Indice

0 Introduzione

Comunicazione : scambio di informazioni fra due o più dispositivi grazie all'uso di mezzi trasmissivi.

Protocollo : insieme di regole che consentono la comunicazione.

Flussi Trasmissivi : tra mittente e destinatario può essere di tre tipi:

- **Simplex**: solo uno dei due dispositivi spedisce informazioni mentre l'altro le riceve
- **Duplex**: ogni dispositivo può sia trasmettere che ricevere, ma non contemporaneamente
- **Full Duplex**: ogni dispositivo può sia trasmettere che ricevere contemporaneamente tramite due collegamenti fisici

DTE Data Terminal Equipment, dispositivo che permette la comunicazione dati (es. computer)

DCE Data Communication Equipment, dispositivo che consente di convertire i segnali nella forma migliore per l'invio sul canale di comunicazione tra due DTE

Reti insieme di dispositivi connessi da canali di comunicazione. Due modelli di elaborazione:

- *Reti ad elaborazione* ***concentrata***: un potente DTE è messo a disposizione per più DTE che ne sfruttano la capacità di calcolo
- *Reti ad elaborazione* ***distribuita***: il calcolo da elaborare è suddiviso in più sotto problemi affidati ognuno a diversi nodi della rete

Informazione : grandezza misurabile con unità di misura il **bit**. Vale:
$Q=\log_2 M$
con M= *numero dei possibili stati di un sistema,* Q= *bit necessari per distinguerli.*

Carattere : in un sistema elaborativo è associato a sequenze significative di bit, esistenti all'interno di *codici* come:

- BCD (Binary Decimal Code)
- EBCDIC (Extended Binary Coded Decimal Code): codice a 8 bit
- ASCII (American Standard Code for Information Interchange): *codice a 7 bit*

0.1 Qualità rete

0.1.1 Affidabilità

Affidabilità: capacità di una rete di consegnare l'informazione priva di errori, rimediare a malfunzionamenti ed essere robusta.

0.1.2 Sicurezza

Sicurezza: protezione dei dati della rete, al fine di impedire accessi non autorizzati, modifiche non autorizzate o perdite dati.

0.1.3 Prestazioni

Valutate misurando:

- **Ritardo**: tempo di transito dei dati
- **Tempo di risposta**: intervallo di tempo compreso tra la richiesta e l'arrivo della risposta
- **Throughput**: quantità effettiva di dati spediti nell'unità di tempo *(velocità)*
- **Banda**: banda passante di frequenze utilizzabili per la trasmissione di segnali *(massima velocità di trasferimento)*

Valutazione velocità

- **ping**: il comando `ping` indica se un host remoto è raggiungibile; Usa l'*echo message* del protocollo ICMP per *forzare l'host a rispedire indietro il pacchetto inviato.*
- **traceroute**: comando che indica l'instradamento dei pacchetti in uscita, visualizzando tutti i dispositivi di rete attraversati
- Speedtest, Pingtest, NetIndex, Ne.Me.Sys

0.2 Tipi di rete

0.2.1 PAN

Personal Area Network: Rete personale che non si estende per più di 10-20 metri. Il termine si riferisce propriamente a reti con connessioni via cavo.

0.2.2 LAN

Local Area Network: Rete con un raggio limitato ad una abitazione o un edificio.

0.2.3 WAN

Wide Area Network: Rete che copre ampie aree geografiche connettendo tra loro più sottoreti locali.

0.2.4 MAN

Metropolitan Area Network: Rete metropolitana caratterizzata da una velocità di trasmissione molto elevata (tipicamente fibra ottica).

0.2.5 GAN

Global Area Network: Internet.

0.3 Topologie di rete

0.3.1 Rete a dorsale

I dispositivi sono connessi tutti ad una via di trasmissione principale detta appunto dorsale. Un'interruzione in qualunque punto della dorsale compromette tutta la rete.

0.3.2 Rete ad albero

La trasmissione avviene in modo gerarchico tra i nodi padre e figlio, fino ad arrivare alla *root* dalla quale dipende tutto il funzionamento della rete.

0.3.3 Rete a stella

Tutti i dispositivi sono connessi ad un *hub* (router o swith di rete), più economico da sostituire in caso di rottura. Inoltre, in caso di danni ad un cavo, viene disconnesso solo un terminale. Questa topologia è tipicamente utilizzata per la realizzazione di reti LAN.

0.3.4 Rete ad anello

Le informazioni vengono passate da un dispositivo all'altro in modo ciclico, la trasmissione è unidirezionale anche se questo si può ovviare con un secondo anello in direzione opposta. Questa topologia era tipica delle LAN TokenRing, ora viene utilizzata principalmente nelle MAN in fibra ottica.

0.3.5 Rete a maglia

In inglese mesh, è una rete in cui ogni dispositivo può essere connesso ad ogni altro dispositivo ottenendo di fatto un grafo connesso. É la topologia di rete meno vulnerabile, ma è poco utilizzata nelle reti cablate a causa dei costi. É diffusa invece nelle WLAN, spesso nella versione *ad hoc*, dove i collegamenti nascono e muoiono dinamicamente. In questa topologia il Routing viene effettuato da ogni nodo.

0.3.6 Grid

Rete di computer incentrata sulla condivisione dinamica delle risorse, nel contesto di calcolo distribuito e HTC (*High Troughput Computing*).
Rispetto ad un cluster, la grid coinvolge anche l'hardware, grazie a librerie *middleware* (*software glue*) che si collocano tra il S.O. e lo strato fisico della macchina.

0.4 Protocolli Basilari

0.4.1 RTS-CTS

Request-to-Send/Clear-to-Send, basato su interfaccia seriale.
Una *stazione* che vuole trasmettere, invia una richiesta (**RTS**) alla stazione master; Se il *master* può rispondere, autorizza (**CTS**) la stazione a trasmettere, la quale invierà i suoi dati e solo dopo il master le invierà un messaggio di ACK.
Questa procedura temporizzata è detta **handshake**.

0.4.2 XON-XOFF

Usato tra DTE/DCE vicini;
Quando una stazione trasmette dati, questi vengono memorizzati dalla stazione che li riceve. Se la *memoria di questa stazione è piena*, questa invierà a quella che trasmette il carattere **XOFF** per impedirne la trasmissione. Appnea il DTE torna in grado di memorizzare, invia il carattere **XON** e la stazione primaria torna a trasmettere.

0.4.3 ARQ

Automatic Repeat Query, protocollo *Full-Duplex*.
Usa il concetto di finestra scorrevole[1], dividendo il messaggio spedito in sequenza di **frame**: ARQ spedisce più frame prima di ricevere un *riscontro* (costituito da due numeri di sequenza -0 e 1- per distinquere un frame da quello successivo).

Gestione errori

- **Go-back-N**: un frame danneggiato implica la ritrasmissione di N frames
- **Selettivo**: vengono ritrasmessi solo i frames danneggiati

[1] La *finestra del mittente* specifica i frame che può spedire in quel momento, *la finestra del destinatario* quelli che può ricevere

1 Livello Fisico

1.1 Segnali

Lo strato fisico è adibito al trasporto dei segnali (e quindi dei dati); questi, proprio come i dati, possono essere **analogici** *(rappresentazione continua nel tempo, con infiniti livelli di intensità)* o **digitali** *(rappresentazione discreta, con livelli di intensità pari a 0 o 1)*

Segnali analogici segnale sinusoidale che varia nel tempo secondo la legge:

$$u = Usen(\omega t + \varphi)$$

dove

- u è l'ampiezza istantanea
- U è l'ampiezza massima
- ω è la *velocità angolare*, ovvero la variazione dell'angolo nel tempo, espressa in radianti al secondo
- φ è la *fase*, ossia lo sfasamento rispetto all'origine, espresso in radianti
- t è il tempo (variabile)
- T è il *periodo*, cioè l'intervallo di tempo (in secondi) impiegato dall'onda per effettuare un'oscillazione completa
- $f = 1/T$ è detta *frequenza*, misurata in Hz $(1/s)$
- $\lambda = c/f = cT$ è detta *lunghezza d'onda* (dove c è la velocità di propagazione del segnale), ossia la distanza tra due massimi relativi.

Spettro: insieme di frequenze che un segnale contiene;
Larghezza di banda: intervallo di frequenze contenute in un segnale composto (max_Hz - min_Hz)

Teorema di Fourier afferma che un segnale può essere rappresentato come somma di sinusoidi (potenzialmente infinite) con caratteristiche differenti.
Quindi un segnale digitale altro non è che un segnale analogico composto da banda (teoricamente) infinita. Il problema è come trasmettere un segnale analogico in formato digitale tra due punti. Due possibilità: Trasmissione del segnale di base o con modulazione del segnale

1.1.1 Filtri

Sistema che blocca o lascia passare diversi range di frequenze di un segnale.
Possono essere *attivi o passivi* a seconda dei materiali utiizzati. Abbiamo 4 tipi di filtro:

Filtro passa basso permette il passaggio di frequenze **al di sotto** di una determinata frequenza (*frequenza di taglio*), la quale in genere è scelta in base alla relazione V_out/V_in = $(1/2)^{1/2}$, con V_out e V_in rispettivamente segnale in uscita ed in ingresso nel filtro.

Filtro passa alto permette il passaggio di frequenze al di sopra della *frequenza di taglio*.

Filtro passa banda fa passare le frequenze all'interno di un intervallo di taglio (*banda passante)* ed attenua le frequenze al di fuori di esso.

Filtro elimina banda non permette il passaggio di frequenze in un dato intervallo.

1.2 Modulazione del segnale

Operazione reversibile secondo la quale *il segnale del mezzo trasmissivo*(**portante**) viene modificato in *frequenza (f), ampiezza (U) o fase (φ)* in accordo al *segnale d'ingresso (con i relativi dati da trasmettere)* (**modulante**).

1.2.1 Modulazione a Onda Continua

Modulazione del segnale in cui la modulante e la portante sono segnali analogici

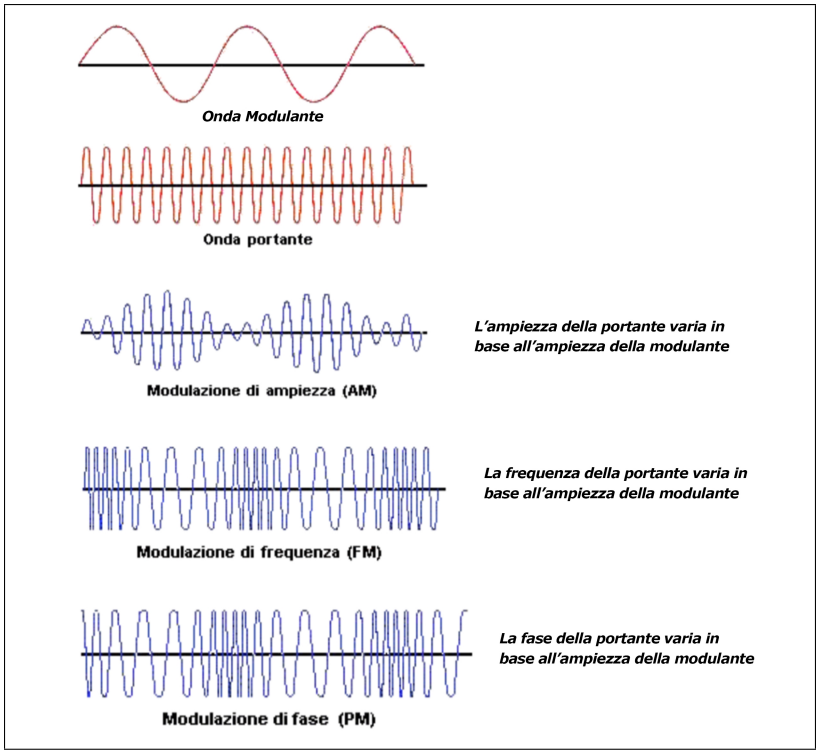

Figura 1: Modulazione segnale a onda continua

1.2.2 Modulazione Impulsiva

Modulazione di segnale in cui la modulante è un segnale analogico mentre la portante è un segnale digitale.

