





# Architetture Hardware di Laboratorio

Alberto Dai Prè  
(vr067036)

Corso di Laurea in Bioinformatica

Università degli Studi di Verona  
A.A. 2008/2009



# Indice

<b>I</b>	<b>La Comunicazione tra sistemi automatici</b>	<b>1</b>
<b>1</b>	<b>Reti di calcolatori e loro importanza</b>	<b>3</b>
1.1	Classificazione delle reti sulla base dell' estensione geografica .	4
1.2	Classificazioni delle reti per tipo di trasmissione . . . . .	4
1.2.1	Principali topologie di rete . . . . .	5
1.3	Modello client/server e modello peer-to-peer . . . . .	6
1.4	Tempo di propagazione, capacità del canale e bitrate . . . . .	7
<b>2</b>	<b>Il modello ISO/OSI</b>	<b>9</b>
2.0.1	I livelli . . . . .	10
2.1	Suite di protocolli Internet (TCP/IP) . . . . .	12
2.1.1	Come funziona il TCP/IP? . . . . .	12
2.2	Commutazione di circuito . . . . .	13
2.3	Commutazione di pacchetto . . . . .	14
2.3.1	Pacchetti . . . . .	14
2.3.2	Commutatori . . . . .	14
2.3.3	Tempi di ritardo . . . . .	15
2.3.4	Perdita di pacchetti . . . . .	16
2.4	Confronto tra commutazione di pacchetto e commutazione di circuito . . . . .	16
2.5	Multiplexazione . . . . .	16
2.6	Il Protocol Data Unit . . . . .	17
2.7	Servizio con connessione e servizio senza connessione . . . . .	17
2.7.1	La modalità connessa . . . . .	18
2.7.2	La modalità non connessa . . . . .	19
2.7.3	Applicazioni connesse e non . . . . .	19
<b>3</b>	<b>Livello Fisico</b>	<b>21</b>
3.1	La trasmissione . . . . .	21
3.1.1	I nemici della trasmissione . . . . .	22
3.2	Trasmissione su cavo . . . . .	23

3.2.1	Il doppino di rame . . . . .	24
3.2.2	La fibra ottica . . . . .	25
3.3	Il cablaggio . . . . .	28
3.3.1	Cablaggio strutturato . . . . .	28
3.4	RS-232 . . . . .	29
3.4.1	Le unità di misura . . . . .	29
3.4.2	Successori di RS-232 . . . . .	30
3.4.3	Limiti di RS-232 e dei suoi successori . . . . .	31
<b>4</b>	<b>Livello DataLink</b>	<b>33</b>
4.1	Il framing . . . . .	33
4.2	Controllo degli errori . . . . .	34
4.3	Controllo di flusso . . . . .	35
4.4	Protocolli . . . . .	36
4.4.1	Sottolivello LLC . . . . .	36
4.4.2	Sottolivello MAC . . . . .	37
4.5	Il frame Ethernet . . . . .	39
<b>5</b>	<b>Il Livello Rete</b>	<b>41</b>
5.1	Livello Rete in IP . . . . .	42
5.2	L'Internet Protocol . . . . .	42
5.3	Indirizzo IP . . . . .	42
5.3.1	Domain Name System . . . . .	43
5.3.2	Indirizzamento IP statico e dinamico . . . . .	43
5.3.3	Classi di indirizzi IP . . . . .	44
5.4	Il router . . . . .	44
5.5	La Virtual LAN (VLAN) . . . . .	46
5.5.1	Realizzazione e terminologia in IEEE 802.1Q . . . . .	46
<b>6</b>	<b>Il Livello Trasporto</b>	<b>49</b>
6.1	Il Transmission Control Protocol ed il User Datagram Protocol . . . . .	50
<b>7</b>	<b>Il Livello Applicazione</b>	<b>53</b>
7.1	La storia di Internet . . . . .	53
7.2	Tipi di connessione . . . . .	54
7.3	Il World Wide Web . . . . .	54
<b>II</b>	<b>Acquisizione e memorizzazione delle informazioni</b>	<b>57</b>
<b>8</b>	<b>Acquisizione da sensori</b>	<b>59</b>

---

8.1	Tipologie di sensori . . . . .	60
8.2	Segnale analogico e segnale digitale . . . . .	61
8.3	Cenni sulla compressione delle informazioni . . . . .	63
8.3.1	Codifica entropica . . . . .	64
<b>9</b>	<b>Memorizzazione su supporti di massa</b>	<b>67</b>
9.1	Memoria di massa . . . . .	67
9.1.1	Disco magnetico . . . . .	69
9.1.2	Disco ottico . . . . .	70
9.1.3	Memoria flash . . . . .	71
9.1.4	Il disco a stato solido . . . . .	73
9.2	Redundant Array of Inexpensive/Indipendent Disks (RAID) .	73
9.2.1	Implementazioni RAID: il Data Striping . . . . .	74
9.2.2	Livelli RAID standard . . . . .	74
9.3	Storage Area Network (SAN) . . . . .	76
<b>10</b>	<b>File e File System</b>	<b>79</b>
10.1	Concetto di file . . . . .	79
10.1.1	Attributi del file . . . . .	79
10.1.2	Operazioni sui file . . . . .	80
10.2	Struttura del File System . . . . .	82
10.3	Realizzazione del File System . . . . .	83
10.4	Metodi di allocazione . . . . .	84
10.4.1	Allocazione contigua . . . . .	85
10.4.2	Allocazione concatenata . . . . .	87
10.4.3	Allocazione indicizzata . . . . .	88
10.4.4	Prestazioni . . . . .	89
10.5	Gestione dello spazio libero . . . . .	90
10.5.1	Vettore di bit . . . . .	90
10.5.2	Lista concatenata . . . . .	90
10.5.3	Raggruppamento . . . . .	90
10.5.4	Conteggio . . . . .	91
10.6	Efficienza e prestazioni . . . . .	91
10.7	Cache del disco . . . . .	92
10.7.1	Problematiche . . . . .	93
<b>11</b>	<b>Alcuni formati di memorizzazione</b>	<b>95</b>
11.1	Grafica raster . . . . .	95
11.1.1	Proprietà della grafica raster . . . . .	95
11.1.2	Formati di immagini raster . . . . .	96
11.2	Grafica vettoriale . . . . .	97

11.2.1 Vantaggi . . . . .	98
11.2.2 Svantaggi . . . . .	98
11.3 Joint Photographic Experts Group (JPEG) . . . . .	99
11.4 Tagged Image File Format (TIFF) . . . . .	100
11.5 Graphics Interchange Format (GIF) . . . . .	101
11.6 Portable Network Graphics (PNG) . . . . .	101
11.7 Scalable Vector Graphics (SVG) . . . . .	102
 <b>III Famiglie di CPU e Multiprocessori</b>	 <b>105</b>
<b>12 La CPU</b>	<b>107</b>
12.1 Struttura della CPU . . . . .	107
12.2 Architetture CISC ed architetture RISC . . . . .	110
12.2.1 Complex Instruction Set Computer (CISC) . . . . .	110
12.2.2 Reduced Instruction Set Computer (RISC) . . . . .	111
12.2.3 RISC vs CISC . . . . .	111
12.3 Intel 80286I/80386/80486 . . . . .	112
12.4 Intel Pentium . . . . .	113
12.4.1 Pentium MMX . . . . .	114
12.4.2 Intel P6 . . . . .	115
12.5 Hyper-Threading . . . . .	116
 <b>13 Il multiprocessore</b>	 <b>119</b>
13.1 Tassonomia dei calcolatori . . . . .	119
13.1.1 Processore vettoriale . . . . .	124
 <b>IV Tracciabilità</b>	 <b>127</b>
<b>14 I codici a barre</b>	<b>131</b>
14.1 Tipi di codici a barre . . . . .	131
14.2 Lettura dei codici a barre . . . . .	131
14.2.1 Tipo di collegamento . . . . .	132
14.2.2 Tecnologia di lettura . . . . .	132
 <b>15 Radio Frequency Identification (RFID)</b>	 <b>133</b>
15.1 ZigBee . . . . .	135
15.1.1 Introduzione . . . . .	135
15.1.2 Usi . . . . .	135
15.1.3 Tipi di dispositivo . . . . .	135



---

<b>16 EPCglobal</b>	<b>137</b>
16.1 EPCIS . . . . .	137
16.1.1 Interfaccia di programmazione di un' applicazione . . .	138



# Parte I

## La Comunicazione tra sistemi automatici



# Capitolo 1

## Reti di calcolatori e loro importanza

Una **rete di calcolatori** è un sistema che permette la condivisione di informazioni e risorse (sia hardware che software) tra diversi calcolatori. Il sistema fornisce un servizio di trasferimento di informazioni ad una popolazione di utenti distribuiti su un' area più o meno ampia.

La costruzione di reti di calcolatori può essere fatta risalire alla necessità di condividere le risorse di calcolatori potenti e molto costosi (i mainframe). La tecnologia delle reti, e in seguito l' emergere dei computer personali a basso costo, ha permesso sviluppi nell' organizzazione delle risorse di calcolo. Si possono indicare almeno tre vantaggi di una rete di calcolatori rispetto al mainframe tradizionale:

- resistenza ai guasti: il guasto di una macchina non blocca tutta la rete, ed è possibile sostituire il computer guasto facilmente;
- economicità: come accennato sopra, hardware e software per computer costano meno di quelli per i mainframe;
- gradualità della crescita e flessibilità: l' aggiunta di nuove potenzialità ad una rete già esistente e la sua espansione sono semplici e poco costose.

Tuttavia, una rete ha alcuni svantaggi rispetto a un mainframe:

- scarsa sicurezza: chiunque può avere accesso più facilmente ad una rete di computer che ad un mainframe! Inoltre, una volta che un worm abbia infettato un sistema della rete, questo si propaga rapidamente a tutti gli altri, e l' opera di disinfezione è molto lunga, difficile e non offre certezze di essere completa;

- alti costi di manutenzione: con il passare del tempo e degli aggiornamenti, e con l'aggiunta di nuove funzioni e servizi, la struttura di rete tende ad espandersi ed a diventare sempre più complessa, ed i computer che ne fanno parte sono sempre più eterogenei, rendendo la manutenzione sempre più costosa in termini di ore lavorative.

## 1.1 Classificazione delle reti sulla base dell'estensione geografica

A seconda dell'estensione geografica, si distinguono diversi tipi di reti:

- si parla di **rete personale** o **PAN** (Personal Area Network) se la rete si estende intorno all'utilizzatore con una estensione di alcuni metri;
- si parla di **rete locale** o **LAN** (Local Area Network) se la rete si estende all'interno di un edificio o di un comprensorio, con una estensione entro alcuni chilometri. Un caso particolare la **rete senza fili** o **WLAN** (Wireless Local Area Network) se la rete locale è basata su una tecnologia in radio frequenza (RF), permettendo la mobilità all'interno dell'area di copertura;
- si parla di **rete metropolitana** o **MAN** (Metropolitan Area Network) se la rete si estende all'interno di una città;
- si parla di **rete geografica** o **WAN** (Wide Area Network) se la rete si estende in un'ampia area geografica;
- si parla di **Internetwork** se collega più reti differenti (in termini sia di hardware che di software) mediante opportuni elementi di interfaccia che si possono estendere su tutto il pianeta.

## 1.2 Classificazioni delle reti per tipo di trasmissione

Due sono le topologie principali, in base alla tecnologia assunta come modalità per il trasferimento dei dati: *reti punto a punto* e *reti broadcast*. Le **reti punto a punto** (**point-to-point**) consistono in un insieme di collegamenti tra coppie di elaboratori, che formano grafi di vario tipo (stella, anello, albero, grafo completo). Per passare da una sorgente ad una destinazione, l'informazione deve attraversare diversi elaboratori intermedi. La strada

che i dati devono seguire per arrivare correttamente a destinazione è data dai *protocolli di routing*. Il *routing* è l'insieme delle problematiche relative al corretto ed efficace instradamento dei dati sulla rete.

Le **reti broadcast**, invece, sono formate da un unico mezzo fisico condiviso da più elaboratori, dove i messaggi inviati da un elaboratore sono ricevuti da tutti gli altri. All'interno del messaggio vi è l'indirizzo del destinatario, in modo che tutte le altre macchine in ascolto possano scartare il messaggio in arrivo. Alcune reti prevedono indirizzi speciali di tipo *broadcast* e *multicast*. Il broadcast permette di inviare messaggi a tutte le stazioni collegate al mezzo fisico, mentre il multicast permette di farlo solo ad un gruppo di stazioni, ma non a tutte. Un esempio di una tale rete è la comunissima Ethernet.

### 1.2.1 Principali topologie di rete

Le moderne reti broadcast sono realizzate con una **topologia fisica a stella** in cui le connessioni fanno capo ad un unico nodo centrale; consente un controllo centralizzato delle comunicazioni. Tra i suoi vantaggi si ricordano le prestazioni elevate (ogni computer ha il suo collegamento con il nodo centrale), la facilità di controllo centralizzato del server e la semplicità del protocollo di comunicazione. Tra i suoi svantaggi si ricordano la possibilità di sovraccarico in caso di traffico elevato (con possibilità di un blocco delle comunicazioni) e la dipendenza dall'affidabilità del nodo centrale (dato che un guasto blocca l'intera rete).

Esiste, anche, la *topologia a bus* in cui tutti gli elaboratori sono connessi ad un unico canale comune di comunicazione; richiede un mezzo trasmissivo intrinsecamente bidirezionale. Tra i suoi vantaggi si ricordano la semplicità, la flessibilità, i bassi costi e l'affidabilità. L'unico svantaggio è che tutte le stazioni dipendono da un solo mezzo trasmissivo condiviso: le prestazioni possono divenire un fattore critico nel momento di traffico elevato.

Infine, nella *topologia ad anello* la connessione è circolare tra tutte le stazioni collegate: l'informazione transita in modo unidirezionale ed è ricevuta a turno da ogni stazione, che verifica se essa è la destinataria del messaggio. In caso negativo, la stazione rigenera il segnale e lo trasmette alla stazione successiva. Ha come vantaggio che un unico anello può estendersi su distanze elevate. Tra i suoi svantaggi si ricordano la limitata flessibilità e la limitata affidabilità della rete (per aumentarla, si realizzano reti a doppio anello, con due collegamenti, uno per direzione, in modo che la rete mantenga la sua funzionalità anche in caso di guasto di una stazione).

### 1.3 Modello client/server e modello peer-to-peer

Il **modello client-server** è un tipo di applicazione di rete nel quale un *computer client* richiede un servizio ad una *server application* o ad un sistema di database. La presenza di un server permette ad un certo numero di client di dividerne le risorse, lasciando che sia il server a gestire gli accessi alle risorse per evitare conflitti. Il software client, in genere, è di limitata complessità, limitandosi normalmente ad operare come interfaccia verso il server. In generale, nel campo informatico, il termine ‘client’ indica una componente che accede ai servizi o alle risorse di un’ altra componente, detta ‘server’. In questo contesto si può, quindi, parlare di ‘client’ riferendosi all’ hardware o al software. Il software server, oltre alla gestione logica del sistema, deve implementare tutte le tecniche di gestione degli accessi, allocazione e rilascio delle risorse, condivisione e sicurezza dei dati o delle risorse.

Il client ed il server sono in collegamento tramite un protocollo di comunicazione attraverso una rete di comunicazione. Il protocollo può essere in chiaro o, in certi casi, crittografato. Nell’ ambito delle telecomunicazioni, due o più macchine o host (computer, telefono, stampante, ecc...) possono comunicare tra loro rispettando norme che sono dette **protocolli di rete**. L’ aderenza ai protocolli garantisce che due macchine possano comunicare correttamente, anche se sono state realizzate indipendentemente.

Generalmente, per **modello peer-to-peer** si intende una rete di computer o qualsiasi rete informatica che non possiede nodi gerarchizzati come nel modello client/server, ma un numero di nodi equivalenti che fungono sia da cliente che da servente verso altri nodi della rete. Mediante questa configurazione, qualsiasi nodo è in grado di avviare o completare una transazione. I nodi equivalenti possono differire nella configurazione locale, nella velocità di elaborazione, nella ampiezza di banda e nella quantità di dati memorizzati.

Vantaggi e svantaggi del modello peer-to-peer sono relativi al tipo di ambiente in cui si decide di installare questo tipo di rete, ma sicuramente si deve tenere presente che:

- non si deve acquistare un server con potenzialità elevate e, quindi, non se ne deve sostenere il costo, ma ogni computer deve avere i requisiti per sostenere l’ utente in locale e in rete, ma anche gli altri utenti che desiderano accedere alle risorse di questo in remoto;
- ogni utente condivide localmente le proprie risorse ed è amministratore del proprio client-server. Questo da un lato può essere positivo per una



questione di “indipendenza”, ma dall’ altro richiede delle competenze ad ogni utente, soprattutto per quel che riguarda la protezione;

- la velocità media di trasmissione dei dati è molto più elevata di una classica rete con sistema client/server, dal momento che l’ informazione richiesta da un client può essere reperita da numerosi client connessi in modo paritario, anziché da un unico server (questo tipo di condivisione diventa tanto più efficace tanti più sono i client connessi, in antitesi con la rete tradizionale client/server, dove un elevato numero di client connessi riduce la velocità di trasmissione dati per utente).

## 1.4 Tempo di propagazione, capacità del canale e bitrate

Il **tempo di propagazione** è il tempo tra l’ entrata di un bit nel canale e il suo arrivo nel sistema di destinazione; la *capacità del canale* è il numero massimo di bit che possono essere trasmessi nell’ unità di tempo; il **bitrate** è il numero di bit trasmessi da un nodo nell’ unità di tempo. La capacità del canale ed il bitrate possono cambiare nel tempo.

La capacità totale di un percorso in rete tra due end-system è condizionata dalla minima tra le capacità dei link che lo formano e la capacità di accettazione del ricevitore (dipende dalla sua potenza). Se il bitrate emesso da un end-system è maggiore della capacità totale del percorso si perde informazione.



# Capitolo 2

## Il modello ISO/OSI

Alla fine degli anni Settanta, l' ISO sentì la necessità di proporre una serie di standard per le reti di calcolatori e avviò il *Progetto OSI* (Open System Interconnection), uno standard che propose un modello di riferimento per l' interconnessione di sistemi aperti. Il modello di riferimento OSI ha due scopi:

1. fornire una base comune su cui sviluppare standard per l' interconnessione di sistemi informatici;
2. fornire un modello rispetto a cui confrontare le architetture di rete proprietarie.

Il modello di riferimento OSI non ha come scopo la definizione di servizi o protocolli specifici. A questo sono stati delegati altri enti (es. IEEE, CCITT) o l' ISO stessa, in tempi successivi.

