmissione. Tuttavia, il protocollo di linea effettua la trasmissione solo fino al nodo intermedio di Milano, per cui è quest’ultimo che effettua il controllo di correttezza della trasmissione. Supponiamo allora che non ci siano stati errori: in questo caso, il nodo di Milano risponde con una conferma positiva (detta **ACK**, che sta per *ACKnowledgement*) che viene inviata a Torino. Questo messaggio significa semplicemente che, a livello di linea, la trasmissione è andata bene. Non ha però niente a che vedere con l’esito dell’operazione complessiva: infatti, lo stesso messaggio che da Torino è arrivato a Milano, deve ora andare a Verona.
Supponiamo che anche su questa seconda tratta non si verifichino errori, per cui l’applicazione di Verona invia un ACK al nodo di Milano. Questo secondo ACK, unito a quello Milano→Torino, significa di fatto che tutto è andato bene, ma non arriva a Torino, in quanto è un messaggio a livello del protocollo di linea, che quindi si ferma a Milano. A questo punto, il terminale di Torino non sa ancora se il proprio messaggio è effettivamente arrivato a Verona e, se sì, con o senza errori. E’ allora necessario un protocollo di livello superiore che invii un ACK direttamente da Verona a Torino, cioè da destinazione a sorgente. Questo è il cosiddetto **protocollo di transport**.
La differenza è dunque evidente: un *protocollo di linea*, che agisce sulle singole tratte, è di tipo **box-to-box**, mentre un *protocollo di transport* è di tipo **end-to-end**.
Viene subito da pensare che i *protocolli end-to-end* possano comportare un traffico maggiore sulla rete: infatti, se il protocollo di transport aggiungesse, ai dati dell’utente, dei messaggi dedicati (che includano appunto un ACK o l’analogo negativo NCK), il numero di informazioni in transito sarebbe sicuramente maggiore. Si ovvia allora a questo inconveniente inserendo le informazioni end-to-end in specifici campi di messaggi contenenti `anche dati d’utente.` Quando questo è possibile (e non sempre lo è), l’esito è effettivamente di non appesantire la rete.

TIPI DI RETE

Vi sono due tipi fondamentali di reti, dalle caratteristiche molto diverse: le **LAN** (*Local Area* Network) e le **WAN** (Wide Area Network).

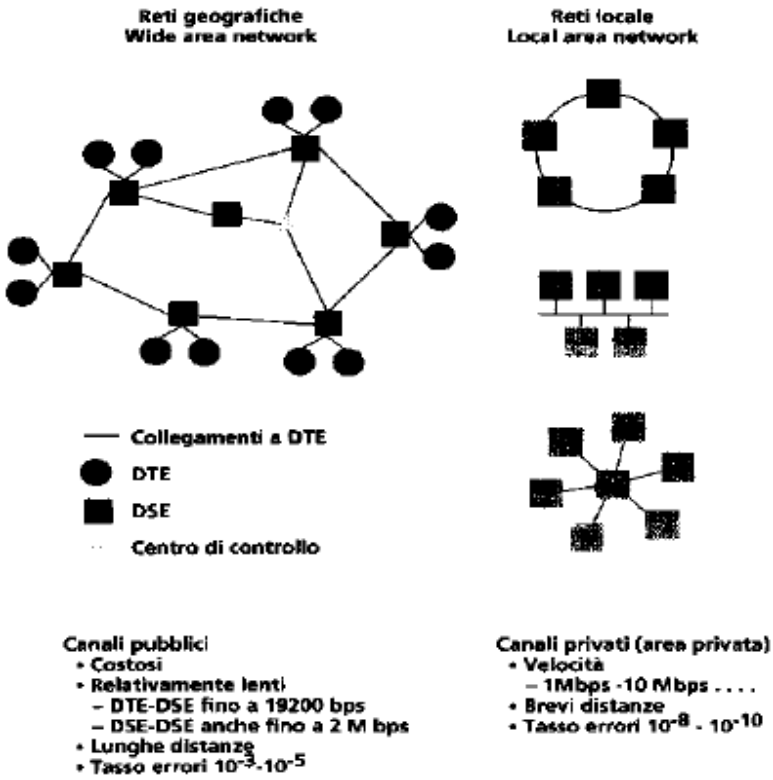
Per le WAN possiamo citare le seguenti caratteristiche generali:

- collegano diversi sistemi elaborativi, spesso distanti centinaia o anche migliaia di chilometri (per cui si parla di **reti geografiche**);
- spesso il numero di terminali collegati è molto elevato (dell'ordine delle migliaia);
- hanno spesso una **struttura a maglia** ed una configurazione dei collegamenti a volte complessa;
- le linee vengono affittate dal gestore pubblico, per cui si tende ad ottimizzarne lo sfruttamento, nei limiti delle possibilità tecnologiche e normative, collegando quanti più DTE possibile sulla stessa linea; in questi casi, la struttura a maglia serve a garantire strade alternative nel caso di indisponibilità di qualche componente o per ripartire il traffico su più percorsi;
- in alternativa alla configurazione a maglia, possono avere una topologia che fa capo ad un sistema principale (**mainframe**), dal quale partono diverse linee, dirette o a stazioni terminali (**host**) oppure a nodi intermedi (**switch**); a loro volta, i nodi intermedi hanno altre linee che vanno verso stazioni utente o altri componenti di livello inferiore;
- utilizzano linee che, date le notevoli distanze, operano spesso a bassa velocità; si tratta inoltre di linee con *tasso d'errore*[1] spesso non trascurabile.

Passando alle LAN, abbiamo invece le seguenti caratteristiche:

- i canali sono privati e non escono perciò dall'ambito di un'area privata; di conseguenza, queste reti hanno una estensione massima dell'ordine di decine di km; i costi principali sono dunque quelli delle apparecchiature, mentre sono trascurabili quelli relativi alle linee stesse;
- usano velocità molto maggiori rispetto alle WAN: in generale, si può dire che esiste almeno un ordine di grandezza in più per la velocità delle LAN rispetto alla velocità delle WAN;
- hanno alta affidabilità e quindi bassissimo tasso di errore.

Nella figura seguente sono indicate le configurazioni di rete più comuni:

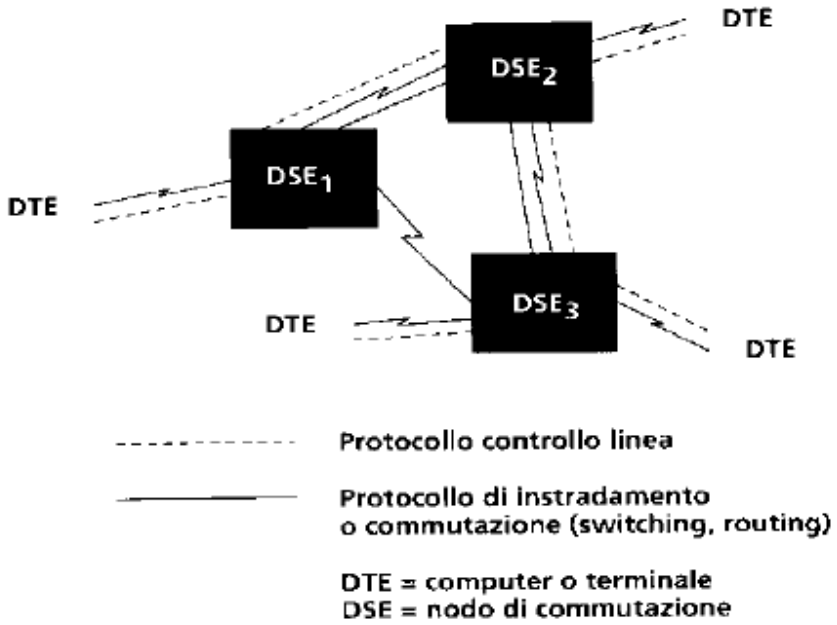


Una caratteristica comune di WAN e LAN riguarda il modo con cui può avvenire la comunicazione tra due DTE della RETE. Ci sono infatti due modi:

- il primo modo è detto **connection oriented** (*orientato alla connessione*): in questo caso, i due DTE, prima di effettuare lo scambio di dati, si assicurano della presenza reciproca in linea; fatta questa verifica, viene instaurata la *connessione* (o *colloquio* o *sessione*), la quale dura per tutto il tempo necessario allo scambio dati; non appena tale scambio è terminato, anche la connessione viene abbandonata. La connessione è continuamente gestita dal software dei due DTE, il quale svolge diverse funzioni: gestione del ritmo di interscambio (quindi essenzialmente della velocità di trasmissione), controllo delle regole dello scambio, capacità di interrompere la controparte (ad esempio quando c'è da inviare un messaggio urgente), controllo degli errori ed eventuale loro correzione. Tutti questi controlli assumono importanza critica nelle WAN, data la bassa affidabilità delle linee;
- il secondo modo è detto **connectionless mode** (*non orientato alla connessione*): in questo caso, un DTE può inviare un messaggio all'altro DTE anche se questo non è presente in linea; è come affidare le lettere alla posta, sperando che vengano consegnate. Il vantaggio è che non sono necessari servizi di controllo o di supporto, il che può essere vantaggioso per le LAN, mentre non è molto opportuno per le WAN, per i citati problemi di scarsa affidabilità.

Il problema principale del *connectionless mode* riguarda il controllo degli errori che, sia pure raramente, possono verificarsi: infatti, non essendoci controlli immediati durante la trasmissione, il DTE sorgente non può sapere come è andata la trasmissione. D'altra parte, l'onere dei controlli ripetitivi spesso diventa inutile sulle reti ad alta affidabilità, dove gli errori sono decisamente pochi. La soluzione cui si può pensare è allora quella di affidare il controllo degli errori direttamente alle applicazioni, il che alleggerisce i protocolli di linea, che possono occuparsi solo del trasporto dei dati, nonché anche i nodi intermedi, che devono occuparsi sono di instradare i dati sui percorsi desiderati.

Quest'ultimo concetto è di importanza cruciale. Consideriamo infatti nuovamente lo schema:



*Rete magliata costituita da 4 computer (DTE) connessi tramite 3 nodi di commutazione (DSE, detti anche **switch**).*

Supponiamo che due DTE della rete vengano posti in comunicazione tramite due nodi intermedi, ad esempio DSE_1 e DSE_2. Se affidiamo il controllo degli errori ai protocolli di linea, in pratica imponiamo che ciascun DSE, ricevendo un *pacchetto* di dati, ne controlli sempre la correttezza. non ci sono errori, il pacchetto viene instradato, altrimenti viene verosimilmente inviato al mittente (che può essere la stazione sorgente oppure un DSE precedente) un messaggio che richieda la ritrasmissione. Ma, se il collegamento è ad alta velocità, il DSE non può concedersi il lusso di effettuare questi controlli; l'unica sua funzione deve essere quella di prendere i dati in arrivo ed instradarli dove necessario, senza operazioni intermedie (o comunque senza operazioni intermedie di eccessiva complessità). Da qui l'opportunità di demandare alle applicazioni il controllo degli errori, lasciando ai DSE solo compiti marginali, eseguibili mediante *circuiti dedicati* facilmente realizzabili e soprattutto molto veloci. Questi problemi rientrano nel vasto campo di problemi di **controllo di congestione del flusso** di una rete di telecomunicazioni.

Reti locali (LAN)

Le reti locali (***Local Area Network, LAN***), in genere:

- sono possedute da una organizzazione (reti private);
- hanno un'estensione che arriva fino a qualche km;
- si distendono nell'ambito di un singolo edificio o campus (non si possono, di norma, posare cavi sul suolo pubblico);
- sono usatissime per connettere PC o workstation.

Esse si distinguono dagli altri tipi di rete per tre caratteristiche:

- ***dimensione***: la dimensione non può andare oltre un certo limite, per cui si può agevolmente calcolare a priori il tempo di trasmissione nel caso peggiore. Questa conoscenza permette di utilizzare delle tecniche particolari per la gestione del canale di comunicazione;
- ***tecnologia trasmissiva***: le LAN sono in generale *reti 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;

- ***topologia***: sono possibili diverse topologie, le più diffuse sono il ***bus*** ed il ***ring***;
- topologia bus:

in ogni istante solo un elaboratore può trasmettere, mentre gli altri devono astenersi, in maniera del tutto analoga a quanto avviene in un singolo calcolatore, dove il busè a disposizione di un dispositivo (CPU o periferica) per volta;

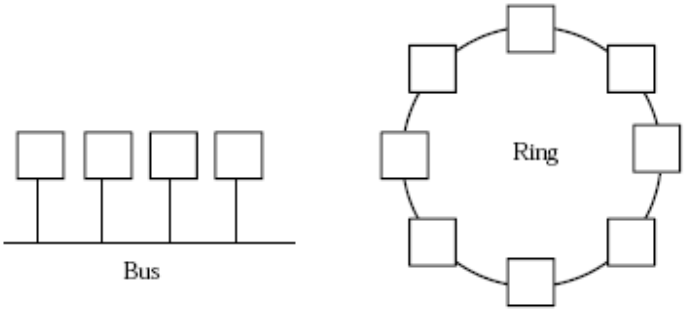
è necessario un meccanismo di ***arbitraggio*** per risolvere i conflitti quando due o piùelaboratori vogliono trasmettere contemporaneamente;

l'arbitraggio può essere centralizzato o distribuito;

- topologia ring:

in un ring ogni bit circumnaviga l'anello;

anche qui è necessario un meccanismo di arbitraggio (spesso basato sul possesso di un ***gettone*** , detto anche ***token***, che abilita alla trasmissione);



Topologie di rete a bus oppure a ring per una rete locale (LAN)

Per quanto riguarda, specificamente, le **reti locali di tipo broadcast**, esse possono essere classificate a seconda del meccanismo scelto per l'arbitraggio:

- ***allocazione statica***: le regole per decidere chi sarà il prossimo a trasmettere sono fissate a priori, ad esempio assegnando un *time slot* ad ogni elaboratore (tecnica **TDM**, *Time Division Multiplexing*). Lo svantaggio è rappresentato dallo spreco dei time slot assegnati a stazioni che non devono trasmettere;
- ***allocazione dinamica***: si decide di volta in volta chi sarà il prossimo a trasmettere, il che significa, ancora una volta, che è necessario un *meccanismo di arbitraggio delle contese*, che può essere:
- ***arbitraggio centralizzato***: un apposita apparecchiatura (ad esempio una *bus arbitration unit*) accetta richieste di trasmissione e decide chi abilitare;
- ***arbitraggio distribuito***: ognuno decide per conto proprio, il che presuppone, ovviamente, la presenza di strumenti per evitare un prevedibile caos.

Reti metropolitane (MAN)

A metà tra le LAN e le WAN si collocano le **reti metropolitane** (*Metropolitan Area Network*, ***MAN***), che hanno un'estensione tipicamente urbana (quindi anche molto superiore a quella di una LAN) e sono generalmente pubbliche (cioè un'azienda, ad es. *Telecom Italia*, mette la rete a disposizione di chiunque desideri, previo pagamento di una opportuna tariffa).

Fino a qualche anno fa erano basate essenzialmente sulle tecnologie delle reti geografiche, utilizzate però su scala urbana. Recentemente, e' stato invece definito un apposito standard, lo ***IEEE 802.6*** o ***DQDB*** (***Distributed Queue Dual Bus***), che e' effettivamente utilizzato in varie realizzazioni.

Sostanzialmente, lo standard DQDB prevede un mezzo trasmissivo di tipo broadcast a cui tutti i computer sono attaccati.

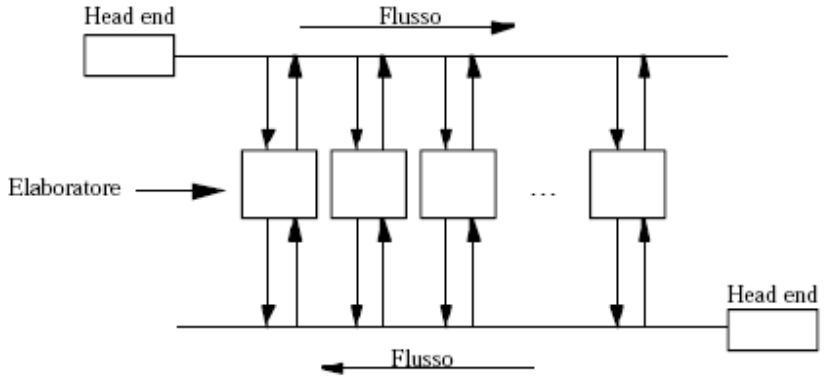


Figura 4 - Distributed Queue Dual Bus

Ogni bus (cavo coassiale o fibra ottica) e' unidirezionale, ed ha una ***head-end*** che cadenza l'attivita' di trasmissione.

Reti geografiche

Le **reti geografiche** (*Wide Area Network, **WAN***) si estendono a livello di una nazione, di un continente o dell'intero pianeta. Una WAN è tipicamente costituita da due componenti distinte:

- un insieme di elaboratori (***host*** oppure ***end system***) sui quali girano i programmi usati dagli utenti;
- una ***comunication subnet (***o ***subnet)***, che connette gli *end system* fra loro. Il suo compito è trasportare messaggi da un *end system* all'altro, così come il sistema telefonico trasporta parole da chi parla a chi ascolta.

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

- linee di trasmissione (dette anche circuiti, canali, trunk):
- ***elementi di commutazione*** (***switching element***): gli elementi di commutazione 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. Non esiste una terminologia standard per identificare gli elementi di commutazione. Termini usati sono:
- sistemi intermedi;
- nodi di commutazione pacchetti;
- ***router*** (quello che utilizzeremo noi).

Una tipica WAN è utilizzata per connettere più LAN fra loro:

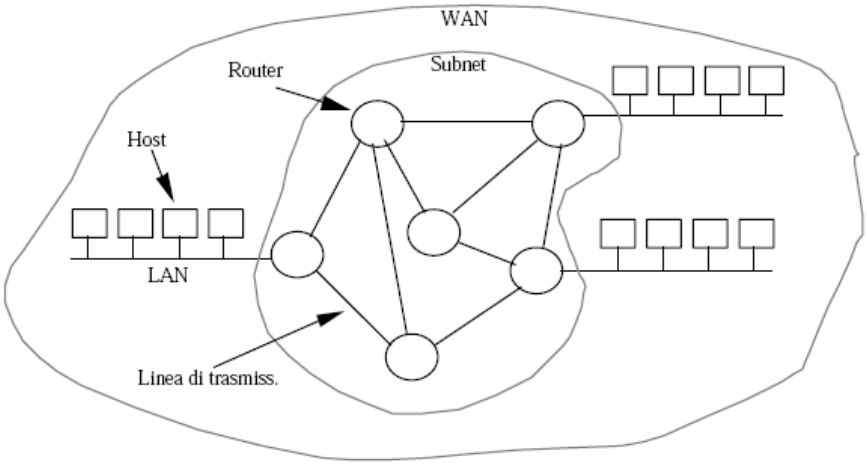


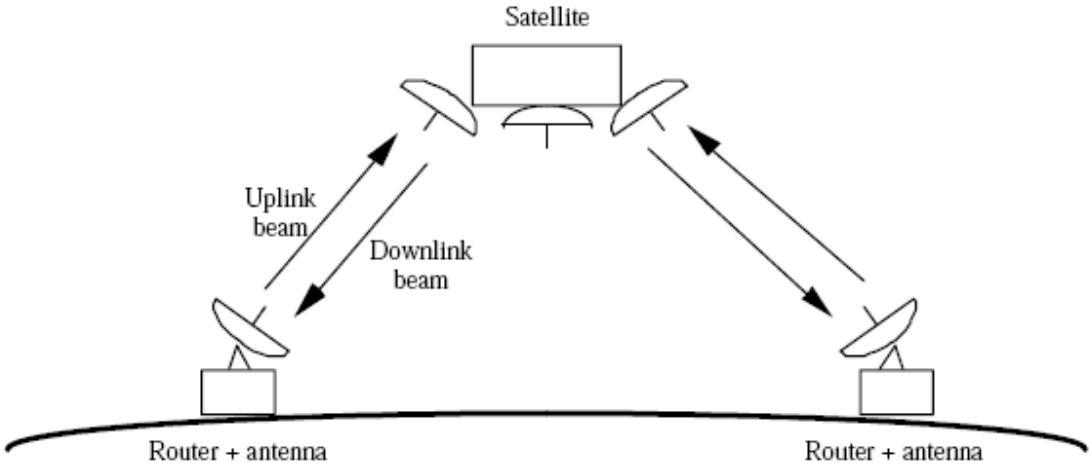
Figura 5 - Struttura tipica di una WAN

In generale, una WAN contiene numerose linee (spesso telefoniche) che congiungono coppie di **router**. Ogni router, in generale, deve:
1. ricevere un pacchetto da una linea in ingresso;
2. memorizzarlo per intero in un buffer interno;
3. appena la necessaria linea in uscita è libera, instradare il pacchetto su essa.

Una subnet basata su questo principio si chiama:
- punto a punto;
- store and forward;
- a commutazione di pacchetto (packet switched).

Molte topologie di interconnessione possono essere impiegate fra i router:
- a stella;
- ad anello;
- ad albero;
- magliata.

Tutte queste topologie sono state descritte in precedenza.
Un'altra possibilità è una WAN basata su **satellite** oppure **radio al suolo**:
- **Satellite**: ogni router *sente* l'output del satellite e si *fa sentire* dal satellite. Questo significa l'esistenza di due possibilità:
- *broadcast downlink* (cioè dal satellite a terra);
- *broadcast uplink* (cioè da terra al satellite) se i router possono "sentire" quelli vicini, point to point altrimenti.



Interconnessione di router via satellite

- **Radio al suolo**: ogni router sente l'output dei propri vicini (entro una certa distanza massima): anche qui siamo in presenza di una rete broadcast.

