1- INTRODUZIONE

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

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

Caratteristiche di una rete:

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

Modalità di comunicazione:

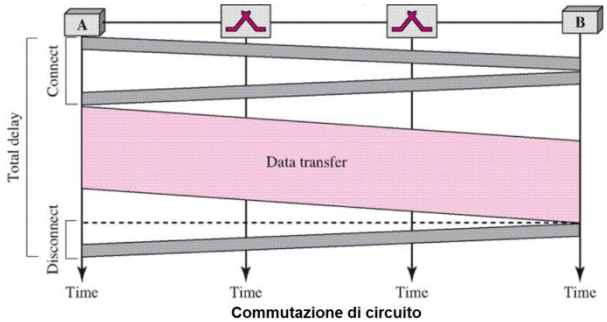
- *punto-punto*: comunicazione tra due soli dispositivi ;
- *multi-punto*: comunicazione multicast (o broadcast).

Topologie:

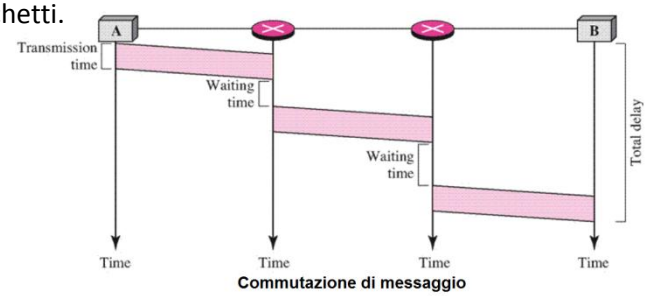
- *Maglia*: $O(n^2)$ collegamenti tutti dedicati ma scarsamente utilizzati, poco sensibile a guasti. Da usare quando le cose che contano sono l'affidabilità e le prestazioni.
- *Stella*: riduce il numero di collegamenti, resistente ai guasti ma ha un punto critico sul nodo centrale.
- *Bus*: supporta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

Commutazioni:

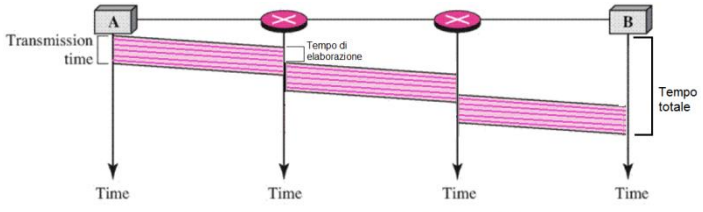
- **Circuito**: caratterizzato dalle fasi di *set-up* (individuazione del percorso), *utilizzo* (scambio dati dedicato) e *abbattimento* (rilascio risorse). Consigliata per comunicazioni di emergenza o massicce.
  Non è previsto lo *store&forward*, quindi deve esserci compatibilità fisica lungo tutto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



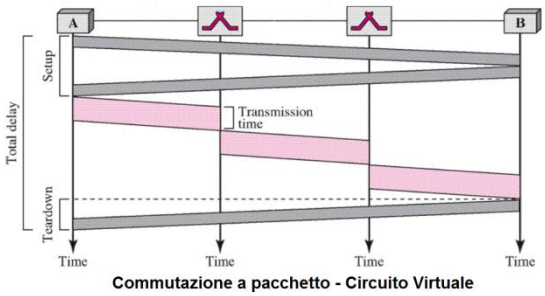
Commutazione di circuito



Commutazione di messaggio



Commutazione a pacchetto - Datagramma



Commutazione a pacchetto - Circuito Virtuale

4- RETI PER TRASMISSIONE DATI

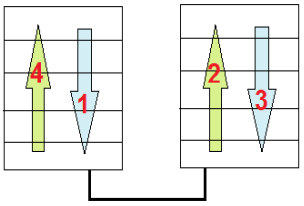
Un *flusso dati* è un flusso di bit che tramite tecniche di trasmissione dati (modulazione, codifica,ecc…) viene messo in grado di attraversare un canale di comunicazione per connettere sorgente e destinazione.
Le caratteristiche di un traffico dati sono:

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

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

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

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



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

Modello ISO/OSI:

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

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

TCP/IP

Suite protocollare alla base di Internet .