OSI introduce il concetto di *sistema* (system) come un insieme di uno o più elaboratori con il relativo software, periferiche, terminali, operatori umani, processi, ecc. che complessivamente è in grado di elaborare dati. Nell' ambito di un sistema, un' *applicazione* è l' elemento che effettivamente svolge l' elaborazione dei dati.

Lo standard OSI tratta lo scambio di informazioni tra i sistemi e non come i sistemi sono realizzati o funzionano al loro interno.

L' architettura del modello di riferimento OSI è stata progettata pensando a tre componenti principali:

1. il processo applicativo che deve scambiare le informazioni;
2. la connessione che permette lo scambio delle informazioni;
3. i sistemi.

Per ridurre la complessità progettuale, OSI introduce un' *architettura a livelli* (layered architecture) i cui componenti principali sono:

- i *livelli* (layers);
- le *entità* (entities);
- i *punti di accesso al servizio* (Service Access Points);
- le *connessioni* (connections).

### 2.0.1 I livelli

In una tale architettura, ciascun sistema è decomposto in un insieme ordinato di livelli: in ciascun livello è usato uno specifico protocollo.

La divisione in livelli è fatta in modo che ciascun livello utilizzi i servizi offerti dal livello inferiore, e fornisca servizi più “ricchi” al livello superiore. I diversi livelli in un host comunicano tra loro tramite le **interfacce**. Ogni livello parla solo con quello immediatamente superiore e con quello immediatamente inferiore. Ogni livello è, poi, composto da una o più **entità**. Entità appartenenti allo stesso livello, su sistemi diversi, sono dette **entità peer-entities**. Lo scopo di ciascun livello è quello di fornire servizi alle entità del livello superiore, mascherando il modo in cui questi sono implementati. Ad eccezione del livello più alto, un livello  $N$  fornisce servizi di livello  $N$  alle entità di livello  $N + 1$ . Le entità di livello  $N$ , eccetto il livello 1, per comunicare usano servizi di livello  $N - 1$ . Le entità di livello 1 comunicano direttamente tramite i mezzi trasmissivi che le interconnettono. Le entità usano e forniscono servizi tramite i **Service Access Points**. Le operazioni specifiche di un livello, cioè la cooperazione tra le entità appartenenti a quel livello, sono realizzate da un insieme di protocolli. Affinché due entità di livello  $N$  su sistemi diversi possano scambiarsi informazioni, una connessione deve essere stabilita nel livello  $N - 1$  usando un protocollo di livello  $N - 1$ . Tale connessione di livello  $N - 1$  è stabilita tra due Service Access Point di livello  $N - 1$ .

I vari livelli sono organizzati in *pile di protocolli*. Le pile di protocolli sono un modo flessibile per combinare componenti per realizzare un servizio.

La struttura serve ad adempiere ad alcuni compiti:

- controllo dell' errore;
- controllo del flusso;
- frammentazione e riassemblaggio;

- multiplexing, in modo che sessioni dello strato più alto possano condividere una singola connessione dello strato più basso;
- instaurazione della connessione.

Il livello più basso è detto *Livello Fisico* e si occupa di gestire la trasmissione dei segnali attraverso il mezzo di trasporto. Il livello più elevato è chiamato *Livello Applicativo* ed è quello che permette all'utente di creare il messaggio da comunicare. In dettaglio, i livelli sono:

- **Livello 1: fisico.** È responsabile della gestione dei particolari meccanici ed elettrici della trasmissione fisica di un flusso di bit.
- **Livello 2: datalink.** È responsabile delle funzioni di gestione dei frame, o di parti di pacchetti di lunghezza fissa, comprese l'individuazione e la correzione degli errori che si sono verificati nello strato fisico.
- **Livello 3: rete.** È responsabile della fornitura dei collegamenti e dell'instradamento dei pacchetti nella rete di comunicazione.
- **Livello 4: trasporto.** È responsabile delle funzioni d'accesso a basso livello alla rete e del trasferimento dei messaggi tra i client, comprese la suddivisione dei messaggi in pacchetti, l'ordinamento dei pacchetti, il controllo del flusso e la generazione degli indirizzi fisici.
- **Livello 5: sessione.** È responsabile della realizzazione di sessioni, o di protocolli di comunicazione da processo a processo.
- **Livello 6: presentazione.** È responsabile della risoluzione delle differenze di formato che si possono presentare tra diversi siti della rete.
- **Livello 7: applicazione.** È responsabile dell'interazione diretta con gli utenti. Tratta il trasferimento di file, la posta elettronica, ecc.

La divisione in livelli è piuttosto rigida a livello di specifica dei protocolli, mentre nell'implementazione spesso diversi livelli sono implementati insieme in uno stesso modulo software.

In una rete a pacchetto, ciascun livello della pila protocollare aggiunge ai pacchetti una intestazione, attraverso una operazione detta *imbustamento* (descritta nel paragrafo 2.6. Il termine si applica anche ad alcune *reti a commutazione di circuito*, come SDH, dove l'imbustamento è un circuito dedicato a trasmettere informazioni di controllo.

## 2.1 Suite di protocolli Internet (TCP/IP)

La **suite di protocolli Internet** è un insieme di protocolli di rete che implementa la pila di protocolli su cui funziona Internet. È chiamata suite di protocolli TCP/IP in funzione dei due più importanti protocolli in essa definiti: il Transmission Control Protocol (TCP) e l' Internet Protocol (IP). Tale suite può essere descritta per analogia con il modello OSI, che descrive i livelli della pila di protocolli.

### 2.1.1 Come funziona il TCP/IP?

L' IP è uno dei principali protocolli del livello InternetWorking del modello DOD/DARPA (o, nel caso del modello OSI, del Livello Network) serve per la comunicazione su una rete a commutazione di pacchetto. Un indirizzo IP identifica ogni nodo della rete ed è un identificativo numerico utilizzato in combinazione con l' indirizzo MAC, questo per identificare in modo univoco ogni scheda di rete. Il TCP fornisce al protocollo FTP, che lavora a livello di applicazione, un canale di trasferimento dati che sia affidabile. Il TCP, nell' invio dei pacchetti, usa il *meccanismo della Window*. Una serie di pacchetti è inviata seguendo delle regole ben precise:

1. ad ogni pacchetto spedito il trasmettitore fa partire un time-out;
2. il ricevitore invia per ogni pacchetto ricevuto un ACK (è il simbolo che identifica un segnale di *Acknowledge* emesso in risposta alla ricezione di un' informazione completa), indicando la sequenza dell' ultimo pacchetto ricevuto correttamente. L' ACK può essere trasmesso in un messaggio a sé stante, o essere inviato in un campo di un pacchetto utente in direzione opposta, ossia in *piggyback*;
3. il trasmettitore considera, quindi, spediti tutti i pacchetti successivi.

Quando ci vogliamo collegare con il nostro browser ad un server web, stabiliamo un collegamento (virtuale) a Livello Applicazione. Il Livello di Trasporto si occupa dei dettagli del Livello Applicazione.

Al Livello di Trasporto, il protocollo TCP mette in coda i messaggi delle applicazioni (browser e server) e li trasmette sotto forma di pacchetti; il buon fine della spedizione è attestato da una ricevuta di ritorno. Anche questo è un collegamento virtuale tra le due applicazioni, i cui dettagli sono demandati al Livello di Rete.

Al Livello di Rete, il protocollo IP decide quale strada seguire per trasmettere effettivamente i messaggi da un computer all' altro. Un computer

spedisce, l'altro riceve, ma è un collegamento virtuale tra i due computer remoti, dei quali i dettagli se ne occupa il Livello di Collegamento.

Al Livello di Collegamento, si decide come fare il trasferimento del messaggio per ogni singolo tratto del percorso: dal computer del browser al primo router, dal primo router al secondo, dal secondo al terzo e dal terzo al computer del server. Questo è un collegamento virtuale tra due computer (o router) adiacenti. I dettagli fisici sono lasciati all'ultimo livello, ovvero al Livello Fisico che trasmette il messaggio sul cavo sotto forma di impulso elettrico. Questo è l'unico livello in cui avviene una trasmissione effettiva.

Per consentire queste comunicazioni, ad ogni singola macchina sulla rete è assegnato un indirizzo per ogni livello: un indirizzo MAC per ogni scheda di rete (Livello di Collegamento), un indirizzo IP (Livello di Rete) ed un numero di porta (Livello di Trasporto).

## 2.2 Commutazione di circuito

Si ha **commutazione di circuito** quando una frazione fissa della capacità trasmissiva è stabilmente allocata a ciascun canale. Ciascun utilizzatore ha a disposizione un canale trasmissivo dedicato, con la garanzia di poter utilizzare tutta la sua capacità.

L'allocazione della capacità può essere **statica** (configurata manualmente dal gestore della rete) o **dinamica**. In questo caso, per stabilire un canale di comunicazione, è necessaria una fase di *creazione della connessione*, nella quale sono impegnate le risorse nei collegamenti e nei nodi. Segue la fase di utilizzo del canale ed, alla fine, una fase di *abbattimento della connessione*, che libera le risorse.

Come esempio, si può considerare una conversazione telefonica che utilizza un circuito su una giunzione tra centrali: un utente compone un numero, e la rete telefonica lo elabora allocando una serie di circuiti fino al destinatario della chiamata (creazione della connessione); se il destinatario risponde, i due possono parlare utilizzando il canale; quando uno dei due abbassa il ricevitore, la rete libera le risorse impegnate (abbattimento della connessione).

La capacità del collegamento può essere suddivisa in circuiti con diversi meccanismi:

- a **divisione di tempo** o TDM;
- a **divisione di frequenza** o FDM;
- a **divisione di lunghezza d'onda** o WDM.

Ogni conversazione impegna delle risorse nella rete che la trasporta. Nel caso non siano disponibili le risorse per stabilire una comunicazione, la rete rifiuta la richiesta.

L'eventuale frazione di capacità trasmissiva non utilizzata (ad esempio, le pause di una conversazione telefonica) è persa, e questo è uno dei grossi limiti della commutazione di circuito. Tra i principali vantaggi, la garanzia che se la chiamata è stabilita essa godrà, per tutta la sua durata, delle prestazioni richieste (banda passante, ritardo costante).

## 2.3 Commutazione di pacchetto

La **commutazione di pacchetto** è una tecnica di accesso multiplo a ripartizione nel tempo, utilizzata per condividere un canale di comunicazione tra più stazioni in modo non deterministico. Tale tecnica non comporta l'attivazione di una linea di comunicazione dedicata fra un elaboratore ed un altro, ma consente lo svolgimento simultaneo di più comunicazioni fra computer, massimizzando così l'utilizzazione dei mezzi trasmissivi impiegati.

### 2.3.1 Pacchetti

In una rete a commutazione di pacchetto (**PBN**, *Packet Based Network*), l'informazione da trasmettere è suddivisa in pacchetti di dimensione abbastanza piccola; ad ognuno di essi è aggiunta un'intestazione che contiene tutta l'informazione necessaria affinché il pacchetto possa essere inoltrato alla sua destinazione finale e sulla sua posizione all'interno dell'informazione che è trasferita. I pacchetti sono inviati individualmente attraverso la rete e sono poi riassemblati nella loro forma originale quando arrivano sul computer di destinazione.

L'intera capacità trasmissiva disponibile è impegnata per la trasmissione di ciascun pacchetto. Se ci sono più pacchetti da trasmettere contemporaneamente, questi sono memorizzati in una coda, subendo un ritardo di accodamento e rischiando di essere scartati in caso di esaurimento della memoria disponibile per la coda.

### 2.3.2 Commutatori

Quando un nodo intermedio, detto **commutatore di pacchetto**, generalmente un router o uno switch, riceve un pacchetto, esso decide quale è il percorso migliore che il pacchetto può prendere per raggiungere la sua destinazione. Questa strada può cambiare da pacchetto a pacchetto, dipenden-



temente dalle condizioni della rete. Per cui, pacchetti appartenenti ad uno stesso messaggio possono intraprendere anche percorsi distinti.

### 2.3.3 Tempi di ritardo

Un pacchetto che attraversa una rete subisce un ritardo, legato in parte alle caratteristiche del percorso ed in parte allo stato di carico della rete.

Le componenti del ritardo sono:

- **ritardo di elaborazione:** è il tempo necessario a ciascun commutatore per processare il pacchetto. Se un commutatore agisce a diversi livelli, ciascun livello aggiungerà una sua componente di ritardo.
- **ritardo di trasmissione:** è il tempo necessario per trasmettere il pacchetto alla velocità della linea di trasmissione.
- **ritardo di propagazione:** è il tempo necessario al segnale fisico per propagarsi lungo una linea di trasmissione. Nei mezzi trasmissivi, come rame o fibra ottica, i segnali si propagano ad una frazione della velocità della luce nello specifico mezzo trasmissivo, indicativamente 200.000 km/s. Quindi, il ritardo di trasmissione è stimabile grossolanamente in  $5 \mu\text{s}/\text{km}$ .
- **ritardo di coda** (*queuing delay*): è dovuto al fatto che i pacchetti in uscita non sempre sono trasmessi immediatamente. Infatti, la linea in uscita può essere occupata da altri pacchetti in corso di trasmissione. In questo caso, il pacchetto è salvato in una memoria temporanea del commutatore, detta **coda**, per essere trasmesso appena possibile. Il tempo atteso dal pacchetto nella coda è, appunto, detto *ritardo di coda*, e non è possibile stabilire a priori quanto ritardo di coda un pacchetto subirà. Il ritardo di coda può essere, però, caratterizzato statisticamente, e di questo si occupa la teoria delle code.

Se sono applicate tecniche di qualità di servizio, alcune classi di pacchetti otterranno un trattamento preferenziale alla coda in uscita, per cui avranno diverse distribuzioni del ritardo di coda.

È anche possibile che i pacchetti inviati tra due stessi host in momenti diversi subiscano ritardi differenti. La variazione del ritardo subito dai pacchetti è detta *jitter*.

### 2.3.4 Perdita di pacchetti

La perdita di pacchetti può avvenire in diverse occasioni. Un pacchetto può perdersi in quanto è ricevuto con un errore e quindi scartato, oppure quando il buffer di un commutatore, sia nelle porte di ingresso che in quelle di uscita, risulta saturo e quindi si trova costretto a scartarlo. Per questa ragione, una rete a pacchetto non può, generalmente, garantire che tutti i pacchetti inviati arrivino a destinazione. TCP risolve questo problema reinviando i pacchetti persi (i pacchetti per i quali non riceve un ACK positivo).

## 2.4 Confronto tra commutazione di pacchetto e commutazione di circuito

In una **rete a commutazione di circuito** la capacità del canale trasmissivo è interamente dedicata ad una specifica comunicazione. In una rete di questo tipo si stabilisce, dunque, una connessione diretta tra sorgente e ricevente al momento di inizio della comunicazione. Invece, la *commutazione di pacchetto* si rivela molto più efficiente nonostante la maggior quantità di dati inviata, in quanto i canali fisici sono utilizzati solo per il tempo strettamente necessario. Inoltre, poiché ogni pacchetto porta con sé la sua identificazione, una rete può trasportare nello stesso tempo pacchetti provenienti da sorgenti differenti.

La commutazione di pacchetto permette quindi a più utenti di inviare informazioni attraverso la rete in modo efficiente e simultaneo, risparmiando tempo e costi mediante la condivisione di uno stesso canale trasmissivo.

La commutazione di pacchetto è uno dei possibili metodi di *multiplazione*, ovvero una tecnica per suddividere la capacità trasmissiva di un canale tra diversi utilizzatori.

## 2.5 Multiplazione

Nelle telecomunicazioni, in elettronica e reti di computer, la **multiplazione**, o **multiplexing**, è il meccanismo per cui la capacità disponibile di un collegamento è condivisa tra diversi canali trasmissivi, combinando più segnali analogici o flussi di dati digitali in un solo segnale trasmesso su un singolo collegamento fisico.

La multiplazione permette di risparmiare nella comunicazione dei dati ed, in particolare, di ridurre il numero di linee di segnale ed il numero di componenti. Ad esempio, in elettronica, il multiplexing permette a diversi segnali analogici di essere elaborati da un unico convertitore analogico-digitale

(ADC) e in telecomunicazioni diverse chiamate possono essere trasmesse usando un solo cavo.

Il componente che opera il multiplexing è chiamato **multiplexer**, mentre quello che opera il processo inverso è chiamato **demultiplexer**.

## 2.6 Il Protocol Data Unit

Il **Protocol Data Unit**, abbreviato con l' acronimo PDU, è una locuzione tipica delle telecomunicazioni, utilizzata per indicare:

- un' unità d' informazione scambiata tra due network peer entities, ossia: informazioni di controllo, campi di indirizzamento, ecc. sia header che trailer;
- un' unità d' informazione inviata/ricevuta tra peer entities protocollari che in più può contenere *payload* (dati), in un contesto di schema protocollare a strati.

Nello schema protocollare a strati, si intendono dati le SDU (*Service Data Unit*) scambiate con il livello protocollare gerarchicamente superiore.

Nelle reti di calcolatori, i dati da trasmettere sono suddivisi in segmenti chiamati pacchetti, costituiti da dati di controllo (intestazione e *checksum*) e dai dati veri e propri (carico pagante o payload). Le intestazioni sono aggiunte da ciascun protocollo di rete utilizzato.

L' operazione di inserire un payload tra dati di controllo è detta **imbustamento**, e può essere ripetuta più volte, inserendo un pacchetto di livello  $N$  tra intestazioni di livello  $N - 1$ . Il risultato è un pacchetto di livello  $N - 1$ , che diventa il carico pagante per il protocollo di livello  $N - 2$ , e così via.

## 2.7 Servizio con connessione e servizio senza connessione

Un'analisi comparata tra i due tipi di protocollo è fornita in tabella 2.1.

Per tutti i livelli superiori al Livello Fisico sono definite due modalità operative: una **modalità connessa** (**CONS: Connection Oriented Network Service**) ed una **modalità non connessa** (**CLNS: ConnectionLess Network Service**). Un dato livello può fornire al livello superiore servizi di tipo connesso, non-connesso o entrambi. Questa è una scelta progettuale che varia per ogni livello, da architettura ad architettura. Lo standard originale ISO 7498 prevedeva solo la modalità connessa ma, vista l' importanza della

Caratteristiche	Connection-Oriented	Connctionless
Initial setup	Richiesto	Impossibile
Indirizzo di destinazione	Durante il setup	In ogni pacchetto
Ordine dei pacchetti	Garantito	Non garantito
Controllo degli errori	Si	No
Controllo di flusso	Si	No
Negoziazione di opzioni	Si	No
Identificatore di connessione	Si	No

Tabella 2.1: Analisi comparata tra servizio connesso e servizio non connesso

modalità non connessa, è stata aggiunta in seguito come emendamento allo standard stesso (ISO 7498/Addendum 1).