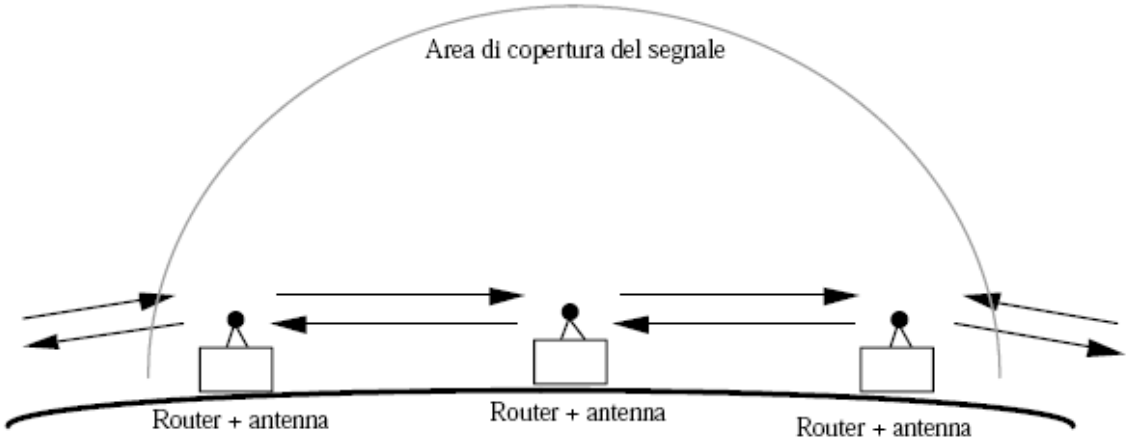
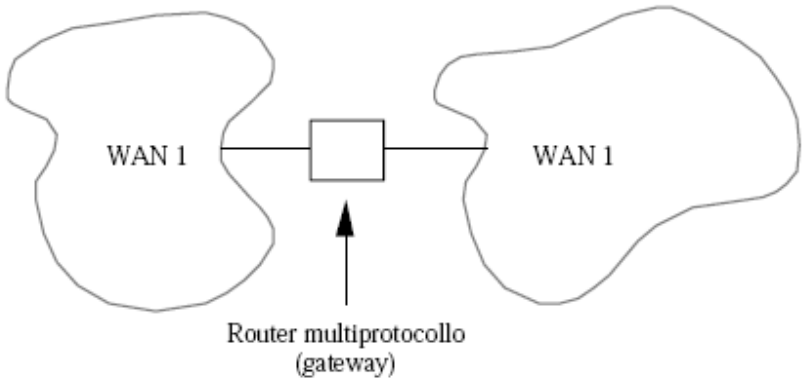


Figura 7 - interconnessione di router via radio al suolo

Una WAN può essere anche realizzata in maniera mista: in parte cablata, in parte basata su radio o satellite.

INTERCONNESSIONE DI RETI (INTERNETWORK)

Una ***internetwork*** è formata quando reti diverse (sia LAN che MAN o WAN) sono collegate fra loro. A prima vista, almeno in alcuni casi, la cosa è apparentemente uguale alla definizione di WAN vista precedentemente. Alcuni problemi, però, sorgono quando si vogliono connettere fra di loro reti progettualmente diverse (spesso incompatibili fra loro). In questo caso si deve ricorrere a speciali attrezzature, dette **gateway** (o ***router multiprotocollo***), che, oltre ad instradare i pacchetti da una rete all'altra, effettuano le operazioni necessarie per rendere possibili tali trasferimenti.



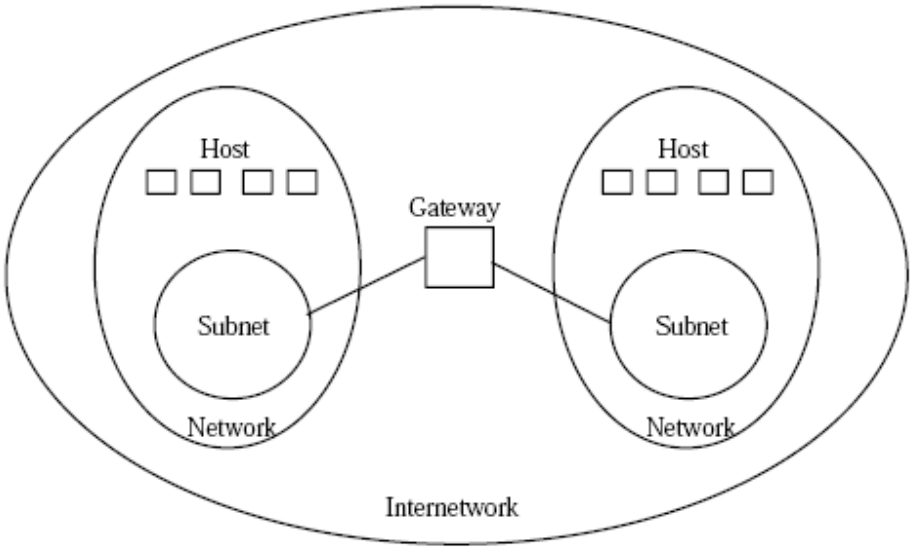
Interconnessione di reti WAN

E’ bene sottolineare una differenza importante

- **internet** (con la *i* minuscola) è sinonimo di internetwork, cioè la interconnessione di più reti generiche;
- **Internet** (con la *I* maiuscola) per riferirci alla specifica internetwork, basata su protocollo **TCP/IP**, che ormai tutti conoscono.

Bisogna inoltre evirare la confusione sui seguenti termini:

- **sottorete (subnet)**: nel contesto di una WAN è l'insieme dei router e delle linee di trasmissione;
- **rete (network)**: è l’insieme costituito da una subnet e da tutti gli host collegati;
- **internetwork**: è una collezione di più network, anche non omogenee, collegate per mezzo di gateway.

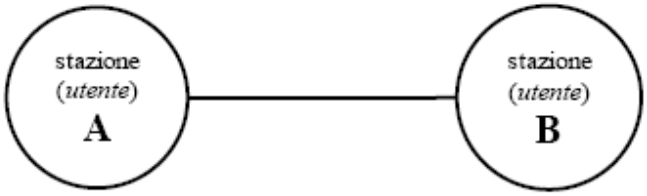


relazioni fra subnet, network e internetwork

LE RETI LAN

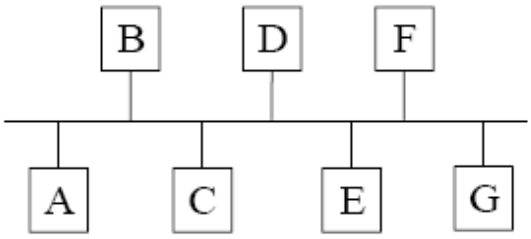
INTRODUZIONE AI PROTOCOLLI DI ACCESSO MULTIPLO

I *protocolli di linea* che abbiamo considerato nei paragrafi precedenti erano specificamente relativi a **connessioni punto-a-punto**, nelle quali cioè si considerano due sole stazioni, A e B, connesse tramite una linea di trasmissione:



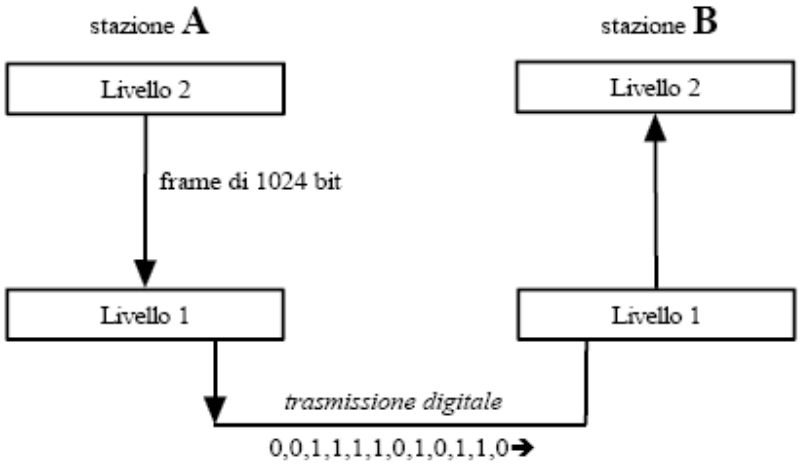
Generica connessione punto-a-punto tra due stazioni

In questo contesto, si trattava solo di analizzare le tecniche con cui le due stazioni devono scambiarsi i dati ed il problema principale era quello di ottimizzare l'uso della risorsa di comunicazione assegnata alle due stazioni.
Il problema invece si complica quando abbiamo più stazioni che devono comunicare tra loro e, per farlo, hanno a disposizione un'unica risorsa trasmissiva. Un esempio banale può essere una rete fatta nel modo seguente:



Topologia di rete a dorsale

Come diremo tra poco, questa è una rete cosiddetta **a dorsale**: c'è un unico mezzo trasmissivo (tipicamente su cavo), cui sono connesse un certo numero di stazioni. Dato che il mezzo trasmissivo accetta una sola trasmissione per volta, è necessario stabilire con quali criteri una data stazione può acquisire il diritto a trasmettere, prevalendo sulle altre.
A questo scopo, riprendiamo per un attimo lo schema generale di due stazioni connesse secondo il *modello di riferimento ISO-OSI*, limitandoci solo ai due livelli più bassi (livello 1 per l'*interfaccia fisica* e livello 2 per i *protocolli di linea*):



GENERALITÀ SUI PROTOCOLLI DI ACCESSO MULTIPLO

Dovendo gestire l'accesso di più stazioni ad un'unica risorsa trasmissiva, si può farlo sostanzialmente in due modi:

- il primo modo è quello dei **protocolli ad accesso multiplo casuale (con contesa)**: genericamente, una rete che funzioni con questo tipo di protocolli prevede che una stazione, appena abbia qualcosa da trasmettere, acceda immediatamente al canale, disinteressandosi del fatto che il canale poteva essere già occupato; se il canale era già occupato, dato che la trasmissione della stazione va inevitabilmente a sovrapporsi (fenomeno della **collisione**) alla trasmissione che già era in corso, entrambe le trasmissioni risultano indecifrabili.

Questo è evidentemente un caso estremo. Più realisticamente, si può invece pensare di imporre ad una stazione di "osservare" il canale prima di trasmettere e di trasmettere solo se il canale risulta libero; in questo caso, le collisioni, pur essendo possibili (vedremo in quali casi), sono sicuramente ridotte: si avrà collisione, ad esempio, quando due stazioni tentano contemporaneamente di accedere al canale (esse lo vedono libero e quindi trasmettono, senza sapere che qualcun altro sta facendo la stessa cosa e quindi i dati andranno persi) oppure quando una stazione A "osserva" il canale e lo vede libero solo perché il segnale già inviato da un'altra stazione B non è arrivato ancora al punto in cui A è connessa;

- il secondo modo è invece quello dei **protocolli ad accesso multiplo ordinato (senza contesa)**: in questo caso, le collisioni vengono del tutto evitate semplicemente ordinando l'accesso, ossia stabilendo un preciso criterio per cui ciascuna stazione viene autorizzata a trasmettere solo in dati momenti, nei quali ovviamente nessun'altra sta trasmettendo.

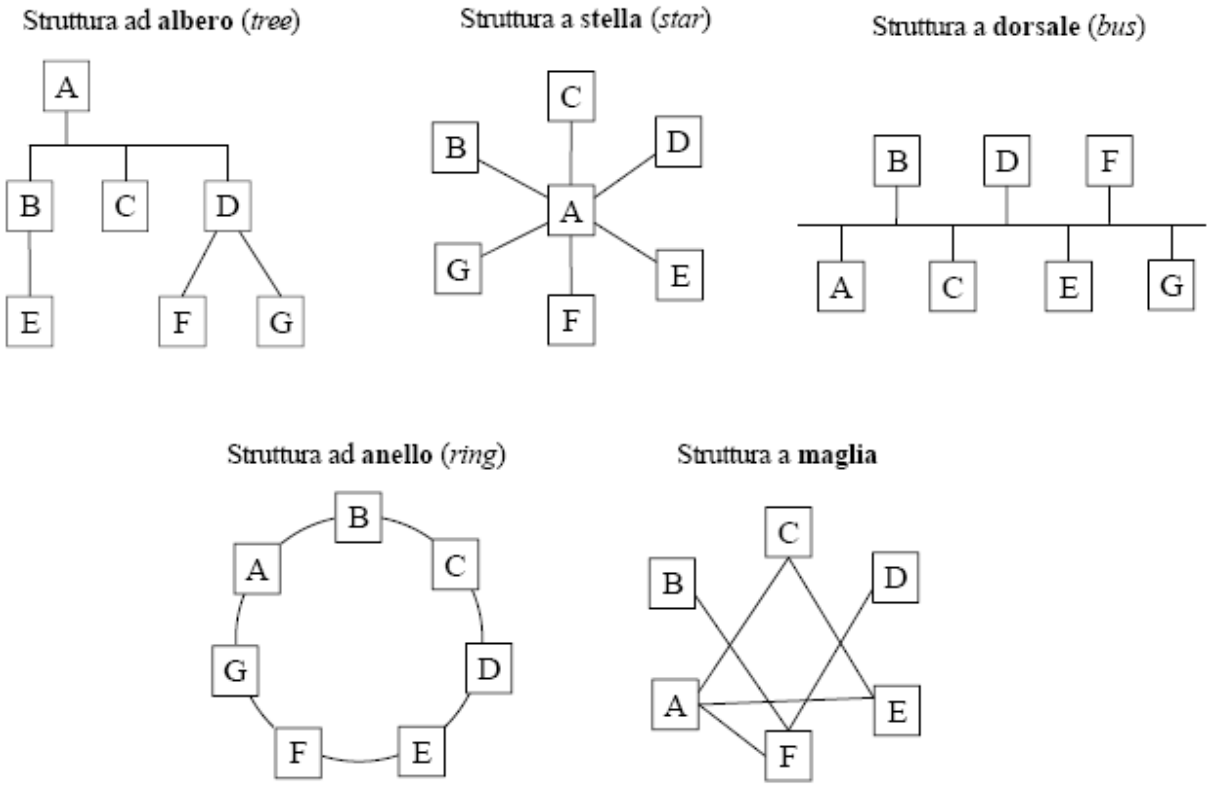
Come vedremo, queste sono caratteristiche solo generali dei vari protocolli, in quanto le varianti storicamente proposte (alcune standardizzate e diffuse con successo, altre standardizzate ma senza riscontrare grande diffusione) sono parecchie.

TOPOLOGIA DELLE RETI LOCALI

Prima di addentrarci nello studio dei singoli protocolli di accesso multiplo, è opportuno fare una panoramica delle principali topologie di rete. A tal proposito, diamo subito la seguente definizione: prende il nome di **topologia di rete** la configurazione geometrica dei collegamenti tra le varie stazioni (in generale i vari componenti) della rete.

La topologia di una rete di telecomunicazioni rappresenta una delle scelte fondamentali nella progettazione della rete stessa, specialmente se si tratta di una rete locale (**LAN**). La topologia determina infatti le dimensioni e la "forma" di una rete, con particolare riferimento al numero massimo di stazioni collegabili, al numero di linee di interconnessione ed alla lunghezza complessiva del cavo utilizzato. La topologia influenza inoltre i costi, l'affidabilità, l'espandibilità e la complessità della rete.

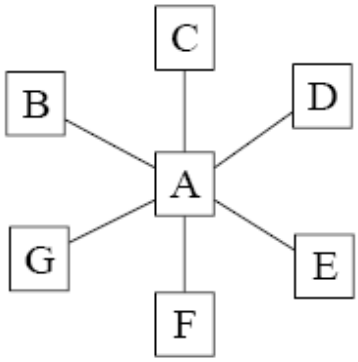
Ci sono essenzialmente 5 tipi di tipologie di rete, rappresentate nella figura seguente, ed un numero piuttosto grande di varianti:



Principali topologie di rete

Tra tutte le tipologie, come vedremo, tre hanno avuto ampia accettazione di mercato: sono le strutture a stella, ad anello e a dorsale. Accettazione anche buona c'è stata per la topologia ad albero. Per quanto riguarda, invece, la struttura a maglia, essa è tipicamente usata solo nelle reti geografiche (**WAN**),

Topologia a stella
Consideriamo per prima la topologia a stella:

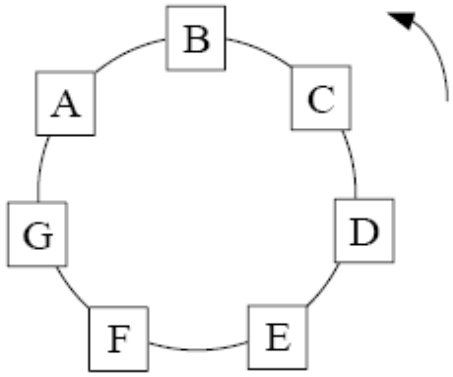


Topologia di rete a stella
Tutte le stazioni sono collegate ad una stazione centrale (**centro-stella**) e tali connessioni sono di tipo punto-a-punto. Si adotta una scelta di questo tipo tipicamente quando si vuole mantenere un controllo centrale di tutte le connessioni tra coppie di interlocutori.
Tipici esempi di topologie di questo tipo sono le centrali telefoniche o i sistemi di smistamento messaggi.
I vantaggi della topologia a stella sono essenzialmente i seguenti:
- *alte prestazioni*: essendo i collegamenti di tipo punto-a-punto, non c'è mai contesa sul mezzo trasmissivo, il quale quindi, a differenza di altre soluzioni, è praticamente sempre disponibile per una stazione che voglia trasmettere;
- *semplicità di protocollo*, per lo stesso motivo di cui al punto precedente;
- *facilità di controllo*: il controllo è tutto concentrato (centralizzato) in un unico punto della rete, che è appunto il *centro stella*;
- l'eventuale andata fuori uso di una stazione (che non sia ovviamente il centro stella) non ha alcuna influenza sul funzionamento della rete.

A fronte di questi vantaggi, ci sono i seguenti svantaggi:
- in caso di intenso traffico, il nodo centrale può risultare sovraccaricato di lavoro e questo potrebbe portare al blocco delle richieste di connessione. Questo rischio può anche essere accettato nel sistema telefonico, ma non può esserlo di certo in una rete di trasmissione dati;
- analogamente, l'affidabilità dell'intero sistema dipende tutta dall'affidabilità del componente centrale: si parla di **single point of failure**, nel senso appunto che l'andata fuori servizio del componente centrale compromette il funzionamento dell'intera rete.

Essenzialmente, possiamo affermare che il pregio principale di questa topologia è nel controllo centralizzato. Proprio per questo, vedremo che *reti locali basate su altri tipi di topologie, mantengono tali topologie solo a livello logico, mentre a livello fisico le connessioni rispettano comunque una topologia a stella.*

Topologia ad anello
Nella topologia ad anello, tutte le stazioni sono collegate in una caratteristica configurazione circolare, chiusa su se stessa, nella quale le stazioni sono tra loro collegate tramite linee punto-apunto:

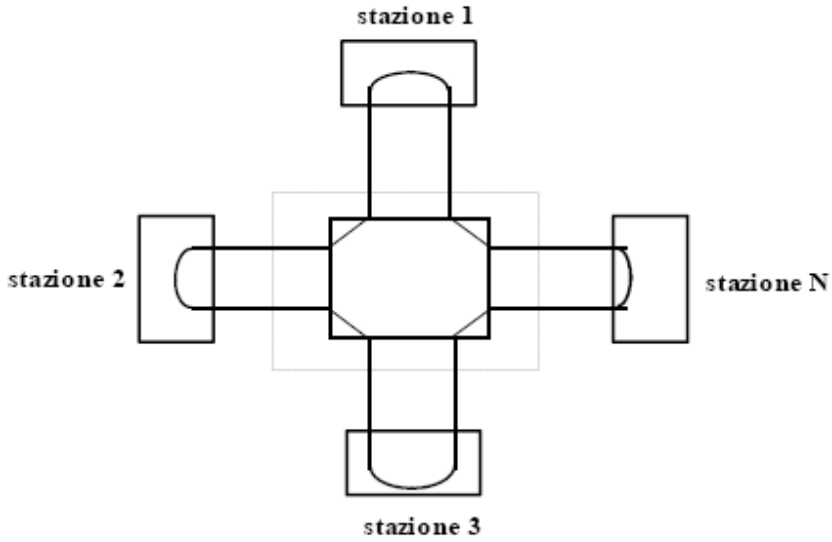


Topologia di rete ad anello

La trasmissione avviene in un unico senso, ad esempio quello antiorario indicato in figura.
Tutte le stazioni prendono parte alla trasmissione: quando una stazione invia sulla linea il proprio pacchetto, questo percorre l'intero anello, in quanto ciascuna stazione riceve il pacchetto, lo memorizza, lo rigenera e lo ritrasmette sulla linea successiva.
Proprio il fatto per cui ogni stazione provvede a rigenerare il segnale, l'anello può avere anche una elevata estensione. Al contrario, i limiti di estensione riguardano la distanza massima tra stazione e stazione.
Questa soluzione si rivela ottima se vengono usate le **fibre ottiche** (che sono notoriamente mezzi trasmissivi unidirezionali).
Il numero di stazioni può variare da poche decine fino a migliaia di unità.
Gli svantaggi fondamentali sono i seguenti:

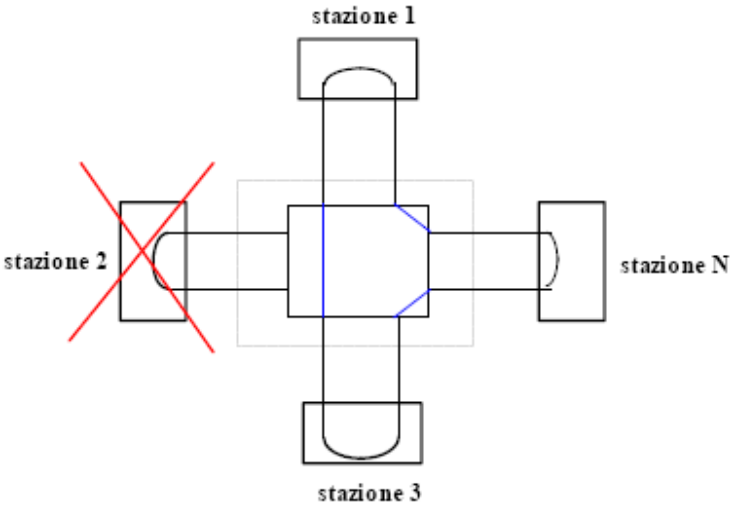
- la lunghezza complessiva del cavo non è minimizzata;
- l'affidabilità dell'intero sistema è critica (a meno di accorgimenti speciali che vedremo): la caduta o il malfunzionamento di una singola stazione o di una linea provoca la caduta dell'intera rete;
- l'inserimento di una eventuale nuova stazione rende necessario interrompere il funzionamento dell'intera struttura.