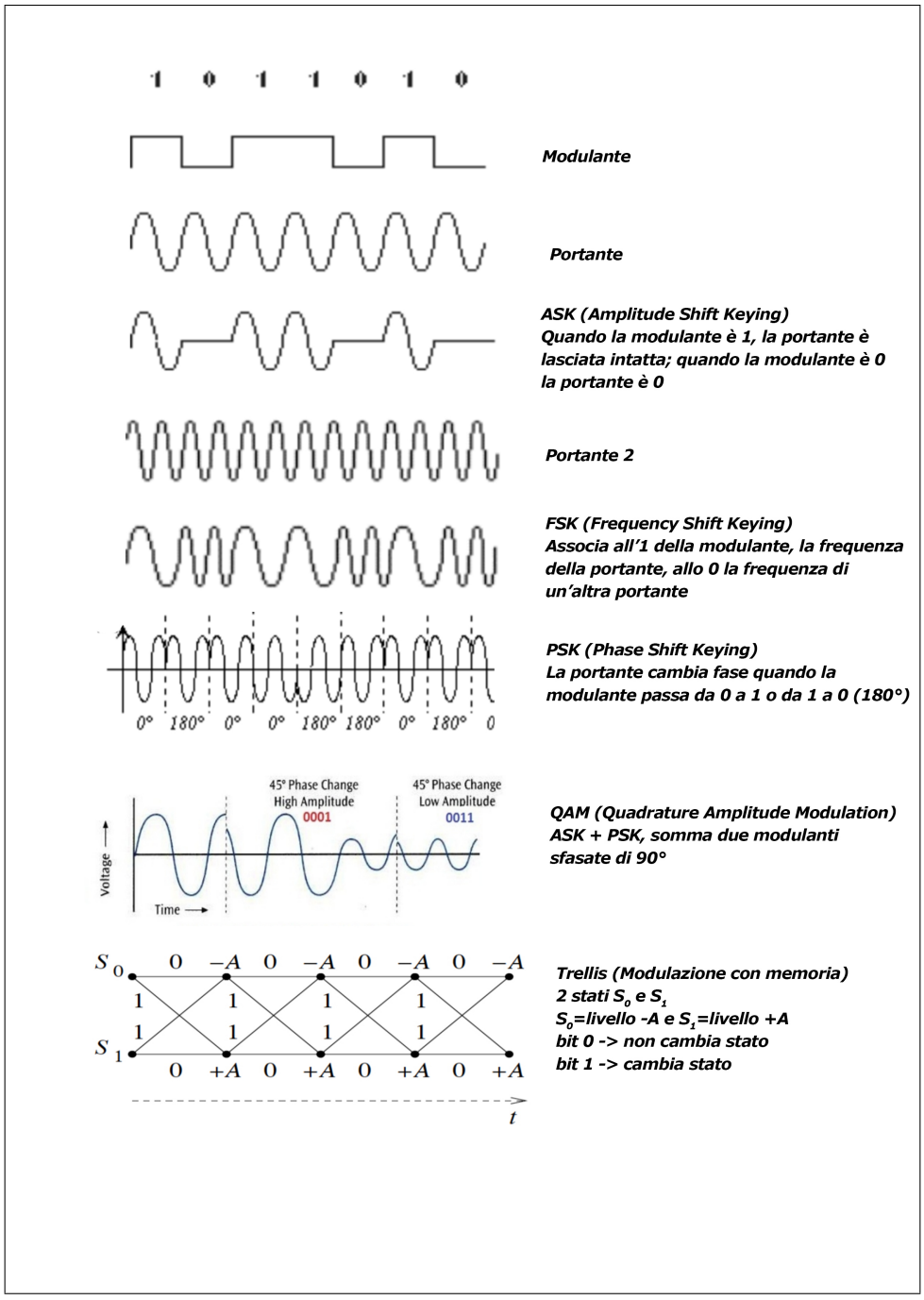

Figura 2: Modulazione segnale a onda digitale

1.2.3 Modulazione Digitale

Modulazione del segnale in cui sia modulante che portante sono segnali digitali.

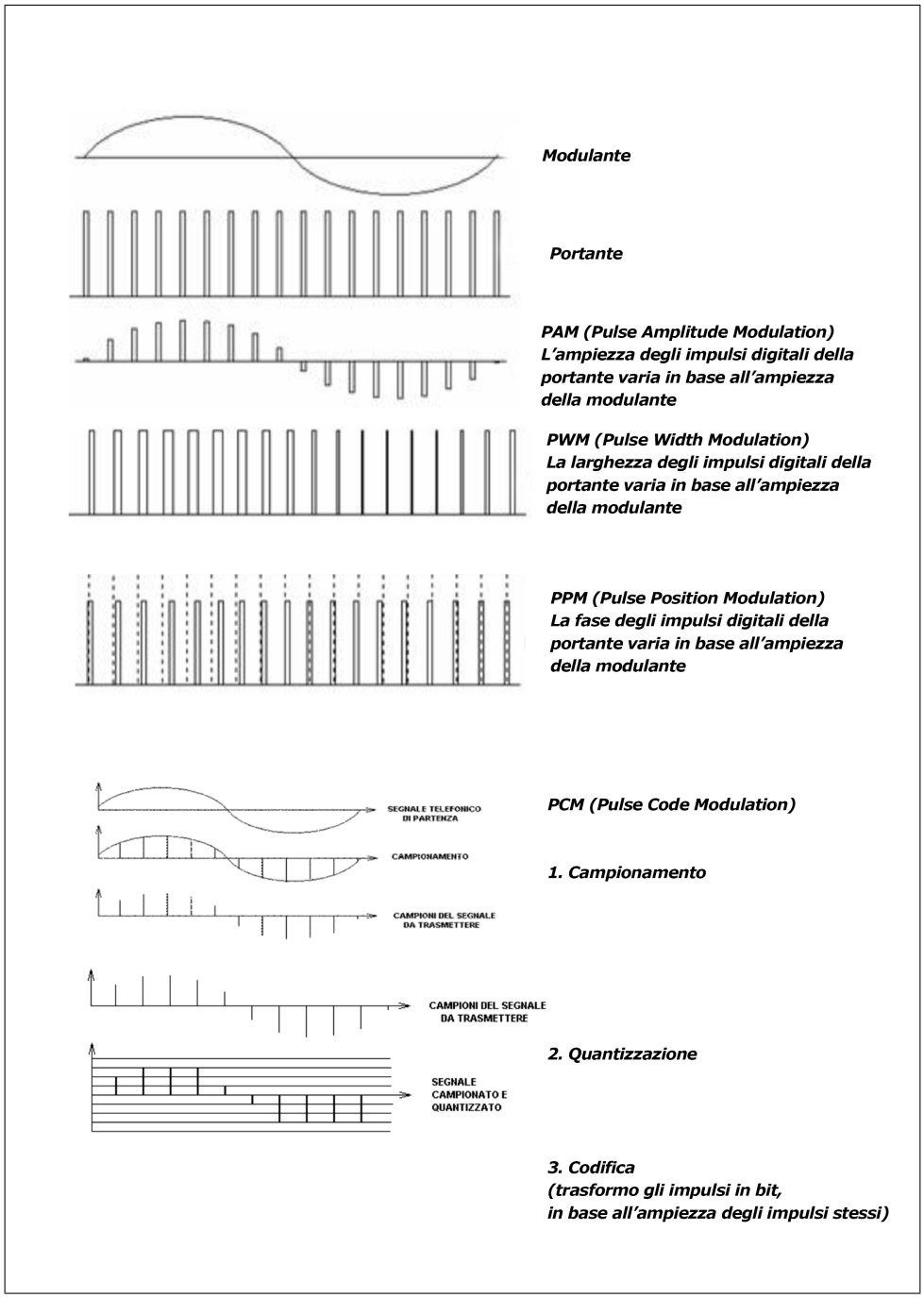

Figura 3: Modulazione segnale a onda impulsiva

1.3 Alterazione del segnale

Processi di modifica del segnale che lo portano a differenziarsi da quello originale. Le principali cause sono:

Attenuazione Perdita di energia del segnale [dB]. Si risolve con gli **Amplificatori** per il sgnale *analogico* e con i **Rigeneratori** per il segnale *digitale*.

Distorsione Cambiamento della forma del segnale

Rumore Insieme dei segnali *indesiderati* che si sovrappongono al segnale utile

Interferenze Sovrapposizioni di altre informazioni che creano disturbo.

- interferenza **ISI** (intersimbolica): causata dalle limitazioni della banda

1.4 Multiplazione

La multiplazione è una tecnica che prevede l'*impiego di un unico canale* per più comunicazioni differenti contemporanee.

1.4.1 FDM

Frequency Division Multiplexing: usata nelle comunicazioni telefoniche, prevede che in fase di trasmissione le comunicazioni subiscano uno **shift di frequenza** e che la frequenza originaria sia *ripristinata* all'arrivo alla destinazione. Il numero di canali multiplabili dipenderà dalla capacità del mezzo trasmissivo.

1.4.2 WDM

Wavelength Division Multiplexing: utilizzata per i segnali ottici, consiste nel *modulare la lunghezza d'onda* del raggio luminoso, così da inviare *diversi raggi* contemporaneamente.

DWDM *Dense WDM*: capace di modulare 16 lunghezze d'onda alla distanza di 0.8 nm.

CWDM *Coarse WDM* (grossolana): utilizza maggiori spaziature tra i canali, con risparmio a livello economico.

1.4.3 TDM

Time Division Multiplexing: nella TDM, i dispositivi *ottengono a turno l'uso esclusivo del canale* di comunicazioni e delle risorse ad esso dedicate, ad esempio la banda.
La TDM si suddivide in:

- **sincrona**: gli intervalli di tempo sono indipendenti dalla presenza di dati da spedire;
- **statistica**: gli intervalli vengono allocati solo quando ci sono dati da inviare. La *velocità* complessiva è minore della somma delle velocità dei canali.

1.5 Limiti di velocità

I canali possono essere *perfetti* (non vi sono alterazioni), *ideali* (vi è al più un'alterazione costante) o *reali*.
Fattori che limitano la velocità:

- **Latenza**: tempo necessario ad un messaggio per arrivare a destinazione. Tiene conto del:
  - Tempo di propagazione (tempo di transito)
  - Tempo di trasmissione (tempo necessario per immettere i bit in rete)
  - Tempo di inoltro (tempo necessario ai nodi per consegnare il messaggio)
  - Tempo di attesa (tempo di attesa nelle code, dipende dal carico di rete)
- **Jitter**: variabilità del ritardo con cui i pacchetti vengono ricevuti
- **Velocità di modulazione**: numero di simboli trasmessi in un secondo [baud=simboli/sec]

Riguardo la massima velocità in un canale (Max_Data_Rate), vi sono due teoremi; In genere si usa prima Shannon per calcolare il Max_Data_Rate, per poi sostituirlo in Nysquist per calcolareil numero dei livelli da usare in un canale.