In un servizio non connesso la spedizione di un pacchetto è simile alla spedizione di una lettera ordinaria con il sistema postale. Tutto avviene in una sola fase, lasciando cadere la lettera nella buca delle lettere. La lettera deve contenere sulla busta l'indirizzo completo del destinatario. Non vi è alcun riscontro diretto che la lettera giunga a destinazione correttamente.

In un servizio connesso lo scambio di dati tramite pacchetti ricorda le frasi scambiate tra due interlocutori al telefono. Vi sono tre momenti principali:

1. creazione della connessione (il comporre il numero telefonico e il «pronto» alla risposta);
2. trasferimento dei dati (la conversazione telefonica);
3. chiusura della connessione (i saluti finali ed il posare il microtelefono).

### 2.7.1 La modalità connessa

Nella modalità connessa lo scambio di dati avviene tramite le tre fasi viste prima. Durante la fase di creazione della connessione (*initial setup*) due peer-entities concordano che trasferiranno delle Protocol Data Unit. Solo durante tale fase devono essere specificati gli indirizzi completi del mittente e del destinatario: successivamente, le entità coinvolte specificheranno soltanto l'identificativo della connessione stabilito durante la prima fase.

Un servizio connesso fornisce una modalità di trasferimento delle Protocol Data Unit affidabile e sequenziale. Per tutta la durata della connessione, le Protocol Data Unit inviate sono ricevute correttamente nello stesso ordine. Se qualcosa non funziona correttamente, la connessione può essere riavviata (reset) o terminata (released). Per verificare che tutte le Protocol Data Unit inviate giungano a destinazione correttamente, un servizio connesso utilizza

degli schemi di numerazione dei pacchetti e di verifica dell'avvenuta corretta ricezione (ACK). Quindi un protocollo connesso è, in generale, in grado non solo di rilevare la presenza di errori, ma anche di correggerli tramite ritrasmissioni.

### 2.7.2 La modalità non connessa

Con una modalità non connessa la comunicazione ha luogo in una fase singola: il pacchetto è inviato e deve contenere l'indirizzo completo del destinatario. Non essendo i pacchetti organizzati in una connessione, un pacchetto non può fare riferimento ad altri pacchetti trasmessi precedentemente o in seguito. Quindi, un protocollo non connesso può solo rilevare la presenza di errori (scartando quindi le Protocol Data Unit errate), ma non correggerli in quanto non si possono realizzare meccanismi di ritrasmissione (in un pacchetto non è possibile fare riferimento ad altri pacchetti).

Un protocollo non connesso è, in generale, più efficiente di un protocollo connesso, specialmente se bisogna trasferire piccole quantità di dati: in quest'ultimo caso, infatti, l'overhead della creazione e distruzione della connessione è rilevante.

Un protocollo non connesso (detto anche *datagram*), non potendo garantire l'affidabilità del trasferimento dati, necessita che almeno un protocollo di livello superiore sia di tipo connesso.

### 2.7.3 Applicazioni connesse e non

Anche le applicazioni possono operare in modo connesso oppure no. Un'applicazione, infatti, può non essere interessata a sapere se i propri dati sono giunti a destinazione o può implementare suoi schemi proprietari di controllo end-to-end. Si pensi, ad esempio, ad una applicazione che fornisca l'ora esatta a tutti i calcolatori della rete: essa può addirittura ignorare quali siano i calcolatori a cui fornisce il servizio e, quindi, operare in modo *connectionless*. Le applicazioni connesse sono molte, ad esempio quelle di trasferimento file e di posta elettronica.

Chiaramente, sia applicazioni connesse che non connesse devono poter operare su di una rete che ai livelli inferiori ha protocolli di tipo connesso oppure no.

Le possibili combinazioni tra livelli connessi e non nel modello OSI sono riportate in figura 2.1.

Il livello 1 non può essere considerato né connesso né non connesso; i livelli 2, 3 e 4 possono operare in entrambe le modalità ed in tutte le possibili

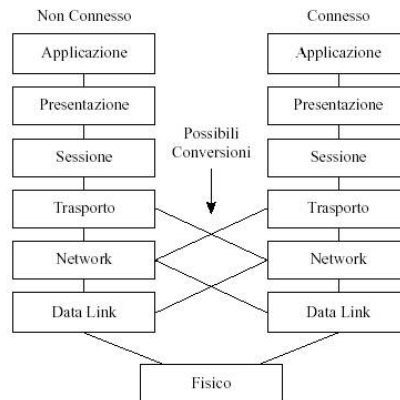


Figura 2.1: Combinazione tra livelli connessi e non

combinazioni. I livelli 5, 6 e 7 devono comportarsi globalmente in modo connesso o non connesso.

A livello 2 l'operatività è normalmente di tipo non connesso quando si opera su reti locali, dove il mezzo trasmissivo è intrinsecamente affidabile, mentre è di tipo connesso su reti geografiche che sono intrinsecamente caratterizzate da un più alto tasso di errore.

A livello 3 vi è sempre stata controversia tra gli informatici che vogliono un livello 3 non connesso e i telecomunicazionisti che lo vogliono connesso. Occorre anche evidenziare come negli ultimi anni il livello 3 sia sempre più stato considerato di competenza degli informatici e, quindi, si utilizzino sempre più protocolli non connessi.

Il livello 4 è, in pratica, sempre connesso: infatti, anche quando opera su un livello 3 connesso, l'affidabilità che tale livello 3 fornisce non è ritenuta soddisfacente.

# Capitolo 3

## Livello Fisico

### 3.1 La trasmissione

S' intende per **trasmissione** l' invio di informazioni su un canale fisico. Esempi di trasmissione sono: la *trasmissione seriale*, la *trasmissione parallela*, la *trasmissione sincrona* e la *trasmissione asincrona*.

La **trasmissione seriale** è una modalità di comunicazione tra dispositivi digitali nella quale le informazioni sono comunicate una di seguito all' altra e giungono sequenzialmente al ricevente nello stesso ordine in cui le ha trasmesse il mittente. Richiede un minor numero di fili con conseguente riduzione dei costi ed è più tollerante rispetto alle interferenze e agli errori di trasmissione.

La **trasmissione parallela** è una tipologia di collegamento utilizzata per trasmettere informazioni. È utilizzata prevalentemente per trasmettere informazioni digitali anche se esistono dei cavi capaci di trasmettere parallelamente informazioni di carattere analogico. Nella trasmissione parallela sono utilizzati più conduttori per trasmettere simultaneamente informazioni. Quindi, un cavo che effettua una trasmissione parallela a  $n$  bit è formato da almeno  $n$  conduttori separati. Nella realtà, il cavo sarà dotato quasi sicuramente di un cavo aggiuntivo per la massa e di altri cavi di controllo (come quello di clock).

Rispetto alla trasmissione seriale, la trasmissione parallela risulta avere prestazioni più alte a parità di frequenza, ma è anche più costosa. Infatti, normalmente per comunicazioni a lunga distanza si preferisce utilizzare delle trasmissioni seriali, dato che i cavi necessari alla trasmissione parallela renderebbero non economicamente conveniente il progetto.

Nella **trasmissione sincrona** il trasmettitore ed il ricevitore hanno una base di tempi comune che scandisce l' invio dei bit. Nella **trasmissione asincrona** non c' è una base di tempi comune tra trasmettitore e ricevitore,

e ciascuna trasmissione deve essere identificata da un *bit di Start* e da un *bit di Stop*. Il bit di Start indica che il “dato” sta per iniziare, ed il bit di stop indica quando che il “dato” è terminato.

### 3.1.1 I nemici della trasmissione

Ogni segnale durante il suo viaggio verso l’utente ricevente subisce l’aggressione di tre terribili nemici: l’*attenuazione*, la *distorsione* e la *interferenza*. Per **attenuazione** s’ intende la diminuzione in ampiezza che subisce un segnale che passa per un circuito, in dipendenza delle sue caratteristiche e di quelle del circuito che attraversa. Per esempio, si parla di attenuazione del segnale laser in una fibra ottica, di attenuazione nelle onde sismiche propagantesi nella terra o di attenuazione dell’ energia elettromagnetica nella trasmissione in un ponte radio, anche se in questo caso il motivo della perdita di potenza non si deve a perdite in dispositivi elettronici. La **distorsione** è il cambiamento di una forma d’ onda a causa del passaggio attraverso un componente fisico non lineare. Infine, il fenomeno dell’ **interferenza** è dovuto alla sovrapposizione, in un punto dello spazio, di due o più onde. Quello che si osserva è che l’ intensità dell’ onda risultante in quel punto può essere diversa rispetto alla somma delle intensità associate ad ogni singola onda di partenza; in particolare, essa può variare tra un minimo, in corrispondenza del quale non si osserva alcun fenomeno ondulatorio, ed un massimo coincidente con la somma delle intensità. In generale, si dice che l’ interferenza è **costruttiva** quando l’ intensità risultante è maggiore rispetto a quella di ogni singola intensità originaria, e **distruttiva** in caso contrario.

Questa aggressione cresce con la strada percorsa dal segnale e lo rende sempre più difficilmente riconoscibile. Prima che diventi assolutamente irriconoscibile, lo si estrae dal suo supporto trasmissivo, lo si rielabora per riportarlo al livello iniziale (**rigenerazione**), lo si reimmette nel supporto trasmissivo per fargli riprendere la strada interrotta verso l’ utente finale o verso un altro punto di rigenerazione.

L’ effetto dell’ attenuazione è un abbassamento del livello del segnale (è come se il segnale si presentasse un po’ sfocato); l’ effetto della distorsione è quello di un’ immagine un po’ mossa); l’ effetto dell’ interferenza è come se si sovrapponesse l’ alone di un’ altra immagine.

Non potendoli eliminare, l’ unica soluzione è quella di scegliere, per ogni segnale, il supporto più idoneo in base alla gamma di frequenza occupata dal segnale e quello di costruire questo supporto con le dimensioni, gli accorgimenti ed i componenti più opportuni.



## 3.2 Trasmissione su cavo

Il **cavo** è un componente elettrico che consiste in un fascio di più fili conduttori o fibre ottiche rivestito da uno strato di materiale isolante. La funzione del cavo è la trasmissione di corrente elettrica o luce per il trasporto di energia elettrica (solo nel caso di cavi dotati di fili elettrici) o per lo scambio di informazioni.

In quasi tutti i cavi elettrici si impiegano metalli con una resistività elettrica molto bassa (per esempio rame o alluminio) per i fili conduttori. Questi ultimi hanno uno spessore variabile a seconda della portata di corrente ed a seconda che sia richiesta una maggiore resistenza alle sollecitazioni meccaniche, o una maggiore flessibilità; possono, inoltre, essere intrecciati a spirale, appiattiti o sagomati.

I cavi sono manufatti sviluppati essenzialmente in lunghezza, usati per collegare due punti al fine di trasferire dall'uno all'altro energia elettrica o informazioni mediante campi elettromagnetici. Nel primo caso si parla di cavi energia, nel secondo di cavi di telecomunicazione.

Esiste poi una categoria intermedia di cavi, detti **cavi di comando e segnalamento**, usati per trasmettere informazioni insieme ad energia elettrica atta ad attivare i dispositivi cui sono collegati.

Infine, quando i cavi di cui sopra sono oggetto di specifiche soluzioni al fine di conferire loro caratteristiche particolari (quali, ad esempio, l'operabilità anche in situazioni ambientali particolarmente aggressive) sono detti **cavi speciali**.

In base al sistema di posa, tutti i cavi sono classificati in:

- **cavi aerei** se sono posati mediante appositi sostegni a palificazioni in legno o metallici;
- **cavi terrestri** se sono interrati direttamente o entro tubazioni o appositi cunicoli ad una profondità dal suolo tale da tenerli sufficientemente protetti dalle azioni che si possono compiere inavvertitamente in superficie;
- **cavi sottomarini** se sono distesi sul fondo del mare, dei laghi o dei corsi d'acqua dopo aver eventualmente predisposto un corridoio privo di particolari asperità. Dove il cavo potrebbe essere soggetto a sollecitazioni particolari si procede a ricoprirli con opportuni strati protettivi.

Tutti i cavi di cui sopra si dicono poi **armati** se, oltre all'involucro protettivo esterno di cui sono normalmente dotati, sono ulteriormente protetti applican-

do su di essi tubi di piombo o di alluminio o nastri/fili di ferro/acciaio avvolti ad elica ed opportunamente trattati per proteggerli dalla corrosione.

### 3.2.1 Il doppino di rame

Tipicamente, il **doppino di rame** è costituito da una coppia di conduttori ritorti (*twisted pair*) mediante un processo di binatura. La binatura del doppino di rame ha lo scopo di fare in modo che i campi elettromagnetici esterni agiscano mediamente in egual modo sui due conduttori. Impiegando, poi, una tecnica di trasmissione differenziale, è possibile eliminare ulteriori disturbi.

Il doppino può essere singolo (una sola coppia) oppure in una treccia di una serie più o meno numerosa di coppie. In questo caso, ogni coppia presenta una binatura diversa, per ridurre il più possibile il fenomeno di diafonia tra le varie coppie di doppino tra loro contigue.

I doppini di rame sono utilizzati, anche, in un intreccio di quattro coppie, per trasmettere (attraverso il protocollo ethernet e gli spinotti RJ-45) dati in una rete locale. Questi doppini di rame possono essere schermati per ridurre l' interferenza elettromagnetica.

#### Unshielded Twisted Pair

L' **Unshielded Twisted Pair** (UTP) identifica un cavo non schermato utilizzato comunemente per il collegamento nelle reti ethernet. È composto da otto fili di rame intrecciati a coppie. Ciascuna coppia è intrecciata con un passo diverso ed è intrecciata con le altre. L' intreccio dei fili ha lo scopo di ridurre le interferenze ed i disturbi.

La lunghezza massima di un cavo UTP nello standard ethernet è di 100 m.

Un cavo UTP termina con dei connettori di tipo RJ-45 che si innestano direttamente nell' interfaccia del dispositivo (scheda di rete, Hub, Switch, Router, ecc).

La sigla *RJ-45* indica una interfaccia fisica usata per l' attestazione di cavi elettrici a coppie di conduttori incrociati. Si tratta di un connettore 8P8C, ovvero a otto posizioni e otto contatti (pin), che può essere usato per varie applicazioni. A differenza delle apparecchiature telefoniche, che utilizzano l' RJ-11, il connettore RJ-45 è utilizzato in particolare per il cablaggio delle reti locali secondo gli standard Ethernet / IEEE 802.3 10Base-T, 100Base-TX e 1000Base-T, anche se con una configurazione dei contatti (pinout) leggermente diversa da quella originale.

Se si devono collegare due dispositivi simili (ad esempio PC-PC o SWITCH-HUB) si utilizza un cavo di tipo “cross” mentre se si devono connettere dispositivi diversi (ad esempio PC-SWITCH) uno “diritto”. I cavi “diritti” presentano gli otto fili nello stesso ordine in entrambi i due connettori, mentre quelli “cross” presentano una sequenza diversa, e sono usati per collegare tra loro due host ethernet.

### Shielded Twister Pair

Il **Shielded Twisted Pair** (STP) identifica un cavo schermato utilizzato comunemente per il collegamento nelle reti ethernet. È composto da otto fili di rame intrecciati a coppie; inoltre, ogni coppia è intrecciata con le altre. L’ intreccio dei fili ha lo scopo di ridurre le interferenze ed i disturbi.

La distanza massima di un cavo STP è di 100 m.

Ogni coppia è avvolta da fogli di materiale conduttivo che fungono da schermo per le onde elettromagnetiche; esternamente alle quattro coppie c’ è un ulteriore foglio di materiale conduttivo.

I cavi STP seguono le specifiche standardizzate in TIA/EIA che li dividono in varie categorie in base ad esempio al numero di intrecci ed alle capacità di trasportare segnali.

Un cavo STP termina con dei connettori di tipo RJ-45 (anch’ essi schermati) che s’ innestano direttamente nell’ interfaccia del dispositivo (scheda di rete, Hub, Switch, Router, ecc). Se si devono collegare due dispositivi simili (ad esempio PC-PC o SWITCH-HUB), si utilizza un cavo di tipo “cross”, mentre se si devono connettere dispositivi diversi (ad esempio PC-SWITCH), si usa un cavo di tipo “diritto”. I cavi “diritti” presentano gli otto fili nello stesso ordine in entrambi i due connettori, mentre quelli “cross” presentano una sequenza diversa.

### 3.2.2 La fibra ottica

Le **fibre ottiche** sono filamenti di materiali vetrosi o polimerici, realizzati in modo da poter condurre la luce. Sono flessibili, immuni ai disturbi elettrici ed alle condizioni atmosferiche più estreme, e poco sensibili a variazioni di temperatura. Hanno un diametro di  $125\ \mu\text{m}$  e pesano molto poco: una singola fibra pesa, infatti, circa 60 g/km, compresa la guaina che la ricopre.

Le fibre ottiche sono classificate come guide d’ onda dielettriche. Esse, in altre parole, permettono di convogliare al loro interno un campo elettromagnetico di frequenza sufficientemente alta (in genere in prossimità dell’ infrarosso) con perdite estremamente limitate. Sono comunemente impiegate

nelle telecomunicazioni anche su grandi distanze e nella fornitura di accessi di rete a larga banda (dai 10 Mbit/s al Tbit/s usando le tecnologie WDM).

Ogni singola fibra ottica è composta da due strati concentrici di materiale trasparente estremamente puro: un nucleo cilindrico centrale, o *core*, ed un mantello, o *cladding*, attorno ad esso. Il core presenta un diametro molto piccolo di circa 10  $\mu\text{m}$  per le Monomodali e 50  $\mu\text{m}$  per le Multimodali, mentre il cladding ha un diametro di circa 125  $\mu\text{m}$ . I due strati sono realizzati con materiali con indice di rifrazione leggermente diverso: il cladding deve avere un indice di rifrazione minore rispetto al core. Come ulteriore caratteristica, il mantello (buffer) deve avere uno spessore maggiore della lunghezza di smorzamento dell'onda evanescente, caratteristica della luce trasmessa in modo da catturare la luce che non è riflessa nel core.

La fibra ottica funziona come una specie di specchio tubolare. La luce che entra nel core ad un certo angolo (angolo limite) si propaga mediante una serie di riflessioni alla superficie di separazione fra i due materiali del core e del cladding.

All'esterno della fibra c'è una guaina protettiva polimerica, detta *jacket*, che serve a dare resistenza alla corrosione ed evitare il contatto fra la fibra e l'ambiente esterno.

Il core e il cladding della fibra ottica possono essere realizzati in silice oppure in polimeri plastici.

Nelle fibre ottiche avviene un fenomeno di riflessione totale interna, per cui la discontinuità dell'indice di rifrazione tra i materiali del nucleo e del mantello intrappola la radiazione luminosa finché questa mantiene un angolo abbastanza radente, in pratica finché la fibra non compie curve troppo brusche.