I problemi principali sono proprio gli ultimi due, ossia l'affidabilità e l'inserimento di nuove stazioni. Per eliminare entrambi questi problemi si può adottare il seguente accorgimento: si inserisce, nell'anello, un **centro di commutazione** (detto **relay**), al quale si connettono tutte le stazioni:



Topologia ad anello con centro di connessione centrale: il centro di connessione è connesso a ciascuna stazione con un cavo di andata ed uno di ritorno; in caso di caduta di una stazione, all'interno del centro di connessione si usano appositi circuiti che escludono la stazione stessa dall'anello, mantenendo quest'ultimo perfettamente funzionante

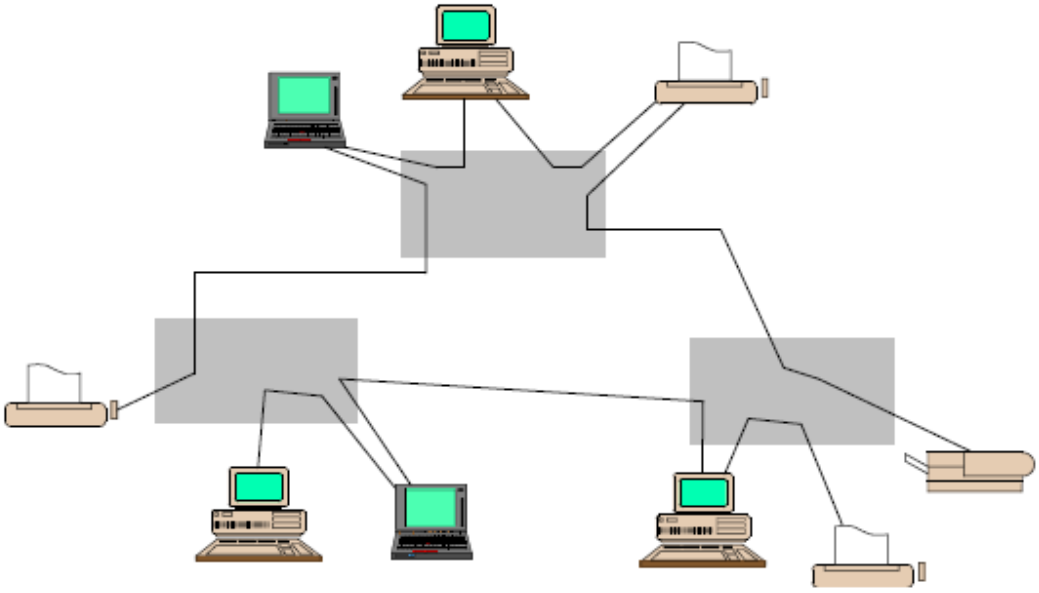
Così facendo, *la configurazione ottenuta è ad anello solo a livello logico, ma invece è a stella a livello fisico.* Questo risolve il problema dell'affidabilità: infatti, il centro ha la capacità di mettere fuori rete una qualsiasi stazione, usando appositi circuiti elettrici:



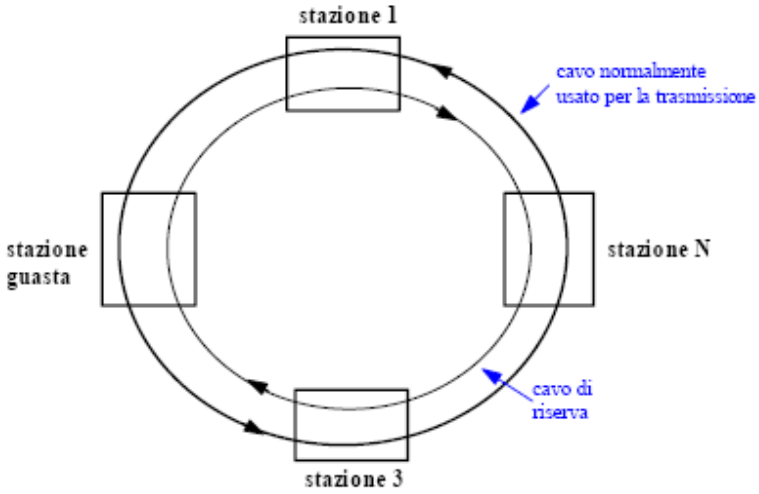
Esempio di funzionamento del centro di connessione: nel caso la stazione 2 dovesse interrompere il proprio funzionamento, il centro di connessione modifica i collegamenti al suo interno in modo da connettere direttamente la stazione precedente e quella successiva della stazione fuori uso. In tal modo, le stazioni ancora in funzione non si accorgono di niente

In caso di guasto o spegnimento di una determinata stazione, il relay disinserisce dall'anello il relativo cavo di giunzione tra centro e stazione (detto **lobo**), in modo da ricostruire l'anello senza la stazione.
Da notare che i relay possono essere attivati e disattivati dalle stazioni stesse: una stazione che volesse staccarsi temporaneamente dalla rete, manda un *segnale elettrico di inizializzazione*, il quale

fa scattare il relativo circuito nel centro di connessione, escludendo la stazione dall'anello; quando la stazione vuole rientrare, procede in modo analogo, inviando un segnale che riporti la configurazione nella situazione originale, reinserendo cioè la stazione stessa.
Uno svantaggio che appare subito evidente è che la lunghezza complessiva del cavo praticamente si raddoppia.
Talvolta, può capitare che si debbano connettere alla rete più stazioni di quante il centro di connessione possa gestire. In questo caso, si può far uso di più elementi centrali, tra loro collegati come nella figura seguente:

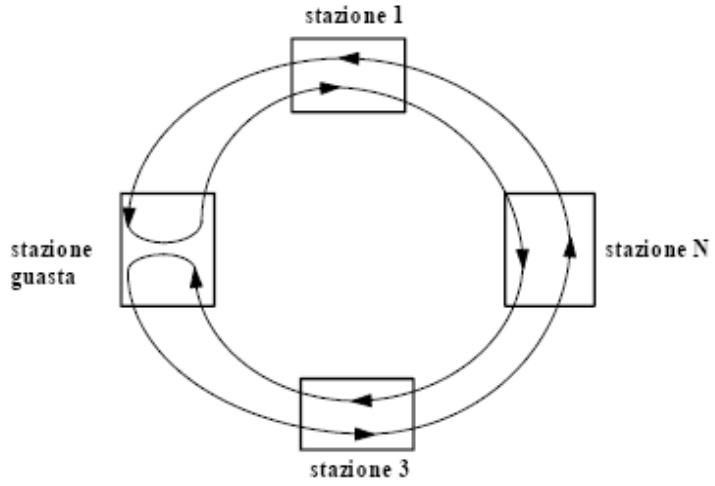


C'è anche un diverso approccio con cui risolvere il problema dell'affidabilità dell'anello, approccio che non prevede l'uso di un nodo centrale. *Esso consiste nel realizzare la connessione tra le stazioni non più con un unico cavo, ma con un doppio cavo,* secondo uno schema del tipo seguente:

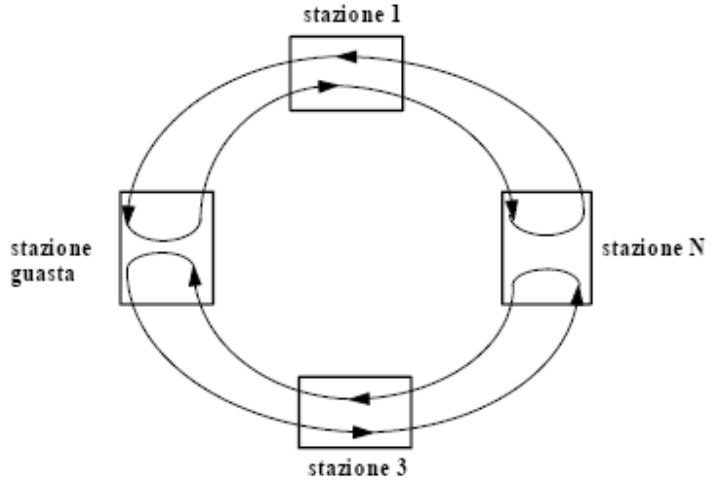


I due cavi sono entrambi monodirezionali ed uno solo di essi viene normalmente usato per la trasmissione (ad esempio quello che in figura è stato disegnato con maggiore spessore). L'altro, di riserva, permette la comunicazione in verso opposto.
Se una data stazione, ad esempio la numero 2, si guasta o viene volontariamente spenta, i due cavi

vengono automaticamente connessi all'interno della stazione stessa, in modo da ricostruire l'anello logico grazie al cavo riserva:

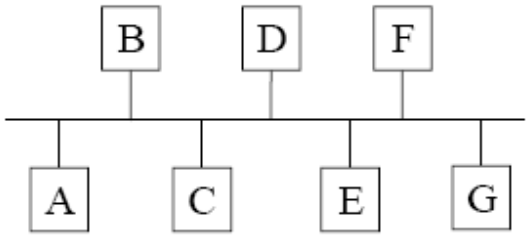


Ancora una volta, lo svantaggio fondamentale è nel fatto che la lunghezza complessiva dei cavi è raddoppiata rispetto alla semplice topologia ad anello. Oltre a questo, è chiaro che, in caso di caduta di due o più stazioni, si formano anelli parziali fra loro non connessi. Ad esempio, se si guastano sia la stazione 2 sia la stazione N, succede quando segue:



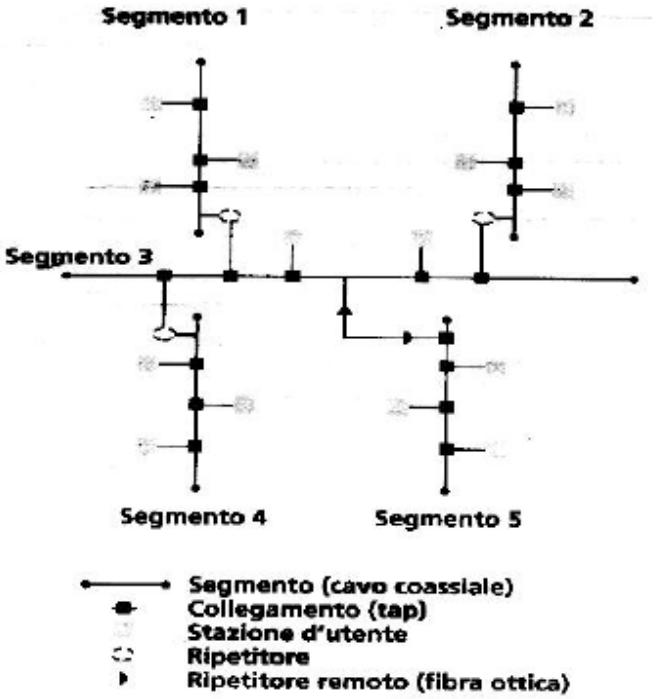
Si sono formati due anelli, che però non sono connessi tra di loro.
Per ovviare a questo problema, vedremo che le **reti token-ring** e le **reti FDDI** usano insieme l'accorgimento del doppio anello e quello del centro stella.
Il metodo del doppio anello può essere chiaramente utilizzato anche quando, come abbiamo visto prima, la rete è formata da più centri di interconnessione, a ciascuno dei quali sono collegate un certo numero di stazioni. In questo caso, il doppio anello si usa per connettere i centri di interconnessione. *Il vantaggio, in questo caso, è che, essendo i centri di interconnessione delle stazioni puramente passive, il rischio di andata fuori servizio è molto basso,* per cui è anche molto basso il rischio di formazione di anelli parziali del tipo visto nell'ultima figura.

Topologia a dorsale
Nella topologia a dorsale (adottata dalle reti locali di tipo **Ethernet**), c'è un unico cavo che si estende su tutta l'area in cui sono situate le stazioni:



Topologia di rete a dorsale

I dati viaggiano sul cavo e sono quindi leggibili da parte di tutte le stazioni. Quindi, la trasmissione di una stazione viene ricevuta da tutte le altre, nonostante possa esser in realtà diretta ad una sola stazione destinataria.
I maggiori vantaggi consistono nella semplicità, nei bassi costi e nell'affidabilità di questa topologia. Non solo, ma è evidente che il guasto di una qualsiasi stazione non provoca la disattivazione dell'intera rete, dato che le stazioni sono passive quando non trasmettono, al contrario di quanto abbiamo visto nel caso della topologia ad anello (nella sua forma più semplice), dove invece ciascuna stazione deve ricevere, rigenerare e ritrasmettere ogni pacchetto.
E' molto facile anche inserire nuove stazioni su cavo.
Oltre ad aggiungere nuove stazioni, si può anche pensare di collegare varie dorsali, dette in questo caso **segmenti**, secondo uno schema del tipo seguente:



In questa figura, abbiamo una rete formata da 5 segmenti; si distinguono allora una serie di "entità":
- in primo luogo, a ciascun segmento sono collegate (tramite i cosiddetti **tap**) un certo numero di stazioni d'utente;

- anche il collegamento tra segmenti è effettuato tramite *tap*, il che significa che un segmento vede un altro segmento come semplicemente una stazione d'utente; la differenza tra un collegamento segmento-stazione ed un collegamento segmento-segmento è nel fatto che, in quest'ultimo, è presente un **ripetitore**, allo scopo di aumentare la potenza del segnale prima di fornirlo alle varie stazioni connesse. Il motivo è chiaramente nell'inevitabile attenuazione subita dal segnale durante la propagazione;
- i ripetitori possono essere dei semplici *amplificatori elettronici* nel caso di usi, come mezzo trasmissivo, un doppino telefonico oppure un cavo coassiale, ma possono anche essere qualcosa di più complesso se si usano, per collegare i segmenti, delle fibre ottiche: in questo caso, il segnale elettrico prelevato da un segmento viene prima convertito in segnale ottico, poi trasmesso sulla fibra ottica ed infine riconvertito in segnale elettrico da immettere sul nuovo segmento. Servono dunque dispositivi di **conversione elettro-ottica** dei segnali;
- infine, alle estremità di ciascun segmento è necessario sistemare un adattatore di impedenza (detto anche **tappo**) che realizzi l'adattamento perfetto e quindi impedisca la riflessione del segnale.

Concludendo, i principali inconvenienti di una rete a dorsale sono i seguenti:

- i potenziali problemi di prestazioni dovuti al fatto che unico cavo serve tutte le stazioni: le prestazioni possono peggiorare quando il carico trasmissivo delle stazioni è elevato;
- una eventuale interruzione del cavo mette fuori uso l'intera rete;
- la mancanza di punti di concentrazione rende difficoltosa l'individuazione di eventuali punti di malfunzionamento;
- dato che le stazioni sono puramente passive, le distanze raggiungibili sono piuttosto ridotte anche con segnali di buon livello, a meno ovviamente di far uso dei ripetitori, che risolvono il problema a prezzo però di maggiori spese.

Generalmente, le LAN a dorsale supportano da alcune decine fino al massimo di un migliaio di stazioni.

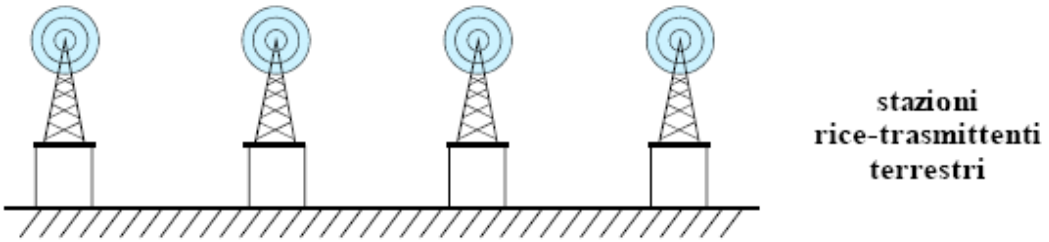
Considerazioni generali sulle topologie

Abbiamo dunque visto che ogni topologia ha caratteristici punti di forza e di debolezza. La scelta della topologia va pertanto fatta tenendo presente fondamentalmente l'affidabilità, l'espandibilità, la complessità dell'installazione, le possibilità di controllo, i costi, l'ampiezza di banda disponibile.
Le configurazioni ad anello ed a stella appaiono come quelle più vulnerabili a causa della ripercussione sull'intera rete della caduta, rispettivamente, della stazione singola o del nodo centrale, mentre la soluzione a dorsale non sembra presentare questo rischio. In realtà, abbiamo osservato sia che esistono opportuni accorgimenti atti a risolvere i problemi delle reti ad anello ed a stella sia che anche la soluzione a dorsale cessa di funzionare nel momento in cui si verifica una interruzione del cavo.
In generale, quindi, *possiamo affermare che ogni topologia presenta i propri inconvenienti, ma è sempre possibile pensare ad accorgimenti che risolvano tali inconvenienti.* Naturalmente, l'implementazione pratica di questi accorgimenti potrà poi risultare più o meno conveniente, soprattutto da un punto di vista economico.
A proposito degli aspetti economici, un elemento cui si è dato in passato una certa rilevanza è il costo dei cavi; tuttavia, grazie alla progressiva diminuzione dei prezzi, questo aspetto è attualmente di secondo piano.
Osserviamo inoltre che *le strutture ad anello ed a stella sono quelle che si prestano maggiormente a collegamenti ad alta o altissima velocità, grazie al fatto che esse utilizzano solo collegamenti punto-a-punto.* Disponendo di mezzi trasmissivi idonei (tipicamente le **fibre ottiche**), le

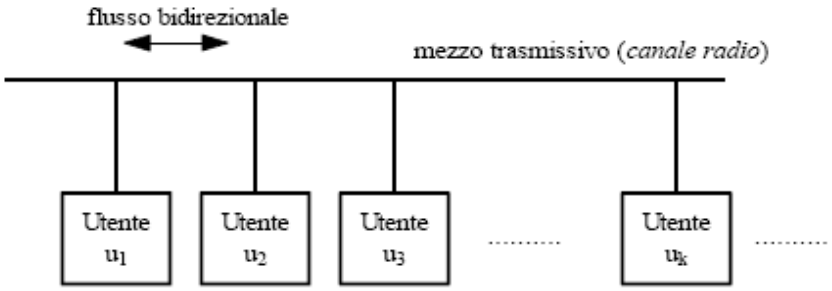
velocità di trasmissione raggiungibili sono molto elevate. D'altra parte, se consideriamo una struttura a stella con collegamenti punto-a-punto ad altissima velocità, appare evidente che le elevate prestazioni sono raggiungibili solo se il centro stella ha una elevata velocità di commutazione; in caso contrario, se cioè il centro stella fosse lento, l'alta velocità con cui le stazioni scambiano i dati con il centro stella verrebbe compensata dalla bassa velocità con cui il centro stella smista i dati sulle varie linee.

PROTOCOLLO ALOHA

Cominciamo l'analisi dei protocolli di accesso multiplo con un esempio di protocollo di accesso casuale, nel quale cioè non venga stabilito alcun ordine particolare con cui le varie stazioni devono accedere all'unica risorsa trasmissiva disponibile.
Consideriamo perciò una rete di telecomunicazione basata sul cosiddetto **protocollo ALOHA**. Questa è una tecnica che deriva da una rete per la trasmissione dati via radio, sviluppata dall'Università delle Hawaii (da cui il nome *ALOHA*), all'inizio degli anni '70:



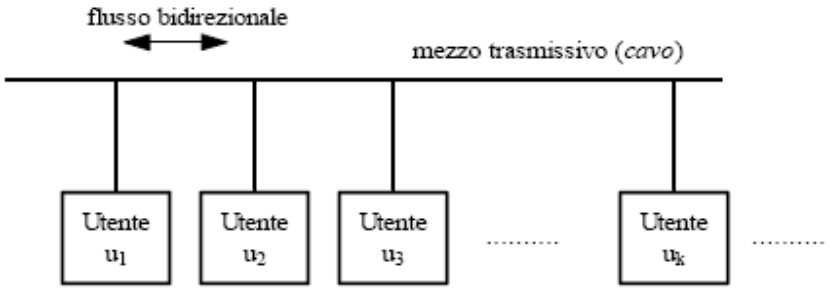
Il mezzo trasmissivo è dunque l'atmosfera: in particolare, i dati vengono inviati sulle **frequenze UHF** (vale a dire frequenze comprese nell'intervallo `300÷3000 MHz`) utilizzando il classico schema a **pacchetti** (di lunghezza fissa). In particolare, tutte le stazioni accedono all'unica banda di frequenza disponibile, il che equivale ad avere un mezzo trasmissivo generico nel quale sia tollerabile una trasmissione per volta:



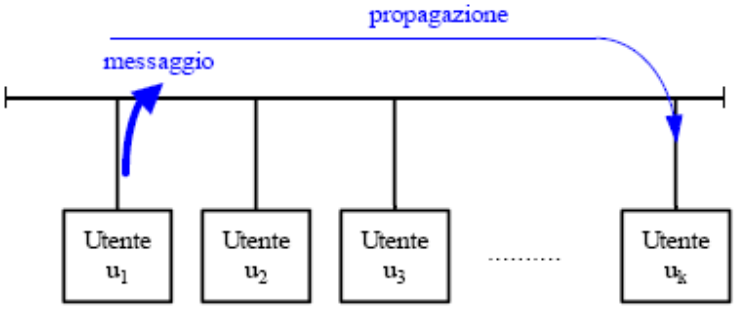
Quando una stazione emette un messaggio (cioè un singolo pacchetto di bit o un insieme di pacchetti), questo messaggio, se la trasmissione va a buon fine, viene ricevuto indistintamente da tutte le stazioni (cosa che, in figura, è stata indicata con la freccia a due punte, indicante appunto un *flusso bidirezionale* dei dati sul mezzo trasmissivo). Ciascuna stazione, una volta ricevuto il generico pacchetto, esamina l'indirizzo del destinatario: se si riconosce in tale indirizzo[3], compie sul pacchetto le necessarie elaborazioni, altrimenti lo scarta visto che non le interessa.
C'è evidentemente il problema per cui, se due stazioni trasmettono contemporaneamente o, comunque, se una stazione trasmette quando già un'altra lo sta facendo, i due messaggi, essendo allocati sulle stesse frequenze, si sovrappongono e diventano indecifrabili: quando questa situazione si verifica, si dice che c'è stata una **collisione**.