1.5.1 Nysquist

$\texttt{Max\_Data\_Rate} = 2B * \log_2 V$,
con V=numero di livelli di tensione nel segnale.

1.5.2 Shannon

$\texttt{Max\_Data\_Rate} = C = B * \log_2(1 + S/N)$,
con S=potenza del segnale e N=potenza del rumore

1.6 Trasmissioni

Le trasmissioni di dati possono essere

1.6.1 Asincrone

Non governata da un *segnale di clock*, ma con segnali inviati a frequenze e fasi diverse. La trasmissione di dati inizia con l'invio di uno *start-bit*, la trasmissione in *idle* è rappresentata da una sequenza di *1*, e la fine della trasmissione da uno *stop-bit*.

1.6.2 Sincrona

Scandita da un *clock* che sincronizza trasmettitore e ricevitore; i dati vengono raggruppati in blocchi e trasmessi secondo il tempo dato dal clock.
Il ricevitore ha sia il *sincronismo a bit* (estrae singoli bit dai dati in ricezione), sia il *sincronismo di carattere* (estrae interi caratteri dal flusso di bit)

1.6.3 Orientata al carattere

Usata per trasmettere informazioni testuali, prevede la lettura dei bit a gruppi di otto.
La sincronizzazione *iniziale* è ottenuta tramite una *serie di caratteri di controllo* **SYN**, il primo dei quali viene ricercato spostandosi di un bit alla volta.
Una volta raggiunta la sincronia si ricerca il carattere di controllo **STX** che *sancisce l'inizio della trasmissione*, ed infine il carattere **ETX** che ne *sancisce la fine*.
Per *evitare che i dati vengano scambiati per caratteri di controllo* si fa precedere ai caratteri *STX* e *ETX*, un altro carattere di controllo chiamato **DLE** (tutti **1** o tutti **0** a seconda del sistema) e in invio viene eseguito il **byte stuffing** dei dati: *occorrenze di DLE* nei dati vengono *duplicate* in modo da renderle riconoscibili.

1.6.4 Orientata al bit

La sincronizzazione si basa su degli **idle bytes** (`011111111`), inviati nei *periodi di inattività*, e dei **flag bytes** (`01111110`) che indicano l*'inizio e fine della trasmissione*;
il rischio di scambiare i flag con i dati è scongiurato dall'utilizzo del **bit stuffing**: ogni volta che si incontrano *5 bit uguali a 1* viene *aggiunto uno 0* che verrà poi rimosso dal ricevente.

1.7 Codifiche

1.7.1 NRZ

Not Reduced Zero: lo stato digitale **1** viene rappresentato da *un segnale alto* mentre un lo stato **0** viene rappresentato da *un segnale basso*;
Buona resistenza agli errori, il problema è che *su lunghe trasmissioni si perde la sincronia* nella trasmissione di lunghe serie di bit di uguale valore.

1.7.2 RZ

Return to Zero: simile all'NRZ con la differenza che **a metà di ogni impulso** il segnale *torna sempre a zero*; il *clock* ha quindi *frequenza doppia* per dimezzare la durata di un impulso. Questo metodo non causa desincronizzazione ma ha un più alto rischio di errore.
Può essere anche a *tre livelli* (**RZ bipolare**), ed in tal caso il *livello inferiore* (**-V**) rappresenta lo **0** logico, il *livello superiore* (**+V**) l'**1** logico e *a metà di ogni impulso* si ritorna al *livello intermedio* (**0**), che non rappresenta di per sé alcun valore logico.

1.7.3 Manchester

Come nella RZ, la frequenza del *clock è raddoppiata.* Allo **0** logico corrisponde una *transizione dal basso verso l'alto*, mentre all'**1** logico una *transizione dall'alto verso il basso* (la transizione avviene a metà dell'impulso).

1.7.4 AMI

Alternative Mark Inversion: come la RZ bipolare, utilizza *tre stati*, con la differenza che lo **0** logico corrisponde allo *stato 0* mentre l'**1** logico *si alterna fra +V e -V*. Questa codifica è usata nella *PCM*.

1.7.5 Scrambling

Lo scrambling è un metodo che consente di risolvere alcuni problemi nelle *trasmissioni di lunga distanza*, consiste nel **mescolare** in modo "intelligente" **i bit** per mantenere attiva la linea, è utilizzata nella codifica 2B1Q (two Binary, one Quaternary).

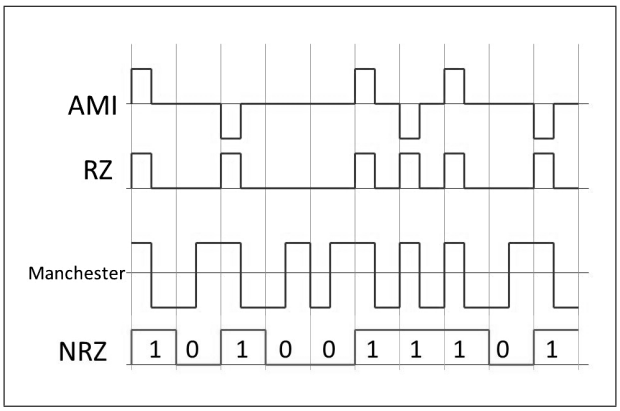

Figura 4: Encoding

1.8 Modem

Un modem (contrazione di modulatore/demodulatore) è un DCE con funzionalità di **Modulazione** in trasmissioni *analogiche e digitali*. Esistono svariati tipi di modem:

- **Modem in banda fonica**;
- **Modem ISDN** (128 kbps);
- **Modem xDSL** (640 kbps-100 Mbps);
- **Modem per PLC** (Power Line Communications), comunicazioni su linea elettrica (640 kbps-200 Mbps);
- **Modem GPRS, UMTS e HSDPA**, spesso integrati nei cellulari o come PC card;
- **Modem in banda base**, utilizzati per scopi industriali, che mettono in comunicazione diretta due utenti su doppino telefonico.

Ognuno di questi modem può essere inoltre **interno** o **esterno** al *DTE*.
Tra i modem **esterni**, in base al tipo di collegamento al DTE, si distinguono:

- **Modem seriali**, collegati con cavo seriale tramite interfaccia *RS-232* o *USB*;
- **Modem paralleli**, collegati con cavo parallelo alle porte LPT1 o LP2.

Tra i modem **interni** si distinguono invece:

- **Modem PCI**, che lavorano appunto sul BUS PCI (Peripheral Component Interconnect);
- **Modem PCMCIA, PC card o Express card**, utilizzati esclusivamente nei portatili.

Modem in banda fonica Si usano quando c'è necessità di trasmettere i *segnali digitali* sulla *linea telefonica* e viceversa.
Dato che la banda disponibile sulla linea telefonica è di $4kHz$, ciò rende lenta la trasmissione di segnali digitali (anche per via dell'*Attenuazione* delle alte frequenze); Un *modem fonico* opera dunque una **Modulazione digitale** volta a comprimere la banda dei segnali emessi dal *DTE*.

1.9 Interfacce Hardware

Dal punto di vista **fisico**, un'interfaccia è caratterizzata da **un canale di trasmissione** (*identificato da un mezzo trasmissivo, due connettori e due porte poste agli estremi*);
Dal punto di vista **logico**, da una **modalità di trasmissione** (*seriale o parallela*).

1.9.1 Interfacce parallele

Le interfacce parallele trasmettono segnali *da più pin in contemporanea.*

Interfaccia Centronics Ormai obsoleta, originariamente monodirezionale, è stata impiegata soprattutto per collegare i DTE alle stampanti, ma esiste anche uno standard bidirezionale per il collegamento di dispositivi di input.
Tale interfaccia consente di trasferire **8 bit in parallelo** nello standard **TTL**[2].
La porta parallela del DTE è un *connettore femmina* detto "a vaschetta" o **DB 25**, con **25 pin**, mentre sulle periferiche è presente un connettore differente, chiamato appunto **Centronics**, dal nome del primo costruttore. Alle due porte corrispondono chiaramente due diversi connettori maschio.

1.9.2 Interfacce seriali

le interfacce seriali trasmettono *da un solo pin alla volta.*

RS-232 L'RS-232 è uno standard che permette la realizzazione di una trasmissione seriale tra un DTE e un DCE, in *modalità sia sincrona che asincrona.*
Per questa interfaccia esistono due diversi tipi di connettore: quello a **25 pin** e quello ridotto, **a 9 pin**.

USB L'USB (*Universal Serial Bus*) ha soppiantato le interfacce precedentemente descritte. Essa risulta vantaggiosa, oltre che in termini di velocità e versatilità, poiché può *fornire direttamente l'alimentazione* alle periferiche e consente di *creare collegamenti a caldo* (interfacce *plug-and-play*).
I connettori USB sono molteplici (*tipi A, B e C con rispettive varianti miniaturizzate*), ma tutti con **quattro poli** ed uguali funzionalità.
Il sistema USB è **asimmetrico**: consiste in *un singolo gestore* e *molte periferiche* collegate ad albero, attraverso hub. Supporta fino a un massimo di *127* periferiche per gestore. max velocità = 7,2 Gbps (versione USB 3.1).

IEEE 1394 La IEEE 1394 è un'*interfaccia bidirezionale* in grado di gestire fino a *63 dispositivi* sulla stessa linea ad alta velocità, miscelando dati *sincroni ed asincroni.*
Come la USB, consente di collegare e scollegare i dispositivi a caldo tramite due diversi tipi di connettore, l'uno a **quattro pin**, senza alimentazione, l'altro a **6 pin**.
Il limite principale di tale interfaccia è *la portata*, limitata a *pochi metri.*

RS422 Utilizzato principalmente per realizzare *collegamenti punto-a-punto* tra due apparecchiature con *alta immunità ai disturbi* anche a distanze considerevoli e a velocità anche superiori a 10 Mbps. Prevede, per ogni coppia di fili, *trasmissione unidirezionale* e non reversibile.

[2]Transistor-Transistor Logic: prima tecnologia di circuiti integrati diffusa su scala globale

1.9.3 Strutture di rete

Il doppino telefonico è il mezzo trasmissivo più utilizzato per la realizzazione di collegamenti. Nel caso di reti **PSTN** (Public Switching Telephone Network) vi sono tratti di cavo che contengono centinaia di doppini (*cavi multicoppia)* connessi da *punti fisici di flessibilità* (armadio ripartilinea e distributore) usati per il controllo.
I tratti di doppino che formano la rete sono distinti in:

- **rete di accesso primaria**: t*ra centrale ed armadio*, formata da due cavi con centinaia di coppie
- **rete di accesso secondaria**: *tra armadio e distributore*, formata da cavi con decine di coppie
- **cablaggio verticale**: *tra distributore e borchia di accesso*, domicilio utente

FTTF Nel caso di Fiber To The Home in FIbra ottica, si hanno:

- **OLT**: *Optical Line Termination*, interfaccia lato rete
- **ONU**: *Optical Network Unit*, interfaccia con terminale, vicino all'utente
- **ONT**: *Optical Network Termination*, terminazione di rete ottica

Reti AON *Active Optical Network* sono strutture di rete **P2P** (point-to-point), nel senso che *ogni utente ha il proprio tratto di fibra*, che giunge direttamente al proprio *ONT*.
Nelle AON si utilizzano **componenti attivi**, come amplificatori, ripetitori, router o switch. SOno caratterizzate da una *topologia a stella*; Garantisce, a costi elevati, la massima velocità di trasmissione.