All'interno di una fibra ottica il segnale può propagarsi in modo rettilineo oppure essere riflesso un numero molto elevato di volte. Il modo di propagazione rettilineo si dice di **ordine zero**. Le *fibre monomodali* consentono la propagazione di luce secondo in un solo modo, e hanno un diametro del core compreso tra 8  $\mu\text{m}$  e 10  $\mu\text{m}$ , quelle *multimodali* consentono la propagazione della luce in più modi, e hanno un diametro del core di 50  $\mu\text{m}$  o 62,5  $\mu\text{m}$ . Il cladding ha tipicamente un diametro di 125  $\mu\text{m}$ . Le fibre multimodali permettono l'uso di dispositivi più economici, ma subiscono il fenomeno della dispersione intermodale, per cui i diversi modi di propagazione impiegano velocità leggermente diverse: questo limita la distanza massima a cui il segnale può essere ricevuto correttamente. Le fibre monomodali, diversamente, hanno un prezzo molto più elevato rispetto alle multimodali, ma riescono a coprire grandi distanze ed a raggiungere velocità nettamente superiori. Le fibre multimodali possono essere divise ulteriormente in *fibre step index* e *graded index*:

- nelle fibre step index l'indice di rifrazione è costante lungo tutta la sezione del core e cambia improvvisamente quando si incontra il cladding;
- nelle fibre graded index l'indice di rifrazione cambia gradualmente dal core al cladding, permettendo l'uso di luce multicromatica.

Se negli anni settanta le fibre ottiche erano usate come oggetto decorativo per la produzione di lampade, oggi sono un componente essenziale nell'industria delle telecomunicazioni, ancora in corso di evoluzione tecnologica. I principali vantaggi delle fibre rispetto ai cavi in rame nelle telecomunicazioni sono:

- bassa attenuazione, che rende possibile la trasmissione su lunga distanza senza ripetitori;
- grande capacità di trasporto di informazioni;
- immunità da interferenze elettromagnetiche, inclusi gli impulsi elettromagnetici nucleari (ma possono essere danneggiate da radiazioni alfa e beta);
- alta resistenza elettrica, quindi è possibile usare fibre vicino ad equipaggiamenti ad alto potenziale, o tra siti a potenziale diverso;
- peso e ingombro modesto;
- bassa potenza contenuta nei segnali;
- assenza di diafonia;
- ottima resistenza alle condizioni climatiche avverse;
- bassi valori di BER (in un sistema di trasmissione digitale, il *Bit Error Ratio* (BER) è il rapporto tra i bit non ricevuti correttamente ed i bit trasmessi. Il BER evidenzia quanto della originaria trasmissione è perso o giunge distorto all'apparecchio ricevente a causa, ad esempio, di disturbi nel canale di trasmissione, di problemi degli impianti, di malformazioni originarie del flusso dati).

Un cavo di fibra ottica, in quanto contiene più fibre ottiche, è solitamente molto più piccolo e leggero di un filo o cavo coassiale con simili capacità di canale. È più facile da maneggiare e da installare. Il cavo in fibra ottica è ideale per le comunicazioni sicure, in quanto è molto difficile da intercettare e altrettanto facile da monitorare.

### 3.3 Il cablaggio

Il **cablaggio** è costituito dagli impianti fisici (cavi, connettori, permutatori, infrastrutture di supporto) che permettono di realizzare una rete di calcolatori, tipicamente nell'ambito di un edificio o un gruppo di edifici.

Le caratteristiche elettriche e le lunghezze dei cavi e dei connettori impiegati influenzano le tipologie di reti locali realizzabili.

Esempi di cablaggio sono le prime reti ethernet (dette *10Base5, thick cable*), costituite da grossi cavi coassiali in rame, a cui i calcolatori dovevano essere collegati perforando la guaina esterna fino a raggiungere il connettore interno. Questi cablaggi avevano una topologia a bus.

#### 3.3.1 Cablaggio strutturato

Negli edifici moderni destinati ad uffici sono realizzati impianti di **cablaggio strutturato**, destinati a supportare la realizzazione di tipi diversi di reti locali, inclusa ad esempio la rete telefonica. Gli impianti sono basati su cavi di categoria 5 o superiore e connettori RJ-45. I cavi hanno una lunghezza massima di 90 m, a cui vanno aggiunti 10 m per i cavi di permuta. Questo vincolo è dettato dalle caratteristiche della rete Ethernet.

Per ogni postazione da servire, sono posati uno o più cavi in apposite canalizzazioni nelle pareti, nei controsoffitti o nei pavimenti dell'edificio, fino a raggiungere un armadio di distribuzione di piano (nel gergo del cablaggio strutturato, *Floor Distributor*, FD), solitamente si tratta di un rack standard da 19 pollici, che può ospitare sia permutatori che apparati attivi.

Questi cavi sono attestati da una parte in un pannello di permutazione nell'armadio, dall'altra in una placca a muro o a pavimento in prossimità della postazione utente. Collegando un cavo di permuta dal calcolatore alla presa a muro, e un altro dal permutatore ad un apparato di rete (come un hub o uno switch), si crea un collegamento elettrico che permette di collegare il calcolatore alla rete.

Nel caso le dimensioni dell'edificio non permettano di servire tutte le utenze con un solo FD, i vari FD sono collegati ad un armadio di edificio (*Building Distributor*, BD), tramite cavi in rame e/o in fibra ottica, anche questi attestati in permutatori. Allo stesso modo, i diversi edifici di un campus sono collegati ad un armadio di permutazione di campus (*Campus Distributor*, CD).

I locali che ospitano gli armadi di distribuzione dovrebbero avere caratteristiche adeguate per alimentazione elettrica (meglio se protetta da un gruppo di continuità), condizionamento, controllo d'accesso (sono luoghi privilegiati per intrusioni o per provocare malfunzionamenti della rete).

## 3.4 RS-232

Lo **standard EIA RS-232** nacque nei primi anni '60 per opera della “Electronic Industries Association” ed era orientato alla comunicazione tra i main-frame e i terminali (*Data Terminal Equipment*) attraverso la linea telefonica, utilizzando un modem (*Data Communication Equipment*). Oggi la porta seriale EIA RS-232 è presente in quasi tutti i PC desktop, anche se è stata soppiantata dall'interfaccia USB (o da PS/2) in quasi tutti gli utilizzi. La gran parte dei PC portatili, invece, non è ormai più dotata di questa interfaccia.

Tra gli utilizzi della porta seriale, si possono citare:

- connessione di terminali ad un calcolatore (tradizionalmente un main-frame, ma anche un PC);
- connessione di periferiche:
  - la porta seriale è stata usata per collegare i mouse ai primi PC;
  - stampante (soppiantato dalla porta parallela, e poi da USB e dalle stampanti di rete);
  - dispositivi specializzati, come ad esempio lettori di codici a barre e di tessere magnetiche (soppiantato da USB).
- connessione a dispositivi embedded per scopi di configurazione e monitoraggio. In questo utilizzo, RS-232 è ancora ampiamente usato, anche se spesso è necessario dotarsi di un adattatore seriale/USB per utilizzare come terminale un computer privo di porta seriale.

L'interfaccia EIA RS-232 ridotta (ovvero solo asincrona) utilizza un protocollo di trasmissione seriale di tipo asincrono. “Seriale” significa che i bit che costituiscono l'informazione sono trasmessi uno alla volta su di un solo “filo”; “asincrono” significa, in questo contesto, che i dati sono trasmessi, byte per byte, in modo anche non consecutivo e senza l'aggiunta di un segnale di clock, cioè di un segnale comune che permette di sincronizzare la trasmissione con la ricezione. Ovviamente, sia il trasmettitore che il ricevitore devono comunque essere dotati di un clock locale per poter interpretare i dati. La sincronizzazione dei due clock è necessaria ed è fatta in corrispondenza della prima transizione sulla linea dei dati.

### 3.4.1 Le unità di misura

Le unità di misura della velocità di trasmissione sono essenzialmente due: il **baud** ed il **bit per secondo** (bps), spesso trattate erroneamente come

sinonimi. Il baud rate indica il numero di transizioni al secondo che avvengono sulla linea; il bps indica, come dice il nome, quanti bit al secondo sono trasmessi lungo la linea.

Nel caso di trasmissione binaria (cioè, è presente un livello alto ed uno basso) le due cose ovviamente coincidono numericamente, da cui la parziale equivalenza dei due termini. Nel caso di trasmissioni a più livelli, invece, è possibile trasmettere con una sola transizione più bit.

Nel caso dello standard EIA RS-232 i livelli utilizzati sono due. Quindi, il baud rate coincide numericamente con il bps.

### 3.4.2 Successori di RS-232

I successori di RS-232 sono: l' *Universal Serial Bus* (USB), il *Firewire* ed il *Serial ATA*. L' **Universal Serial Bus** (USB) è uno standard di comunicazione seriale che consente di collegare diverse periferiche ad un computer. È stato progettato per consentire a più periferiche di essere connesse usando una sola interfaccia standardizzata ed un solo tipo di connettore, e per migliorare la funzionalità *plug-and-play* (termine che indica tecnologie che possono essere messe in uso all' interno di un sistema hardware e software, senza una specifica procedura di installazione), consentendo di collegare/scollegare i dispositivi senza dover riavviare il computer (*hot swap*). Il sistema USB è asimmetrico, consiste in un singolo gestore e molte periferiche collegate da una struttura ad albero attraverso dei dispositivi chiamati *hub* (concentratori). Supporta fino ad un massimo di 127 periferiche per gestore: nel computo vanno però inclusi anche gli hub ed il gestore stesso, quindi, in realtà, il numero totale di dispositivi collegabili è sensibilmente inferiore. Lo standard prevede che il connettore porti anche un cavo (chiamato VBUS) per alimentare le periferiche a basso consumo. Le periferiche che hanno richieste energetiche elevate sono alimentate a parte. USB può collegare periferiche quali mouse, tastiere, memoria di massa a stato solido ed a disco rigido, scanner d' immagini, macchine fotografiche digitali, stampanti, casse acustiche, microfoni e altro ancora. Per i componenti multimediali ormai lo standard USB è il metodo di collegamento più utilizzato, mentre nelle stampanti sopravvivono ancora molti modelli dotati anche di porta parallela per questioni di compatibilità. All' interno del computer, l' USB non ha rimpiazzato gli standard ATA o SCSI per via della sua lentezza. L' USB è molto usato negli hard disk esterni dove si preferisce privilegiare la praticità di poter collegare e scollegare a caldo il componente rispetto alla velocità di una connessione tipo ATA. Il **FireWire** (nome con il quale è noto lo standard IEEE 1394) è un' interfaccia standard per un bus seriale che supporta due diverse modalità di trasferimento dati: **asincrona** e **isocrona**. La modalità asincrona avviene



quando il dato spedito è ricevuto dall' altra parte del cavo. Nel caso in cui la linea non fosse libera, è nuovamente inviato. La modalità isocrona prevede un invio di dati attraverso il flusso continuo in tempo reale. In questa modalità si possono acquisire dati dagli apparecchi digitali come videocamere e macchine fotografiche. La connessione FireWire è comunemente usata per collegare dispositivi di archiviazione o dispositivi di acquisizione video. È utilizzato, anche, in apparecchiature di acquisizione audio e video professionali per via della ampiezza di banda della connessione, della sua predisposizione a trattare flussi multimediali, della capacità di sopportare potenze maggiori e della possibilità di stabilire una connessione tra dispositivi senza il tramite di un computer. L' interfaccia FireWire è tecnicamente superiore all' interfaccia USB, ma questa è molto più diffusa per via dei brevetti. La FireWire supporta fino a 63 periferiche organizzate in una rete non ciclica. Permette una comunicazione peer-to-peer tra i dispositivi. Quindi, i vari dispositivi possono comunicare tra loro senza dover utilizzare il computer come arbitro. Supporta il collegamento a caldo e la presenza di più host tramite una gestione degli IP software. Quindi, una connessione FireWire può essere utilizzata per creare una rete locale tra due computer quattro volte più veloce di una normale rete Ethernet a 100 Mbit/s. Il cavo FireWire supporta fino a 45 watt ed è, quindi, in grado di alimentare la maggior parte dei dispositivi portatili. Il **Serial ATA** (SATA - *Serial Advanced Technology Attachment*) è un' interfaccia per computer generalmente utilizzata per connettere un hard disk oppure un masterizzatore dvd nell' ambito di un computer. È l' evoluzione dell' ATA (più conosciuto come IDE). I vantaggi principali sono tre:

- la velocità;
- la gestione dei cavi;
- la funzione di hot swap.

### 3.4.3 Limiti di RS-232 e dei suoi successori

L' RS-232, l' USB ed il FireWire sono protocolli punto-punto, cioè collegano coppie di macchine. Per collegare tra loro  $N$  macchine occorrerebbero  $N(N - 1)$  cavi e connettori. Perciò, occorre introdurre un canale condiviso e delle regole per identificare univocamente le macchine e decidere il turno di utilizzo.



# Capitolo 4

## Livello DataLink

Il **Livello DataLink** è il secondo livello dei protocolli del modello ISO/OSI per l'interconnessione di sistemi aperti. Questo livello riceve pacchetti dal Livello Rete e forma i *frame* che sono passati al successivo livello. Il Livello DataLink deve svolgere diverse funzioni specifiche:

- raggruppa i bit del Livello Fisico in pacchetti chiamati frame (operazione di framing);
- controllo e gestione degli errori di trasmissione;
- regolare il flusso della trasmissione fra sorgente e destinatario (controllo di flusso).

### 4.1 Il framing

Al fine di fornire servizi al Livello Rete, il Livello DataLink deve usufruire dei servizi fornitigli dal Livello Fisico. L'approccio consueto del Livello DataLink è quello di dividere il flusso dei bit in *pacchetti* e calcolarne la *checksum*. Quando un pacchetto arriva a destinazione, la checksum è ricalcolata. Se il nuovo risultato è diverso da quello contenuto nel pacchetto, il Livello DataLink riconosce che deve essere stato commesso un errore e prende adeguati provvedimenti (ad esempio, scarta il pacchetto e spedisce in risposta un messaggio di errore).

Sono utilizzati vari metodi per la suddivisione dei bit in pacchetti:

- conteggio dei caratteri;
- caratteri di inizio e fine;
- indicatori (flag) di inizio e fine;

Il metodo del conteggio di caratteri (specificando in un campo dell' intestazione il numero di caratteri del frame) è raramente utilizzabile perché, se il campo che contiene il conteggio si rovina durante la trasmissione, non si può più individuare dove comincia il frame successivo; sono quindi utilizzate le altre tecniche.

Nella trasmissione orientata al byte, il frame (che mantiene la suddivisione in byte) è preceduto dalla sequenza di caratteri ASCII DLE STX (DataLink Escape Start of TeXt) e finisce con la sequenza DLE ETX (DataLink Escape End of TeXt). Se un frame si rovina e la destinazione perde la sincronizzazione basta trovare il successivo DLE STX o DLE ETX. Il carattere DLE, però, può comparire casualmente dentro al frame quando sono trasmessi dati binari (ad esempio, programmi oggetto o numeri in virgola mobile); perché questi caratteri non interferiscano è aggiunto un ulteriore DLE (che è rimosso a destinazione prima di passare al frame al Livello Rete) in modo che solo i DLE singoli siano interpretati come delimitatori. Questa tecnica si chiama *character stuffing*. Nella trasmissione orientata al bit, ogni frame (che può contenere un numero qualsiasi di bit) inizia e finisce con la sequenza 01111110 chiamata flag: questa sequenza può comparire casualmente nei dati, perciò in trasmissione dopo cinque 1 consecutivi è sempre inserito uno 0 nel flusso di bit, indipendentemente dal fatto che il bit successivo sia 1 o 0, mentre in ricezione bisogna provvedere ad eliminare i bit inseriti, rimuovendo sempre uno 0 dopo cinque. Questa tecnica è chiamata *bit stuffing*.

## 4.2 Controllo degli errori

La stazione mittente di questo livello riceve i dati dal livello superiore e li suddivide in frame, aggiungendo ad esso un codice per il controllo degli errori di trasmissione (checksum).

Si hanno due tipi di codici di controllo: i **codici rilevatori**, che permettono soltanto di capire che il frame non è corretto, ed i **codici correttori**, che permettono non solo di capire che si è verificato un errore, ma anche di individuare la posizione dell' errore e, quindi, di correggerlo. Quest' ultimi richiedono molti più bit dei codici rilevatori e, quindi, sprecano ampiezza di banda; di solito, perciò, sono usati i codici rilevatori.

In caso di errore, se il servizio è inaffidabile il frame può essere semplicemente scartato; se la linea deve essere affidabile bisogna che tutti i frame arrivino correttamente; se si usa un codice rilevatore il ricevente deve richiedere la ritrasmissione dei frame errati.

Il modo consueto per assicurare una consegna affidabile è quello di fornire al mittente un riscontro di quello che sta accadendo all' altro capo della

linea. Tipicamente, il protocollo richiede che il ricevente rispedisca alcuni speciali pacchetti di controllo con valore positivo o negativo a seconda dei pacchetti ricevuti. Se il mittente riceve un riscontro positivo su di un pacchetto spedito, sa che esso è arrivato correttamente. Se, invece, ottiene un riscontro negativo significa che qualcosa è andata male e che occorre ritrasmettere il pacchetto. Una complicazione aggiuntiva potrebbe derivare dalla possibilità che i problemi hardware causino la sparizione totale del pacchetto. Se un pacchetto non arriva a destinazione il mittente non aspetterà all'infinito; infatti, è utilizzato un timer, che è avviato quando i dati sono trasmessi: se il timer supera la soglia limite senza ricevere l'ack rimanderà di nuovo i pacchetti. Tuttavia, se il pacchetto o il messaggio di riscontro sono persi, il timer scade e la stazione mittente, quindi, rinvia i dati. Così, però, il mittente riceverebbe due o più volte lo stesso pacchetto. Per risolvere questo problema sono numerati i pacchetti inviati, così il ricevente, nel caso in cui dovesse ricevere un numero uguale a quello precedente, lo scarta. Questa tecnica è nota come *Stop and Wait*; le altre tecniche maggiormente utilizzate per il controllo degli errori sono il *Codice di Hamming* ed il *CRC* (*Codice di Ridondanza Ciclica*).