Proprio perché fa parte dei *protocolli ad accesso multiplo casuale*, il protocollo ALOHA non prevede niente di particolare per evitare le collisioni. La tecnica è detta infatti, in gergo, "*trasmetti e prega*": essa prevede che, appena una stazione debba trasmettere, lo faccia immediatamente, a prescindere da quello che stanno facendo le altre stazioni; se, nel frattempo, un'altra stazione sta trasmettendo a sua volta, si verifica la collisione.
Ciascuna stazione che abbia inviato un proprio messaggio sul canale, per sapere quale sia stato l'esito della propria trasmissione, non può fare altro che aspettare la *risposta di avvenuta ricezione* da parte dei propri destinatari; se si è verificata una collisione, è ovvio che i destinatari non hanno ricevuto niente di decifrabile e quindi non hanno inviato alcuna risposta. La generica stazione mittente, quindi, non ricevendo alcuna risposta dai destinatari, capisce che è avvenuta una collisione. A questo punto, essa aspetta un *periodo casuale di tempo* e poi ritenta la trasmissione. Il motivo per cui il tempo di attesa, prima del nuovo tentativo, deve essere casuale è evidente: se due stazioni, che hanno colliso, hanno cessato la propria trasmissione nello stesso istante e poi aspettano lo stesso tempo prima di riprovare, ricadrebbero in una nuova collisione; al contrario, se ciascuna stazione sceglie un tempo casuale di attesa, la probabilità di collisione è di gran lunga ridotta.

SCHEMA CSMA/CD
Il protocollo **CSMA** (*Carrier Sense Multiple Access*, che sta per *accesso multiplo a rivelazione di portante*) è una importante evoluzione dello schema *un-slotted ALOHA* ed è perciò un altro protocollo a contesa, nel quale cioè sono possibili le collisioni.
La differenza sostanziale rispetto all'ALOHA è quella per cui una stazione, prima di trasmettere, "ascolta" se v'è già una trasmissione in corso (da cui la terminologia "*rivelazione di portante*") e trasmette solo in caso negativo. Inoltre, durante la trasmissione, la stazione stessa continua a monitorare il canale e si interrompe immediatamente nel caso rilevi una collisione (così come ALOHA).
Si tratta di una tecnica molto diffusa, che presenta una serie di varianti legate soprattutto alla ripresa del tentativo di trasmissione dopo che una stazione ha rilevato che il canale è già occupato. Una di queste varianti, detta **CSMA/CA** (*Carrier Sense Multiple Access - Collision Avoidance*) prevede che la stazione debba attendere un intervallo di tempo, di durata casuale o fissa, prima di ritentare. Grazie all'intervallo di tempo cui sono comunque costrette, le stazioni tendono ad evitare le collisioni, ma questo determina un degrado delle prestazioni.
La variante sicuramente più diffusa è quella denominata **CSMA/CD** (*Carrier Sense Multiple Access - Collision Detected*). Essa è stata resa popolare dalle reti locali di tipo **Ethernet** ed è stata standardizzata (col nome di **IEEE 802.3**). *Il mezzo trasmissivo usato può essere sia il mezzo radio sia un cavo (mezzo passa-basso):* nel primo caso, si usa certamente una portante alla quale agganciare i messaggi (ci si riferisce alle tecniche di modulazione numerica, in cui una delle caratteristiche della portante - ampiezza, fase, frequenza - viene fatta variare proporzionalmente al segnale modulante, costituito dalla sequenza di bit da trasmettere), per cui il monitoraggio di una stazione sul canale si basa semplicemente sulla rivelazione della presenza o meno della portante (canale libero significa assenza di portante); se, invece, si usa un cavo, generalmente non si usa alcuna portante in quanto si trasmette in *banda base*, il che significa che il monitoraggio del canale consiste nel verificare la presenza o meno di segnali qualsiasi (che non siano ovviamente rumore).
Per semplicità, consideriamo il caso di rete cablata, per cui il mezzo trasmissivo è un cavo (ad esempio il **cavo coassiale standard Rg 8** per lo standard **IEEE 802.3**):
mezzo trasmissivo (*cavo*)

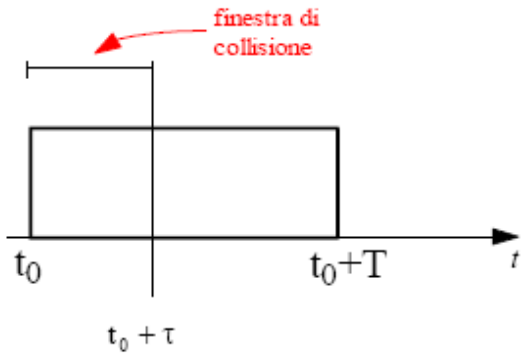


Come detto, ogni stazione monitora continuamente il canale. Quando ha qualcosa da trasmettere, verifica se il canale è libero: appena lo rileva libero, trasmette il proprio messaggio su entrambi i segmenti di cavo (in modo cioè che il messaggio giunga a tutte le stazioni) e continua a monitorare il cavo stesso. Il messaggio raggiunge tutte le stazioni, ma viene conservato solo dal destinatario (o dai destinatari).
Un meccanismo di questo tipo non evita il pericolo delle collisioni e ce ne possiamo rendere conto con un esempio molto semplice:

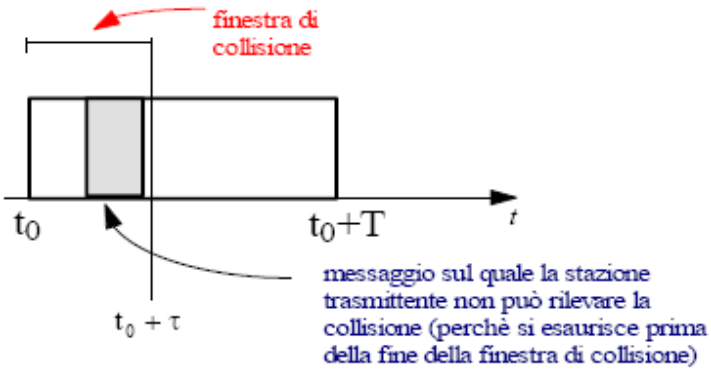


Supponiamo, come indicato in figura, che la prima stazione abbia appena rilevato il canale libero ed abbia così deciso di immettere sulla linea il proprio pacchetto. Essa impiega un tempo T per immettere il pacchetto completo sul canale:

Il segnale elettrico immesso sul cavo impiega a sua volta un certo tempo per propagarsi e raggiungere tutte le stazioni. Questo tempo è evidentemente funzione della distanza tra le stazioni e della velocità di propagazione v_P. La stazione che per ultima riceve il messaggio è quella più lontana dalla stazione mittente. Supponiamo allora che la prima stazione abbia cominciato la propria trasmissione nell'istante t_0 e che il corrispondente segnale elettrico impieghi un tempo τ per percorrere tutta la linea, fino all'ultima stazione. E' evidente che, nell'intervallo $[t_0,t_0+\tau]$, il segnale non ha ancora raggiunto l'ultima stazione; di conseguenza, se questa stazione va a monitorare il canale in questo intervallo, lo trova libero, per cui si sente autorizzata a trasmettere, non sapendo che già un'altra stazione aveva cominciato a farlo. Ancora una volta si crea una collisione:



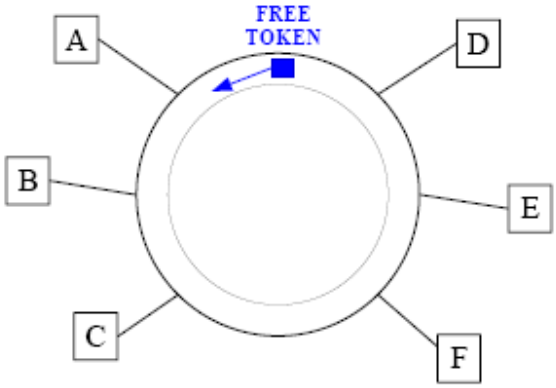
Dato che le stazioni, anche durante la propria trasmissione, continuano a monitorare il canale, esse si accorgono della collisione e cessano di trasmettere.
L'intervallo $[t_0,t_0+\tau]$ prende il nome di **finestra di collisione** e cambia evidentemente al variare della stazione che sta già trasmettendo e della stazione che a sua volta tenta di trasmettere. La massima finestra di collisione è proprio quella che coinvolge le due stazioni estreme della linea. In generale, la durata della finestra di collisione è direttamente proporzionale alla lunghezza del tratto di cavo considerato ed è inversamente proporzionale alla velocità di propagazione del segnale elettrico sul cavo stesso. Anche in questo caso, in seguito ad una collisione, dopo che tale collisione è stata rilevata, ciascuna stazione aspetta un tempo casuale prima di riprovare la trasmissione, in modo da ridurre la probabilità di una collisione successiva.
Come vedremo in dettaglio, la variante CSMA/CD è quella che permette, tra le tecniche a contesa, le prestazioni migliori. Se il carico trasmissivo è basso, il tempo di accesso alla trasmissione è inferiore a quello delle LAN del tipo **Token-Ring** (che vedremo più avanti), le collisioni sono poche e quindi le prestazioni effettive brillanti. Inoltre, essendo lo schema molto semplice, anche il corrispondente hardware è abbastanza semplice e facile da realizzare ed è questo il motivo per cui lo schema CSMA/CD ha riscosso grande successo. Oltre a questo, non essendoci alcuna stazione centrale, si tratta di un sistema completamente distribuito.
E' interessante notare che c'è un preciso vincolo che consenta ad una stazione che ha provocato una collisione di capire che la propria trasmissione non è andata a buon fine. Infatti, consideriamo una stazione che, rilevando libero (erroneamente) il canale, comincia la propria trasmissione. La stazione emette un certo numero di bit, che è proporzionale alla velocità di immissione dei bit in rete. Se il messaggio non è sufficientemente lungo, la sua immissione sul canale potrebbe esaurirsi prima che si concluda la finestra di collisione: in questo caso, la stazione non si accorgerebbe minimamente della collisione.



Si possono fare alcun semplici conti: per una **LAN Ethernet standard**, con un cavo lungo al

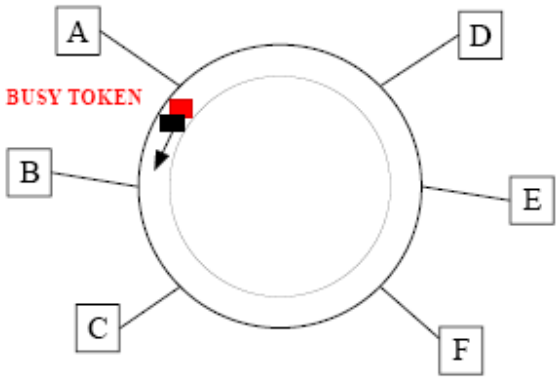
massimo 2.5 km, una velocità di immissione dei bit in linea di 10 Mbps ed una velocità di propagazione dei segnali elettrici di circa $2*10^8$ m/s, , si stima che ogni messaggio deve essere lungo almeno **64 byte**. Se esso fosse più corto, come può accadere in alcune applicazioni, bisognerebbe necessariamente inserire dei **byte di riempimento**, privi di contenuto informativo. Questo è un grosso limite di sviluppo: per esempio, se si volesse estendere la rete a lunghezze 10 volte maggiori e contemporaneamente si volesse aumentare dello stesso rapporto la velocità di immissione, la lunghezza minima del messaggio diventerebbe di 6400 byte (cioè 64*10*10) e questa è una lunghezza accettabile solo per determinate applicazioni. Anche per questi motivi, quando è stato progettato lo schema di reti locali a 100 Mbps su fibra ottica (**reti FDDI**), è stato seguito uno schema (detto di *token-passing*) diverso dallo schema CSMA/CD. Per gli stessi motivi, le LAN di tipo **Ethernet a 100 Mbps** presentano dimensioni complessivamente ridotte.
Tutti questi discorsi evidenziano comunque una cosa fondamentale: `al contrario dello schema ALOHA, lo schema CSMA/CD è fortemente dipendente dalla distanza tra le stazioni.` Anzi, abbiamo osservato che tale distanza non può essere troppo elevata, perché questo aumenterebbe la durata della finestra di collisione e quindi, in media, il numero di collisioni. Deduciamo che una rete di telecomunicazioni basata sullo schema CSMA/CD deve necessariamente avere una limitata copertura geografica.

TECNICA DEL PASSAGGIO DEL TOKEN (TOKEN PASSING)
La tecnica detta del **token passing** è una tecnica *non a contesa* (o di *accesso multiplo ordinato*), nella quale cioè non sono possibili collisioni. Essa conviene soprattutto nelle reti con *topologia ad anello* (**ring**). Consideriamo allora proprio una rete ad anello, schematizzata come nella figura seguente:

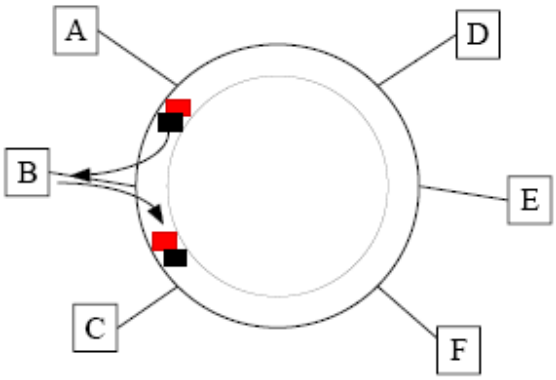


Il principio di base è che una generica stazione, per poter trasmettere (cioè accedere al canale di trasmissione) deve ricevere il cosiddetto **token**, ossia una particolare configurazione di bit che l'autorizzi a trasmettere. In particolare, essa aspetta il cosiddetto **free token**, che, come vedremo, si distingue dal **busy token**. Il token ha generalmente una lunghezza di 16 bit; il *free token* si differenzia del *busy token* per uno o più bit. Per semplicità, possiamo supporre che il token sia lungo 8 bit e che il *free token* si differenzi dal *busy token* per l'ultimo bit, che vale 0 per il free token ed 1 per il busy token
Ogni stazione riceve a turno, dalla stazione che la precede secondo il senso della trasmissione (che è unidirezionale), il *free token*. Nell'ultima figura, è la stazione A che riceve il free token. Appena riceve questo token, la stazione deduce che può trasmettere. Immette allora sulla rete il proprio pacchetto, accodando ad esso il *busy token*, in modo da segnalare alle altre stazioni che il

diritto a trasmettere è stato acquisito:

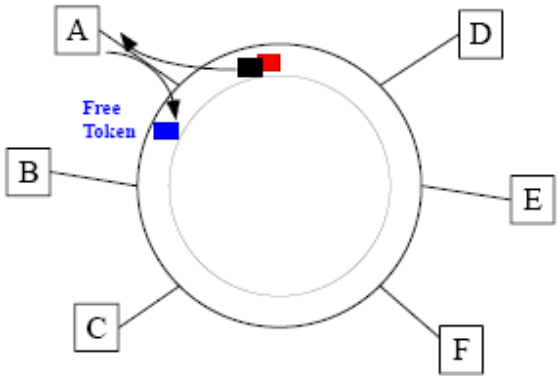


L'insieme dei bit rappresentativo del pacchetto e del busy token si propaga dunque lungo l'anello, attraversando tutte quante le stazioni. Ogni stazione che riceve i bit, si comporta nel modo seguente: man mano che i bit arrivano, li memorizza, li rigenera e li ritrasmette uno ad uno verso la stazione successiva:



Il messaggio (pacchetto + token busy) emesso dalla stazione A viene ricevuto per primo dalla stazione B, la quale lo memorizza, lo rigenera e lo ritrasmette sul canale verso la stazione successiva. Stesso comportamento per tutte le stazioni

Per semplicità, supponiamo per il momento che la stazione non faccia alcuna elaborazione dei bit tra l'istante in cui comincia a riceverli e l'istante in cui finisce di ritrasmetterli. Al contrario, una volta che la stazione ha finito di ritrasmettere il pacchetto, ne possiede una copia, sempre insieme al token, nella propria memoria (**buffer**), per cui può farne una elaborazione: tipicamente, la stazione deve andare a leggere l'indirizzo del destinatario del pacchetto e verificare se si tratta del proprio indirizzo (il che significa che il pacchetto è destinato proprio ad essa), oppure no (nel qual caso può tranquillamente scartare il pacchetto).
Con questo meccanismo è evidente che il pacchetto percorre tutto l'anello, attraversando tutte le stazioni, fino a giungere nuovamente alla stazione che lo aveva trasmesso:



A questo punto, la stazione trasmittente si rende conto che il pacchetto è sicuramente passato per tutte le stazioni, inclusa quella o quelle cui era destinato. A seconda delle scelte di progetto, la stazione trasmittente può adesso trasmettere un secondo pacchetto oppure liberare il canale. Supponiamo che sia obbligata a liberare il canale, come indicato nell'ultima figura. Per fare questo, è sufficiente che essa emetta nuovamente un *free token*, il quale raggiungerà così la stazione successiva, conferendole il diritto a trasmettere. Se questa stazione ha da trasmettere un proprio pacchetto, allora si comporta così come visto prima; se invece non ha niente da trasmettere, emette anch'essa un free token passando il diritto a trasmettere alla stazione ancora successiva.
A queste considerazioni dobbiamo però aggiungerne delle altre. Consideriamo infatti la stazione che si riconosce essere destinataria del pacchetto appena ricevuto. In questo caso, è opportuno che la stazione compia una certa elaborazione sui bit ricevuti:

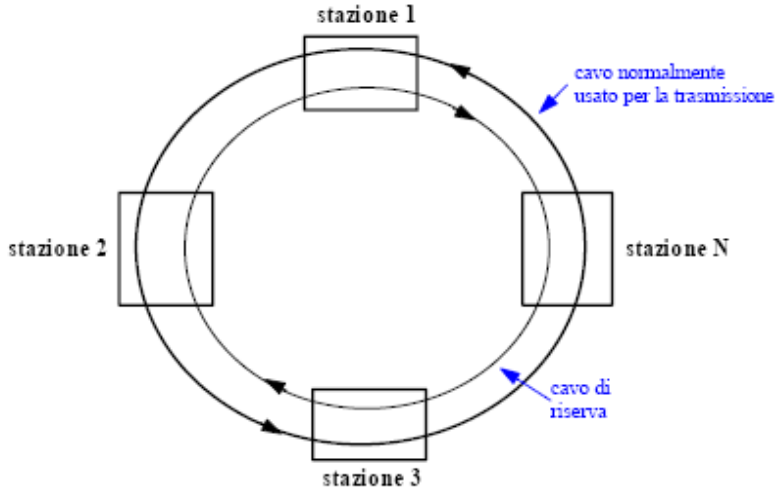
- in primo luogo, essa imposta su ON un particolare **bit di indirizzo riconosciuto**, in modo da segnalare, alla stazione mittente, che il suo pacchetto è stato effettivamente ricevuto;
- in secondo luogo, essa effettua anche un controllo degli errori: l'esito di questo controllo viene indicato ancora una volta imposta un opportuno **bit di corretta ricezione**, posto generalmente in coda al pacchetto.

In questo modo, quando il pacchetto ritorna alla stazione che lo ha trasmesso, quest'ultima è in grado di sapere sia se il pacchetto è stato ricevuto sia, in caso affermativo, se ci sono stati o meno errori. Essa, quindi, elimina il pacchetto dalla rete, annota se la trasmissione è andata a buon fine oppure no, ma comunque, a prescindere dall'esito della trasmissione e dall'eventuale necessità di trasmettere altri pacchetti, emette un nuovo free token e libera il canale.
La scelta di consentire a ciascuna stazione la possibilità di emettere un solo pacchetto per volta deriva da una considerazione molto semplice: evitare che una stazione monopolizzi il canale di trasmissione.
E' inoltre evidente che, con questo meccanismo, viene evitata del tutto la possibilità di contese, in quanto solo la stazione in possesso del token è autorizzata a trasmettere. Si tratta dunque di un metodo deterministico, che non presenta rischi di collisione o, peggio, di collasso di rete (che si può verificare invece nella reti con accesso a contesa come quelle basate sul protocollo *ALOHA*), che offre prestazioni elevate e stabili anche in presenza di carico trasmissivo elevato.