Reti PON *Passive Optical Network*, utilizzano solo **componenti passivi** e sono caratterizzate da una *topologia ad albero* .
In esse, *tratti di fibra vengono condivisi tra più utenti*, e solo alla fine uno **splitter ottico** suddivide il segnale in più segnali uguali ma di potenza minore in modo da distribuirlo.

1.10 Protocolli di primo livello

1.10.1 PDH

Plesiochronous Digital Hierarchy: tecnologia utilizzata per trasportare *grandi quantità di dati* su apparecchiature di trasporto digitali. Rappresenta il primo metodo di **Multiplazione** per trasmettere su molti canali contemporaneamente (*30*).
Tramite un multiplexer TDM si uniscono i diversi canali; questo dovrà inserire *bit aggiuntivi* (i **dummy bits**) per rendere sincrono il flusso in entrata al **demultiplexer** ricevente, che li riconosce e li elimina.
Il PDH *non monitorale prestazioni* della rete e *non garantisce buone performance per trasferimento contemporaneo* di audio, video e dati.

1.10.2 SDH

Synchronous Digital Hierarchy: protocollo usato per la *trasmissione di fonia e dati su fibra e rete* elettrica, anch'esso ha il compito di aggregare flussi di dati con bitrate diversi e spedirli tutti insieme a grandi distanze.
A differenza del PDH, prevede che *tutti gli elementi della rete siano sincronizzati da uno stesso **clock***.
Inoltre permette di trasferire informazioni essenziali per la corretta gestione della rete, non presenta limiti di distanza e può offrire una velocità di *140Gb/s*. L'unico vincolo è dovuto alla multiplazione TDM, che quindi fornisce ad ogni utente una capacità costante nel tempo.

1.10.3 DSL

Digital Subscriber Line, famiglia di tecnologie che fornisce trasmissione digitale di dati su **Doppino telefonico**, comunemente utilizzata nella connessione ad Internet da utenza domestica tramite l'**ADSL** *(Asymmetrical DSL).*
Il *protocollo ADSL* sfrutta completamente la banda passante del doppino telefonico, utilizzando due tecniche di modulazione: la **CAP** (*Carrierless Amplitude Phase*, variante della QAM) e la **DMT** (*Discrete Multi Tone*).
Con un *filtro DSL*, chiamato **splitter**, le bande di frequenza vengono *tripartite*, per l'uso contemporaneo di un'unica linea telefonica *sia per il servizio ADSL che per le chiamate telefoniche.*
Velocità teoriche: *8Mbit/s in download* e *800kbit/s in upload*; **Velocità reali**: 1,5Mbit/s in download e 256kbit/s in upload.

1.11 Dispositivi livello Fisico

1.11.1 Ripetitore

Dispositivo di rete che si limita a ripetere i pacchetti che gli arrivano.
Si utilizza quando esiste una limitazione fisica alla lunghezza della *LAN*: le reti collegate possono in tal caso essere viste, da un punto di vista logico, come una rete unica.

1.11.2 Hub

Uno hub è un dispositivo di rete che funge da nodo di smistamento dati di una una *Rete a dorsale* o in una *Rete a stella.* Poiché non è in grado di ri-inviare i pacchetti se non in *broadcast*, è attualmente impiegato in misura molto minore rispetto allo *Switch.*

2 Livello Collegamento

2.1 Specifiche protocolli

I protocolli di secondo livello si dividono in due principali categorie:

- **Asincroni**: consentono soltanto trasmissioni di caratteri singoli; Sono asincroni in quanto non si ha un preciso intervallo di tempo tra l'invio di due caratteri. È comunque obbligatorio definire un *bit time* di durata del carattere.
  I protocolli asincroni sono detti anche **start/stop**: ogni carattere è composto da un **bit di start**, dal *bit del carattere vero e proprio* da trasmettere, da un **bit di parità** (volto al controllo dell'errore) e da uno o due **bit di stop**.
  L'utilizzo di questi bit in forma, rispettivamente, di *header* e *trailer* consente al ricevente di distinguere i singoli byte trasmessi.

- **Sincroni**: mittente e destinatario sincronizzano i loro clock grazie a particolari caratteri. A loro volta si suddividono a seconda della trasmissione:

  - Orientata al carattere: utilizzano il carattere di sincronismo **SYN**. Vi appartiene *BSC*.
  - Orientata al bit: la sincronia è garantita da 2 byte detti **flag** posti uno all'inizio e l'altro alla fine del pacchetto. Vi appartiene *HDLC*.

2.2 BSC

Binary Synchronous Communication è un protocollo sincrono orientato al carattere. Il flusso trasmissivo è di tipo **Half-Duplex** con velocità tra *1200 e 19200 bps*.
Il frame è composto da circa 100 byte, divisi tra messaggio da trasmettere e caratteri di controllo.
La codifica binaria usata può essere *ASCII, EBCDIC* o *SBT* (Six Bit Transcode).
In base alla rete su cui opera, il BSC si classifica in:

- **BSC1**: rete *dedicata* punto-punto;
- **BSC2**: rete *commutata* punto-punto;
- **BSC3**: rete multipunto.

Nei due casi di **rete punto-punto** il trasmettitore invia *caratteri di sincronismo (**PAD** o **SYN**) seguiti da **ENQ***; [3] Il ricevente risponderà con **ACK** *se è pronto*, oppure **NAK** se non può acquisire i dati. Il *collegamento viene terminato* con il messaggio **EOT**.
Nel caso di **rete multi-punto** l'elaborazione centrale effettua un'*interrogazione ciclica* (**pollic**) per individuare il terminale a cui collegarsi.

[3] Dato che i DTE sono sia trasmettitori che riceventi, può succedere che vogliano trasmettere nello stesso momento: uno dei due diventerà una *stazione primaria* che ripete l'invio, mentre l'altro sarà una *stazione secondaria* che dovrà rinunciare.

2.3 HDLC

Il protocollo **High Level Data Link Control** costituisce lo standard ISO per trasmissioni **sincrone Full-Duplex**; è *orientato ai bit* e utilizzato su reti di grandi dimensioni.
Prevede 3 tipi di terminali:

- **Stazione primaria**: detta anche *master*, ha il compito di controllare il collegamento inviando i comandi di controllo;
- **Stazione secondaria**: agisce in base ai comandi della stazione primaria e può spedire soltanto *pacchetti di risposta*;
- **Stazione combinata**: può inviare sia comandi sia risposte.

La connessione fra mittente e destinatario è detta **bilanciata** se sono entrambi stazioni combinate; al contrario, si parla di connessione **sbilanciata**: in questo caso il protocollo lavora in modalità **Half-Duplex** ed i messaggi inviati dal master prendono il nome di **command**, mentre quelli delle stazioni secondarie sono detti **response**.
Nello specifico, HDLC può lavorare in 3 diverse modalità:

- **NRM** *(Normal Response Mode):* connessione half-duplex sbilanciata.
  Le stazioni secondarie possono trasmettere anche senza autorizzazione esplicita del master;
- **ABM** *(Asynchronous Balanced Mode):* bilanciata full-duplex tra due stazioni paritetiche;
- **ARM** *(Asynchronous Response Mode):* come NRM ma limitata a due stazioni.

Frame: Il protocollo HDLC prevede che le stazioni possano scambiarsi **frame** di 3 tipologie:

- **I-frame**: il tipo ***I****nformation* è usato per trasportare i dati dal *Livello di rete.*
- **S-frame**: il tipo ***S****upervisory* è usato per controllare il flusso e gli errori;
- **U-frame**: il tipo ***U****nnumbered* fornisce funzioni di controllo aggiuntive, come informazioni per iniziare/terminare la connessione.

I frame sono composti dai seguenti *sottocampi*:

- **flag**: due sequenze di 8 bit `01111110` usate come *inizio e fine della trasmissione.* La stessa sequenza è usata come *idle byte.*
  La forma del flag rende necessario l'uso del **bit stuffing** *(aggiunta di un bit a `0` in mezzo a lunghe sequenze di`1`.*
- **indirizzo**: è un campo di 8 bit: identifica la *stazione che ha trasmesso* o che *deve ricevere* il frame;
- **controllo**: 8 bit o 16 bit contenenti *informazioni di controllo* o definizione del pacchetto. (N.B: il bit a sinistra è il meno significativo).
  Di seguito le possibile trame:
  - ***Information***: la forma è `0SSSPRRR`, dove `SSS` *conta i frame trasmessi*, `RRR` *conta quelli ricevuti.*
    `P` è detto bit P/F (*Poll/Final*):

- ***Supervisory***: la forma è `10TTPRRR`, in cui i primi due bit sono fissi ed identificano il tipo S-frame; `P` ha lo scopo visto in precedenza.
  I due bit `TT` comunicano che:
  - 00 = *Received Ready*: ricevuti tutti i frame, stazione pronta a ricevere;
  - 01 = *Reject*: problema di acquisizione, necessario ritrasmettere da `RRR`;
  - 10 = *Receive Not Ready*: la stazione non può ricevere;
  - 11 = *Selective Reject*: invito a rinviare il frame numero `RRR`.
- *Unnumbered*: la forma è `11MMPMMM`.

- **campo informativo**: contiene i dati significativi da trasmettere. La lunghezza è arbitraria .N.B: questo campo è assente nei S-frame;
- **campo FCS** (Frame Check Sequence): 16 o 32 bit utilizzati per rilevare eventuali errori di trasmissione.

2.3.1 SDLC

HDLC deriva da SDLC **(Synchronous Data Link Control)**; Rispetto ad HDLC, il protocollo SDLC:

- ha il campo FCS di 8 bit;
- supporta configurazioni a loop, come il token ring;
- può lavorare solo in NRM;

2.4 PPP

Il **Point-to-Point Protocol** è un protocollo usato nelle *connessioni punto-punto* e trova la sua più ampia diffusione in ambito *WAN*.
Si può definire un'estensione del protocollo *HDLC* in quanto il funzionamento è analogo. Utilizza solo **U-frame**, la cui struttura differisce da quelli in HDLC nei seguenti campi:

- **indirizzo**: i suoi 8 bit sono sempre `11111111` dato che le trasmissioni avvengono esclusivamente in *broadcast*;
- **controllo**: di 8 bit che, in PPP sono del valore fisso `11000000` e rappresentano il comando UI (*unnambered information*), ovvero un messaggio che contiene dati;
- **campo informativo**: la lunghezza è limitata tra 0 e 1500 ottetti ma può essere ampliata facendo uso del campo supplementare **padding**.

Inoltre, il PPP prevede un secondo campo che non compare in HDLC:

- **protocol**: fatto di 1 o 2 *byte*, serve ad identificare il protocollo incapsulato nel frame.

Per le operazioni del *Livello di collegamento* il PPP si avvale del protocollo **LCP** *(Link Control Protocol)*, mentre per le negoziazioni con il *Livello di rete* impiega protocolli del tipo **NCP** *(Network Control Protocol)*.