### 4.3 Controllo di flusso

Un altro importante problema di progettazione che si ritrova nel Livello DataLink è quello di gestire una linea condivisa quando più nodi vogliono inviare messaggi nello stesso tempo, e decidere cosa fare di un mittente che sistematicamente tende a trasmettere pacchetti più velocemente di quanto il ricevente li accetti. Questa situazione può facilmente essere riscontrata quando il mittente è dislocato su una macchina veloce ed il ricevente su una macchina lenta. Il mittente continua a spedire pacchetti ad alta velocità, fino a quando il ricevente non è completamente sopraffatto. Anche se la trasmissione è esente da errori, ad un certo punto, il ricevente non sarà in grado di gestire i pacchetti in arrivo ed inizierà a perderli. La tipica soluzione è quella di introdurre un controllo di flusso per obbligare il mittente a rispettare la velocità del ricevente nello spedire i pacchetti. Questa imposizione solitamente richiede un certo tipo di meccanismo di riscontro in modo che il mittente possa essere avvisato se il ricevente è in grado di ricevere o meno. Nel caso in cui invece più nodi vogliono inviare contemporaneamente dei messaggi, si tende ad introdurre un controllo centralizzato, creando un singolo nodo di controllo responsabile di determinare chi ottiene la priorità all'interno della rete; il nodo successivo, quindi, controllerà quando la rete non sarà più occupata, così da poter inviare il messaggio appena questa diventerà libera. Può

accadere, però, che più nodi monitorizzano la rete e che appena questa sia libera inviano immediatamente i messaggi. In questo caso si avranno dei problemi di collisione; per ovviare a questo problema, i nodi che monitorizzano la rete attenderanno un tempo casuale prima di inviare i messaggi, poiché è improbabile che i nodi scelgano lo stesso istante per inviare i dati.

## 4.4 Protocolli

Nello stack IP, in alcuni casi, il Livello DataLink consiste nell' utilizzo di una rete realizzata con un altro protocollo per il trasporto di pacchetti IP. Sono protocolli del livello datalink **Ethernet** (per le LAN), **PPP**, **HDLC** e **ADCCP** per le connessioni punto-punto. Molti protocolli di DataLink non utilizzano conferme ed alcuni potrebbero addirittura non controllare se sono stati commessi errori di trasmissione. In questo caso, devono essere i livelli superiori ad effettuare il controllo di flusso, il controllo degli errori e gestire le conferme (e relative ritrasmissioni).

In alcune reti, come le LAN IEEE 802, questo livello è diviso nei sottolivelli MAC e LLC.

### 4.4.1 Sottolivello LLC

Il **sottolivello** superiore è **Logical Link Control (LLC)**, e può fornire servizi di controllo di flusso, conferma, rilevazione (o correzione) degli errori. I protocolli PPP e HDLC fanno parte di questo sottolivello.

I protocolli di sottolivello LLC che forniscono il servizio di “conferma di” o “garanzia di” ricezione dei dati devono prevedere messaggi di conferma avvenuta ricezione (*acknowledge*, o ACK). Il trasmittente può attendere il riscontro di ciascun messaggio prima di trasmettere il successivo, oppure può continuare a trasmettere fino al raggiungimento di un numero massimo di messaggi non ancora confermati dal ricevente, nei cosiddetti protocolli *finestrati*.

Nei protocolli con finestra ciascun pacchetto trasmesso è identificato con un numero progressivo all' interno della *finestra*, detto *numero di sequenza* (*sequence number*); i messaggi di conferma devono riportare il numero di sequenza del pacchetto che riscontrano.

I messaggi di conferma possono essere cumulativi (“ricevuti i pacchetti fino a  $N$ ”), o richiedere la ritrasmissione cumulativa (“ritrasmettere i pacchetti fino a  $N$ ”) o selettiva dei soli pacchetti non ricevuti correttamente. In alcuni casi, il riscontro dei messaggi ricevuti utilizza un messaggio dedicato, in altri

casi il riscontro è inserito in campi specifici dei messaggi trasmessi in direzione opposta (*piggyback*).

#### 4.4.2 Sottolivello MAC

Il **sottolivello inferiore** è **Media Access Control**. A volte, questo si riferisce al sottolivello che determina chi può utilizzare il mezzo trasmissivo in un certo momento (di solito *CSMA/CD*); altre volte, si riferisce in una struttura a frame che contiene indirizzi MAC. Ci sono essenzialmente due tipi: **distribuito** e **centralizzato**.

Il sottolivello MAC stabilisce dove iniziano i frame di dati e dove finiscono.

Il **CSMA/CD** è l'acronimo inglese di *Carrier Sense Multiple Access with Collision Detection*, ovvero **accesso multiplo tramite rilevamento della portante con rilevamento delle collisioni**. È un'evoluzione del protocollo MAC del CSMA ed è nato per la risoluzione dei conflitti di trasmissione dovuti al CSMA puro.

L'algoritmo è il seguente:

- l'adattatore sistema il frame in un buffer;
- se il canale è inattivo si procede alla trasmissione, se è occupato si attende prima di ritrasmettere;
- mentre si trasmette, l'adattatore monitora la rete (è questo il vero e proprio *Collision Detection*); se non riceve segnali da altri adattatori considera il frame spedito. Tale segnale si ricava confrontandolo con quello che trasmette: se i due differiscono è avvenuta una collisione, quindi va interrotta la trasmissione;
- se l'adattatore riceve, durante una trasmissione, un segnale da un altro adattatore, arresta la trasmissione e trasmette un segnale di disturbo (*jam*);
- dopo aver abortito la trasmissione attende in maniera esponenziale (algoritmo di backoff esponenziale binario).

L'attesa esponenziale funziona in questo modo: gli adattatori aspettano un tempo casuale entro un valore massimo  $d$  (il protocollo che usa il CSMA/CD, ad esempio Ethernet, fissa tale valore). Se è generata nuovamente una collisione il valore  $d$  è raddoppiato, così fino a che questo è sufficientemente grande. Questa tecnica è chiamata *recessione binaria esponenziale*. Avviene perché se altri adattatori sono contemporaneamente in attesa, tutti simultaneamente tenteranno di trasmettere provocando altre collisioni.

Il segnale di disturbo è inviato per avvertire tutti gli adattatori che è avvenuta una collisione.

L' **indirizzo MAC**, detto anche indirizzo fisico, indirizzo ethernet o indirizzo LAN, è un codice di 48 bit (6 byte) assegnato in modo univoco ad ogni scheda di rete ethernet prodotta al mondo. Rappresenta un nome per un particolare dispositivo di rete.

L' originale Indirizzo MAC IEEE 802, ora chiamato ufficialmente "MAC-48", deriva dalla specifica dell' Ethernet. Poiché chi inizialmente progettò l' Ethernet decise di usare uno spazio indirizzi a 48 bit, adesso disponiamo potenzialmente di ben 248 (cioè 281.474.976.710.656) possibili indirizzi MAC. In tale formato (quello attualmente più diffuso), i 48 bit del codice sono suddivisi in 12 cifre esadecimali: le prime 6 cifre individuano il produttore dell' interfaccia di rete, mentre le successive corrispondono al numero di serie della scheda stessa.

L' indirizzo MAC si scrive normalmente in sei ottetti separati da un trattino (ad esempio 00-50-FC-A0-67-2C) ed i primi tre ottetti sono detti *OUI* (*Organizationally Unique Identifier*). Per questo tipo di indirizzi, di solito, si preferisce la notazione esadecimale anche per differenziarla dagli indirizzi IP che usano la notazione decimale.

I tre sistemi di numerazione utilizzano lo stesso formato, differendo soltanto nella lunghezza dell' identificatore. Gli indirizzi si dividono in *indirizzi universally administered* e *indirizzi locally administered*. Gli **indirizzi universally administered** sono assegnati ai dispositivi dal loro produttore; sono talvolta chiamati "indirizzi burned-in". I primi tre ottetti (in ordine di trasmissione) identificano l' organizzazione o il produttore che ha emesso l' identificatore e rappresentano l' Organizationally Unique Identifier (OUI). I successivi tre o cinque ottetti sono assegnati dal produttore che deve rispettare il solo vincolo dell' univocità. Gli **indirizzi locally administered** sono assegnati ad un componente dall' amministratore di rete, annullando l' "indirizzo burned-in". Gli indirizzi locally administered non contengono gli ottetti OUI. Gli indirizzi universally administered e locally administered si distinguono per mezzo del settaggio del secondo bit meno significativo del byte più significativo dell' indirizzo. Se il bit è a 0, l' indirizzo è universally locally administered (o local scope); se è a 1, esso è locally globally administered (o global scope). Tale bit vale 0 in tutti gli OUI.

Ogni scheda ha un indirizzo unico perché i primi 24 bit sono identificativi della casa produttrice ed i successivi della scheda. In questo modo, ogni casa produttrice ha a disposizione 224 indirizzi, e, quindi, può produrre più di 16 milioni di schede; se un produttore ne produce meno, gli indirizzi (a 48 bit) non assegnati sono persi, non potendo essere utilizzati da altri costruttori. Si comprende, quindi, come l' indirizzo MAC non cambi se si sposta una



scheda di rete da una LAN ad un'altra, mentre può cambiare l'indirizzo IP. La conversione tra indirizzo MAC ed indirizzo IP avviene mediante alcuni protocolli (il più conosciuto è ARP).

## 4.5 Il frame Ethernet

Il **frame Ethernet** costituisce l'unità elementare di informazione per il sottolivello MAC di IEEE 802.3. La struttura di un frame è riassunta nella Tabella 1, che riporta il nome di ciascun campo e la relativa lunghezza:

Campo	Byte
PRE	7
SFD	1
DA	2-6
SA	2-6
L/T	2
Dati	0-1500
PAD	0-46
FCS	4

Tabella 4.1: Struttura del frame Ethernet

I campi hanno il seguente significato:

- PRE (Preamble) è il preambolo del frame. Si tratta semplicemente di una sequenza di segnali 1 e 0 che consente al ricevente di sincronizzare la comunicazione.
- SFD (Starting Frame Delimiter). Questo campo è composto da un byte, la cui sequenza di bit è 10101011 (in esadecimale AB). Lo SFD dichiara che dal prossimo byte avrà inizio il frame vero e proprio, a partire dall'indirizzo di destinazione del frame (DA).
- DA (Destination Address) è l'indirizzo destinazione. Si tratta di sei byte, spesso rappresentati nella forma aa:bb:cc:dd:ee:ff. Il primo bit ha un significato particolare: se vale 0, la destinazione è una singola unità, altrimenti è un gruppo. Anche il secondo bit ha un significato speciale: se vale 0, l'indirizzo ha valore globale, altrimenti ha soltanto valore locale.
- SA (Source Address) è l'indirizzo sorgente. Ha la stessa struttura del DA, ma rappresenta sempre una singola unità, per cui il primo bit è sempre 0.

- L/T (Length/Type) rappresenta la lunghezza o tipo del frame. Possono esistere diversi tipi di frame. Il tipo normale serve a trasferire dati, ma in certi casi è necessario trasmettere informazioni estranee ai dati veri e propri, per segnalare qualche particolare situazione creatasi nella rete locale. In questo caso il campo L/T assume un valore da 1536 a crescere; valori differenti superiori o uguali a 1536 determinano un tipo diverso di frame. Se invece il valore è inferiore a questa soglia (al massimo 1500), questo indica esattamente il numero di byte di dati forniti dal livello superiore (il MAC client) che saranno trasmessi in questo frame.
- Dati (Payload) sono i dati veri e propri che, nel caso di frame normale, saranno trasmessi con questo frame. I dati veri e propri possono ammontare al massimo a 1500 byte, ma se sono meno di 46 byte, occorre aggiungere dei byte supplementari di riempimento per arrivare almeno a 46 byte. Questo garantisce in ogni caso una lunghezza totale del frame di almeno 64 byte, essenziale per evitare che la trasmissione di frame troppo corti sui più lunghi segmenti ammessi provochi la mancata individuazione delle collisioni nei casi peggiori.
- FCS (Frame Check Sequence). Il mittente calcola su tutta la parte precedente del frame un valore di controllo secondo un algoritmo CRC (Cyclic Redundancy Check), ed inserisce in questo campo il risultato. Il ricevente farà lo stesso non appena ricevuto l'intero frame, e potrà così confrontare il valore di questo campo con quello da lui calcolato. In questo modo si elevano notevolmente le possibilità di riscontrare errori di trasmissione nei frame, che provocano lo scarto del frame errato.

# Capitolo 5

## Il Livello Rete

Il **Livello Rete** è il livello 3 della pila ISO/OSI. Questo livello riceve datagrammi dal Livello Trasporto e forma pacchetti che sono passati al Livello DataLink.

Il compito del Livello Rete è la trasmissione di pacchetti tra due host arbitrari, che in generale non sono direttamente connessi (ovvero non hanno un collegamento diretto tra di loro). Nel modello ISO/OSI, il Livello Rete è presente in tutti i commutatori della rete, mentre i livelli superiori sono presenti solo nei nodi terminali. In particolare, le funzioni del Livello Rete sono:

- **inoltro**, ovvero ricevere un pacchetto su una porta, immagazzinarlo e ritrasmetterlo su un' altra. Questa funzione è presente in tutti i nodi della rete.
- **frammentazione**: se un pacchetto ricevuto ha una dimensione eccessiva per la rete su cui deve essere trasmesso, il Livello Rete lo divide in frammenti e si occupa di riassemblare i frammenti ricevuti al momento della consegna.
- **instradamento** (*routing*), ovvero determinare il percorso per la trasmissione dei dati attraverso la rete. Nella maggior parte dei casi, questa funzione è svolta dinamicamente tramite appositi algoritmi, che analizzano le condizioni della rete, le tabelle di instradamento, la priorità del servizio ed altri elementi secondari.
- alcuni protocolli di rete forniscono un servizio di gestione delle **connessioni**, ovvero richiedono che sia stabilito un canale di comunicazione prima che due host possano scambiarsi dati; altri trasportano semplicemente i datagrammi a destinazione. I protocolli orientati alla connessione possono offrire garanzie di consegna in ordine dei pacchetti,

mentre questo non avviene normalmente nei protocolli non orientati alla connessione.

- funzioni talvolta presenti nel Livello Rete sono il **controllo della congestione**, o garanzie di **qualità di servizio**, tipicamente basate sulla prenotazione delle risorse su tutti i nodi della rete.
- nelle reti geografiche (WAN o MAN) può essere gestita la **tariffazione**, calcolata sulla base dei tempi di connessione e/o di altri parametri.

## 5.1 Livello Rete in IP

Nel modello TCP/IP, il livello 3 è detto **livello internet** oppure **livello di internetworking**, in quanto interconnette reti eterogenee, che possono essere basate su protocolli di livello collegamento o su protocolli di rete, per realizzare un' unica rete in modo trasparente agli utilizzatori. La forza di IP sta proprio in questo agnosticismo rispetto al Livello Rete, che permette di usare o riusare tecnologie già disponibili, e di adattarsi con naturalezza a nuove tecnologie.

## 5.2 L'Internet Protocol

L' **Internet Protocol (IP)** è un protocollo di rete a pacchetto, non connesso. IP è un protocollo di interconnessione di reti (Inter-Networking Protocol), nato per interconnettere reti eterogenee per tecnologia, prestazioni, gestione.

I protocolli di trasporto utilizzati su IP sono soprattutto TCP e UDP.

## 5.3 Indirizzo IP

Un indirizzo IP può essere visto come l' equivalente di un indirizzo stradale o un numero telefonico riferito a dispositivi collegati ad una qualsiasi rete telematica. Infatti, così come un indirizzo stradale o un numero telefonico identificano rispettivamente un edificio o un telefono, un indirizzo IP identifica univocamente uno specifico computer o dispositivo di rete. Gli indirizzi IP possono essere assegnati localmente per realizzare una LAN, come succede con la numerazione degli interni di un edificio. Ma, al contrario degli indirizzi stradali, gli indirizzi IP possono mutare il loro valore a seconda di molti fattori (diversa LAN, indirizzamento dinamico) o a seconda della volontà dell' utente. Più esattamente, l' indirizzo IP è assegnato ad una *interfaccia* (ad esempio una scheda di rete) che identifica l' host di rete. Va considerato,

infatti, che un host può contenere più di una interfaccia: ad esempio, un router ha diverse interfacce (minimo due) e per ognuna occorre un indirizzo IP.

Gli indirizzi IP pubblici ed i range di indirizzi sono rilasciati e regolamentati dall' ICANN tramite una serie di organizzazioni delegate. Tuttavia è da tener presente che a livello mondiale e nazionale i primi provider di connessione Internet si sono accaparrati un numero non controllato di indirizzi IP. Ciò ha portato a prevedere la terminazione degli indirizzi entro pochi anni dall' uscita del protocollo e, per ovviare a tale disagio, si è proceduto alla ridefinizione della versione 6 del protocollo IP (attualmente la versione largamente in uso è la 4). Tale versione è basata su indirizzi a 128 bit anziché a 32 e ciò permette l' assegnazione di un numero maggiore di indirizzi. Ma la difficile implementazione a livello globale dell' IPv6 ha portato all' introduzione di nuovi concetti che hanno rivoluzionato la teoria delle reti. Vanno citati l' abbandono del concetto di classi di indirizzi IP e l' utilizzo sempre maggiore di indirizzi classless (privi del concetto di classe), il subnet mask, la riorganizzazione gerarchica degli indirizzi mediante utilizzo massivo di NAT.

### 5.3.1 Domain Name System

Il **Domain Name System (DNS)** è un servizio di directory utilizzato per la risoluzione dei nomi dei server da indirizzi logici e testuali (URL) in indirizzi IP. Questa funzione è essenziale per l' usabilità di Internet, visto che gli esseri umani hanno più facilità a ricordare nomi testuali, mentre i dispositivi di instradamento lavorano su indirizzi binari. Permette, inoltre, ad una qualsiasi entità di cambiare o riassegnare il proprio indirizzo IP, senza dover notificare tale cambiamento a nessuno, tranne che al proprio server DNS di riferimento.

Un' altra delle peculiarità del DNS è quella di consentire, ad esempio ad un sito web, di essere ospitato su più server (ognuno con il proprio indirizzo IP), con una conseguente divisione del carico di lavoro.

### 5.3.2 Indirizzamento IP statico e dinamico

[modifica] Gli indirizzi IP possono essere assegnati in maniera permanente (ad esempio un server che si trova sempre allo stesso indirizzo) oppure in maniera temporanea, da un range di indirizzi disponibili.

Gli **indirizzi dinamici** sono utilizzati per identificare dispositivi non permanenti in una LAN. Un server DHCP presente nella LAN assegna dinamicamente ed automaticamente l' indirizzo scegliendolo casualmente da un range preimpostato. Si può scegliere il range di indirizzi a seconda del

numero delle utenze della rete impostando la *netmask*, ossia dicendo al server DHCP quanti bit dell' indirizzo sono assegnabili dinamicamente ad ogni singolo client che fa accesso. Ad esempio, se la netmask ha valore 255.255.255.0 (dove ogni blocco separato da puntini denota un gruppo di 8 bit) solo gli ultimi 8 bit sono assegnabili agli host.

Gli **indirizzi statici** sono utilizzati per identificare dispositivi semi-permanenti con indirizzo IP permanente. L' indirizzo statico può essere configurato direttamente sul dispositivo, oppure come parte di una configurazione DHCP che associa all' indirizzo MAC il corrispondente indirizzo IP statico.