SCHEMA FDDI

L'**FDDI** è uno standard appositamente definito dall' **ANSI** (l'acronimo **ANSI** sta per *American National Standards Institution* ed è l'ente che rappresenta gli Stati Uniti nell'**ISO** (**International Standard Organization**), ossia l'agenzia dell'ONU responsabile degli standard internazionali, inclusi quelli delle comunicazioni. per le reti locali ad alta velocità basate su fibre ottiche. Sostanzialmente, l'FDDI (*Fiber Distributed Data Interface*) è una evoluzione del token-ring. Infatti, osserviamo per

prima cosa che la topologia utilizzata è ancora una topologia ad anello (*ring*), con la differenza, però, che gli anelli sono due e che il verso di percorrenza delle informazioni è opposto sui due anelli:



I due cavi sono entrambi monodirezionali ed uno solo di essi viene normalmente usato per la trasmissione.
Come abbiamo già avuto modo di dire, il problema fondamentale di questo schema si ha in caso di caduta di due o più stazioni: infatti, in questo caso si formano anelli parziali fra loro non connessi. Per ovviare a questo problema, le **reti FDDI** , così come le *token ring*, usano insieme l'accorgimento del doppio anello e quello del **centro stella**, il quale è in grado in ogni momento di monitorare lo stato sia delle linee sia delle stazioni e di prendere gli opportuni accorgimenti per mantenere attivo l'anello logico in caso di malfunzionamenti.
Quindi, l'uso del doppio anello è una prima fondamentale somiglianza tra una rete FDDI ed una rete token ring. Un'altra somiglianza riguarda la tecnica di accesso al mezzo trasmissivo. Infatti, la tecnica usata nelle reti FDDI è una variante del passaggio del token ed è identificata dall'acronimo **E.T.R.** (ossia *Early Token Release*). Secondo questa tecnica, ogni stazione è ancora una volta autorizzata a trasmettere solo quando riceve un *free token*; la differenza, però, rispetto al token ring, è che la stazione può rilasciare il free token appena ha terminato di trasmettere il proprio pacchetto, senza quindi dover aspettare di ricevere lo stesso pacchetto (come invece accadeva nel token ring). La conseguenza di questo fatto è che, in un dato istante, nonostante sia in corso la trasmissione di un pacchetto, ci sia comunque un token libero, il che permette di sfruttare meglio il canale.
Esistono una serie di varianti a questo schema, nelle quali si cerca essenzialmente di sfruttare l'elevata velocità di trasmissione, che è **100 Mbps**, e la brevità della "lunghezza d'onda" dei bit in trasmissione, che è di circa **2 m**. In particolare, citiamo due importanti possibilità:

- in primo luogo, c'è la possibilità di avere diversi token in circolazione simultanea sulla rete, ossia quindi di autorizzare contemporaneamente, sia pure in ordine opportuno, più stazioni alla trasmissione simultanea;
- in secondo luogo, si può attribuire alla generica stazione il diritto a trasmettere quanti più pacchetti possibile all'interno di un tempo (slot) prefissato, anziché un solo pacchetto alla volta.

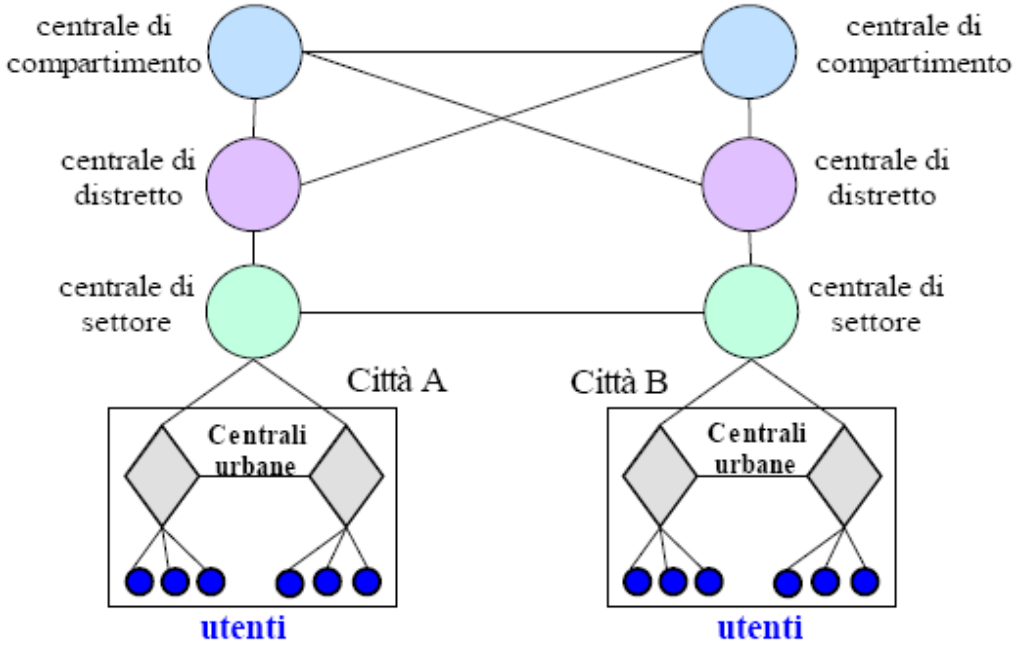
Lo svantaggio principale delle reti FDDI è nei costi elevati di implementazione e gestione, in particolare rispetto alle reti basate sulle tecnica CSMA/CD.
Per concludere, ricordiamo che lo standard per reti FDDI prevede un cavo (fibra ottica appunto) che può essere lungo fino a **200 km** e che può collegare fino a **1000 nodi**, distanti non più di **2 km** uno dall'altro.

LE RETI WAN

LA COMMUTAZIONE

COMMUTAZIONE DI CIRCUITO

La necessità della commutazione risulta evidente se pensiamo che, senza di essa, ogni utente della **rete telefonica pubblica** dovrebbe avere tante linee punto-a-punto permanenti quanti sono gli utenti della rete con cui esso vuole parlare (al limite tutti gli utenti della rete). Al contrario, essendo le centrali telefoniche pubbliche dotate della **funzione di commutazione**, la cosa è più semplice: l'apparecchio telefonico di ogni utente è collegato, tramite una *linea dedicata*, solo alla **centrale urbana** più vicina:

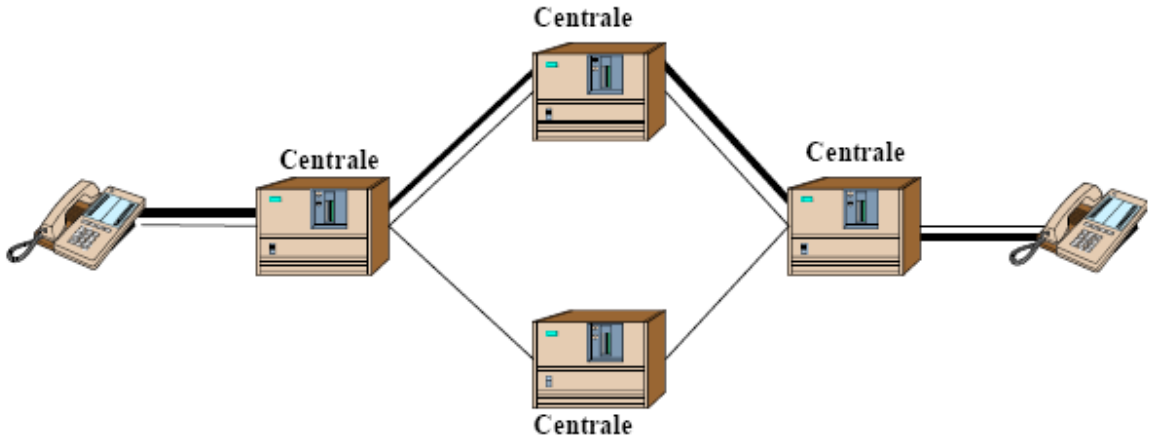


All'atto di una *chiamata*, la centrale interpreta il numero dell'utente chiamato e provvede a realizzare un **circuito fisico** che collega l'apparecchio di chi chiama a quello di chi risponde. Questa è la tecnica cosiddetta a **commutazione di circuito** ed è valida non solo per le conversazioni telefoniche, ma anche per la trasmissione dati: il terminale dell'utente è collegato alla centrale telefonica (o a qualche altra apparecchiatura con funzioni simili, come ad esempio un PABX), la quale provvede a stabilire un circuito fisico tra il DTE chiamante ed il DTE remoto che si vuol contattare.
L'alternativa a questo modo di procedere è la cosiddetta tecnica a **commutazione di pacchetto**, che sarà descritta più avanti.

La più antica forma di commutazione è proprio quella tramite centrale telefonica pubblica: c'è una fase iniziale di chiamata, necessaria affinché il numero dell'utente chiamato arrivi alla prima **centrale urbana** e questa inoltri il segnale su una linea di uscita diretta verso un'altra centrale; il procedimento segue fino a raggiungere la centrale cui è connesso l'utente chiamato; in questo modo, le varie centrali (*urbana*, di *settore*, di *distretto*, di *compartimento*) mettono in serie segmenti di cavo trasmissivo (o, in alternativa, il loro equivalente su un cavo a banda larga, nel quale quindi ad ogni conversazione viene riservato un canale, ossia una certa banda di frequenze) fino a formare il circuito completo tra utente chiamante e utente chiamato.
Una volta realizzato tale circuito, la comunicazione può avere luogo. La tariffa di tale circuito può essere applicata in vari modi; in generale, essa cresce proporzionalmente alla distanza ed al tempo di attivazione del circuito.
Da notare che le centrali hanno un ruolo puramente passivo: ad esempio, nel caso della trasmissione dati, esse non memorizzano i messaggi trasmessi né, salvo funzioni di valore aggiunto espressamente richiamate dall'utente, convertono il protocollo di linea utilizzato. Se, a causa di un sovraccarico di lavoro, non dispongono di linee di uscita libere, esse bloccano il collegamento direttamente in fase di chiamata[1].

CENNI AL FUNZIONAMENTO DI UNA CENTRALE TELEFONICA URBANA

Possiamo dare dei cenni sulla logica con cui una centrale urbana controlla le linee degli utenti locali per determinare e realizzare il circuito fisico tra utente chiamante e utente chiamato.



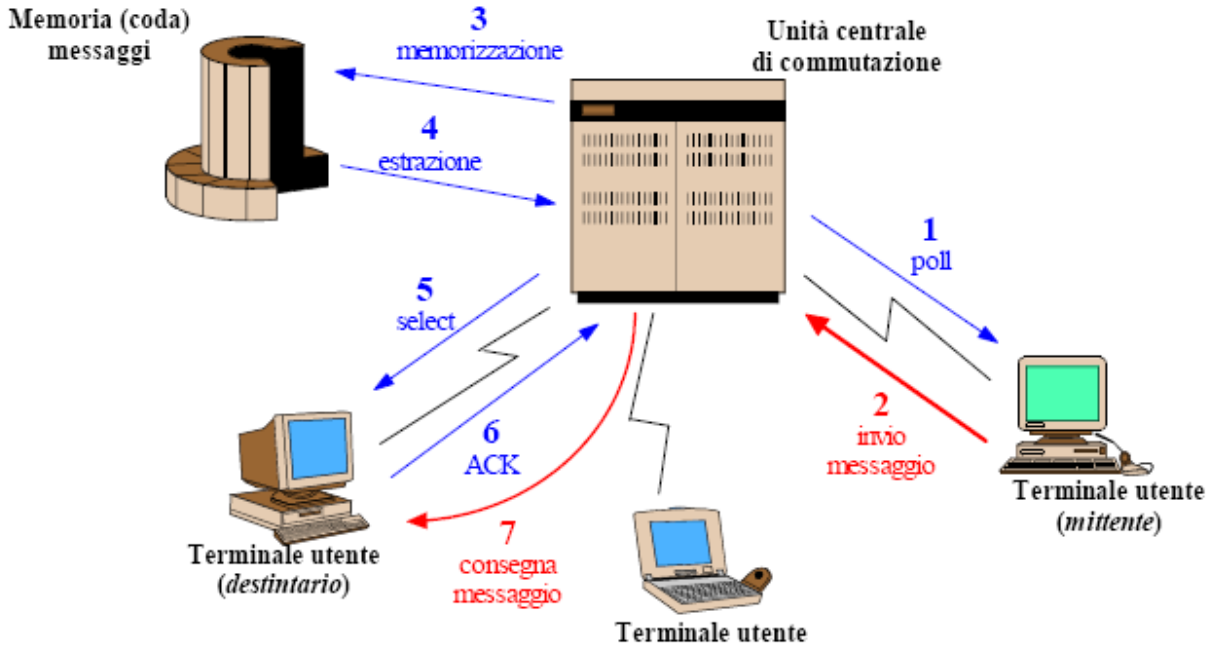
Si definisce **off-hook** l'azione con cui si avverte la centrale che si intende fare una chiamata: se si tratta di una conversazione telefonica, per cui si usa l'apparecchio telefonico, l'*off-hook* consiste semplicemente nel sollevare la cornetta, il che lascia partire un impulso elettrico di avviso verso la centrale; nel caso, invece, di una trasmissione dati, per la quale si usa tipicamente un modem collegato alla linea telefonica, la funzione di *off-hook* è svolta dal modem stesso.
In centrale è presente un **registro** in cui viene memorizzato il numero dell'utente chiamato: se c'è

[1] Facciamo osservare che questo è un criterio assolutamente generale, valido per qualsiasi rete di telecomunicazioni: quando una rete riceve una richiesta di servizio da parte di un utente, deve per prima cosa rendersi conto che sono disponibili le risorse necessarie a soddisfare la richiesta con la qualità richiesta; in caso affermativa, la rete impegna tali risorse e soddisfa la richiesta; in caso contrario, la rete deve necessariamente respingere in partenza la richiesta. Ovviamente, se al momento della richiesta di servizio dovessero risultare disponibili le risorse richieste, non è comunque escluso che, una volta accettata la richiesta, tali risorse vengano a mancare, per periodi di tempo più o meno lunghi: in queste situazioni, la rete deve necessariamente garantire il proseguimento del servizio e di questo bisogna assicurarsi in sede progetto.

un registro disponibile, significa che la centrale è in grado di effettuare la chiamata, per cui la centrale restituisce all'utente un segnale di **linea pronta**, indicante appunto la possibilità di procedere. Se invece non c'è alcun registro disponibile, bisogna riprovare.
Una volta ottenuta la linea libera, si compone il numero dell'utente con cui si vuol comunicare e tale numero viene registrato nell'apposito registro selezionato nella fase precedente. Segue l'importante fase per la **determinazione del percorso**, che assume una particolare rilevanza nel caso in cui la chiamata è interurbana, il che significa che saranno coinvolte almeno due centrali urbane. Per fare un esempio concreto, supponiamo che tra la centrale X dell'utente chiamante e la centrale Y dell'utente chiamato si possano scegliere due strade: quella che passa per le centrali intermedie M ed N e quella che passa per le centrali intermedie P, Q ed R. A prescindere dalla distanza fisica tra le centrali, è ovvio che la seconda possibilità comporta sia un allungamento del tempo complessivo della fase di chiamata, dato che sono coinvolte 3 centrali anziché 2, sia anche un maggior onere della rete, dato che il circuito fisico da realizzare impegna parte delle risorse di 5 centrali anziché di 4.
Questo esempio aiuta a capire la criticità degli **algoritmi di instradamento** delle chiamate: un buon algoritmo di instradamento dovrebbe cercare sempre di individuare il percorso con il minor numero di nodi intermedi e ricorrere agli altri solo quando la prima scelta non risultasse agibile (ad esempio a causa del malfunzionamento di una o più centrali). Il cosiddetto **segnale di occupato** viene inviato all'utente chiamante solo quando nessuna strada in uscita risulta disponibile.
Una volta riscontrata la disponibilità di una linea tra l'utente chiamante e la centrale dell'utente chiamato, quest'ultima riceve il numero dalla centrale dell'utente chiamante; questo consente di individuare direttamente l'utente chiamato. A questo punto, segue una ulteriore verifica: bisogna vedere se la linea dell'utente chiamato è libera. In caso negativo, il chiamante riceve i toni di **occupato**. Se, invece, la linea è libera, viene inviato al chiamato un segnale che faccia suonare il telefono. Non appena l'utente chiamato alza la cornetta, la connessione viene istantaneamente stabilita e quindi può partire la conversazione.
Come è noto, le centrali di commutazione della rete telefonica pubblica sono state per lungo tempo di tipo elettromeccanico, mentre solo recentemente sono state introdotte prima quelle elettroniche, ma sempre analogiche, e finalmente quelle numeriche, che al giorno d'oggi sono la quasi totalità. Nelle centrali elettromeccaniche, tutto si basava sul movimento di organi elettromeccanici: in particolare, a muoversi erano i **connettori** che collegavano la linea in ingresso a quella di uscita. I principali inconvenienti di questa soluzione erano nel tempo non trascurabile necessario per effettuare la connessione e nel fatto che il movimento di organi elettromeccanici causa disturbi sulle linee. Tali disturbi possono anche essere tollerati in una conversazione telefonica, ma non certamente in una trasmissione dati. Con le centrali elettroniche, invece, e soprattutto con quelle numeriche, i problemi di rumore sono scomparsi e si sono inoltre ottenuti altri grandi vantaggi: oltre alle elevate velocità di trasmissione ed agli elevati volumi di traffico gestibili, ottenibili grazie al fatto che il collegamento tra centrali numeriche viene effettuato tramite linee ad altissime velocità ed a larga banda (tipicamente fibre ottiche), sono possibili funzioni aggiuntive come il *log* (registro) degli errori, la contabilizzazione, i servizi aggiuntivi (voice mailing,...) e l'assistenza agli utenti (numero chiamato disabilitato, cambiato, ...). Non solo, ma risultano anche più agevoli gli interventi di gestione e manutenzione.

IL MESSAGE SWITCHING

Negli anni '60 e '70, il metodo più comune per commutare dati tra diversi utenti è stato il cosiddetto **message switching**, tra le cui principali applicazioni c'era la *posta elettronica*.
Il meccanismo è illustrato nella figura seguente:



Descrizione del meccanismo con cui avviene la consegna di un messaggio: 1) poll del terminale 2) invio messaggio 3) memorizzazione del messaggio in coda 4) estrazione del messaggio dalla coda 5) select del terminale destinatario 6) ACK da parte del terminale destinatario 7) consegna del messaggio

L'**unità di commutazione** è un computer, che riceve i messaggi dai terminali collegati (tramite linea dedicata o commutata), esamina l'indirizzo del destinatario indicato nella testata (*header*) del messaggio ed instrada infine il messaggio verso il destinatario.
A differenza delle centrali che realizzano la commutazione di circuito (come le centrali telefoniche prima esaminate), i sistemi basati su *message switching* sono di tipo **store and forward**: i messaggi in arrivo vengono memorizzati in code su disco, che possono essere più di una se vi sono diversi livelli di *priorità*. Non si tratta dunque di sistemi che lavorano in tempo reale, anche se in certi casi è possibile evitare l'accodamento di certi messaggi: infatti, i messaggi accodati ad alta priorità subiscono un tempo di coda inferiore a quelli a bassa priorità.
La memorizzazione in code su disco permette di gestire ordinatamente il traffico, dato che i messaggi a bassa priorità giunti durante un periodo di intenso traffico non vengono subito inviati, ma semplicemente memorizzati: solo quando il traffico lo consente, questi messaggi vengono estratti dalla coda e inviati.
Si tratta inoltre di sistemi di tipo **master/slave**: l'unità di commutazione periodicamente interroga i terminali (azione di *poll*), i quali, solo quando interrogati, possono inviare i propri messaggi; questi vengono accodati e, quando arriva il loro turno, vengono spediti (dopo aver fatto una operazione di *select* sul terminale destinatario, cioè dopo che si è verificato se tale terminale può ricevere oppure no).
Oltre a commutare i messaggi, il programma di controllo permette anche l'accesso dei singoli terminali ad *applicazioni conversazionali in tempo reale*. E' ovvio, però, che, in questo caso, le transazioni interessante evitano l'accodamento.
I limiti principali del message switching sono essenzialmente i seguenti:

- l'affidabilità dell'intero sistema dipende da quella del computer centrale (come del resto in tutti i sistemi costituiti da una stazione centrale cui fanno riferimento tutte le altre): per evitare che una caduta del computer centrale provochi l'interruzione del servizio, si provvedeva talvolta ad installare al centro un sistema duplicato (*mirror*), pronto a rilevare le funzioni del sistema principale qualora questo cadesse;

- in caso di carico trasmissivo alto, il sistema centrale può aumentare notevolmente il tempo impiegato dai messaggi per attraversare la rete;
- le linee di connessione ai terminali sono poco utilizzate.

In definitiva, sistemi di questo tipo presentano generalmente costi abbastanza alti rispetto al servizio fornito.

COMMUTAZIONE DI PACCHETTO (PACKET SWITCHING)

Negli anni '70 fu introdotta la tecnica della **commutazione di pacchetto** (**packet switching**), che aveva i seguenti obbiettivi fondamentali:

- multiplazione del traffico su più linee;
- bilanciamento del traffico;
- tempi ottimali di attraversamento della rete;
- alta affidabilità dei collegamenti (tramite l'uso di strade alternative);
- distribuzione dei rischio mediante l'uso di svariati nodi intermedi;
- condivisione delle risorse.

Vediamo di spiegare bene questi aspetti.

In primo luogo, la tecnica a commutazione di pacchetto prevede l'uso di diversi **nodi intermedi**, il che diminuisce l'incidenza della eventuale caduta di un singolo componente sull'efficienza dell'intera rete.

Il termine **pacchetto** significa semplicemente che la lunghezza massima dei messaggi è prefissata. Il pacchetto contiene, in una propria testata (*header*) di livello 3 (*livello network* della **pila ISO-OSI**), l'indirizzo del destinatario. Il generico nodo che riceve il pacchetto dall'utente mittente decide verso quale altro nodo intermedio inoltrare il pacchetto, a seconda dell'indirizzo del destinatario, degli **algoritmi di instradamento** di cui è dotato e delle condizioni di traffico della rete.

In queste reti sono dunque possibili percorsi alternativi, il che aumenta l'affidabilità dell'intera struttura: infatti, nel caso un determinato pacchetto debba passare per un determinato nodo e quest'ultimo, però, vada fuori servizio, è sempre possibile trovare per il pacchetto una strada alternativa che lo farà comunque arrivare al destinatario.

Inoltre, una qualsiasi linea tra due nodi, dopo essere stata utilizzata per l'inoltro di un pacchetto di un utente, viene successivamente utilizzata per trasmettere pacchetti di altri utenti: il traffico sfrutta quindi in modo condiviso le risorse di rete, massimizzando il rendimento delle linee. Sulle linee che collegano vari nodi viene resa possibile la trasmissione di pacchetti di utenti diversi, secondo una tecnica che, in pratica, risulta molto simile al Time Division Multiplexing (TDM).