3 Livello Rete

3.1 IP

L'IP definisce l'esatto formato dei dati mentre attraversano l'internet TCP/IP. Svolge la funzione di **routing** e definisce regole.
L'unità fondamentale è il **datagram IP**, diviso in *header* (intestazione) e *data* (blocco dati). Nell'**header** sono presenti i seguenti campi:

- **VERS**: 4 bit, indica la versione dell'IP del datagram (IPv4 o IPv6)
- **HLEN**: 4 bit, indica la lunghezza dell'header in parole da 32 bit
- **Lunghezza totale**: 16 bit, indica la lunghezza totale del datagram in ottetti (incluso il blocco data). max=2 alla 16.
- **Servizio**: 8 bit, indica come deve essere gestito il datagram. E' diviso in 5 sottocampi:
  - 3 bit di **precedenza**, per specificare l'importanza del datagram (0-7)
  - 3 bit suddivisi in D, T, R per specificare il tipo di trasporto. D chiede un basso ritardo, T un alto throughput, R alta affidabilità
- **ID, FLAG, OFFSET**:controllano la frammentazione e il riassemblaggio del datagram a seguito dell'incapsulamento degli stessi in frame al liv. fisico.
- **TTL**: Time To Live, durata (s) concessa al datagram di restare in trasporto.
- **PROTOCOL**: indica quale protocollo di più alto livello ha generato la porzione Data
- **CHECKSUM**: garantisce l'integrità dell'header mediante CRC sii bit dell'header
- **SOURCE**: indirizzo IP a 32 bit dell'host che ha generato il datagram
- **DESTINATION**: indirizzo IP a 32 bit dell'host al quale è destinato il datagram
- **DATI**: dati trasportati dal datagram
- **OPTIONS** debugging
- **RIEMPIMENTO** area riempita dibit = 0 per garantire lunghezza multipla di 32 bit.

3.1.1 IPv4

Un IP su 32 bit (4 byte) identifica univocamente una rete ed un host appartenente alla rete `x.y.z.w`;
L'indirizzo è diviso in due parti: **host** e **rete**.
Esistono 5 classi di IP:

- **Classe A**: 0xxxxxxxx.y.z.w,
  subnet mask: 255.0.0.0 (parte rete=x, host=y.z.w)
- **Classe B**: 10xxxxxxx.y.z.w,
  subnet mask: 255.255.0.0 (parte rete=x.y, host=z.w)
- **Classe C**: 110xxxxxx.y.z.w,
  subnet mask: 255.255.255.0 (parte rete=x.y.z, host=w)
- **Classe D**: 1110xxxxx.y.z.w (multicast)
- **Classe E**: 11110xxxx.y.z.w (riservata).

NB: L'indirizzo IP indica la *connessione di un host alla rete*, non l'host in sè. INfatti esso è **assegnato alle interfacce di rete**, non agli host!

3.1.2 Indirizzi speciali

Gli IP possono far riferimento a *reti* o *host*.

- **indirizzo di rete**: IP in cui i bit della parte *host* sono tutti a 0, denota la rete stessa;
- **indirizzo di broadcast**: Se tutti i bit della parte *host* sono a 1, riservato a tutti gli host della rete;
- **default route**: 0.0.0.0
- **loopback address**: 127.0.0.1
- **Broadcast locale**: 255.255.255.255

3.1.3 Interfaccia di rete

Ad ogni *scheda di rete (hardware)* di un dispositivo corrisponde un'*Interfaccia di Rete (software)* sulla quale operano protocolli di rete come Ethernet. [4]
Quando un'interfaccia di rete viene configurata ad essa viene assegnato un indirizzo IP, ciò avviene automaticamente attraverso il protocollo *DHCP* o può essere effettuato manualmente.
Nei sistemi Unix-Linux, il comando per la configurazione manuale dell'IP è `ifconfig` (dove "if" sta per "interface"). La sintassi è la seguente:
`$ ifconfig <nome interfaccia> <indirizzo IP> netmask <netmask x.y.z.q> broadcast <indirizzo broadcast>` [5] Più di recente, `ifconfig`, `route` (vedi *Routing*) ed `arp` sono stati rimpiazzati dal comando `ip`, preinstallato nei sistemi arch.

[4] Esistono anche interfacce virtuali come quella di loopback locale presente in ogni computer (`lo` su Unix)
[5] Il comando `ifconfig` senza argomenti mostra lo stato delle interfacce.

3.2 Routing

Il routing consiste nella scelta del cammino migliore da percorrere per trasmettere un datagram da un host all'altro, passando attraverso i nodi di una rete basata sul protocollo IP. Si divide in:

- **Routing Minimale**: la tabella di routing viene definita al momento della configurazione dell'Interfaccia di rete;
- **Routing Statico**: utilizzato quasi solamente per gli host, prevede la definizione manuale delle varie *route*. In Unix-Linux le route possono essere aggiunte usando il comando
  `$ route add -net <ind. di rete> netmask <netmask x.y.z.q> gw <ind. gateway>`
  Per rendere permanenti le modifiche, i comandi `route` devono essere salvati in un file di configurazione eseguito all'avvio della macchina (in sistemi Unix-Linux, il path di tale file è `/etc/init.d/rc.local`);
- **Routing Dinamico**: utilizzato nei router, sfrutta i diversi **Protocolli di routing**.

Alla ricezione di un pacchetto, ogni nodo delle rete esegue le seguenti operazioni:

- determina la classe dell'Indirizzo IP di destinazione del pacchetto;
- controlla se tale indirizzo è locale ed eventualmente vi applica la Netmask per poi inviarlo direttamente all'host destinatario;
- se l'indirizzo non è locale, cerca la rete di destinazione nella Tabella di routing e, se presente, instrada il datagram verso il Gateway corrispondente.

3.2.1 Tabella di routing

Presente in ogni nodo di rete, contiene le informazioni per il routing.
Nella tabella di routing ogni riga rappresenta una "strada", composta da:

- indirizzo dell'host o sottorete di destinazione;
- indirizzo del prossimo gateway da attraversare;
- "distanza" dalla destinazione, detta **Metric**.

Nei sistemi Unix-Linux, si visualizza tramite il comando `netstat -r`[6] o `route`.
'E possibile utilizzare la specifica `-n` per ottenere gli indirizzi di destinazione in forma numerica.
Il significato delle *flag* è il seguente:

- U (Up) indica che l'Interfaccia di rete è attiva;
- G indica un'uscita verso un'altra rete tramite Gateway;
- H indica che la destinazione è l'indirizzo completo di un host;
- D indica una route aggiunta da un ICMP redirect.

[6] Attenzione: il campo `Genmask` indica la netmask.

3.2.2 Algoritmi di routing

Devono essere: ottimali, semplici e con basso overhead, robusti e stabili, rapidi nella convergenza, flessibili.
Classificati in: statici/dinamici, single/multi-path, piatti/gerarchici, Host/Router-intelligent, Intra/inter-domain, **Distance-vector**[7]**/Link-state**[8].

3.3 Protocolli di Address Resolution

Quando un pacchetto di livello 3 (Network) deve essere incapsulato in un protocollo di livello 2 (Data Link), questo deve inserire nell'header del pacchetto *l'indirizzo Data Link*.
Quindi per comunicare si deve conoscere l'**indirizzo fisico** dell'host di destinazione se questo appartiene *alla stessa rete del mittente*; O l'**indirizzo del Gateway** se l'host destinazione *appartiene ad un'altra rete*.

3.3.1 ARP

Address Resolution Protocoll: associa (risolve) la corrispondenza indirizzo ip (generalmente conosciuto) - indirizzo fisico di un host.
L'host A che vuole conoscere l'indirizzo fisico di B, invia un pacchetto broadcast contenente l'IP dell'host B, il quale rissponderà fornendo il suo indirizzo fisico.
In ogni macchina è presente una **cache** che salva gli indirizzi risolti via ARP per consultazioni successive (cache=*soft state - indirizzi possono diventare vecchi == timer di scadenza)*
L'host che effettua richiesta ARP via broadcast include il proprio indirizzo fisico, cosicchè tutti gli host possono aggiornare la propria cache.

3.3.2 RARP

Reverse Address Resolution Protocol: l'host spedisce la richiesta RARP ad un server mediante pacchetto broadcast con specificato indirizzo fisico, ed attende una risposta (che includerà l'IP relativo a quell'indirizzo fisico trasmesso mediante richiesta RARP).
Usato per workstation diskless (devono caricare il S.O. da un server ad ogni avvio).

3.3.3 Autonomous System

Le reti e gli IS (Intermediate System) si suddividono in **interni** ad un dominio di routing ed **esterni** ad un dominio di routing [9]. Il dominio di routing prende il nome di **AS** (Autonomous System), identificando la politica di routing adottata.
GLi AS si dividono in : **Multihomed AS** (mantiene connessioni con più di un AS), **Stub AS** (connesso solamente con un altro AS) e **Transit AS** (fornisce attraverso di sè connessioni con altre reti).
I *router* che instradano messaggi all'interno dello stesso AS sono **Interior Router**, quelli che instradano anche tra AS diversi sono **Exterior Router**.
I primi eseguono una famiglia di protocolli detta **IGP** (Interior Gateway Protocol), i secondi **EGP** (Exterior Gateway Protocol).

[7] *Periodicamente* i router raggiungibili si scambiano le istruzioni di routing; *il ricevente sostituisce le proprie istruzioni se quelle ricevute sono più ottimali*, le quali verranno propagata.
[8] Informazioni inviate in *broadcast*. Routers= nodi di un grafo connesso
[9] insieme delle reti soggette all'amministrazione di una stessa organizzazione

3.4 Protocolli di Routing IGP

3.4.1 OSPF

Open Shortes Path First, protocollo standard per routing all'interno di un AS, è open source e basato sull'omonimo algoritmo di Dijkstra.
E' un **link-state routing protocol**, in quanto invia *link-state advertisements* (LSA) a **tutti** i routers di una stessa **area gerarchica**. Un router OSPF accumula LSA e calcola lo Shortest Path.
Quindi il routing può essere Inter-Area o Intra-Area. Il responsabile del routing Inter-Area è l'*OSPF Backbone*, che può essere anche spezzato e ricollegato tramite *virtual-links* (questo per definire topologie logiche diverse da quelle fisiche). OSPF distingue 4 tipi di router:

- **Internal Router**: interni ad un'area
- **Are Border Router**: che connettono 2 o più aree
- **Backbone Router**: appartenenti alla dorsale (area 0)
- **Border AS Router**: router di confine tra AS

OSP si basa sull'invio di pacchetti (router non adiacenti non scambiano informazioni):

- **Hello**: usato per scoprire i neighbors all'avvio del router, il designed router (DR) ed il backup designed router (BDR)
- **LS Update**: fornisce i propri criteri per la selezione del costo del link
- **LS ACK**: conferma un LS update
- **Database Description**: comunica gli aggiornamenti che conosce
- **LS Request**: richiesta di info di stato ai neighboars routers

OSPF Packet format
Header (24 Bytes): version number (1), Type (1), Length (2), Router ID(4), Area ID (4) Checksum (2), Autentication Type (2), Autentication (8);
Data (Variable).

3.4.2 RIP

Routing Information Protocol, implementato nel programma *routed*, si basa sull'algoritmo di Bellman-Ford (vettore-distanza), quindi è un **Distance-vector routing protocol**. Ha un limite di *15 hops*: reti più distanti sono irraggiungibili.
Un router RIP invia *tutta la routing table* o una porzione di essa ai *router vicini* (neighbors distanti 1 hop) ad intervalli di tempo.
Abbiamo due forme di RIP:

- **Attiva**: usata dai *routers*, invia in *broadcast* aggiornamenti periodici di routing ed usa i msg in arrivo per aggiornare la propria routing table.
- **Passiva**: usata dagli *hosts* (ma anche dai routers[10], usa i msg in arrivo per aggiornare la routing table, ma non invia aggiornamenti[11].