### 5.3.3 Classi di indirizzi IP

Le **classi di indirizzi IP** sono un modo per caratterizzare lo spazio di indirizzamento IPv4. Il *Classful Addressing* (indirizzamento basato sulla classe) prevedeva che dai primi bit di un indirizzo si potesse determinare la maschera di rete. Questa scelta, con il crescere dell' utenza di internet, si è rivelata troppo rigida, ed è stata abbandonata a favore dell' indirizzamento senza classe (CIDR). Gli indirizzi IP all' origine erano stati suddivisi in classi. Il tipo di classe si può determinare sulla base dei bit più significativi:

- classe A: il primo byte rappresenta la rete, gli altri l'host. La maschera di rete è 255.0.0.0, o /8. Questi indirizzi iniziano tutti con un bit a 0.
- classe B: i primi due byte rappresentano la rete, gli altri l'host. La maschera di rete è 255.255.0.0, o /16. Questi indirizzi iniziano con la sequenza 10.
- classe C: i primi 3 byte rappresentano la rete, gli altri l'host. La maschera di rete è 255.255.255.0, o /24. Questi indirizzi iniziano con la sequenza 110.
- classe D: riservata agli indirizzi multicast. Questi indirizzi cominciano con la sequenza 1110. Non hanno maschera di rete, essendo tutti e 32 i bit dell' indirizzo utilizzati per indicare un gruppo, non un singolo host.
- classe E: riservata per usi futuri. Questi indirizzi cominciano con la sequenza 11110 e non è definita una maschera di rete.

## 5.4 Il router

Nella tecnologia delle reti informatiche, un **router** è un dispositivo di rete che si occupa di instradare pacchetti lavorando al livello 3 del modello OSI.

La caratteristica fondamentale dei router è che la funzione di instradamento è basata sugli indirizzi di livello 3 del modello OSI (corrispondente al livello IP dello stack TCP/IP), a differenza dello switch che instrada sulla base degli indirizzi di livello 2 (collegamento) MAC. Gli elementi della tabella di instradamento (o *Routing Table*) non sono, quindi, singoli calcolatori, ma intere reti, ovvero sottoinsiemi anche molto ampi dello spazio di indirizzamento. Questo è fondamentale per la scalabilità delle reti, in quanto permette di gestire reti anche molto grandi facendo crescere le tabelle di instradamento in modo meno che lineare rispetto al numero di host.

In generale, i router necessitano di essere configurati, non essendo dispositivi plug and play. A seconda della tipologia del router, per essere configurato esso fornisce un'interfaccia basata su web (accessibile digitando l'indirizzo del gateway nel browser) o attraverso un'apposita console a linea di comando su porta seriale.

Per garantire la massima affidabilità e lo sfruttamento ottimale dei collegamenti in caso di reti complesse costituite da molte sottoreti diverse e variamente interconnesse, i router possono costruire le loro tabelle di instradamento in modo dinamico, scambiandosi periodicamente informazioni su come raggiungere le varie reti che collegano l'un l'altro. Per fare questo, sono stati messi a punto dei protocolli di routing appositi, come OSPF, RIP e BGP, attraverso i quali i router si scambiano informazioni sulle reti raggiungibili.

Un router può interconnettere reti di livello 2 eterogenee, come ad esempio una LAN Ethernet con un collegamento geografico in tecnologia *Frame Relay* o ATM. Inoltre, rispetto ad un bridge, un router blocca le tempeste broadcast. Molti router destinati al mercato domestico incorporano la funzionalità di *access point* per reti wireless Wi-Fi.

Alcuni router possiedono anche un firewall incorporato, poiché il punto di ingresso/uscita di una rete verso l'esterno è il luogo migliore dove effettuare controlli sui pacchetti in transito.

I router possono essere normali computer che fanno girare un software apposito (gateway), o (sempre più spesso) apparati specializzati, dedicati a questo solo scopo. I router di fascia più alta sono basati su architetture hardware specializzate per ottenere *prestazioni wire speed*, letteralmente alla velocità della linea. Un *router wire speed* può inoltrare pacchetti alla massima velocità delle linee cui è collegato. I router di fascia media ed alta hanno normalmente una costruzione modulare, che permette di aggiungere interfacce verso reti di tipo diverso secondo la necessità.

## 5.5 La Virtual LAN (VLAN)

Il termine **VLAN (Virtual LAN)** indica un insieme di tecnologie che permettono di segmentare il dominio di broadcast, che si crea in una rete locale (tipicamente IEEE 802.3) basata su switch, in più reti non comunicanti tra loro. Le applicazioni di questa tecnologia sono tipicamente legate ad esigenze di separare il traffico di gruppi di lavoro o dipartimenti di una azienda, per applicare diverse politiche di sicurezza informatica.

Le prime versioni proprietarie permettevano di realizzare su un singolo switch diverse reti VLAN, assegnando ciascuna porta ad una di queste reti. Gli host collegati ad una rete potevano comunicare solo tra di loro e non con quelli collegati alle altre reti, se non per mezzo di un router connesso ad entrambe le VLAN. Il risultato è lo stesso che si otterrebbe utilizzando un diverso switch “tradizionale” per ciascuna rete, ma con alcuni vantaggi:

- costi ed ingombri: invece di diversi switch, è possibile utilizzare un solo switch con molte porte, risparmiando in costi di acquisizione e manutenzione, spazio occupato, prese di alimentazione elettrica ed indirizzi IP per la gestione remota;
- flessibilità: le porte dello switch possono essere spostate da una VLAN ad un’altra per mezzo di semplici operazioni di riconfigurazione software, spesso effettuabili da remoto. Altre VLAN possono essere aggiunte utilizzando le porte esistenti, e quindi a costo nullo.

In seguito, è stata sviluppata la possibilità di collegare tra loro due switch unendo le VLAN presenti su di essi (**VLAN trunking**). Questo permette di realizzare VLAN che si estendono nelle diverse parti di una rete aziendale, anche su scala geografica. Questa tecnologia è stata standardizzata come IEEE 802.1Q, in modo che apparati di rete di diversi fornitori possano essere collegati.

### 5.5.1 Realizzazione e terminologia in IEEE 802.1Q

Ogni VLAN è identificata da un numero, detto **VID (Vlan ID)**, compreso tra 1 e 4094 (0 e 4095 sono riservati).

Per realizzare il trunking di VLAN presenti su switch diversi, è necessario che sui collegamenti tra switch si possa identificare a quale VLAN appartiene ciascun pacchetto. Per fare questo, prima del pacchetto IEEE 802.3 viene aggiunto un preambolo di 2 byte, detto **VLAN TAG**, che contiene il VID relativo a quel pacchetto. Il VLAN TAG (detto, anche, **tag control information** o **TCI**) è così suddiviso:



- **user priority** è un campo a 3 bit utilizzato per indicare un livello di priorità per il frame. L' utilizzo di questo campo è definito in IEEE 802.1P.
- **CFI** è il campo di 1 bit che indica se i MAC address nel frame sono in forma canonica.
- **VID** è il campo di 12 bit che indica l' ID delle VLAN, che possono così essere fino a 4096 (ossia 2<sup>12</sup>). Di queste, la prima (VLAN 0) e l' ultima (VLAN 4095) sono riservate, quindi gli ID realmente usabili sono 4094.

Una porta di uno switch su cui viaggiano pacchetti con il VLAN TAG è detta **tagged** o **trunk port**. Viceversa, una su cui viaggiano pacchetti senza VLAN TAG è detta **access port**. Alcuni switch accettano anche un traffico misto di pacchetti tagged e non tagged, ed una porta configurata in questo modo è detta **hybrid port**.



# Capitolo 6

## Il Livello Trasporto

Il **Livello Trasporto** è il quarto dei livelli nel modello OSI. Il suo compito è di fornire servizi al Livello Sessione, sfruttando i servizi del Livello Rete. Lo scopo del Livello Trasporto è fornire un canale di comunicazione end-to-end per pacchetti. Di seguito sono riportati i servizi, in genere, offerti dal Livello Trasporto; nessuno di tali servizi è obbligatorio. Di conseguenza, per ciascuna applicazione è possibile scegliere il protocollo più adatto allo scopo.

- Servizio orientato alla connessione. In genere, il Livello Rete non stabilisce una connessione persistente verso l'host di destinazione. Il Livello Trasporto si incarica, quindi, di realizzare una connessione persistente che è poi chiusa quando non è più necessaria.
- Corretto ordine di consegna. Poiché i pacchetti possono seguire percorsi diversi all'interno della rete, non c'è alcuna garanzia che i dati siano recapitati nello stesso ordine in cui sono stati inviati. Il Livello Trasporto verifica che i pacchetti siano riordinati nella giusta sequenza in ricezione prima di passarli al livello superiore.
- Trasferimento affidabile. Il protocollo si occupa di garantire che tutti i dati inviati siano ricevuti; nel caso il servizio di rete utilizzato perda pacchetti, il protocollo di trasporto si occupa di ritrasmetterli.
- Controllo di flusso. Se gli host coinvolti nella comunicazione hanno prestazioni molto differenti può capitare che un host più veloce "inondi" di dati uno più lento. Mediante il controllo di flusso, un host in "difficoltà" può chiedere di abbassare il tasso di trasmissione in modo da poter gestire le informazioni in ingresso.
- Controllo di Congestione. Il protocollo riconosce uno stato di congestione della rete, e adatta di conseguenza la velocità di trasmissione.

- Orientamento al Byte. Invece che gestire i dati in base ai pacchetti, è fornita la possibilità di vedere la comunicazione come uno stream di byte, in modo da semplificarne l' utilizzo.
- Multiplazione. Il protocollo permette di stabilire diverse connessioni contemporanee tra gli stessi due host, tipicamente utilizzando l' astrazione delle porte. Nell' uso comune, diversi servizi utilizzano porte diverse.

I protocolli di trasporto più utilizzati sono *TCP* ed *UDP*. *TCP* è il più complicato fra i due e fornisce un servizio end-to-end orientato alla connessione ed al byte, con verifica del corretto ordine di consegna, controllo di errore e di flusso. *UDP*, invece, è un protocollo più snello e fornisce un servizio a datagrammi, senza connessione, con un meccanismo di riduzione degli errori e con porte multiple.

## 6.1 Il Transmission Control Protocol ed il User Datagram Protocol

Il **Transmission Control Protocol (TCP)** è un protocollo di Livello Trasporto della suite di protocolli Internet. Il TCP può essere classificato al Livello Trasporto (OSI level 4) del modello di riferimento OSI, e di solito è usato in combinazione con il protocollo di Livello Rete (OSI level 3) IP. La corrispondenza con il modello OSI non è perfetta, in quanto il TCP e l' IP nascono prima. La loro combinazione è indicata come TCP/IP e, alle volte, è erroneamente considerata un unico protocollo.

Il servizio offerto da TCP è il trasporto di un flusso di byte bidirezionale tra due applicazioni in esecuzione su host differenti. Il protocollo permette alle due applicazioni di trasmettere contemporaneamente nelle due direzioni, quindi il servizio può essere considerato "Full Duplex". Il flusso di byte è frazionato in blocchi per la trasmissione dall' applicazione a TCP (che normalmente è implementato all' interno del sistema operativo), per la trasmissione all' interno di segmenti TCP, per la consegna all' applicazione che lo riceve, ma questa divisione in blocchi non è per forza la stessa nei diversi passaggi.

TCP è un protocollo orientato alla connessione, ovvero prima di poter trasmettere dati deve stabilire la comunicazione, negoziando una connessione tra mittente e destinatario, che è esplicitamente chiusa quando non più necessaria. Esso, quindi, ha le funzionalità per creare, mantenere e chiudere una connessione.

TCP garantisce che i dati trasmessi, se giungono a destinazione, lo facciano in ordine e una volta sola. Più formalmente, il protocollo fornisce ai livelli superiori un servizio equivalente ad una connessione fisica diretta che trasporta un flusso di byte. Questo è realizzato attraverso vari meccanismi di acknowledgment e di ritrasmissione su timeout.

TCP possiede funzionalità di controllo di flusso e di controllo della congestione sulla connessione, attraverso il meccanismo della finestra scorrevole. Questo permette di ottimizzare l' utilizzo della rete anche in caso di congestione, e di condividere equamente la capacità disponibile tra diverse sessioni TCP attive su un collegamento.

TCP fornisce un servizio di moltiplicazione delle connessioni su un host, attraverso il meccanismo delle porte.

A differenza del TCP, l' **User Datagram Protocol (UDP)** è un protocollo di tipo connectionless, inoltre non gestisce il riordinamento dei pacchetti né la ritrasmissione di quelli persi, ed è perciò generalmente considerato di minore affidabilità. È, in compenso, molto rapido ed efficiente per le applicazioni "leggere" o time-sensitive. Ad esempio, è usato spesso per la trasmissione di informazioni audio o video. Dato che le applicazioni in tempo reale spesso richiedono un ritmo minimo di spedizione, non vogliono ritardare eccessivamente la trasmissione dei pacchetti e possono tollerare qualche perdita di dati, il modello di servizio TCP può non essere particolarmente adatto alle loro caratteristiche. L' UDP fornisce soltanto i servizi basilari del livello di trasporto, ovvero:

- moltiplicazione delle connessioni, ottenuta attraverso il meccanismo delle porte;
- verifica degli errori mediante una checksum, inserita in un campo dell' intestazione del pacchetto.

mentre TCP garantisce anche il trasferimento affidabile dei dati, il controllo di flusso e il controllo della congestione.

L' UDP è un protocollo stateless, ovvero non tiene nota dello stato della connessione, dunque ha rispetto al TCP informazioni in meno da memorizzare. Un server dedicato ad una particolare applicazione che scelga UDP come protocollo di trasporto può supportare molti più client attivi.

L' architettura di rete TCP/IP adotta un modello di comportamento chiamato **Best Effort** che descrive uno scenario di rete in cui non è fornita nessuna garanzia sulla consegna dei dati o sul livello di QoS, ma tutte le comunicazioni avvengono con il massimo impegno possibile. La diretta conseguenza di questo approccio è rappresentata da un bitrate ed un tempo di

consegna variabili in base all' attuale carico della rete. Nonostante la rimozione di queste caratteristiche di controllo e preallocazione delle risorse possa sembrare insensata, è in realtà utilissima; infatti, così facendo, la struttura della reti ne risulta semplificata e opera più efficientemente.

# Capitolo 7

## Il Livello Applicazione

### 7.1 La storia di Internet

La storia di Internet inizia negli anni successivi alla seconda guerra mondiale. Nel 1957, il governo degli Stati Uniti d' America fondò *ARPA* (Advanced Research Projects Agency), un' agenzia incaricata di mantenere la leadership americana nel campo della scienza e della tecnologia americana. Fu proprio ARPA che nel 1969 creò *ARPANET*, una rete progettata per sopravvivere ad un' eventuale guerra nucleare che permettesse agli USA di mantenere i collegamenti tra le sue basi militari anche se alcuni siti fossero stati bombardati.

Il primo appalto per la costruzione della rete fu concesso ad una società chiamata BBN (Bolt Beranek and Newman) che collegò quattro università nel sud-ovest degli Stati Uniti, usando linee telefoniche. L' impianto divenne attivo nel Settembre 1969, ponendo le basi per il successivo sviluppo di **Internet**.

Dagli anni Settanta in poi, ARPANET ebbe un enorme sviluppo. L' introduzione dei protocolli TCP/IP e FTP permise di trasferire files da un computer ad un altro indipendentemente dal loro tipo e dal collegamento usato.

Dal 1980, ARPANET si trasformò in uno strumento vitale per le università e per i centri di ricerca americani. La posta elettronica si affiancò al protocollo FTP rendendo semplice lo scambio di informazioni ed il coordinamento delle attività tra gruppi di ricerca geograficamente distanti.

Nel giro di breve tempo, nacquero MILNET (una rete alternativa creata dal dipartimento della Difesa americana per non dover più dipendere da ARPANET), NFS (National Science Foundation, un ente con il duplice scopo di fornire risorse di elaborazione alle università mediante l' uso centralizzato

di supercomputer e di favorire la crescita di un sistema di comunicazione veloce tra queste ultime), CSNET (una rete che univa le diverse facoltà di informatica americane) e moltissime altre reti locali che usavano ARPANET per servirsi della comunicazione elettronica.

Alla fine degli anni Ottanta, NSF costruì **NSFNET** per rimpiazzare ARPANET, che fu definitivamente smantellato nel 1990.

## 7.2 Tipi di connessione

Esistono due modalità principali per collegarsi ad Internet. Nel **collegamento permanente** il computer (o la rete locale di cui fa parte) è collegato direttamente tramite una linea dedicata ad una rete TCP/IP che è parte di Internet. La linea dedicata garantisce un' elevata velocità di accesso ad Internet, ma chiede grandi costi di gestione. Invece, nel **collegamento tramite Internet Service Provider** gli utenti si collegano ad Internet mediante un Internet Service Provider, ovvero un' azienda specializzata nel fornire servizi Internet. Una connessione di questo tipo richiede che l' utente disponga di un computer, di un modem e di un accesso ad una linea telefonica. Gli Internet Service Provider permettono di accedere ad Internet ad una velocità inferiore rispetto a quella prevista se si usa una linea dedicata, ma con costi altamente inferiori.

## 7.3 Il World Wide Web

Il web è un' applicazione client-server. Dal punto di vista dell' utente, esso consiste di un enorme database di documenti multimediali, chiamati *pagine web*. Ognuna di queste pagine può contenere *collegamenti ipertestuali* (link), ovvero puntatori ad altre pagine in qualunque parte del mondo. Pagine logicamente correlate tra loro sono raggruppate in *siti web*. Ogni pagina web è identificata da un nome univoco, chiamato **URL** (Uniform Resource Locator) o *indirizzo internet*. L' URL è composto da tre parti: il *protocollo di comunicazione*, il nome della macchina su cui risiede la pagina, ed un nome locale a quest' ultima che identifica in maniera univoca la pagina.

Le pagine web sono scritte utilizzando il linguaggio **HTML** (Hyper-Text Markup Language); sono visualizzate per mezzo di un *browser*, ovvero un programma in grado di "recuperare" le pagine dal sito web in cui risiedono e di mostrarle nel computer dell' utente. Quando è fornito l' URL di una pagina web, il browser contatta il processo *server* del sito web in cui questa risiede, stabilendo una connessione TCP, ed esegue la richiesta della pagi-



na. Di conseguenza, il server risponde inviando la pagina desiderata, che è visualizzata sul monitor dell'utente; quindi, la connessione viene chiusa.