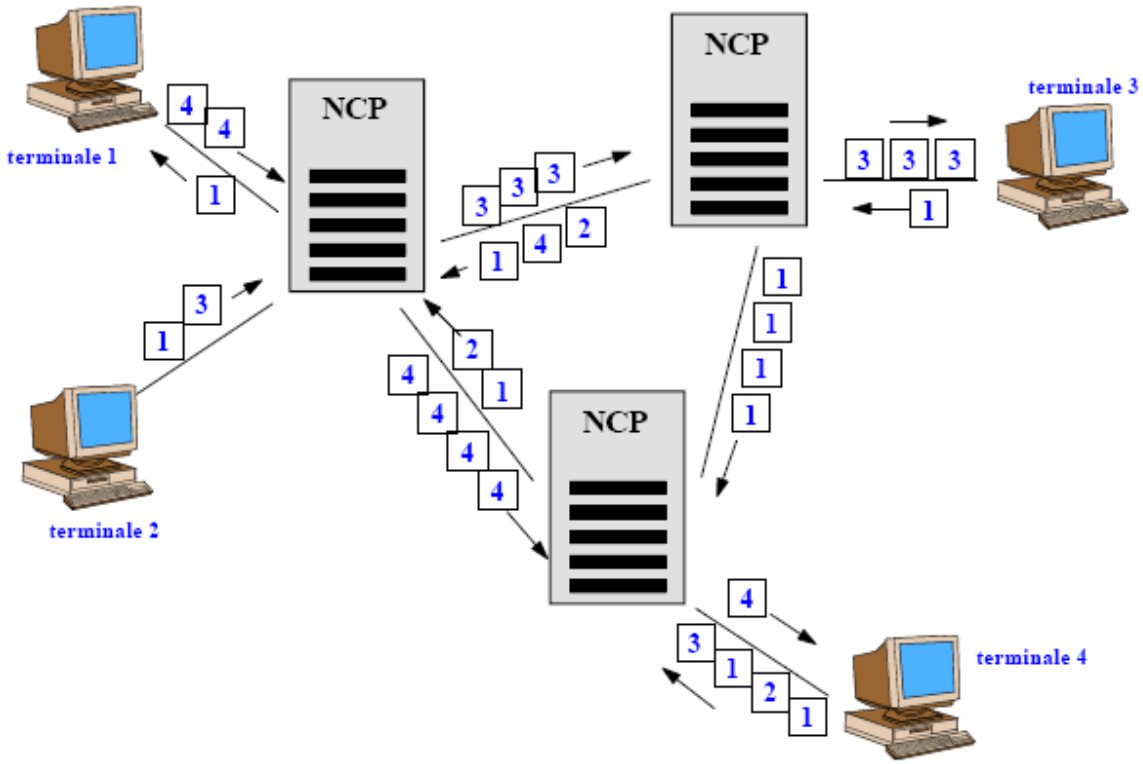
C'è anche un altro vantaggio, che è quello per cui i terminali possono essere collegati in modo permanente al nodo più vicino, eliminando, per quanto li riguarda, il tempo della *fase di chiamata*, che invece abbiamo visto essere una delle principali caratteristiche delle tecniche a commutazione di circuito.

Grazie all'**instradamento dinamico** (detto anche **adattativo**), il traffico complessivo risulta quindi bilanciato sulle varie risorse di rete, premessa anche per un buon tempo di attraversamento della rete.

L'utente ha la certezza che la spedizione dei suoi pacchetti verso un altro utente della rete venga effettuata, anche se non viene dedicato ai due alcun circuito reale: poiché la capacità di instradamento dei nodi assicura la consegna, nelle reti a commutazione di pacchetto si parla di connessioni tra due utenti tramite **circuito virtuale**.

Parecchie aziende ricorrono oggi alla **rete pubblica a commutazione di pacchetto** anziché a quella a commutazione di circuito, poiché tale rete è più affidabile della rete telefonica commutata e sono inoltre maggiori le velocità raggiungibili. Tra l'altro, nel caso di collegamenti su distanze elevate, si ha spesso un significativo vantaggio economico, in quanto sulle reti pubbliche a commutazione di pacchetto il costo del traffico è proporzionale al numero di pacchetti trasmessi e non

alla durata della connessione né alla distanza tra i due utenti connessi dal circuito virtuale. Ci sono poi aziende che, sempre per motivi di convenienza, hanno ritenuto utile impiantare una rete propria a commutazione di pacchetto.
Per comprendere i vantaggi della commutazione di pacchetto, principalmente in termini di utilizzazione delle linee, possiamo considerare la figura seguente:



Abbiamo qui un certo numero di terminali, numerati semplicemente con 1,2,3 e così via, connessi ad un certo numero di nodi di rete, detti **NCP** (*Nodi a Commutazione di Pacchetto*). Ciascun terminale è connesso ad un NCP tramite una linea dedicata permanente. Attraverso questa linea transitano i vari pacchetti (rappresentati da "mattoncini", ognuno contraddistinto dal numero corrispondente all'indirizzo del terminale destinatario), sia quelli spediti dal terminale sia quelli diretti al terminale.
I nodi di rete ricevono via via i pacchetti sia delle **linee d'utente** sia dalle **linee internodali**; memorizzano questi pacchetti e, dopo aver controllato la correttezza della trasmissione, li inoltrano sulle varie linee di uscita, a seconda dell'indirizzo del destinatario e degli algoritmi di instradamento.
Un caso particolare si ha quando un pacchetto appena ricevuto da un nodo è destinato ad una stazione d'utente collegata allo stesso nodo: in questo caso, non c'è alcun instradamento, se non quello sulla linea dell'utente destinatario.

L'INSTRADAMENTO

L'**instradamento in rete** consiste essenzialmente nell'utilizzo delle risorse (software, hardware o microcodice), da parte dei nodi di rete, per trasmettere i pacchetti attraverso la rete, fino alla stazione destinataria.
Un concetto generale dal quale partire è il seguente: una rete di telecomunicazioni, che mantiene connesse un certo numero di stazioni e utilizza per questo un certo numero di risorse, possiede una propria **capacità di smaltimento del traffico**, dove per *traffico* intendiamo sostanzialmente la

quantità di pacchetti che le stazioni immettono nella rete perché siano recapitati ai rispettivi destinatari, mentre per *smaltimento* intendiamo l'effettiva consegna dei pacchetti ai destinatari. E' allora ovvio che il traffico in ingresso alla rete dovrà essere sempre inferiore alla capacità di smaltimento della rete stessa. Viceversa, quanto più il traffico in ingresso si avvicina alla capacità di smaltimento della rete, tanto più la rete è "in difficoltà", in quanto si creano problemi di **congestione**; maggiore è la congestione e peggiori sono le prestazioni (in quanto aumentano notevolmente i tempi di ritardo con cui i pacchetti giungono a destinazione ed aumentano anche le probabilità di perdita dei pacchetti stessi). I problemi di congestione di una rete (o di un singolo nodo) possono essere affrontati in vari modi; per quanto riguarda la prevenzione dalla congestione, un ruolo sicuramente importante è svolto proprio dall'instradamento: quanto più efficienti sono gli algoritmi di instradamento, tanto meno probabile risulta il verificarsi di un congestionamento.
L'instradamento viene effettuato seguendo 3 criteri primari:

- assicurare sia il minor tempo di attraversamento sia il massimo rendimento della rete (**throughput**);
- ridurre al minimo il **costo** dell'attraversamento, cioè impegnare il minor numero di risorse;
- garantire un accettabile livello di sicurezza e di affidabilità.

Ci sono due possibili modi di effettuare l'instradamento: nell'**instradamento centralizzato**, esiste un unico centro (**NNC**, *Network Control Center*) che determina l'instradamento all'interno dell'intera rete; nell'**instradamento distribuito**, invece, ogni nodo di rete assume le decisioni sulle strade da seguire.
La soluzione centralizzata è quella più semplice da implementare, ma ha un punto debole ancora una volta nel fatto che la caduta della stazione principale (l'NNC appunto) comporta l'inutilizzabilità di tutta la rete. Una possibile soluzione a questo problema è quella di avere due NNC identici, di cui uno in funzione e l'altro di riserva. Se l'NNC attivo dovesse andare fuori-servizio, interviene l'altro a soppiantarlo. Lo svantaggio di questa soluzione è però che tutte le informazioni di controllo provenienti dai singoli nodi devono essere inviate ad entrambi gli NNC e devono essere memorizzate da entrambi gli NNC, il che costituisce un appesantimento della rete ed un raddoppio dei costi per gli NNC.
La soluzione distribuita è sicuramente più complessa di quella centralizzata (si pensi, ad esempio, ai problemi di coerenza tra le decisioni sull'instradamento prese dai diversi nodi), ma ha una maggiore affidabilità, proprio perché il tutto non dipende più da un'unica stazione centrale.
Dobbiamo ora capire con quali criteri, sia nel caso centralizzato sia nel caso distribuito, si possa determinare l'instradamento dei pacchetti. L'insieme dei parametri sui quali vengono decisi gli instradamenti è contenuto in apposite *tabelle*, che possono essere di due tipi:

- nelle **tabelle statiche**, le informazioni sui percorsi e sulle risorse vengono fissate al momento della generazione del software di sistema;
- nelle **tabelle dinamiche**, invece, ci sono aggiornamenti periodici delle informazioni.

E' ovvio che la soluzione preferibile è quella dinamica.
Nel caso del calcolo centralizzato dell'instradamento, tutti i nodi devono periodicamente mandare alla stazione centrale (NNC) le informazioni sul loro stato e sullo stato delle linee cui sono connessi. L'NNC, sulla base di queste informazioni, calcola il conseguente instradamento. Nel caso, invece, del calcolo distribuito dell'instradamento, la cosa è più complessa, perché non è proponibile che ciascun nodo debba conoscere lo stato dell'intera rete; infatti:

- in primo luogo, ogni nodo dovrebbe avere grosse risorse di memoria (le stesse che, nella soluzione centralizzata, ha l'NNC), il che può risultare poco conveniente da un punto di vista economico;
- in secondo luogo, il traffico delle informazioni sullo stato della rete diventerebbe un onere troppo pesante per la rete stessa.

Di conseguenza, si fa in modo che ciascun nodo riceva solo le informazioni relative ai nodi più

vicini, in modo che l'instradamento venga calcolato solo con riferimento ai nodi più vicini.
A proposito delle informazioni sullo stato della rete, abbiamo poco fa osservato che esse costituiscono in ogni caso un traffico addizionale che la rete deve gestire. Si tratta anche di un traffico molto "importante", nel senso che, senza queste informazioni, non sarebbe possibile indirizzare i pacchetti all'interno della rete. Si può allora pensare ad almeno due soluzioni per gestire questo traffico di informazioni di controllo:

- la soluzione più banale, ma che può anche risultare meno efficiente, è quella di inviare le informazioni di controllo sugli stessi mezzi trasmissivi usati dai dati veri e propri; in questo caso, però, bisogna necessariamente dare una sorta di "precedenza" a queste informazioni rispetto ai dati: si tratta sostanzialmente di attribuire una **priorità di trasmissione**, in base alla quale, se un nodo deve trasmettere sia dati sia informazioni di controllo, queste ultime saranno le prime ad essere inviate;
- una soluzione più efficiente, ma spesso meno economica, è invece quella di usare dei canali appositi (**canali di segnalazione**) per la trasmissione delle informazioni di controllo: ad esempio, se una data linea di comunicazione è gestita con la tecnica FDM (multiplazione a divisione di frequenza), uno o più canali saranno destinati alle informazioni di controllo e gli altri canali ai pacchetti veri e propri. In modo ancora più pratico, si può pensare ad una **rete dedicata**, che cioè trasmetta solo le informazioni di controllo: si tratterà, ovviamente, di una rete ad alta velocità (tipicamente in fibra ottica), ed è la soluzione adottata, per esempio, nella **rete GSM**, dove le informazioni di segnalazione viaggiano su una rete (cablata) completamente indipendente dalla rete su cui invece vengono trasmesse le conversazioni vere e proprie.

Un altro aspetto da considerare riguarda il criterio con cui viene scelto l'instradamento per un messaggio composto da più pacchetti:

- nella maggior parte dei casi, il percorso tra due stazioni utente viene scelto nel momento della definizione di una sessione tra di esse e mantenuto, salvo modifiche in seguito a problemi di rete, per tutta la durata della connessione; si parla in questo caso di **instradamento a circuito virtuale**;
- in altri casi (ad esempio in **Internet**, dove si usa lo schema **TCP/IP**), il percorso può essere determinato per ogni singolo pacchetto, per cui è possibile che i pacchetti di uno stesso messaggio seguano strade diverse per giungere alla stessa stazione destinataria (la quale, ovviamente, dovrà essere in grado di ricostruire la sequenza esatta, in quanto è possibile che un pacchetto trasmesso dopo arrivi invece prima). Si parla in questo caso di **instradamento a datagramma**: il concetto è quello di scegliere per ogni pacchetto l'instradamento ottimale, dato che lo stato della rete cambia in continuazione e quindi la scelta fatta per un pacchetto potrebbe non essere più quella ottimale per il pacchetto successivo.

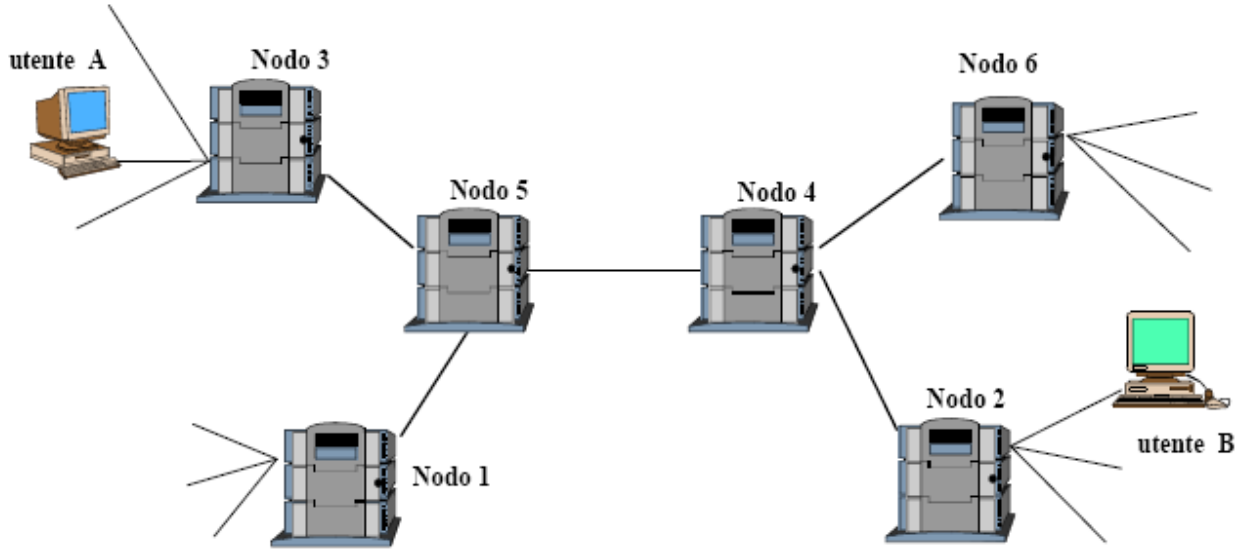
Non è detto che l'instradamento, anche per il singolo pacchetto, sia sempre ottimale. Infatti, bisogna tener conto che le informazioni sulla base delle quali l'instradamento viene calcolato non sempre sono aggiornate. Per rendercene conto, possiamo considerare un esempio molto semplice: mettiamoci nel contesto di un calcolo distribuito dell'instradamento, nel quale cioè sono i nodi a eseguire gli algoritmi di instradamento; supponiamo allora che il nodo A, dovendo inviare un certo pacchetto, vada a leggere le informazioni contenute nelle proprie tabelle e calcoli di conseguenza l'instradamento; anche se le tabelle sono state aggiornate un attimo prima, non è detto che lo stato dei nodi vicini sia rimasto nel frattempo invariato; per esempio, supponiamo che un nodo B vicino sia andato nel frattempo fuori servizio e abbia trasmesso la segnalazione di fuori-uso; questa segnalazione impiega un certo tempo per raggiungere A, per cui potrebbe arrivare dopo che A stesso ha letto le proprie tabelle; succede, perciò, che A calcoli l'instradamento "credendo" che B funzioni perfettamente, quando invece non è così. Lo stesso esempio si potrebbe anche fare nel contesto di un calcolo centralizzato dell'instradamento, dove anzi il problema è ancora maggiore, in quanto le informazioni di un nodo molto lontano dall'NNC possono impiegare parecchio tempo per arrivare fino all'NNC.

INSTRADAMENTO ADATTATIVO ARPANET

Arpanet è la rete sviluppata agli inizi degli anni '70 dal ministero della difesa degli Stati Uniti in collaborazione con enti di ricerca e università. Si tratta di una pietra miliare nella storia delle reti di computer: da essa è stato sviluppato il concetto di **commutazione di pacchetto** ed è ora incorporata nella più nota **rete Internet**. Arpanet ha visto la nascita della struttura TCP/IP ed è anche un interessante esempio di **instradamento distribuito dinamico** (o *adattativo*), dove le responsabilità sono cioè distribuite tra i nodi ed ogni nodo conosce, con un certo grado di aggiornamento, la situazione almeno dei nodi più vicini.

L'elemento base per le scelte è ancora una volta una **tabella**, che ogni nodo possiede e mantiene periodicamente aggiornata sulla base delle informazioni scambiate con gli altri nodi e di opportuni calcoli.

Consideriamo la stessa rete:



Preso un qualsiasi nodo, ad esempio il 5, la sua tabella sarà del tipo seguente:

Destinatario	Prossimo nodo	Ritardo
1	1	2
2	4	8
3	3	1
4	4	1
6	4	11
....	...	...

Per ogni destinatario, la tabella contiene, tra le altre informazioni, l'indicazione del nodo successivo ed il costo globale (inteso come tempo o come qualche altro parametro) per l'attraversamento. Per esempio, si nota che i pacchetti destinati ai nodi 3 e 4 vanno inoltrati agli stessi nodi 3 e 4, visto che essi sono adiacenti al nodo 5. Viceversa, un pacchetto destinato al nodo 6 andrà inoltrato al nodo 4, il quale disporrà di una ulteriore tabella e quindi di ulteriori indicazioni per l'inoltro verso il nodo 6 (che in questo caso è adiacente al 4).

Il generico nodo, tutte le volte che invia un pacchetto verso i suoi nodi adiacenti, calcola il **tempo**

di invio, che può essere valutato come l'intervallo tra tempo di spedizione del pacchetto e tempo di ricezione di ACK. Questi dati vengono memorizzati e periodicamente (ad esempio ogni 10 secondi), ne viene fatta una media, per ciascuna linea. Tale media è quella che viene inserita nella tabella.
Se, in un certo momento, viene rilevato un tempo di invio molto maggiore del valore medio, l'informazione viene comunicata (ad esempio tramite il packet-flooding, ossia con invio su tutte le linee di uscite) a tutti i nodi vicini. In questo modo, anche gli altri nodi vengono a conoscere i dati sperimentali delle varie tratte e sono quindi in grado di modificare le proprie tabelle. Questo è il motivo per cui si parla di **instradamento dinamico** o adattativo: le scelte possono cambiare ad ogni ricalcolo, secondo le attuali condizioni della rete.
Gli obbiettivi principali dell'instradamento dinamico sono sostanzialmente due:

- tenere conto delle mutevoli condizioni della rete;
- predisporre una tecnica (quanto più veloce possibile) che eviti di utilizzare un nodo che manifesti dei problemi: se un nodo dovesse risultare malfunzionante o congestionato, i tempi di attraversamento delle linee ad esso collegate vengono automaticamente impostati al valore massimo possibile, in modo che il nodo non venga più usato fin quando non torna in condizioni normali.

Questi discorsi mostrano una volta di più la necessità di avere dei fitti scambi di informazioni di controllo tra i nodi, con tutti i problemi che ne derivano e che sono stati già accennati in precedenza.
Arpanet ha però anche dei problemi:

- la tecnica adattativa è sicuramente quella più complessa;
- il carico dovuto al packet-flooding appesantisce i nodi;
- ogni tanto qualche pacchetto entra in loop o viene addirittura perso, perché transitato per nodi con tabella non aggiornate;
- pacchetti destinati allo stesso utente, ma viaggianti su percorsi diversi, possono arrivare in sequenza alterata rispetto a quella dell'immissione.

Quest'ultimo problema può essere affrontato sia a livello direttamente dell'utente, al quale viene demandato l'onere di ricostruire la sequenza, sia a livello dell'ultimo nodo: in questo caso, il nodo deve poter memorizzare tutti i pacchetti, al fine di ristabilire la sequenza corretta, dopo di che potrà inviare il messaggio all'utente. E' ovvio che la soluzione di lasciare il riordinamento all'utente consente di alleggerire i nodi.
Per quanto riguarda, invece, l'appesantimento dovuto al packet-flooding, si può pensare di rallentare i ritmi di calcolo delle strade: questo, però, se da un lato riduce il sovraccarico della rete e dei nodi per l'aggiornamento reciproco, ha l'ovvio inconveniente che i nodi risultano, in ogni istante, non perfettamente aggiornati sullo stato attuale della rete.