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

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

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

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

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

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

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

Alcuni indirizzi IP sono riservati:

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

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

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

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

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

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

IPv6 comprende 3 tipi di indirizzamento:

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

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

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

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

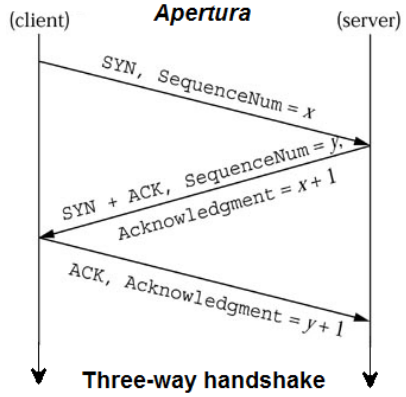
➢ Livello Trasporto

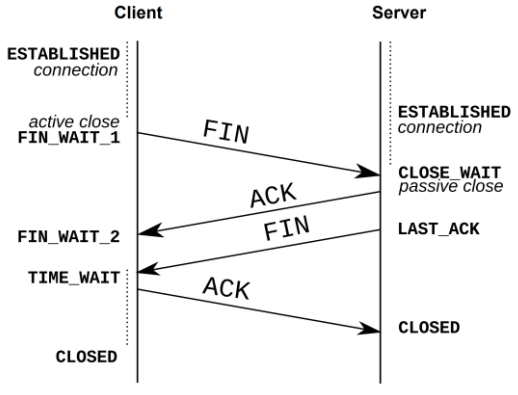
UDP: lo header UDP comprende *numero porta sorgente* e *destinatario*, *lunghezza del campo dati* e *checksum* (opzionale). Supporta comunicazioni connectionless non prevedendo una fase di setup; non è inoltre affidabile.

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

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

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





IEEE 802

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

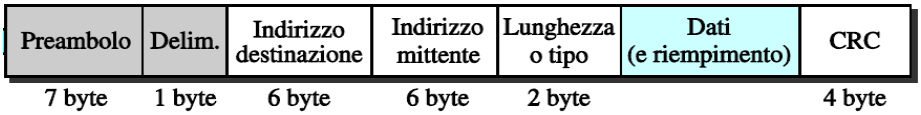
- **Logical Link Control** (**LLC**): Definisce uno standard che si può interfacciare con tutti i MAC di livello inferiore e i protocolli di rete superiori. Esso si occupa di sovrintendere al *trasferimento di pacchetti tra due nodi adiacenti* effettuando controllo di flusso, integrità ecc.. Può essere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

6- RETE ETHERNET

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

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

Formato frame:

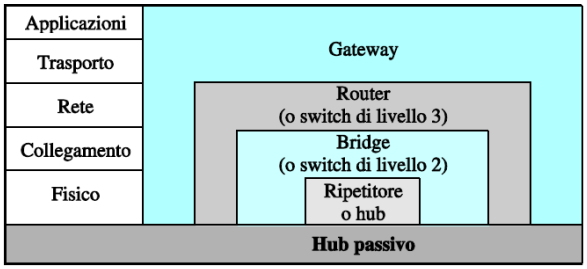


In tutte le generazioni di Ethernet, il formato del frame è identico.
E' imposta una lunghezza minima del frame (64 byte) e lunghezza massima (1518 byte). Per quest'ultima le motivazioni sono: migliore gestione dei buffer ; evitare la monopolizzazione del canale.

Livello Fisico: cambia in relazione alla generazione Ethernet considerata:

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

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



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

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

15- SICUREZZA NELLE RETI

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

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

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

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

La crittografia si distingue in:

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

Crittografia a chiave simmetrica

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

Crittografia a chiave pubblica (asimmetrica)

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

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

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

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