Il protocollo di comunicazione, ovvero l'insieme di regole che governano la connessione tra il browser ed un server web è chiamato **Hyper Text Transfer Protocol** (*http*)



## Parte II

### Acquisizione e memorizzazione delle informazioni



## Capitolo 8

# Acquisizione da sensori

Con il termine **sensore** si definisce un dispositivo che trasforma una grandezza fisica che si vuole misurare in un segnale di natura diversa (tipicamente elettrico) più facilmente misurabile o memorizzabile. Il termine sensore è riferito, di solito, al componente che fisicamente effettua la trasformazione della grandezza d'ingresso in un segnale di altra natura. I dispositivi in commercio spesso integrano al loro interno anche alimentatori stabilizzati, amplificatori di segnale, dispositivi di comunicazione remota, ecc. In quest'ultimo caso si preferisce definirli **trasduttori**.

L'informazione da acquisire tramite un sensore è una grandezza che può essere misurata attraverso una misura di intensità luminosa, una misura di concentrazione ionica oppure una misura di tensione/corrente proveniente da un altro tipo di sensore. Ad esempio, i misuratori di intensità luminosa convertono la luce in segnale elettrico. Spesso, le caratteristiche costruttive di tali misuratori li rendono sensibili ad una specifica frequenza dello spettro luminoso. Tale selettività si può ottenere, anche, antepponendo un filtro ottico davanti al misuratore. Siccome certi atomi (o molecole), se eccitati da energia sotto forma di luce (o di calore), possono rilasciare tale energia sotto forma di luce ad una precisa lunghezza d'onda, la misura di intensità luminosa può determinare la concentrazione di una sostanza.

Se si conosce la legge che lega la grandezza da misurare (ad esempio, la concentrazione di una sostanza) con una grandezza che si riesce a misurare (ad esempio, la corrente elettrica), allora si può automatizzare la misura.

La rilevazione di un fenomeno fisico può consistere nel conteggio di eventi oppure può generare un segnale elettrico con tensione/corrente proporzionale all'intensità della grandezza misurata. In quest'ultimo caso, l'informazione è un segnale continuo e, quindi, necessita di una conversione analogico/digitale.

La curva caratteristica dei sensori (figura 8.1) può subire variazioni do-

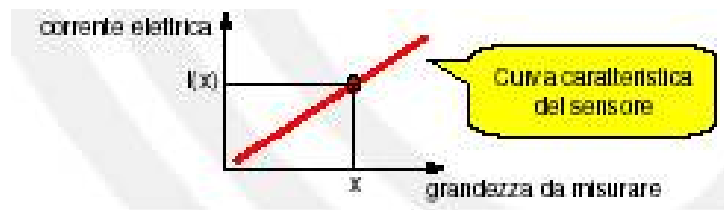


Figura 8.1: Curva caratteristica del sensore

vute a fattori ambientali oppure ad usura: spesso, la curva subisce solo una traslazione oppure una variazione di pendenza (o di scala). A tal proposito, si utilizza la calibrazione per risolvere tali problemi:

- si verifica la presenza di una variazione misurando un campione di valore noto;
- si compensano le variazioni della curva nel sistema di misurazione;
- si lancia un segnale di allarme in caso di compensazioni impossibile.

## 8.1 Tipologie di sensori

A seconda del tipo e dell' utilizzo, i sensori possono:

- dare una lettura direttamente nell' unità ingegneristica d' interesse (esempio nei termometri a mercurio);
- essere collegati ad uno strumento indicatore (display) che provvede a leggere il segnale e tradurlo in una lettura nell' unità ingegneristica;
- essere collegati ad uno strumento registratore che provvede a memorizzare il segnale per una sua successiva elaborazione (il più delle volte, quest' ultimo opera una conversione analogico-digitale che traduce il segnale in dati digitali, che sono immediatamente memorizzati nello strumento stesso o su un computer collegato in remoto).

I sensori possono essere classificati in base al loro principio di funzionamento oppure al tipo di segnale in uscita, ma più comunemente sono classificati in base al tipo di grandezza fisica che misurano, ad esempio:

- sensori di luce (o sensori ottici): fotocellule, fotodiodi, fototransistor, tubi fotoelettrici, CCD, radiometri di Nichols, fotomoltiplicatori;
- sensori di suono: microfoni, idrofoni.

- sensori di accelerazione: accelerometri, sensori sismici.
- sensori di temperatura: termometri, termocoppie, resistori sensibili alla temperatura, termistori, termometri bimetallici e termostati.
- sensori di calore: bolometri, calorimetri.
- sensori di radiazione: contatori Geiger, dosimetri.
- sensori di particelle subatomiche: scintillometri, camere a nebbia, camere a bolle, camere di ionizzazione.
- sensori di resistenza elettrica: ohmmetri, multimetri.
- sensori di corrente elettrica: galvanometri, amperometri.
- sensori di tensione elettrica: elettroscopio, voltmetri.
- sensori di potenza elettrica: wattmetri.
- sensori di pressione: barometri, barografi, misuratori di pressione, altimetri, variometri.
- sensori di movimento: radar, velocimetri, tachimetri, odometri.
- sensori di forza: celle di carico, estensimetri.
- sensori di distanza: sensori ottici.
- ...

## 8.2 Segnale analogico e segnale digitale

Un **segnale analogico** consiste di valori continui esprimibili solo mediante numeri reali; invece, un **segnale digitale** consiste di una sequenza discreta di numeri interi con precisione finita.

La conversione di un segnale analogico in un segnale digitale consiste di due azioni indipendenti: *campionamento* e *quantizzazione*. Nella teoria dei segnali, il **campionamento** è una tecnica che consiste nel convertire un segnale continuo nel tempo in un segnale discreto, valutandone l'ampiezza ad intervalli regolari di tempo. In questo modo, a seguito di una successiva operazione di quantizzazione e conversione, è possibile ottenere una stringa digitale (discreta nel tempo e nell'ampiezza) che approssimi quella continua originaria. Il tempo che intercorre tra una valutazione e l'altra si chiama

**periodo di campionamento.** La **frequenza di campionamento** è l'inverso del periodo. In specifico, la frequenza di campionamento è la misura (espressa in Hertz) del numero di volte al secondo in cui un segnale analogico è misurato e memorizzato in forma digitale. Ogni singola misurazione del segnale corrisponde ad un numero memorizzato, ed è detta **campione**; nella produzione dei campioni avvengono, generalmente, innumerevoli errori di campionamento.

Il teorema che stabilisce quale sia la frequenza minima di campionamento con una determinata caratterizzazione in frequenza (trasformata di Fourier) è il teorema del campionamento. Un segnale può sempre essere rappresentato come somma di sinusoidi. Per ricreare fedelmente un segnale occorre ricreare fedelmente tutte le sinusoidi che lo compongono. Per riprodurre fedelmente una sinusoide occorrono due campioni per periodo. La frequenza di campionamento deve essere doppia della frequenza della sinusoide. Se il segnale da campionare ha larghezza di banda  $f$ , allora la sua componente sinusoidale a frequenza più alta ha frequenza  $f$ . Quindi, la frequenza di campionamento di tale segnale deve essere doppia della larghezza di banda del segnale (**teorema del campionamento di Nyquist**).

Quando si misura una grandezza, l'insieme di valori che essa può assumere in natura è un insieme continuo e composto da infiniti punti. Perché una grandezza sia trasmissibile e codificabile con un numero finito di bit, è necessario far sì che possa assumere solo un numero finito di valori discreti; ciò avviene tramite la **quantizzazione**. Questa tecnica prevede i seguenti passi:

1. Determinazione di una range di variazione.
2. Individuazione di un insieme finito di punti all'interno di tale range.
3. Assegnamento di un numero intero a ciascun punto.
4. Misurazione del segnale analogico e rappresentazione della misura con il numero corrispondente al punto più vicino.

La quantizzazione implica sempre un errore di arrotondamento, detto **errore di quantizzazione**: maggiore è il numero di bit su cui si rappresenta la misura, minore è l'errore di quantizzazione. Questo tipo di errore è irreversibile: quando un procedimento di acquisizione/memorizzazione di dati prevede una o più fasi di quantizzazione, allora non è possibile ritornare alla rappresentazione originale.

Una possibile misura dell'errore di quantizzazione è detta **distorsione di quantizzazione**. Una possibile misura della distorsione su una sequenza



$\{v_i\}$  é data da:

$$MSE = \frac{\sum_1^N (v_i - v'_i)}{N}$$

dove,  $v_i$  è il simbolo prima della quantizzazione,  $v'_i$  è il simbolo dopo la decodifica e  $N$  è il numero di elementi considerati.

## 8.3 Cenni sulla compressione delle informazioni

Una sorgente di informazioni emette una serie ordinata di dati. Tale serie di dati si può rappresentare come una sequenza di simboli dove ciascun simbolo appartiene ad un certo alfabeto dipendente dal tipo di applicazione.

La **compressione dei dati** è una tecnica utilizzata in ambito informatico per la riduzione della quantità di bit necessari alla rappresentazione in forma digitale di un' informazione. La compressione dati è utilizzata sia per ridurre le dimensioni di un file, e quindi lo spazio necessario per la sua memorizzazione e l' occupazione di banda necessaria per la sua trasmissione, sia per ridurre l' occupazione di banda necessaria in una generica trasmissione dati. Le varie tecniche di compressione organizzano in modo più efficiente i dati, spesso perdendo una parte dell' informazione originale, al fine di ottenere una rappresentazione dell' informazione più compatta; quindi, comportante minori risorse per la sua memorizzazione e trasmissione. Come controparte, la compressione dei dati necessita, però, di potenza di calcolo per le operazioni di compressione e decompressione, spesso anche elevata se tali operazioni devono essere eseguite in tempo reale.

Le tecniche di compressione dati si dividono in due grandi categorie:

- La **compressione dati lossy** (*compressione dati con perdita*) è una classe di algoritmi di compressione dati che porta alla perdita di parte dell' informazione originale durante la fase di compressione/decompressione dei dati che la rappresentano. Decomprimendo un file compresso con un metodo "lossy" la copia ottenuta sarà peggiore dell' originale per livello di precisione delle informazioni che codifica, ma in genere abbastanza simile da non comportare perdita di informazioni irrinunciabili. Ciò è possibile poiché i metodi di compressione a perdita di informazioni, in genere, tendono a scartare le informazioni poco rilevanti, archiviando solo quelle essenziali. La compressione dei dati con perdita di qualità è usata in molti settori dell' informatica: su Internet, nell' ambito dello streaming dei media, nella telefonia, per la compressione di immagini o altri oggetti multimediali, ecc. Una volta compres-

so un file con un metodo lossy, le informazioni perse non saranno più recuperabili.

- La **compressione dati lossless** (*compressione dati senza perdita*) è una classe di algoritmi di compressione dati che non porta alla perdita di alcuna parte dell'informazione originale durante la fase di compressione/decompressione dei dati stessi.

Sia data una sequenza ordinata di simboli  $s_0, s_1, \dots, s_n$  di un certo alfabeto emessa da una sorgente d'informazione. Esiste correlazione quando la probabilità che  $a_i$  assuma un certo valore dipende dal valore assunto dagli elementi precedenti della sequenza. Si dice che l'informazione contenuta nella sequenza è ridondante, e si può cambiare alfabeto in modo da usare meno bit.

È meglio rappresentare con meno bit i simboli più frequenti e con più bit quelli meno frequenti.

### 8.3.1 Codifica entropica

Nella teoria dell'informazione, ed in rapporto alla teoria dei segnali, l'entropia misura la quantità di incertezza o informazione presente in un segnale aleatorio. Da un altro punto di vista, l'entropia è la minima complessità descrittiva di una variabile aleatoria, ovvero il limite inferiore della compressione dei dati.

Per stabilire limiti alla massima compressione possibile di un insieme di dati, si utilizza al **teorema della codifica di sorgente di simboli**. Tale teorema definisce il significato operativo dell'entropia. Il teorema stabilisce che, per una serie di variabili aleatorie indipendenti ed identicamente distribuite (i.i.d.) di lunghezza che tende ad infinito, non è possibile comprimere i dati in un messaggio più corto dell'entropia totale, senza rischiare di perdere informazione. Al contrario, compressioni arbitrariamente vicine al valore di entropia sono possibili, con probabilità di perdita di informazione piccola a piacere.

Il **teorema della codifica di sorgente per simboli di codice** stabilisce un limite inferiore e superiore alla minima lunghezza attesa di una serie di parole di codice, in funzione dell'entropia della parola in ingresso (vista come una variabile aleatoria) e della dimensione dell'alfabeto in esame.

Il numero minimo teorico di bit/simbolo per un alfabeto  $s$  è dato da:  $H(s) = \sum_k p_k \log_2 \frac{1}{p_k}$ , dove  $p_k$  è la frequenza di apparizione del simbolo  $k$ -esimo. Tale formula fornisce un valore medio minimo, ma non dice come assegnare i bit ai simboli per ottenerlo.

### Algoritmo di Huffman

In informatica, l' **algoritmo di Huffman** è un algoritmo di codifica dell' entropia usato per la compressione di dati, basato sul principio di trovare il sistema ottimale per codificare stringhe, e basato sulla frequenza relativa di ciascun carattere. La codifica di Huffman usa un metodo specifico per scegliere la rappresentazione di ciascun simbolo, risultando in un codice senza prefissi (cioè, in cui nessuna stringa binaria di nessun simbolo è il prefisso della stringa binaria di nessun altro simbolo) che esprime il carattere più frequente nella maniera più breve possibile. Per un insieme di simboli la cui cardinalità è una potenza di due e con una distribuzione probabilistica uniforme, la codifica di Huffman equivale alla semplice codifica a blocchi binari.

Questa tecnica funziona creando un albero binario di simboli:

1. Iniziare con tanti alberi quanti sono i simboli.
2. Finché c'è più di un albero:
  - Trovare quali sono i due alberi con larghezza totale minore.
  - Combinare i due alberi in uno solo, impostandone uno come il ramo di sinistra e l'altro come quello di destra.
3. Ora, l'albero contiene tutti i simboli. Uno "0" rappresenta il successivo ramo di sinistra, un "1" rappresenta il successivo di destra.



## Capitolo 9

# Memorizzazione su supporti di massa

In ambito informatico, la **memoria** è la parte del computer destinata a conservare informazioni per un certo periodo di tempo. La memorizzazione di informazioni in memoria, ed il successivo recupero delle medesime, sono funzioni fondamentali nel funzionamento del computer. Una memoria può essere considerata astrattamente come una sequenza finita di celle in cui ogni cella contiene una sequenza finita di bit. Normalmente, i bit sono gestiti a gruppi di otto, detti *byte*. Pertanto lo spazio fisico della memoria può essere immaginato come una sequenza di locazioni, ognuna contenente un byte. Ogni locazione è individuata da un preciso indirizzo, normalmente espresso tramite un numero intero positivo.

### 9.1 Memoria di massa

In informatica si distinguono due tipi di memoria: la *memoria principale*, costituita fondamentalmente da memoria RAM, memoria ROM, memoria Cache, e la *memoria secondaria* o **memoria di massa**, i cui maggiori rappresentanti sono gli hard disk, ma anche supporti rimovibili come dischi floppy, CD, DVD, nastri magnetici, memorie flash, ecc. Nell'architettura di von Neumann, quando ci si riferisce alla memoria, s'intende la memoria principale, che è quella sulla quale il calcolatore carica i processi (programmi in esecuzione) ed i dati ad essi relativi, mentre si considera la memoria secondaria alla stregua di un dispositivo d'ingresso (input) e/o di uscita (output) dei dati.

La caratteristica principale della memoria di massa è la “non volatilità”, ovvero la possibilità di memorizzare permanentemente i dati (per questo

si parla anche di memoria di archiviazione). I tempi medi di accesso alla memoria principale sono dell'ordine delle centinaia di nanosecondi, contro i millisecondi delle memorie di archiviazione, che, quindi, necessitano di tempi di accesso maggiori di ben cinque ordini di grandezza.

I dati sono riuniti in entità omogenee dette file. Le memorie di massa sono gestite da un componente fondamentale dei sistemi operativi, il file system: ogni sistema operativo ne utilizza uno diverso. Fondamentalmente, l'organizzazione delle memorie di massa è gestita tramite strutture dati collegate che possono essere liste o, molto più frequentemente, B-Alberi, oppure tabelle di indirizzamento (dette TOC).

Nel caso di file system a TOC, la ricerca è piuttosto semplice e rapida, mentre nel caso di struttura collegata la ricerca può risultare onerosa.

La memorizzazione dei dati, nel caso di memorie non riscrivibili (per esempio CD-R) avviene progressivamente, riempiendo lo spazio in ordine progressivo di indirizzo. Per quanto riguarda le memorie riscrivibili (per esempio gli hard disk) il data-storage è eseguito fondamentalmente secondo tre possibili criteri:

- First-fit: si cerca la prima locazione di memoria abbastanza grande da ospitare il file da salvare.
- Best-fit: si cerca la più piccola locazione di memoria abbastanza grande da ospitare il file da salvare.
- Worst-fit: si cerca la più grande locazione di memoria disponibile.

Statisticamente si può osservare che i primi due metodi sono più efficienti del terzo.

La cancellazione dei dati (nei supporti che lo consentono, ovvero quelli riscrivibili) è spesso intesa dai sistemi operativi come la semplice rimozione del record relativo al file in questione dalla TOC o dall'albero B+, il che rende possibile tramite apposite utility il ripristino di dati cancellati, in quanto, questi risiedono di fatto in memoria, finché la necessità di spazio sul quale memorizzare nuovi dati spingerà il sistema operativo a scrivere nella locazione che l'utente, cancellando il vecchio file, ha di fatto indicato come vuota.

Tutti i file system commerciali utilizzano la tecnica del file paging: un supporto di memoria di massa è suddiviso in pagine o cluster (la cui misura in genere varia da 512 byte a 4 KB), per ottimizzare le operazioni di lettura e scrittura. Questo, però, può causare uno spreco di memoria, soprattutto quando le pagine sono di grandi dimensioni. In questo caso, si parla di *frammentazione interna*. Invece, per *frammentazione esterna* s'intende il fenomeno per il quale tra due file si viene a creare uno spazio vuoto troppo

piccolo per memorizzarvi un altro file: questo fenomeno tipico dei file system con cluster troppo piccoli.

### 9.1.1 Disco magnetico

I **dischi magnetici** sono il mezzo fondamentale di memoria secondaria dei moderni sistemi di calcolo. Concettualmente, i dischi sono relativamente semplici: i **piatti** dei dischi hanno una forma piana e rotonda (come quella dei CD), con un diametro che comunemente varia tra 1,8 e 5,25 pollici, e le due superfici ricoperte di materiale magnetico; le informazioni si memorizzano registrandole magneticamente sui piatti.