RIP è più propenso a generare *loops* rispetto a OSPF, ma ha bisogno di *meno risorse*, è più semplice da implementare ed è disponibile di default su Unix/Linux (**routed**)
RIP deve gestire 3 problemi:

- non rileva loop
- instabilità (risolto usando un numero basso di distanza max)
- problemi di convergenza lenta. Risolto adottando tecniche di:
  - *split horizon update* (un router non propaga info su un'altro che ha ricebuto questo msg)
  - *hold down* (il router ignora aggiornamenti inerenti a una rete dopo che ha ricevuto un msg di rete irraggiungibile),
  - *poison reverse* (se scompare un collegamento, questo gli verrà assegnato distanza infinita) + *triggered update* (i routers aggiornano la scomparsa immediatamente)

RIP1 Packet format: Command (1), Version number (1), Zero (2), Address Family ID (2), Zero (2), Address (4), Zero (4), Zero (4), Metric (4)
RIP2 Packet format: Command (1), Version number (1), Unused (1), Address Format ID (2), Route Tag (2), IP address (4), Subnet Mask (4), Next Hop (4), Metric (4)

[10] Quindi i roter possono essere attivi o passivi, gli host passivi
[11] sono del tipo (indirizzo destinazione, distanza)

3.5 Protocolli di routing EGP

3.5.1 BGP

Border Gateway Protocol, protocollo di inter-domain routing che mette in comunicazione routers appartenenti ad AS differenti (*gateway di confine*).
BGP effettua

- **inter AS routing** tra due o più router BGC appartenenti ad AS diversi.
- **intra AS routing** tra due o più router iBGP appartenenti allo stesso AS
- **pass-through AS routing** tra due o più router BGP che scambiano attraverso un AS che non esegue BGP, il cui traffico non è destinato a nessun nodo interno a quell'AS.

Due router BGC formano una connessione TCP *(peer o neighbor routers)* e si scambiano prima di tutto le routing tables e per ogni rete il prossimo hop, dopodichè si scambiano *messaggi di gestione della connessione (nel campo Type)* (Open, Update, Notification, Keepalive, Refresh).
BGP Packet format: Marker[12](16), Length (2), Type (1), Data (variable).

3.6 ICMP

Internet Control Message Protocol, protocollo progettato per riportare anomalie durante il routing dei pacchetti e per verificare lo stato della rete. I vari tipi di messaggi ICMP sono:

codice	Messaggio
0	Echo Reply *(verifica raggiungibilità nodo)*
3	Destination Unreachable *(segnalazione anomalia)*
5	Redirect *(associa un router diverso da quello di default per un migliore instradamento)*
8	Echo Request *(verifica raggiungibilità nodo)*
11	Time Exceeded for a Datagram *(segnalazione anomalia)*
17	Address Mask Request *(Richiedi quale netmask è usata in una rete)*
18	Address Mask Reply *(Invia all'interfaccia che l'ha richiesta, la propria netmask)*

[12] valore riconosciuto da entrambi i peers per contrassegnare l'inizio del msg.

3.7 Dispositivi di rete

Nel secondo livello OSI troviamo dispositivi di rete plug-and-play ed *"intelligenti"*, che non si limitano alla sola replicazione del segnale ma sono in grado di riconoscere, nei segnali elettrici che ricevono dal mezzo trasmissivo, i dati organizzati in **frame**; Agiscono quindi sui frame ricevuti, gestendoli ed instradandoli, il tutto in modo trasparente.

3.7.1 Bridge

Un bridge *(ponte)* è un dispositivo di rete munito di porte con cui è collegato a diversi ***segmenti di rete*** (generalmente due o più LAN) da cui riceve dati e li instrada selettivamente verso una porta destinataria

Funzionamento Quando riceve un pacchetto su una porta, riconosce i frame, estrae gli indirizzi sorgente e destinazione e cerca di capire dall'indirizzo del *destinatario* se questi si trova nello **stesso segmento** del *mittente* oppure no ***(Filtering)***. Nel primo caso non inoltra il frame (il destinatario condivide lo stesso bus).
Nel secondo caso il bridge inoltra il frame verso il *segmento del destinatario* ***(Forwarding)***; Se non sa dove si trova, il bridge inoltra il frame su tutte le porte ***(Flooding)***.
Per instradare i frames, il bridge mantiene una tabella (di **forwarding**) di indirizzi MAC per ogni porta, così è in grado di capire verso quale porta, e quindi quale segmento, inoltrare il frame.

Tipologie In base alle tipologie delle due reti si distinguono:

- **Transparent Bridge**: collega 2 LAN Ethernet o IEEE 802.3
- **Source Routing Bridge**: collega 2 LAN Token Ring
- **Translational Bridge**: collega 2 LAN di tipo diverso (deve adattarsi alle diverse regole trasmissive)

3.7.2 Switch

Uno switch *(commutatore)* è un dispositivo molto simile al bridge, ma a differenza di quest'ultimo è collegato direttamente agli host ed ha un numero di porte nettamente superiore.

Funzionamento La funzione di instradamento è implementata mantenendo in un buffer locale tutti gli indirizzi *MAC* degli host connessi alla rete, suddivisi per *porta dello switch.*
L'instradamento è analogo a quello del bridge, ma nel caso in cui lo switch invii il pacchetto a tutte le porte (flooding), il nodo destinatario, ricevuto il pacchetto, risponderà facendo sapere allo switch la sua porta ed il suo MAC address.
Infine, usando lo switch, non sono possibili collisioni ed è possibile usare la modalità *Full-Duplex.*

4 Livello Applicazione

4.1 DNS e Gestione dei nomi

Il DNS (*Domain Name System*) è un servizio utilizzato per associare i nomi degli host, più semplici da ricordare per l'utente, ai relativi indirizzi *IP*.

Spazio dei nomi gerarchico basato su un insieme di database distribuiti, garantisce l'aggiornamento di tutta la rete. L'insieme dei nomi viene suddiviso in zone dette domini che possono coprire più host ed essere suddivise in sotto-domini e così via, formando una struttura ad albero in cui i nodi rappresentano i nomi:

- I domini di primo livello o **TLD** (*Top Level Domain*) sono i figli del nodo radice ".", suddivisi in:
  - **gTLD** (*generic TLD*), ad esempio .com, .edu...
  - **ccTLD** (*country-code TLD*) ad esempio .it, .fr, .uk...),
  - **infrastrutturali**: .arpa, usato per la risoluzione inversa dei nomi.
- I domini di secondo livello, in genere, appartengono alle organizzazioni che li hanno registrati e comprendono il loro e il dominio precedente separati da un punto (es. unipg.it).
- I successivi *sotto-domini* vengono creati per rendere la gestione del DNS modulare e seguono la stessa logica dei domini di secondo livello.
- Infine le foglie corrispondono agli host.

Quindi, al contrario degli indirizzi IP, in un nome DNS la parte più importante è la prima partendo da destra (appunto, il TLD). Ad ogni dominio è associato un resource record (RR), file ASCII che contengono records del database DNS.

Server DNS per rendere disponibile lo spazio dei nomi e rispondere alle richieste del resolver risolvendo i nomi (name server)

Client DNS una libreria di funzioni per generare e inviare le richieste sui nomi, interrogando i server DNS (resolver)

Risoluzione La conversione di un nome in un indirizzo è detta *risoluzione*, mentre la conversione di un indirizzo in nome è detta *risoluzione inversa*.
La risoluzione può essere **statica** (mapping stabilito permanentemente tramite una host table) o **dinamica** (mapping stabilito ad ogni avvio dell'host).
Per eseguire la risoluzione il client chiama il risolutore, passandone come parametro il nome.
Questo invia un pacchetto *UDP* a un server DNS locale (*primary server*) che cerca il nome e se è presente nella sua cache lo restituisce; in caso contrario interroga ricorsivamente i server partendo da un root server del TLD fino ad arrivare ai server autorevoli del nome richiesto (*authoritative server*), i quali invieranno la loro risposta al client.

BIND *Berkeley Internet Name Domain* è l'implementazione più comune del DNS su ambiente Unix. BIND è composto da una parte client, il `resolver`, ed una parte server, `named`. La prima è una libreria volta a generare ed inviare le richieste al server; la seconda un demone che risponde alle richieste del resolver. Sia lato server che lato client, la configurazione avviene tramite specifici file di testo.
BIND può essere configurato come

- ***caching-only*** reindirizza ogni richiesta del resolver ad altri server e memorizza il risultato che ritornano in una cache locale
- **authoritative** contiene info su tutta la zona di sua competenza. Può essere
  - ***secondary***: scaricano gli **Zone files** dal *primary server* e li memorizzano in appositi file detti *zone file transfer*;
  - ***primary***: gestiscono le informazioni relative a specifici domini, salvate negli *Zone files*, confgurati dall'amministratore di rete.

Di seguito, le **Configurazioni** del **Resolver**, del **Named** e degli **Zone files** di BIND:

Resolver (/etc/resolv.conf): contiene istruzioni per l'esecuzione delle richieste; Si può usare la configurazione di default, altrimenti è necessario specificare:

- **nameserver <IP-address>**: le richieste saranno inviate all'IP *IP-address*. Si possono specificare al massimo 3 nameserver, nel caso il primo non risponda.
- **domain <name>**: nome del dominio di default che verrà concatenato a sinistra di ogni nome host che non contiene il carattere punto (in caso di fallimento omette i domini meno significativi fino a concatenare solo il TLD).
- **search <domain-1, ..., domain-n>**: come domain ma con la possibilità di avere più domini da provare ad aggiungere al nome host (ma non risale i domini se fallisce).

Named (Server): è necessario configurare più files:

- **/etc/named.conf**: parametri generali di configurazione e puntatori ai file dei domini gestiti dal server (ossia gli zone files)
  - caching-only: si omettono i comandi di configurazione del primary e secondary server tranne il dominio di loopback. `primary 0.0.127.IN-ADDR.ARPA /etc/named.local cache . /etc/named.ca//`
  - primary server: supponendo che il dominio sia *unipg.it* e il primary server *moe*

```
directory                          /etc
primary      unipg.it               named.hosts
primary      250.141.IN-ADDR.ARPA   named.rev
primary      0.0.127.IN-ADDR.ARPA   named.local
cache        .                      named.ca
```

2. il server locale è il primary server per [unipg.it] con zonefile [named.host]
3. puntatore a [named.rev]. Il server locale è il primary server per il reverse domain [205.141.IN-ADDR.ARPA]

– secondary server: supponendo che il dominio sia *unipg.it* e il primary server *moe*

```
directory                                          /etc
secondary    unipg.it               141.250.1.1     unipg.it.hosts
secondary    250.141.IN-ADDR.ARPA   141.250.1.1     250.141.rev
primary      0.0.127.IN-ADDR.ARPA                   named.local
cache        .                                      named.ca
```

2. server locale scarica info su [unipg.it] dal server con IP [141.250.1.1] e memorizza nel file [unipg.it.hosts]
3. il server locale è il secondary server per il reverse domain [250.141.IN-ADDR.ARPA]. I suoi dati vanno scaricati dal server con IP [141.250.1.1] e memorizzati nel file [/etc/250.141.rev]

- **/etc/named.ca**: puntatori ai root domain server. Stabilisce il nome dei root server e i loro indirizzi. ;

```
; formerly NS1.ISI.EDU
.                        3600000          NS          B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000          A          128.9.0.107
...
```

- **/etc/named.local**: zone file per la traduzione del reverse domain 0.0.127.IN-ADDR.ARPA (lookback). Permette quindi la conversione di 127.0.0.1 nel nome "localhost".

```
$TTL  86400
@     IN    SOA        localhost. root.localhost.  (
                       2014030101 ; Serial
                       10800      ; Refresh after 3 hours
                       3600       ; Retry after 1 hour
                       604800     ; Expire after 1 week
                       86400 )    ; Minimum TTL of 1 day
      IN    NS         localhost.
1     IN    PTR        localhost.
```