PROBLEMI TIPICI DELLE RETI A COMMUTAZIONE DI PACCHETTO

Esaminiamo rapidamente i problemi principali di una rete a commutazione di pacchetto, alcuni dei quali già esaminati in precedenza:

- *probabilità di perdita di pacchetti*: la perdita di un pacchetto può derivare sia dal fenomeno del **loop** sia anche da altre situazioni. Una di queste è la seguente: immaginiamo un pacchetto spedito dal nodo A verso il nodo B; il nodo B lo riceve ed invia il corrispondente ACK, ma, allo stesso tempo, non riesce ad inviare il pacchetto verso il nodo C, a causa di una interruzione della linea; in questa situazione, il nodo A non viene avvisato dei problemi riscontrati tra B e C, ma il pacchetto si ferma al nodo B. La soluzione più semplice a questo problema è che B invii l'ACK ad A solo dopo aver ricevuto l'ACK dal nodo destinatario. Un'altra possibilità è invece quella di prevedere un pacchetto specifico tra i due nodi finali, che indichi al mittente quali eventuali pacchetti di una sequenza non sono stati ricevuti;
- *probabilità di duplicazione dei pacchetti*: supponiamo che il nodo A abbia inviato a B un pacchetto e che B abbia risposto con il corrispondente ACK; supponiamo inoltre che, mentre B

provvede all'inoltro del pacchetto verso C, l'ACK inviato ad A venga perso; di conseguenza, il nodo A, non ricevendo alcuna conferma, deduce che il nodo B ha dei problemi, per cui ritrasmette lo stesso pacchetto su altre strade. Il risultato è che nella rete ci sono almeno 2 copie dello stesso pacchetto. Ancora una volta, si può aggirare l'inconveniente tramite una numerazione dei pacchetti: quando la stazione destinataria riceve due o più pacchetti uguali, ne considera uno solo e scarta gli altri;

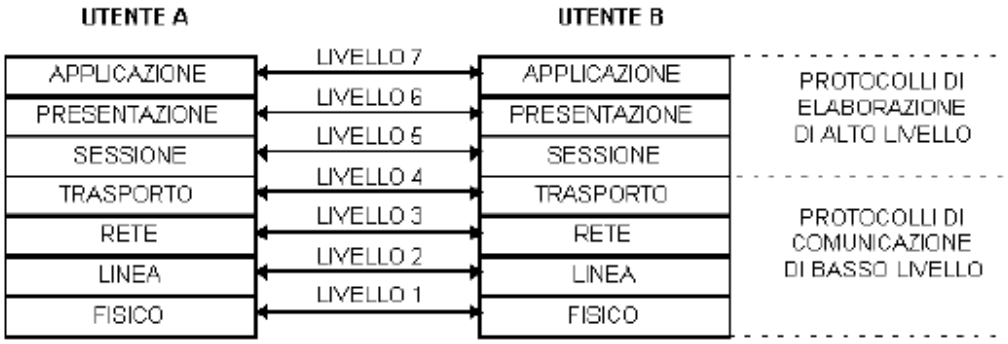
- *pericoli di sovraccarico dei nodi*: dei problemi legati al controllo della congestione parleremo diffusamente più avanti; per il momento, limitiamoci a dire che, per motivi vari, un dato nodo può avere un carico di lavoro eccessivo o troppe code sulle linee di uscita: in altre parole, esso sta ricevendo pacchetti in ingresso ad un ritmo superiore rispetto alle proprie possibilità di output. Questo comporta che la memoria del nodo arrivi a saturarsi: quando risulta piena, ogni ulteriore pacchetto in arrivo risulta perso. Per evitare la congestione (cioè appunto la saturazione di tutti i buffer di Input/Output), il nodo invia uno speciale pacchetto (detto **choke**) col quale invita i nodi vicini a rallentare l'emissione dei pacchetti verso di lui. Il problema è che questo rallentamento può provocare, per identici motivi, la congestione dei nodi vicini. Essi cercano quindi di rimediare allo stesso modo: la congestione si allarga a macchia d'olio e la rete rischia di non operare più. Si intuisce che l'unico rimedio realmente valido contro la congestione sia la prevenzione: bisogna tener continuamente sotto controllo la situazione e avvertire i nodi adiacenti in anticipo, pur continuando ad operare.
- *problemi legati ad operazioni di reset*: quando un nodo sperimenta una situazione di errore non rimediabile, esso emette un comando di **reset** verso gli altri nodi, ossia li invita ad interrompere e reinizializzare tutte le sessioni d'utente. In casi come questi, i pacchetti che sono ancora in rete e che appartengono a sessioni resettare vengono inevitabilmente perduti.

IL MODELLO ARCHITETTURALE OSI

Il collegamento e la cooperazione tra sistemi informatici che utilizzano sistemi operativi incompatibili tra loro è una delle principali esigenze del mercato attuale. I sistemi capaci di interagire tra loro, pur basandosi su sistemi operativi incompatibili, sono detti **aperti** quando permettono le comunicazioni in accordo con gli standard specificati nel modello generale **Open System Interconnection** (**OSI**). Questi standard sono stati definiti da una speciale commissione dell'**International Standard Organization** (**ISO**), ossia l'agenzia dell'ONU responsabile degli standard internazionali, inclusi quelli delle comunicazioni.
Questi standard sono nati come risposta alla diffusa esigenza di interconnettere tra loro sistemi incompatibili. La difficoltà di fondo consiste nel far comunicare tra loro due o più *processi*[1] che usano, internamente, regole e tecniche diverse. Ci si è preoccupati quindi di definire le strutture dei dati trasmessi, le regole e i comandi per la gestione dello scambio dati tra applicazione o tra utenti, i meccanismi di controllo che assicurano uno scambio senza errori.
Il **comitato ISO** ha stabilito le regole e le opzioni per tali interazioni, definendo un **modello di riferimento**. Un *modello di riferimento* è cosa diversa da un'*architettura di rete*:

- un *modello di riferimento* definisce il numero, le relazioni e le caratteristiche funzionali dei livelli, ma non definisce i protocolli effettivi;
- una *architettura di rete* definisce, livello per livello, i protocolli effettivi

Un modello di riferimento, quindi, non include di per sé la definizione di protocolli specifici, che invece vengono definiti successivamente, in documenti separati, come appunto accaduto dopo l'introduzione del **modello ISO/OSI**.
Tale modello suddivide le necessarie funzioni logiche in sette diversi *strati funzionali*, detti **layer** (*livelli*). La figura seguente, che in seguito sarà ampiamente commentata, illustra tali livelli:



- *Schema logico di due utenti (A e B) connessi tramite il modello OSI*

L'insieme dei 7 layer garantisce tutte le funzioni necessarie alla rete comunicativa tra sistemi, nonché una gamma molto ampia di funzioni opzionali (come ad esempio la compressione e la cifratura dei dati): in tal modo, si è in pratica suddiviso un compito complesso in un insieme di compiti più semplici.
I principi di progetto che furono seguiti durante lo sviluppo del modello OSI furono sostanzialmente i seguenti:

- ogni livello deve avere una funzione ben definita;
- la scelta dei livelli deve:
- minimizzare il passaggio delle informazioni fra livelli;
- evitare:

- troppe funzioni in un livello
- troppi livelli.

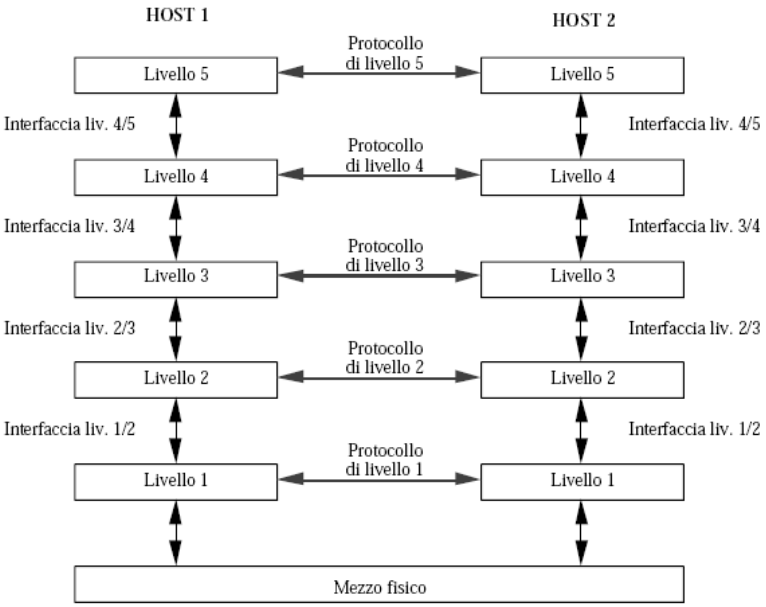
Il modello ha tre capisaldi:

- **simmetria**: la simmetria assicura che le *funzioni logiche* di due qualsiasi sistemi interagenti siano le stesse, in accordo con gli standard. La simmetria consente di bilanciare i carichi elaborativi sui vari sistemi ed assicura che ogni richiesta, da qualunque parte provenga, venga interpretata correttamente dalla controparte. Si tratta dunque della base per i processi elaborativi di tipo cooperativo e di tipo *client-server*;
- **struttura gerarchica**: i vari sottosistemi (livelli) sono organizzati in una rigida gerarchia operativa; ogni livello riceve i comandi ed i dati dal livello superiore, esegue per esso alcune sue specifiche funzioni e, a sua volta, chiede servizi al livello subordinato. Il livello gerarchicamente più alto è quello dell'applicazione (*application layer*), ossia dell'utente: tutte le funzioni ed i servizi del modello sono sempre attivati su esigenza di chi opera al livello più alto (*application layer*); gli altri 6 livelli, in ordine decrescente di gerarchia, sono così denominati: *presentation*, *session*, *transport*, *network*, *data link control* e *interfaccia fisica*;
- **modularità**: la modularità garantisce che ogni livello abbia ben definite non solo le proprie *funzioni interne*, ma anche le *interfacce* con cui riceve o trasmette i comandi ed i dati verso i propri livelli adiacenti. La definizione formale di queste relazioni assicura che ogni livello funzionale abbia una propria caratteristica precisa, che lo distingue nettamente dagli altri. Questo implica che, per aggiungere opzioni o per permettergli prestazioni migliori grazie a nuove tecnologie, non sia necessario modificare anche gli altri livelli, il che significa, in altre parole, garantire la possibilità di sviluppo, al crescere delle possibilità tecnologiche e delle esigenze degli utenti.

Soffermiamoci sulla struttura gerarchica. Per ridurre la complessità di progetto, le reti sono in generale organizzate a **livelli**, ciascuno costruito sopra il precedente. Fra un tipo di rete ed un altra, possono essere diversi:

- il numero di livelli;
- i nomi dei livelli;
- il contenuto dei livelli;
- le funzioni dei livelli.

C'è però un principio generale sempre rispettato: lo scopo di un livello è offrire certi **servizi** ai livelli più alti, nascondendo i dettagli su come tali servizi siano implementati..
Facciamo riferimento alla figura seguente:

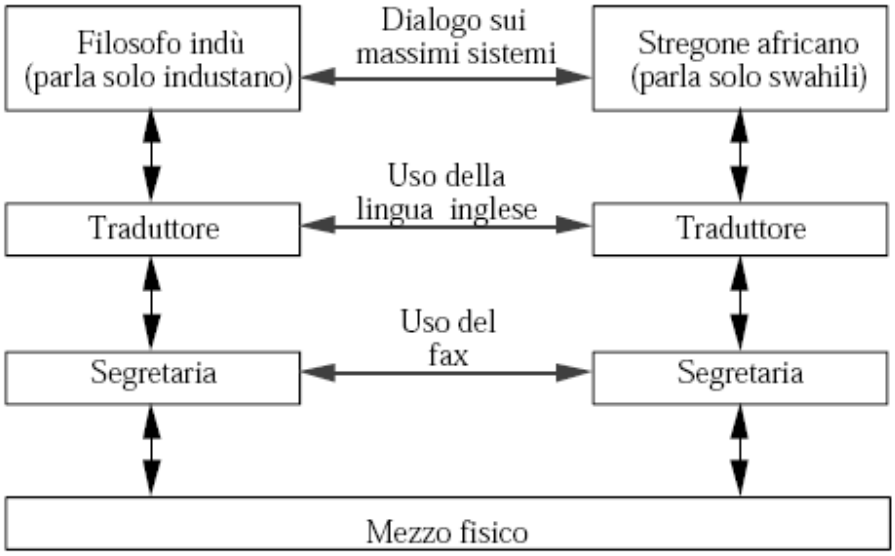


Dialogo fra peer entity (sono considerati, per semplicità, solo 5 livelli, mentre il modello OSI, come visto, ne prevede 7)

Il *livello n* su un host porta avanti una conversazione col *livello n* su di un altro host. Le regole e le convenzioni che governano la conversazione sono collettivamente indicate col termine di **protocollo di livello n**.
Le entità (processi) che effettuano tale conversazione si chiamano **peer entity** *(entità di pari livello)*. Il dialogo fra due *peer entity di livello n* viene materialmente realizzato tramite i servizi offerti dal livello (n-1).
E' bene però precisare subito (e lo faremo anche in seguito) che non c'è un trasferimento diretto dal livello n di host 1 al livello n di host 2: ogni livello di host 1 passa i *dati*, assieme a delle *informazioni di controllo*, al livello sottostante, fino al livello più basso, al sotto del quale c'è il mezzo fisico, attraverso cui i dati vengono effettivamente trasferiti da host 1 ad host 2. Quando arrivano a host 2, i dati vengono passati da ogni livello (a partire dal livello 1) a quello superiore, fino a raggiungere il livello n.
Fra ogni coppia di livelli adiacenti è definita una **interfaccia**, che caratterizza:

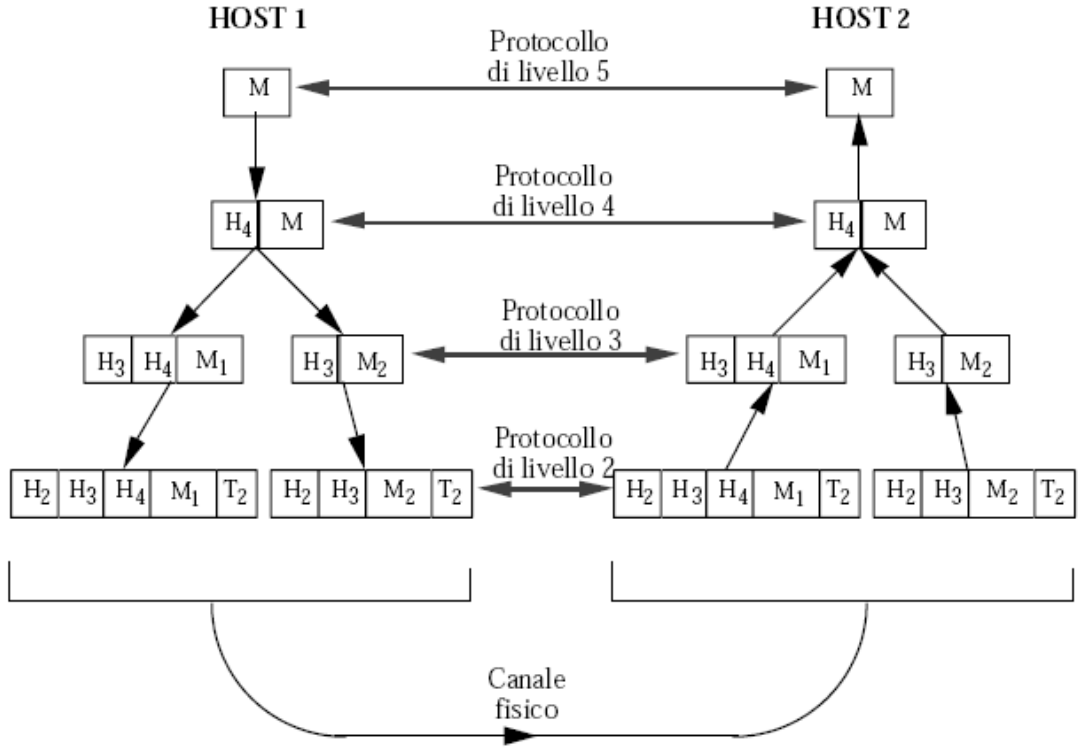
- le *operazioni primitive* che possono essere richieste al livello sottostante;
- i *servizi* che possono essere offerti dal livello sottostante.

Per comprendere ancora meglio i meccanismi basilari di funzionamento del software di rete, possiamo pensare alla seguente analogia umana, nella quale un *filosofo indiano* vuole conversare con uno *stregone africano*:



Dialogo fra grandi menti

Nel caso delle reti, la comunicazione fra le due entità di livello superiore avviene con una modalità che, almeno in linea di principio, è uguale in tutte le architetture di rete:



Flusso dell'informazione fra peer entità

Vediamo cosa accade:

- il programma applicativo (livello 5) dell'*host 1* deve mandare un *messaggio* M alla sua peer entity dell'*host 2*;
- il livello 5 consegna M al livello 4 per la trasmissione;

- il livello 4 aggiunge un suo **header** (*intestazione*) in testa al messaggio; questo header contiene informazioni di controllo, tra le quali il numero di sequenza del messaggio, la dimensione del messaggio e altro.
- il livello 4 consegna il risultato al livello 3;
- il livello 3 può trovarsi nella necessità di frammentare i dati da trasmettere in unità più piccole, (i cosiddetti **pacchetti**) a ciascuna delle quali aggiunge il suo header;
- il livello 3 passa i pacchetti al livello 2;
- il livello 2 aggiunge ad ogni pacchetto il proprio header (e magari un ***trailer***) e lo spedisce sul canale fisico;
- nella macchina di destinazione i pacchetti fanno il percorso inverso, con ogni livello che elimina (elaborandoli) l'header ed il trailer di propria competenza, e passa il resto al livello superiore.

Aspetti importanti sono i seguenti:

- le peer entity pensano concettualmente ad una comunicazione orizzontale fra loro, basata sul protocollo del proprio livello, mentre in realtà comunicano ciascuna solo col livello sottostante, attraverso l'interfaccia fra i due livelli;
- spesso i livelli bassi sono implementati in hardware o *firmware* (per ragioni di efficienza).

I COMPITI DI BASE DEI 7 STRATI FUNZIONALI

Al fine di descrivere i 7 strati funzionali (*layer*) di cui si compone il *modello ISO/OSI*, esaminiamo velocemente i vari comportamenti di una stazione connessa ad una rete (di qualsiasi tipo):

1) **livello applicativo**: il processo elaborativo elabora i dati in accordo sia alle richieste dell'utente sia alle norme applicative prestabilite. In determinati momenti, può capitare che certi dati o certe richieste vadano trasmessi ad una controparte remota: il processo elaborativo si occupa allora di preparare sia i dati da trasmettere sia i motivi della trasmissione.

2) **livello presentation**: i dati vengono strutturati in modo che il processo remoto possa comprenderli ed elaborarli;

3) **livello session**: il sistema esamina se la connessione logica con la controparte è stata già attivata o meno; in caso negativo, prima che venga attivata, è necessario disporre di regole (preventivamente fissate) per il dialogo da instaurare: ad esempio, si deve sapere se una delle due parti può interrompere l'altra oppure se una delle due si dovrà comportare da *slave* nei confronti dell'altra che farà da *master*. Se invece la connessione logica è stata già attivata, occorre esaminare lo stato della stessa, per stabilire se i dati preparati precedentemente possono essere trasmessi subito oppure è necessario attendere: ad esempio, nel caso di una *connessione master-slave* con tecnica di *poll*, una stazione slave deve aspettare che la stazione master effettui la sua interrogazione (cioè richieda se qualcuno deve trasmettere);

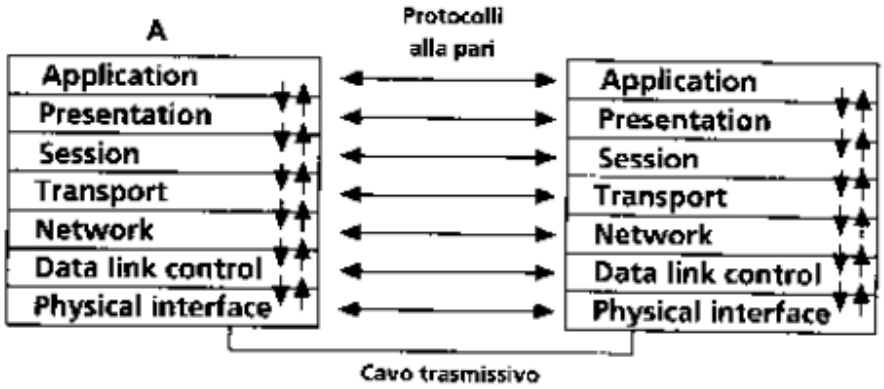
4) **livello transport**: prima ancora di effettuare la trasmissione, devono essere definiti una serie di dettagli tecnici, che dovranno essere in accordo con le regole del dialogo; ad esempio, tra questi dettagli tecnici citiamo il numero di sequenza del messaggio, la specifica se esso può essere suddiviso in rete durante la trasmissione, i provvedimenti da prendere se il messaggio arriva errato e altro ancora.

5) **livello di network**: appurati i dettagli tecnici della trasmissione, perché questa possa avvenire è necessario scegliere il percorso effettivo dei dati in rete, a meno che la scelta non sia stata fatta precedentemente e per tutti i messaggi della connessione. Ad esempio, possiamo pensare a 2 stazioni che possono essere collegate o tramite una *linea dedicata*, nel qual caso la scelta del percorso non si pone, oppure tramite un percorso di rete (utilizzando perciò dei *nodi intermedi*) che può variare ogni volta, cioè per ogni connessione, o anche nel corso della trasmissione stessa. Opportuni algoritmi di scelta (*instradamento*) determinano il percorso basandosi sulle strade di rete esistenti;

6) **data link controllo** (o *livello di protocollo di linea*): l'ultimo passo, prima della trasmissione

vera e propria, è quello di strutturare il messaggio secondo il formato previsto dal protocollo utilizzato sulla linea in uscita. Vanno anche definite le funzioni di controllo della trasmissione;
7) **livello di interfaccia fisica**: a questo punto, la trasmissione del messaggio può finalmente avvenire, per cui il messaggio viene passato all'*adattatore di linea*, il quale provvede ad inviare, uno alla volta, i singoli bit, in accordo con l'interfaccia fisica della linea utilizzata. La trasmissione avviene tramite una collaborazione dell'adattatore con il dispositivo DCE che collega il sistema alla linea trasmissiva (ad esempio il *modem*).
Lo schema appena descritto è molto generale. Di volta in volta, ad esso si possono aggiungere delle funzioni opzionali, tra le quali citiamo soprattutto la *cifratura dei dati*, i processi di *autenticazione* della controparte remota e simili.
Il modello OSI è intervenuto a formalizzare e generalizzare la sequenza logica descritta poco fa, includendo anche una discreta varietà di funzioni opzionali.