Dove ad @ verrà sostituito il nome del dominio corrispondente a questo file, e 1 rappresenta l'ultimo numero dell'IP 127.0.0.1 (i primi tre sono in 0.0.127.IN-ADDR.ARPA)

- **/etc/named.hosts**: zone file per la risoluzione diretta

```
@ IN SOA moe.unipg.it.  root.moe.unipg.it.  (

                                2014030101      ; Serial
                                10800           ; Refresh
                                3600            ; Retry
                                604800          ; Expire
                                86400 )         ; Minimum
//Name servers
                               IN NS moe.unipg.it.
   plant                       IN NS pack.plant.unipg.it.
//Mail server
                               IN MX 10 moe.unipg.it.
                               IN MX 20 larry.unipg.it.
   localhost                   IN A 127.0.0.1
//Name to IP mapping
   moe.unipg.it.               IN A 141.250.1.1
   larry.unipg.it.             IN A 141.250.1.2
   omniw.unipg.it.             IN A 141.250.1.40
//Alias
   www                         IN CNAME moe.unipg.it.
//interface specific name
   ns133.unipg.it.             IN A 141.250.5.51
```

- **/etc/named.rev**: zone file per la risoluzione inversa

```
@ IN SOA moe.unipg.it.  root.moe.unipg.it.  (

                                2014030101      ; Serial
                                10800           ; Refresh
                                3600            ; Retry
                                604800          ; Expire
                                86400 )         ; Minimum
//Name server
   250.141.in-addr.arpa.       IN NS moe.unipg.it.
   250.141.in-addr.arpa.       IN NS larry.unipg.it.
//Address point to canonical names
   1.1.250.141.in-addr.arpa. IN PTR moe.unipg.it.
   2.1.250.141.in-addr.arpa. IN PTR larry.unipg.it
```

Zone files File di testo che descrivono un sottoinsieme di domini (spesso un singolo dominio), ogni riga viene detta *Resource Record* (RR) ed è della forma:

$$[name|ttl|recordclass|recordtype|recorddata]$$

con:

- **name**: nome di dominio (in genere si usa @ per riferirlo al dominio definito nello zone file)
- **ttl (time to live)**: tempo di permanenza del RR nella cache di un sistema remoto
- **record class**: sempre `IN`, indica che il record è un INternet DNS RR
- **record type**: il tipo di RR (v. standard resource record)
- **record data**: info specifiche del tipo di RR

I principali componenti di uno zone file sono chiamati "standard resource record" e sono:

- **SOA** (Start of authority): segna l'inizio di un zone file (in genere è il primo record usato e ne esiste uno per zone file), definendo parametri specifici per questo zone file [data]
- **NS** (Name Server): nome del server *[record data]* che ha autorità su questo dominio *[name]*
- **A** (Address record): associa l'hostname *[name]* ad un indirizzo IP *[record data]*
- **PTR** (domain name PoinTR): associa gli indirizzi IP *[name]* ad un nome di host *[record data]*
- **MX** (Mail eXchanger): definisce il server *[record data]* che gestisce la posta per un host o un dominio *[name]*
- **CNAME** (Canonical NAME): definisce un alias per il nome di un host

Comandi: Per avviare il servizio DNS si usa

```
named [
      -c configfile //path di named.conf (default:  /etc/named.conf)
      -d level      //attiva il debug, salvando i log in $dir/named.run
      -p port       //porta a cui deve rispondere il servizio (default:  53)
      -n ncpus      //per sfruttare i sistemi multiprocessore
      -t directory  //cambio di directory dopo aver letto named.conf
      -u user       //utente che esegue named
]
```

Per avviare un tool di debugging (**dig**) si usa

```
dig @server hostname
```

e permette di interrogare un nameserver per ottenere informazioni e verificarne la configurazione.

4.2 Accesso risorse in rete

4.2.1 Telnet

Servizio di emulazione di un terminale a carattere ASCII attraverso la rete (basato su TCP). Effettua l'astrazione del terminale consentendo l'accesso remoto attraverso la rete.
Client: invocato dall'utente, realizza la connessione col *server remoto* e passa i caratteri digitali alla tastiera dall'utente al server, visualizzando l'output nel server.
Server: accetta connessioni di rete, passa i caratteri digitati dall'utente al S.O., come se fossero digitati da taastiera locale. Invia l'output al client.
Basato su 3 aspetti:

- **NTV**: Network Virtual Terminal, terminale virtuale; ogni client e server traduce i comandi nativi in quelli del NTV
- **Opzioni negoziate**: tra client e server aumenta funzionalità di telnet
- **Viste simmetriche**: fanno si che ai lati della comunicazione ci siano programmi invece di una tastiera e un monitor.

Attualmente sostituito con **SSH** (Secure SHell)

4.2.2 Comandi r

Progettati per i sistemi BSD Unix e anch'essi basati sul *TCP*, i comandi r hanno funzionalità analoghe a quelle di *Telnet*:

- `rlogin` (*remote login*) permette di amministrare una serie di macchine autenticandosi una sola volta (*one time login*);
- `rsh` (*remote shell*) consente di eseguire da remoto singoli comandi;
- `rcp` (*remote copy*) abilita alla copia di file attraverso la rete.

Anche in questi casi non vi è alcuna forma di crittografia ed *SSH* sopperisce a questa mancanza.

4.2.3 NIS

Network Information Service: inizialmente chiamato **YP** (Yellow Pages), è un servizio che permette di **definire delle risorse di amministrazione comuni ad un insieme di host**, in modo che l'utente possa utilizzare host differenti mantenendo i suoi dati principali.
Funziona per mezzo di un database (in formato **NIS map**) collocato nel master server, il che permette un controllo centralizzato e la condivisione automatica delle risorse. Le NIS map sono rese disponibili ai client tramite il processo `ypserv` e vengono aggiornate dinamicamente tramite il demone `ypbind`. Sia il server che i client fanno parte di uno stesso **NIS domanin**, il cui nome può essere stabilito tramite il comando `domainname` *`domain`*, mentre `ypwhich` serve a visualizzare l'indirizzo del server.

4.2.4 NFS

Network File System: permette di **condividere directory e file su rete**.
Lato *client*, l'inserimento di una directory collocata in un host remoto viene detto **mounting** ed è realizzato tramite il comando `mount`.
Lato *server*, la condivisione di una cartella con un host specifico è detta **sharing** ed è ottenuta mediante il comando `export`. NFS è modulare, suddiviso in tre parti indipendenti:

- **NFS**, implementata da:
  - `nsfd [nserver`[13]`]`, demone lato server che gestisce le richieste NSF,
  - `biod [nserver]` demone lato server che gestice l'I/O dei client;
- **RPC** (Remote Procedure Code), implementata da:
  - `rpc.locked` demone che gestisce i *lock files* che bloccano istanze multiple di processi specifici,
  - `rpc.statd` demone che controlla lo stato della rete,
  - `rpc.mountd` demone lato server che risponde alle richieste di mount;
- **XDR** (eXternal Data Representation), che consente la condivisione dei file a prescindere dalla codifica specifica di ognuno.

4.3 Posta elettronica

4.4 Servizi di controllo e gestione delle reti

4.4.1 Programma di trasporto

Il programma di trasporto si occupa del *trasferimento dei messaggi* e del servizio di notifica all'utente.
Per trasferimento s'intende il meccanismo con il quale il messaggio viene effettivamente trasmesso: il Mail User Agent trasferisce l'e-mail tramite il protocollo **SMTP** (Simple Mail Transfer Protocol).

SMTP Il Simple Mail Transfer Protocol é adibito al trasporto di messaggi, efficiente ed affidabile, in ambienti eterogenei.
Ad ogni richiesta da parte dell'utente, tramite il comando `HELO` viene attivato un canale di comunicazione bidirezionale sulla porta 25 tra il server SMTP trasmettitore (*sender*), che invia i comandi, e quello ricevente (*receiver*), che invia le risposte.
Una volta creato il canale, l'SMTP-sender invia il comando `MAIL`[14], che contiene i dati riguardanti il mittente, e il receiver risponde con `OK`.
Il sender invia poi il comando `RCPT`, contenente i dati del destinatario, seguito da `DATA`. Se il receiver non puó rispondere, invia un codice di *reject*.
Ogniqualvolta il receiver interpreta correttamente un messaggio risponde con `OK`. Altri comandi utilizzati durande la sessione sono `NOOP`, `HELP`, `EXPN` e `VRFY`. La sessione termina sempre con `QUIT`. Tale dialogo é bloccante.

[13] Numero di processi da eseguire
[14] Comandi alternativi a `MAIL` sono `SEND`, `SOML` e `SAML`

Configurazione di Sendmail La configurazione manuale di Sendmail (file `sendmail.cf`) è diventata nel tempo talmente complessa da essere sconsigliata. Essa viene semplificata tramite uno pseudolinguaggio compilato, **M4**.

4.4.2 DHCP

Il protocollo DHCP *(Dynamic Host Configuration Protocol)* permette agli *host* di una rete locale di ricevere ad ogni richiesta di accesso a una rete IP tutte le informazioni di configurazione necessarie a connettersi ed operare.
Non è utilizzato per la configurazione dei router.
Il DHCP si basa sul modello *client-server*:

- **client**: host che necessita un indirizzo IP per collegarsi alla sottorete;
- **server**: host designato all'assegnazione degli indirizzi ai client che li richiedono. Anche un *router* può assolvere, tra le altre cose, tale ruolo.

È un protocollo nato come complemento del **BOOTP** (Bootstrap protocol), il quale sfrutta l'*UDP* ed assegna gli indirizzi IP tramite messaggi *broadcast*.
Il DHCP lo estende aggiungendo nuove opzioni di configurazione, tra cui la scelta fra *3 metodi di assegnamento degli indirizzi:*

- **automatic allocation**: i client che si connettono ricevono dal server un indirizzo IP permanente;
- **dynamic allocation**: ad ogni nuova connessione il client riceve un indirizzo IP, il quale ha un tempo di validità (*lease*), al cui termine ritorna nella *pool* degli indirizzi disponibili. Ciò permette di riutilizzare indirizzi non più in uso dai client;
- **manual allocation**: il DHCP si limita a comunicare al client l'indirizzo scelto per lui dall'amministratore di rete.

Il DHCP permette di riservare indirizzi IP per specifici client tramite l'associazione al relativo *Indirizzo MAC* (**DHCP Client Reservation**).
Il processo di assegnamento degli indirizzi si divide in 4 fasi:

1. **Discovering**: il client chiede che gli venga assegnato un indirizzo tramite il messaggio `DHCPDISCOVER` inviato in *broadcast*;
2. **Offering**: i server che ricevono la richiesta rispondono (se hanno indirizzi liberi a disposizione) con il messaggio `DHCPOFFER`, in cui propongono un indirizzo IP e gli altri parametri di configurazione al client;
3. **Requesting**: una volta ricevute le offerte dei server, il client le valuta e risponde con `DHCPREQUEST` (sempre in broadcast) per comunicare quale server ha scelto;
4. **Acknowledgment**: se l'assegnamento è avvenuto con successo, il server invia al client la conferma tramite `DHCPACK`; in caso di errori viene invece inviato il messaggio `DHCPNACK` (*negative acknowlegment*).