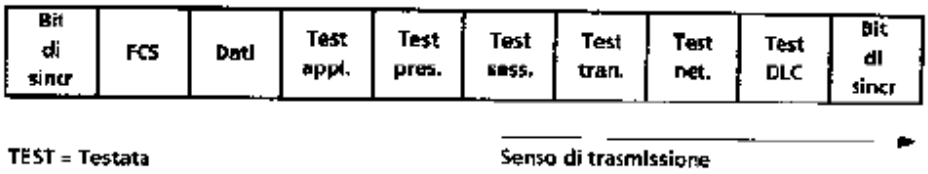
IL MODELLO DI RIFERIMENTO ISO/OSI
Il modello OSI può essere così rappresentato:



Quando c'è da effettuare una trasmissione tra i due utenti, ogni livello funzionale (**layer**) chiamato in causa effettua sostanzialmente tre operazioni:
- in primo luogo, esegue le funzioni richieste;
- successivamente, aggiunge a quanto ricevuto una **intestazione funzionale** (**header**), che è specifica del proprio livello ed è destinata ad essere interpretata dal proprio omologo sul sistema remoto. Questa testata serve per ottenere la cooperazione della controparte funzionale remota oppure per inviare una semplice informazione;
- infine, passa al proprio *strato subordinato* sia ciò che ha preparato sia l'opportuno comando funzionale.

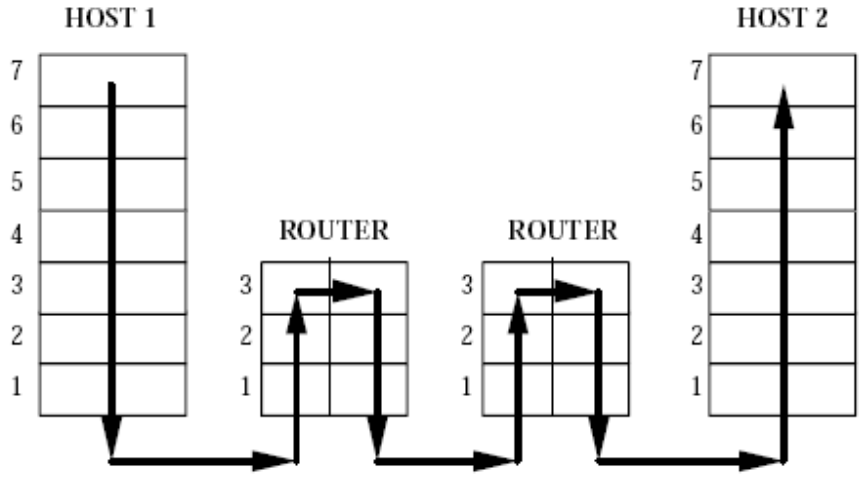
Ciascun livello è attivato a partire dal livello funzionale più elevato, che è quello applicativo (*application layer*); ogni livello, esaurite le proprie funzioni, chiama in causa il proprio subordinato, passandogli i dati ed un opportuno comando. Questo vale per tutti i livelli, il che significa che i dati ed il comando originati dal livello applicativo si arricchiscono, man mano che si scende verso il basso, di testate funzionali. Particolare è il comportamento degli ultimi due livelli:
- il penultimo livello, cioè quello del *protocollo di linea*, aggiunge la propria intestazione ma aggiunge anche una *coda* al messaggio, nella quale è incluso il campo per il controllo degli errori in trasmissione (**FCS**, *Frame Check Sequence*);
- l'ultimo livello, quello dell'interfaccia fisica con il mezzo trasmissivo, non aggiunge nulla al messaggio, almeno nella maggior parte dei casi, ma si occupa solo della trasmissione "brutale" dei bit sulla linea di trasmissione utilizzata.

La figura seguente chiarisce come risulta composto un messaggio che ha seguito i passaggi appena descritti:



Struttura del generico messaggio generato dai 7 livelli del modello ISO/OSI

Come evidenziato dalla freccia indicante il senso di trasmissione, il messaggio viene inviato in linea con i vari campi disposti in senso inverso a quello con cui sono stati generati. Questo serve in sede di ricezione: il sistema remoto (destinatario), infatti, riceve il messaggio al livello inferiore e lo ricostruisce bit per bit passandolo allo strato superiore, quello del controllo di linea; ogni livello, nel passaggio del messaggio verso l'alto, interpreta solo la testata di propria competenza, esegue le proprie funzioni in accordo con quanto specificato nella testata, memorizza eventuali risultati (da utilizzare, ad esempio, per una risposta al messaggio), elimina la propria testata e passa allo strato superiore ciò che rimane. In tal modo, al livello applicativo arrivano solo i dati ed il comando incluso nella *intestazione applicativa*, comando che riguarderà l'azione da eseguire sui dati. A questo punto, il sistema può preparare la propria risposta o, eventualmente, una propria richiesta, dopo di che il procedimento si ripete, ovviamente a ruoli invertiti.
Uno schema logico esemplificativo di quanto detto prima è il seguente:



Qui sono evidenziati non solo la sequenza logica con cui il messaggio viene costruito nell'host numero 1 e poi "interpretato" dall'host numero 2, ma anche il fatto che il messaggio, per la trasmissione vera e propria, ha generalmente bisogno di attraversare un certo *percorso di rete* prima di giungere al destinatario.
Il percorso di rete sarà composto da un certo numero di *nodi intermedi* (detti anche *elementi di commutazione* o **router**) e sicuramente ciascuno dei due host (sorgente e destinazione) è collegato alla rete tramite un proprio nodo intermedio.

DESCRIZIONE DEI SINGOLI LIVELLI DEL MODELLO ISO/OSI
Il modello OSI può essere discusso da vari punti di vista. Una prima descrizione è quella basata sulla figura 1, che riportiamo nuovamente:

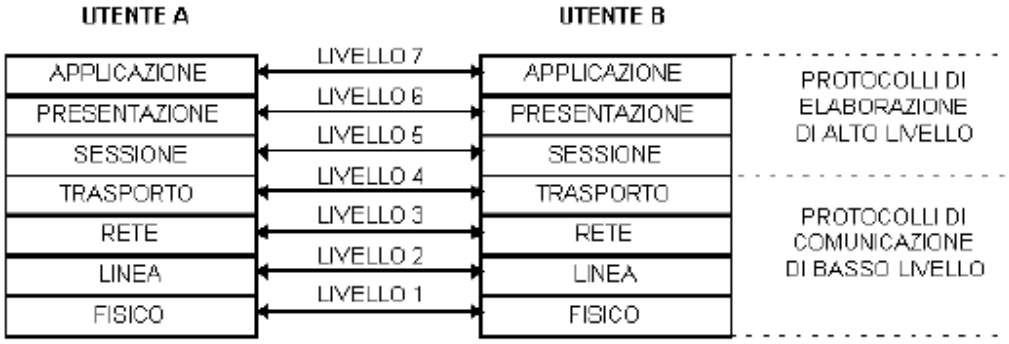
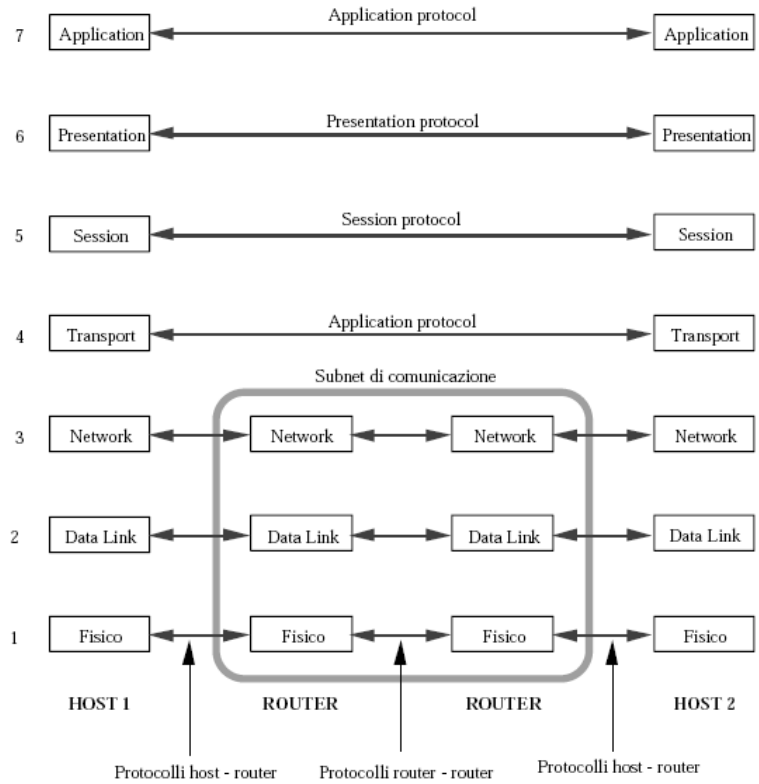


Figura 7 - Schema logico di due utenti (A e B) connessi tramite il modello OSI

In questa figura è evidenziata una possibile suddivisione dei 7 livelli:

- i quattro livelli inferiori (interfaccia fisica, protocollo di linea, network e transport) hanno funzioni prevalentemente trasmissive, cioè si occupano essenzialmente delle tecniche correlate alla trasmissione propriamente detta;
- al contrario i tre livelli superiori (session, presentation e application) sono caratterizzati da funzioni di interazione tra le due applicazioni o i due utenti finali, mentre non intervengono nelle tecniche trasmissive.

E' possibile però vedere la cosa da un altro punto di vista, leggermente diverso. Infatti, se consideriamo gli ultimi 3 livelli (livello di rete, livello di protocollo di linea, livello di interfaccia fisica), essi hanno una importante caratteristica: hanno funzioni che riguardano tutti i *componenti di rete* utilizzati durante una connessione, includendo, tra tali componenti di rete, anche eventuali nodi intermedi (o elementi di commutazione o **router**). Al contrario, le funzioni dei quattro livelli superiori (dal livello di transport al livello applicativo) non riguardano eventuali nodi intermedi, ma stabiliscono solo una cooperazione diretta tra i due sistemi interessati alla connessione. Possiamo allora proporre una ulteriore schematizzazione del modello:



Schema logico dettagliato della connessione di due utenti secondo il modello di riferimento OSI. Viene qui evidenziata una suddivisione tra i 3 livelli inferiori,comuni alla subnet di comunicazione, e i 4 livelli superiori, che appartengono solo ai due sistemi remoti

Come si nota, solo i tre livelli inferiori hanno a che fare con la cosiddetta **subnet di comunicazione** (***comunication subnet*** o semplicemente ***subnet***): essa comprende tutto ciò che serve a connettere tra loro i due utenti finali (i cosiddetti **end system**), in quanto ha il compito di trasportare messaggi da un end system all'altro, così come il sistema telefonico trasporta parole da chi parla a chi ascolta.
Si dice allora che i tre livelli inferiori hanno funzioni di tipo **box-to-box**, mentre quelli di livello superiore hanno funzioni di tipo **end-to-end**.
E' chiaro che la *relazione box-to-box*, tipica dei 3 livelli inferiori, ha una notevole importanza pratica: essa infatti implica che i protocolli usati a questo livello debbano essere comuni a tutte le *box intermedie* del percorso. Questo non è necessario, invece, per i 4 livelli superiori: infatti, per una *sessione* (o dialogo o connessione) tra i due sistemi, è necessario e sufficiente che le norme a questi 4 livelli siano comuni solo ai due sistemi finali.
Detto in altre parole, eventuali nodi intermedi dovranno essere coerenti, con i due sistemi finali, solo per quanto riguarda gli strati trasmissivi: questo è garantito dall'uso di ciò che in figura è stato indicato come **protocolli host-router**. Non solo, ma è chiaro, come si evince dalla figura, che i livelli superiori non sono chiamati in causa nei nodi intermedi; di conseguenza, i nodi intermedi, qualora avessero capacità elaborative autonome, potrebbero sicuramente usare livelli funzionali di tipo end-to-end del tutto diversi da quelli della connessione per cui operano come nodi di transito.
Infine, nella comunicazione tra router e router, non è necessario che la comunicazione avvenga con gli stessi protocolli usati nella comunicazione tra router e host, ma possono essere usati altri protocolli (**protocolli router-router**), purché compatibili con i precedenti.
Da quanto detto, si evince che, tra i 7 livelli, una posizione particolare è assunta dal *livello di transport*, che si trova esattamente a metà delle due classificazioni fatte prima: si è visto, infatti, che tale livello da un lato ha funzioni correlate alla trasmissione ed è quindi assimilabile ai 3 livelli inferiori, ma, dall'altro lato, ha anche funzioni riguardanti il rapporto tra i due sistemi interessati alla

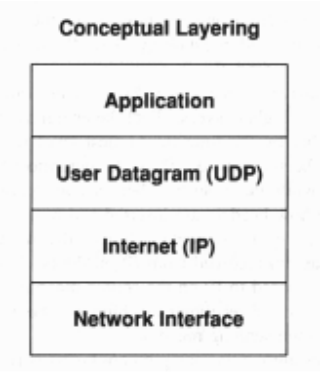
connessione, il che lo rende assimilabile anche ai 3 livelli superiori, quelli che gestiscono il dialogo. Possiamo allora affermare quanto segue:

- il compito primario del livello di transport è quello di scegliere le più idonee procedure di controllo della trasmissione, affinché il traffico venga gestito in modo appropriato. Esso deve tener conto sia delle esigenze applicative sia delle caratteristiche di qualità della rete trasmissiva utilizzata; a questo scopo, il livello di transport mette a disposizione dell'utente un certo numero di **classi di trasporto**, cioè diverse modalità (ciascuna con le proprie opzioni) per il controllo sulla trasmissione e l'utente può scegliere quella che preferisce;
- sempre in termini di controllo sulla trasmissione, il livello di transport si occupa anche delle procedure per la gestione degli errori, della sequenza dei messaggi, della lunghezza dei messaggi e simili.

IL PROTOCOLLO TCP/IP
(TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL)

UDP (User Datagram Protocol)

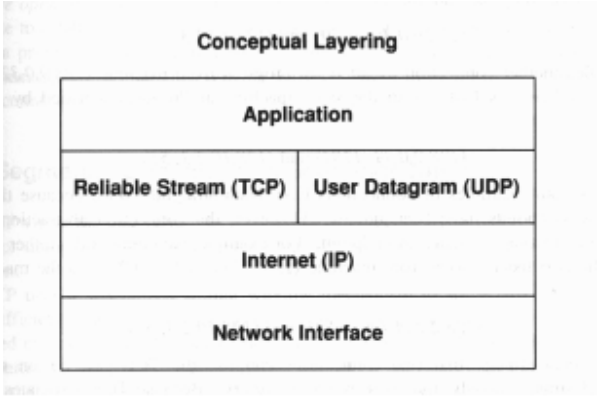
Nel complesso del protocollo TCP/IP, l'**User Datagram Protocol** (UDP) fornisce un servizio di recapito dei datagrammi connectionless ed inaffidabile, usando l'IP per trasportare messaggi da una macchina ad un'altra; prevede delle porte di protocollo usate per distinguere tra più programmi in esecuzione (o *processi*) su una singola macchina.
E' un protocollo di trasporto che si colloca sopra l'Internet Protocol Layer (IP):



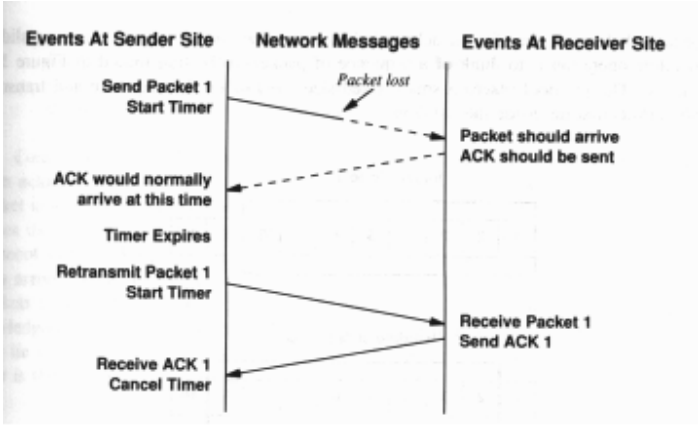
E' simile all'IP, ma oltre ai dati spediti, ciascun messaggio UDP contiene sia il numero di porta di destinazione che quello di origine, rendendo possibile al software UDP di destinazione di recapitare il messaggio al corretto ricevente (programma od utente), ed a quest'ultimo di inviare una replica. L'inaffidabilità è quella propria dell'IP, in quanto l'UDP non prevede nessun protocollo per il controllo dell'errore, a differenza del TCP: non usa acknowledgement per assicurare al mittente che i messaggi siano arrivati, non dispone le sequenze di datagrammi in ordine e non fornisce una retroazione per il controllo del rate del flusso di informazioni tra macchine. Perciò i messaggi UDP possono essere persi, duplicati oppure arrivare fuori dall'ordine; inoltre i datagrammi possono arrivare più velocemente di quanto il ricevente sia in grado di processarli.

TCP (Transmission Control Protocol)

Il **Transmission Control Protocol** (TCP), si assume la responsabilità di instaurare un collegamento tra due utenti, di rendere affidabile il trasferimento di dati e comandi tra essi ed infine di chiudere la connessione. Esso è capace di trasferire un flusso continuo di dati fra due utenti in entrambe le direzioni (*full-duplex*), decidendo quando bloccare o continuare le operazioni a suo piacimento. Poiché il TCP fa veramente poche assunzioni riguardo l'hardware sottostante, è possibile implementarlo sia su una singola rete come una **ethernet** sia su un complesso variegato quale l'**internet**.
Tale protocollo, come l'UDP, si colloca, nel modello a strati, sopra l'Internet Protocol Layer (IP), che gestisce il trasferimento e l'instradamento del singolo pacchetto fino a destinazione, ma, come ulteriore funzionalità, tiene una traccia di ciò che è stato trasmesso ed eventualmente ritrasmette quella parte di informazione che è andata perduta lungo il tragitto.



Come l'UDP, il TCP permette a più programmi applicativi su una stessa macchina di comunicare contemporaneamente, e demultiplexa il traffico dei pacchetti in ingresso a tali programmi; usa i numeri di **porta** per identificare la destinazione finale all'interno di una macchina. La fondamentale differenza con l'UDP è che il TCP garantisce un servizio di trasporto **affidabile** (*Reliable Delivery Service*), ponendo rimedio alle cause di inaffidabilità proprie dell'IP (duplicazione e perdita di dati, caduta di rete, ritardi, pacchetti ricevuti fuori ordine, etc.), anche se ciò comporta una implementazione più complessa. L'importanza dell'affidabilità del flusso permessa da tale protocollo è il motivo per cui il complesso del protocollo TCP/IP ha tale nome.
L'affidabilità di questo servizio è caratterizzata da cinque proprietà:
Stream Orientation: quando due programmi applicativi trasferiscono dati (*stream of bits*), il flusso nella macchina di destinazione passa al ricevente esattamente come è stato originato nella macchina sorgente.
Virtual Circuit Connection: dal punto di vista del programmatore e dell'utente, il servizio che il TCP fornisce è analogo a fornire una connessione dedicata.
Buffered Trasfer: i routers interessati dal trasferimento sono provvisti di buffers per rendere più efficiente il trasferimento e minimizzare il traffico di rete.
Unstructured Stream: il TCP/IP stream service non adotta un flusso di dati strutturato; ovvero non c'è modo di distinguere i records che costituiscono il flusso dati.
Full-duplex Connection: la connessione fornita dal TCP/IP stream service permette un trasferimento di flusso contemporaneo ed indipendente in entrambe le direzioni, senza apparente interazione.
Se un qualunque messaggio è troppo grande per un singolo pacchetto TCP (gli standard consigliano una dimensione di **576 byte** compreso l'header del IP) si procede a dividerlo in segmenti di lunghezza fissa e poi, arrivato a destinazione, si controlla che siano in ordine e si riassemblano, in modo che tale operazione risulti del tutto invisibile ai due utenti.
Poiché queste funzionalità sono necessarie per molte applicazioni, sono state messe tutte insieme in questo protocollo piuttosto che inserirle, come parte del programma, in ogni applicativo che ne ha bisogno.
L'affidabilità è garantita da una tecnica di fondamentale importanza nota come **acknowledgement with retransmission** (riscontro con ritrasmissione). Tale tecnica prevede che il destinatario invii un messaggio di acknowledgement (**ACK**) al mittente, una volta ricevuto un pacchetto. Il mittente mantiene una copia di ciascun pacchetto spedito e la rimuove dal buffer di trasmissione solo dopo aver ricevuto l'ACK relativo ad essa.
Nella configurazione più banale e meno efficiente l'utente sorgente, dopo aver trasmesso un pacchetto, aspetta di ricevere il suo ACK prima di spedire il successivo; inoltre fa anche partire un "cronometro" per il **timeout**, allo scadere del quale, se non ha ricevuto risposta, ritrasmette quello stesso pacchetto:



Le porte del protocollo TCP

Le porte del TCP sono molto più complesse rispetto a quelle dell'UDP, perchè un dato numero di porta non corrisponde ad un singolo oggetto. Infatti nel TCP gli oggetti da identificare sono delle connessioni di circuito virtuali tra due programmi applicativi, e non delle particolari porte. Il TCP usa la connessione, e non la porta di protocollo, come sua fondamentale astrazione; le connessioni sono identificate da una coppia di **end points**, ognuno dei quali è costituito da due interi **host,port**, dove l'*host* è l'indirizzo IP dell'host e *port* è il numero di porta TCP su quell'host (per esempio: l'end point **128.10.2.3,25** specifica la porta 25 sulla macchina di indirizzo 128.10.2.3). Poichè il TCP identifica una connessione con una coppia di valori, uno dato numero di porta può essere condiviso da più connessioni su una stessa macchina, senza che si crei ambiguità. Perciò la macchina identificata da **128.10.2.3,53** può comunicare simultaneamente con le macchine identificate da **128.2.254.139,1184** e **128.9.0.32,1184**.
Si possono così creare servizi concorrenti con connessioni multiple simultanee, senza dover riservare un numero di porta locale per ogni connessione. Per esempio, alcuni sistemi forniscono un accesso concorrente al loro servizio di posta elettronica, permettendo a più utenti di spedire un E-mail contemporaneamente.