Configurazione server DHCP :

1. Installazione del software (dipende dal sistema operativo del server);
2. Configurazione della pool di indirizzi: definizione degli intervalli di indirizzi assegnabili ed eventuale *tempo di lease*;
3. Definizione delle opzioni di configurazione dei client che hanno accettato l'IP offerto

In reti con più segmenti si può evitare di definire un server per ogni sottorete ricorrendo ai **DHCP Relay Agent**: è sufficiente configurarne **uno per ogni segmento di rete** al fine di rilevare i pacchetti inviati in *broadcast* (`DHCPDISCOVER` o `DHCPREQUEST`) ed inoltrarli ai server di destinazione, *aggiungendo ad ogni pacchetto il proprio indirizzo.*

Problemi DHCP Il DHCP presenta dei considerevoli problemi di sicurezza (assenza di autenticazione e di cifratura nei messaggi).

5 Sicurezza delle reti

Minacce Ogni host connesso ad una rete è sottoposto ad un gran numero di minacce, ad esempio **accessi non autorizzati** o **DoS** (*Denial of Services*.
Esistono diversi strategie per proteggere una rete da questi ed altri attacchi, le principali delle quali sono **Oscuramento**, **Hardening** e uso di **Firewall**.

5.1 Oscuramento

Per oscuramento si intende il garantire la sicurezza nascondendo le risorse di rete mediante strumenti come *NAT*, IP Masquerading, GPG e trasmissioni crittografate.

5.1.1 Encryption

Limita gli accessi ai dati trasmessi, crittografando i contenuti.
In ambienti Unix, si usano i comandi `crypt` e `des`.

5.2 Hardening

Per hardening si intende la gestione della sicurezza a livello del singolo host.

5.2.1 TCP-wrapper

I TCP-wrapper consentono di limitare l'accesso ai servizi basandosi sull'IP e l'hostname del client.
Le richieste verso l'host vengono elaborate dal wrapper, che verifica che l'indirizzo del chiamante sia *incluso nell'elenco* di `/etc/hosts.allow`: se è così, o *se non è incluso* in `/etc/hosts.deny`, permette l'accesso.
Tramite le seguenti **keyword** è possibile inoltre specificare host o gruppi di host:

- `ALL` (tutti gli host);
- `LOCAL` (tutti gli host locali);
- `KNOWN` (host riconosciuti dal sistema);
- `UNKNOWN` (host non riconosciuti);
- `PARANOID` (host il cui nome non corrisponde all'indirizzo).

In entrambi i file è anche possibile definire delle **regole di filtraggio**, che hanno effetto dal basso verso l'alto; inoltre quelle di `hosts.allow` hanno la precedenza su quelle di `hosts.deny`.
Il formato delle regole è:
`<elenco-servizi> : <elenco-client> [: spawn`[15] `<comando-shell>]`

[15] permette di eseguire normali comandi da shell qualora una richiesta entrante soddisfi una regola.

5.2.2 xinetd

Demone che estende le funzionalità di `inetd`, non si limita a gestire l'accesso ai servizi di rete, *ma controlla anche i servizi stessi* al Livello delle applicazioni.
Viene configurato mediante `/etc/xinetd.conf` ed altri file (uno per ogni servizio) posti nella directory `/etc/xinetd.d`, da importare nel file di configurazione generale di xinetd mediante il comando `includedir`.

5.3 Firewall

Sistema hardware e/o software che costituisce un *intermediario* tra la rete locale (o singolo host) ed una o più reti esterne (tipicamente internet), **filtrando il traffico**.
Tipicamente, un firewall è un *DTE* con più schede di rete che costituisce l'unico punto di collegamento tra le reti coinvolte.
Nella rete *interna* ci saranno i servizi di base (come *NFS*, *NIS*...) rivolti agli utenti della sottorete interna; nella rete *esterna* i servizi di networking (ad esempio *DNS*, *SMPT*, *FTP* etc.), rivolti sia ad utenti interni che esterni e dunque esposti a rischi.
Si distinguono due tipi di firewall:

- firewall **stateless**, che prendono decisioni basate esclusivamente sulla specifica connessione. Un esempio di questo tipo di firewall è `ipchains`;
- firewall **stateful**, che tengono traccia delle connessioni e prendono dunque decisioni basate sulle varie connessioni. Un esempio è l'attuale `iptable`.

Packet filtering Metodo con il quale un firewall limita il traffico di rete.
In Linux ***NetFilter*** è il filtro di pacchetti implementato nel kernel, la cui interfaccia è rappresentata dal comando `iptables`.
`iptables` contiene alcune **TABLES**, che rappresentano le liste di regole di filtraggio. Queste contengono le **CHAINS** (catene in cui i pacchetti vengono controllati), che a loro volta sono formate da molte **Rules** (composte da un campo **match** e uno **target**).
Ogni pacchetto attraversa almeno una catena, di cui ogni regola controlla (proceduralmente) se il pacchetto soddisfa il campo *match*: se così accade, gli viene applicato il **target**. I principali sono **ACCEPT** viene accettato ed instradato, **REJECT** se lo rifiuta ed avvisa il mittente con un errore o **DROP** se lo blocca e lo scarta senza notifica.
Le tabelle di iptables sono:

- **filter**: regolamenta il firewalling vero e proprio, inteso come filtraggio; permette cioè di far passare i pacchetti `ACCEPT`, rifiutarli ed avvisarne il mittente con un messaggio di errore `REJECT` o scartarli senza alcuna forma di notifica `DROP`. Le catene predefinite di suddetta tabella sono:
  - `INPUT`: vi passano i pacchetti destinati al firewall stesso,
  - `OUTPUT`: vi passano i pacchetti originati dal firewall stesso, diretti altrove,
  - `FORWARD`: vi passano i pacchetti che transitano nel firewall pur provenendo da altri host, diretti altrove;
- **nat**: regola le attività di natting, ossia di modifica degli indirizzi IP nell'ambito dell'*Oscuramento*. Consiste solitamente nel tradurre gli indirizzi di una rete privata, globalmente non univoci, in un unico indirizzo IP pubblico, in modo da consentire alla *LAN* di connettersi ad *Internet* e di nasconderla. Le catene predefinite di suddetta tabella sono:

  - `PREROUTING`: vi passano i pacchetti su cui non sono ancora state fatte scelte di routing,
  - `POSTROUTING`: vi passano i pacchetti su cui sono già state fatte scelte di routing,
  - `OUTPUT`: come in filter, vi passano i pacchetti originati dal firewall stesso, diretti altrove;

  `PREROUTING` e `POSTROUTING` sono state introdotte per consentire di realizzare particolari comportamenti: ad esempio, se si vuole cambiare l'indirizzo di destinazione di un pacchetto sarà importante farlo prima che venga deciso da dove fare uscire il pacchetto (se così non fosse, il pacchetto uscirebbe dall'interfaccia sbagliata, mentre quando si cambia l'indirizzo del mittente di un pacchetto potrebbe essere importante farlo dopo che una decisione di routing è stata presa. I possibili target sono:

  - `SNAT`, che consente di cambiare l'indirizzo ip sorgente di un pacchetto (`POSTROUTING`). Questa azione viene solitamente intrapresa per avere la possibilità di avere indirizzi privati già utilizzati da altre LAN senza creare conflitti o per nascondere la rete privata dall'esterno sotto un indirizzo pubblico (IP Masquerading).
  - `DNAT`, che consente di cambiare l'indirizzo ip destinazione di un pacchetto (`PREROUTING`),
  - `MASQUERADE`: tipo particolare di `SNAT`, che fa in modo che i pacchettini abbiano come mittente l'indirizzo IP della interfaccia di rete dalla quale usciranno (`POSTROUTING`),
  - `REDIRECT`, versione semplificata del `DNAT` che consente di cambiare la porta di destinazione di un pacchetto (`PREROUTING`);

- **mangle**: responsabile delle modifiche alle opzioni dei pacchetti. Ha per catene predefinite:
  - `PREROUTING`: esamina tutti i pacchetti che entrano nel sistema (prima di sapere se sono destinati allo stesso o se devono essere inoltrati);
  - `OUTPUT`: tutti i pacchetti generati dal sistema passano per questa catena.

NetFilter è quindi in grado di stabilire il **contesto** del pacchetto: **NEW** (appartiene ad una nuova connessione), **ESTABLISHED** (ad una connessione esistente), **RELATED** (legato ad una connessione esistente) o **INVALID** (pacchetto sospetto, non legato ad alcuna connessione stabilita).
Il packet filtering opera al *Livello di rete*, per cui non si preoccupa di quali siano le applicazioni che generano il traffico. Tuttavia, non tutti i firewall operano allo stesso livello: ad esempio, un firewall che lavora anche al *Livello delle applicazioni* è il **Proxy**.

Architetture di firewall Un firewall può essere configurato in diverse modalità:

- **Screening router** o **firewall router**: il firewall opera un routing filtrato sul traffico tra gli host interni e l'esterno. Sistema basilare, usato per proteggere sottoreti;
- **Dual-homed host**: un host con almeno due interfacce di rete divide il traffico di rete e garantisce i servizi base di rete. Sistema economico ma a *single point of failure*;
- **Screened host**: oltre al firewall router, il traffico tra la rete interna ed esterna passa attraverso un host della rete interna *(bastion host)* il quale fornisce servizi di base. Lo *screening-router* accetta solo pacchetti da e verso il bastion host.
- **Screened sub-net**: viene creata una *rete intermedia* (**DMZ** - De-Militarized Zone) usando un firewall router esterno ed uno interno; *i bastion host risiedono nella DMZ*, non rappresentando più un punto critico.

5.3.1 Proxy

Con il termine **proxy** si indica un server collocato tra un host e un web server il quale funge da intermediario fra i due, disaccoppiando l'accesso ad internet dal browser.
I proxy sono largamente impiegati come *Firewall* al *Livello delle applicazioni*, oltre a poter essere utilizzati in catena per garantire l'anonimato in internet.

5.3.2 Sicurezza nel WWW

La capillare diffusione del *WWW* lo rende forse il servizio più critico.

Cookies Introdotti con Netscape 2.0, i cookies sono *stringhe di caratteri ASCII* che vengono passate dal web server al web browser.
Hanno la funzione di tenere traccia delle scelte dell'utente per meglio orientare il telemarketing.
I cookies pongono problemi nell'ambito della privacy dell'utente. In risposta è nato *eTrust*, un programma che definisce uno standard per la privacy online.

5.4 Monitoraggio

C'è la possibilità di effettuare un monitoraggio continuo sul sistema usando gli **IDS** *(Intrusion Detecting Systems),* che possono essere:

- **Host-based** IDS: verifica periodicamente i log dei servizi di rete e di sistema e controlla l'integrità dei dati e dei filesystem (comandi `diff` ed `rpm -V`)
- **Network-based** IDS: scandisce il traffico di rete, segnalando quelli sospetti ed assegnandogli un *livello di pericolosità* in base a dei database.

In alternativa agli IDS, si possono utilizzare i comandi Unix-Linux:

- `ps -aux`: mostra i processi attivi;
- `who`: restituisce la lista degli utenti collegati al sistema;
- `last`: mostra il contenuto del file `/usr/adm/wtmp` (registro dei collegamenti al sistema);
- `ls -lR`: crea una lista con le informazioni su ogni file di sistema, la quale può essere poi confrontata con con una precedente tramite il comando `diff`.

Audit trail Detto anche *audit log*, è costituito da uno o più registri che *documentano in ordine cronologico le attività svolte entro un sistema*, utile per permettere la ricostruzione degli eventi.