Le testine di lettura e scrittura sono sospese su ciascuna superficie d'ogni piatto e sono attaccate al **braccio del disco** che le muove in blocco. La superficie di un piatto è divisa logicamente in **tracce** circolari a loro volta suddivise in **settori**; l'insieme delle tracce corrispondenti ad una posizione del braccio (equidistanti dal centro dei piatti) costituisce un **cilindro**. In un'unità a disco possono esservi migliaia di cilindri concentrici ed ogni traccia può contenere centinaia di settori. La capacità di memorizzazione di una comune unità a disco è dell'ordine delle decine di Gigabyte.

Quando un disco è in funzione, un motore lo fa ruotare ad alta velocità; la maggior parte dei dischi ruota a velocità comprese tra 60 e 200 giri/secondo. L'efficienza di un disco è caratterizzata da due valori: la **velocità di trasferimento**, cioè la velocità con cui i dati fluiscono dall'unità a disco al calcolatore, ed il **tempo di posizionamento**, talvolta detto tempo d'accesso diretto, che consiste nel tempo necessario a spostare il braccio del disco in corrispondenza del cilindro desiderato, detto **tempo di ricerca** (*seek time*), e nel tempo necessario affinché il settore desiderato si porti, tramite la rotazione del disco, sotto la testina, detto **latenza di rotazione**. In genere, i dischi possono trasferire parecchi Megabyte di dati al secondo e hanno un tempo di ricerca ed una latenza di rotazione di diversi millisecondi.

Poiché le testine di un disco sono sospese su un cuscino d'aria sottilissimo (dell'ordine dei micron), esiste il pericolo che la testina urti la superficie del disco; in tal caso, nonostante i piatti del disco siano ricoperti da un sottile strato protettivo, la testina può danneggiare la superficie magnetica. Tale incidente, detto **crollo della testina**, di solito non può essere riparato e comporta la sostituzione dell'intera unità a disco.

Un disco può essere **rimovibile**: ciò permette che diversi dischi siano montati secondo le necessità. I dischi magnetici rimovibili consistono generalmente in un piatto contenuto in un involucro di materiale plastico, che serve ad evitare danni che possono verificarsi quando il disco non è inserito nella propria unità.

Un' unità a disco è connessa ad un calcolatore attraverso un insieme di fili detto **bus di I/O**; esistono diversi tipi di tale bus, tra i quali ATA, ATA seriale, USB, ecc. Il trasferimento dei dati in un bus è eseguito da speciali unità di elaborazione dette **controllori**: gli **adattatori** o **controllori di macchina** sono i controllori posti all' estremità relativa al calcolatore del bus; i **controllori dei dischi** sono incorporati in ciascuna unità a disco. Per eseguire un' operazione di I/O il calcolatore inserisce un comando nell' adattatore, generalmente mediante porte I/O mappate in memoria; l' adattatore invia il comando al controllore del disco, che agisce sugli elementi elettromeccanici dell' unità a disco per portare a termine il comando. I controllori dei dischi di solito hanno una cache incorporata: il trasferimento dei dati nell' unità a disco avviene tra le cache e la superficie del disco.

### 9.1.2 Disco ottico

Il **disco ottico** è una tipologia di supporto di memoria. È costituito da un disco piatto e sottile (in genere di policarbonato) trasparente. Al suo interno è inserito un sottile foglio metallico (in genere di alluminio) su cui sono registrate e lette le informazioni tramite un raggio laser.

L' informazione su un disco ottico è memorizzata sequenzialmente in una traccia continua a spirale, dalla traccia più interna a quella più esterna. I dischi ottici sono particolarmente resistenti agli agenti atmosferici, e hanno una grande capacità di memorizzazione.

Esistono diverse tipologie di disco ottico:

- LaserDisc
- CD
  - 700 MB
- DVD
  - Singolo lato, singolo strato (4,7 GB)
  - Singolo lato, doppio strato (8,5 GB)
  - Doppio lato, singolo strato (9,4 GB)
  - Doppio lato, Doppio strato (17 GB)
- DualDisc
- HD DVD



- Blu-ray Disc: è il supporto ottico proposto dalla Sony agli inizi del 2002 come evoluzione del DVD per la televisione ad alta definizione. Grazie all' utilizzo di un laser a luce blu, riesce a contenere fino a 54 GB di dati.
- Digital Multilayer Disk
- Enhanced Versatile Disc
- Fluorescent Multilayer Disc
- Holographic Versatile Disc
- Universal Media Disc
- Versatile Multilayer Disc

### 9.1.3 Memoria flash

La **memoria flash**, anche chiamata **flash memory**, è una tipologia di memoria non volatile, che per le sue prestazioni può anche essere usata come memoria a lettura-scrittura. Quando viene utilizzata come ROM è anche chiamata flash ROM.

La memoria flash, trattandosi di memoria a stato solido, non presenta alcuna parte mobile: quindi, è piuttosto resistente alle sollecitazioni e agli urti. Inoltre, è estremamente leggera e di dimensioni ridotte.

La memoria flash è particolarmente indicata per la trasportabilità, proprio in virtù del fatto che non richiede alimentazione elettrica per mantenere i dati e che occupa poco spazio. È, infatti, molto usata nelle fotocamere digitali, nei lettori di musica portatili, nei cellulari, nei pendrive (chiavette), nei palmari, nei moderni computer portatili e in molti altri dispositivi che richiedono un' elevata portabilità ed una buona capacità di memoria per il salvataggio dei dati.

#### Teoria fisica

Le informazioni sono registrate in un array di transistor chiamati “celle”, ognuna delle quali conserva il valore di un bit. Le nuove flash utilizzano delle celle multilivello che permettono di registrare il valore di più bit attraverso un solo transistor.

Se consideriamo le memorie NOR, le prime ad essere state prodotte, ogni cella è simile ad un MOSFET (il transistor metallo-ossido-semiconduttore a effetto di campo; è una tipologia di transistor usata principalmente nei

dispositivi digitali grazie al basso consumo di potenza dovuto alla ridotta dispersione di calore), ma con due gate anziché uno soltanto. Uno è il solito *CG* (*Control Gate*) mentre l'altro è chiamato *Floating Gate* (*FG*), che risulta essere completamente isolato da uno strato di ossido. Il Floating Gate si trova tra il CG ed il substrato. Siccome il FG è isolato, ogni elettrone che gli passa sopra è intrappolato permettendo così di conservare il bit di informazione. Quando gli elettroni si attestano sul FG, essi modificano il campo elettrico proveniente dal CG e ciò influenza la tensione di soglia ( $V_t$ ) della cella. Durante un'operazione di lettura, applicando una tensione sul CG, la corrente fluisce o meno a seconda della  $V_t$  della cella che è controllata dal numero di elettroni presenti sul FG. Questa presenza o assenza di corrente è tradotta in 0 o 1, riproducendo il valore del bit memorizzato. L'inserimento degli elettroni sfrutta un fenomeno chiamato *iniezione a valanga*, mentre la loro rimozione dal Floating Gate sfrutta l'*effetto tunnel*.

Le memorie flash sono a tutti gli effetti i dispositivi quantistici più diffusi in assoluto. Le memorie flash secondo i postulati della fisica classica non dovrebbero funzionare dato che il Floating Gate non è collegato fisicamente a nessuna linea, e quindi gli elettroni non dovrebbero scorrere. Grazie all'effetto tunnel, invece, applicando un'adeguata differenza di potenziale, alcuni elettroni saltano, raggiungendo il Floating Gate e rimanendovi bloccati.

Per espandere la capacità delle memorie sono state sviluppate le celle a multilivello, dove non si controlla soltanto l'assenza o presenza di corrente, ma si precisa anche il suo valore che dipende naturalmente dal numero di elettroni intrappolati dal FG: in questo modo, si possono memorizzare più bit.

Per programmare il valore di una cella si avvia un flusso di elettroni dal *source* al *drain*; poi, una tensione molto elevata è imposta sul CG che genera un campo elettrico sufficientemente elevato affinché gli elettroni siano intrappolati nel FG. Questo processo è chiamato *hot-electron injection*. È da qui che prende origine la denominazione di flash, perché il CG non fa nient'altro che "flashare" il FG attraverso una tensione molto elevata.

Per la cancellazione è applicata una differenza di tensione tra CG e source per far sì che gli elettroni siano estratti dal FG attraverso un processo chiamato *Fowler-Nordheim tunneling*. Le memorie NOR moderne raggruppano le celle in segmenti chiamati blocchi o settori in maniera che le operazioni di cancellazione avvengano contemporaneamente su tutte le celle appartenenti allo stesso segmento.

Le memorie flash offrono un accesso random in lettura e scrittura, ma non nelle operazioni di modifica e cancellazione. È questa una delle ragioni per la quale, al momento, non è possibile sostituire le RAM dei PC con queste tipologie che permetterebbero di non perdere le informazioni nel caso in cui

## 9.2 Redundant Array of Inexpensive/Indipendent Disks (RAID) 73

ci sia un' interruzione improvvisa dell' alimentazione. Altre due ragioni sono che l' accesso sia in lettura che in scrittura alle memorie flash richiede molto più tempo rispetto ad una RAM attuale, ed il numero di scritture che una memoria flash può supportare non è illimitato seppur molto alto.

### 9.1.4 Il disco a stato solido

Il **drive a stato solido** (in sigla **SSD**) è una tipologia di dispositivo di memoria di massa che utilizza memoria a stato solido (in particolare memoria flash) per l' archiviazione dei dati.

I dischi a stato solido si basano sulla memoria flash di tipo NAND per l' immagazzinamento dei dati, ovvero sfruttano l' effetto tunnel per modificare lo stato elettronico di celle di transistor; per questo, essi non richiedono parti meccaniche e magnetiche, portando notevoli vantaggi per la sicurezza dei dati. L' uso della parola 'disco' è principalmente dovuta al fatto che gli SSD dovranno, per un certo periodo, affiancare i dischi rigidi tradizionali.

#### Vantaggi e svantaggi dei dischi a stato solido

La totale assenza di parti meccaniche in movimento porta diversi vantaggi, di cui i principali sono: rumorosità assente, minore possibilità di rottura, minori consumi durante le operazioni di lettura e scrittura, tempo di accesso ridotto, maggiore resistenza agli urti, minore produzione di calore.

A fronte di una maggiore resistenza agli urti ed ad un minor consumo, i dischi a stato solido hanno due svantaggi principali: un maggiore costo per bit, che fa sì che i dischi a stato solido abbiano un costo circa dodici volte superiore a un omologo disco rigido, e una possibile minore durata del disco, a causa del limite di riscritture delle memorie flash. I dispositivi attuali dichiarano un numero massimo di riscritture consecutive dello stesso bit che va da 10000 a 1000000 di cicli, a seconda del modello e degli utilizzi ipotizzati.

## 9.2 Redundant Array of Inexpensive/Indipendent Disks (RAID)

Un **Redundant Array of Independent Disks** (insieme ridondante di dischi indipendenti, **RAID**) è un sistema informatico che usa un insieme di dischi rigidi per condividere o replicare le informazioni. I benefici del RAID sono di aumentare l' integrità dei dati, la tolleranza ai guasti e/o le prestazioni, rispetto all' uso di un disco singolo. Nella sua implementazione originaria (nella quale l' acronimo era l' abbreviazione di *Redundant Array*

of *Inexpensive Disks*, cioè “insieme ridondante di dischi economici”), il fattore chiave era l’abilità di combinare parecchi dischi a basso costo e obsoleti in modo da rendere il sistema complessivamente migliore di un disco di ultima generazione per capacità, affidabilità e/o velocità.

Nel suo livello pi semplice, il sistema RAID permette di combinare un insieme di dischi in una sola unità logica. In questo modo il sistema operativo, invece di vedere differenti dischi, ne vede solamente uno. Il RAID è tipicamente usato nei server, e di solito è implementato con dischi di identica capacità.

Le specifiche originali suggerivano un diverso numero di “livelli di RAID”, o combinazioni di dischi. Ogni combinazione aveva dei vantaggi e degli svantaggi. Con il passare degli anni, sono nate diverse implementazioni del concetto di RAID. La maggior parte differiscono sostanzialmente nell’implementazione dei livelli RAID ideati inizialmente.

### 9.2.1 Implementazioni RAID: il Data Striping

I dati sono partizionati in segmenti di uguale lunghezza e scritti su dischi differenti. La grandezza della partizione si chiama **unità di striping**. Le partizioni sono solitamente distribuite fra i dischi usando un *algoritmo round robin* (particolare algoritmo di tipo preemptive che esegue i processi nell’ordine d’arrivo, ed esegue la prelazione del processo in esecuzione, ponendolo alla fine della coda dei processi in attesa, qualora l’esecuzione duri più del quanto di tempo stabilito, e facendo proseguire l’esecuzione al successivo processo in attesa). Quando la dimensione dei dati richiesti è superiore ad un’unità di striping, tali dati sono distribuiti su più dischi e possono essere letti in parallelo aumentando le prestazioni (alcuni sistemi di RAID implementano questa funzionalità). Ad esempio, se abbiamo un’unità di striping di 1 bit e abbiamo un array di  $N$  dischi, le sequenze di dati lunghe  $N$  bit o più necessitano di tutti i dischi.

### 9.2.2 Livelli RAID standard

Il **sistema RAID-0** (*striping*) divide i dati equamente tra due o più dischi con nessuna informazione di parità o ridondanza (operazione detta di striping). RAID-0 è usato generalmente per aumentare le prestazioni di un sistema, anche se è molto utile per creare un piccolo numero di grandi dischi virtuali da un grande numero di piccoli dischi fisici. Sebbene il RAID-0 non sia indicato tra i livelli RAID originari, in un sistema ideale di tipo RAID-0 le operazioni di I/O si dividerebbero in blocchi di dimensioni uguali e si applicherebbero equamente su tutti i dischi. Le implementazioni di sistemi

## 9.2 Redundant Array of Inexpensive/Indipendent Disks (RAID) 75

---

RAID-0 su più di due dischi sono possibili, ma l' affidabilità di un dato sistema RAID-0 è uguale all' affidabilità media dei dischi diviso per il numero di dischi presenti. Quindi, l' affidabilità, misurata come tempo medio tra due guasti, è inversamente proporzionale al numero degli elementi; cioè, un sistema di due dischi è affidabile la metà di un disco solo. La ragione per la quale questo succede è che il file system è diviso tra tutti i dischi. Quando un drive si guasta, il file system non può gestire una perdita di dati così grande visto che i dati sono divisi tra tutti i dischi. I dati possono essere spesso recuperati con qualche strumento, anche se saranno sicuramente incompleti e corrotti. Siccome non c' è ridondanza, i dati sono condivisi tra i dischi, ed i dischi non possono essere sostituiti visto che sono tutti dipendenti tra di loro. Questo tipo di progettazione non è, in realtà, un vero e proprio RAID, in quanto non c' è alcuna ridondanza.

Il **sistema RAID-1** (*mirroring*) crea una copia esatta di tutti i dati su due o più dischi. È utile nei casi in cui la ridondanza è più importante che usare tutti i dischi alla loro massima capacità: infatti, il sistema può avere una capacità massima pari a quella del disco più piccolo. In un sistema ideale, formato da due dischi, l' affidabilità aumenta di un fattore due rispetto al sistema a disco singolo, ma è possibile avere più di una copia dei dischi. Poiché ogni disco può essere gestito autonomamente nel caso l' altro si guasti, l' affidabilità aumenta linearmente al numero di dischi presenti. RAID-1 aumenta anche le prestazioni in lettura, visto che molte implementazioni possono leggere da un disco mentre l' altro è occupato.

Il **sistema RAID-2** divide i dati al livello di bit (invece che di blocco) ed usa un codice di Hamming per la correzione d' errore che permette di correggere errori su singoli bit e di rilevare errori doppi. Il RAID-2 è sostanzialmente un RAID-1 maggiormente affidabile. Questi dischi sono sincronizzati dal controllore, in modo tale che la testina di ciascun disco sia nella stessa posizione in ogni disco. Questo sistema si rivela molto efficiente in ambienti in cui si verificano numerosi errori di lettura o scrittura, ma in ambienti più prestanti, data l' elevata affidabilità dei dischi, il RAID-2 non è utilizzato.

Il **sistema RAID-3** divide i dati al livello di byte con un disco dedicato alla parità. Il RAID-3 è estremamente raro nella pratica. Uno degli effetti collaterali del RAID-3 è che non può eseguire richieste multiple simultaneamente. Questo perché ogni singolo blocco di dati ha la propria definizione diffusa tra tutti i dischi del RAID e risiederà nella stessa locazione, così ogni operazione di I/O richiede di usare tutti i dischi.

Il **sistema RAID-4** divide i dati a livello di blocchi con un disco dedicato alla parità. Il RAID-4 assomiglia molto al RAID-3 con l' eccezione che divide i dati al livello di blocchi invece che al livello di byte. Questo permette ad ogni

disco appartenente al sistema di operare in maniera indipendente quando è richiesto un singolo blocco. Se il controllore del disco lo permette, un sistema RAID-4 può servire diverse richieste di lettura contemporaneamente.

Il **sistema RAID-5** divide i dati a livello di blocco con i dati di parità distribuiti tra tutti i dischi appartenenti al RAID. Virtualmente, ogni sistema di storage permette il RAID-5 tra le sue opzioni. Ogni volta che un blocco di dati (chiamato delle volte *chunk*) deve essere scritto nel sistema di dischi, un blocco di parità è generato all'interno della stripe. (Un blocco spesso composto da molti settori di disco, delle volte anche 256. Una serie di blocchi consecutivi chiamato *stripe*). Se un altro blocco, o qualche porzione dello stesso blocco, è scritta nella stessa stripe, il blocco di parità è ricalcolato e riscritto. Il disco usato per memorizzare le parità è modificato tra una stripe e la successiva; in questo modo si riescono a distribuire i blocchi di parità. Il blocco di parità non è letto quando si leggono i dati da disco, visto che sarebbero un sovraccarico non necessario e diminuirebbe le prestazioni. Il blocco di parità è letto, invece, quando la lettura di un settore dà un errore CRC. In questo caso, il settore nella stessa posizione relativa nei blocchi di dati rimanenti della stripe, insieme al blocco di parità, sono usati per ricostruire il blocco mancante. In questo modo, l'errore di CRC è nascosto al computer chiamante. Nella stessa maniera, se un disco dovesse danneggiarsi all'interno del sistema, i blocchi di parità dei dischi rimanenti sono combinati matematicamente "al volo" con i blocchi dati rimasti per ricostruire i dati del disco guasto. Questa procedura viene chiamata *Interim Data Recovery Mode*. Il computer principale non è messo al corrente che un disco si è danneggiato. Le letture e scritture verso il sistema di dischi avvengono tranquillamente come prima, sebbene con qualche calo di prestazioni. In un sistema RAID-5, che ha un solo blocco di parità per stripe, la rottura di un secondo disco comporta la perdita di tutti i dati presenti nel sistema.

### 9.3 Storage Area Network (SAN)

Una **Storage Area Network (SAN)** è un disco di rete ad alta velocità (generalmente Gigabit/sec) di dispositivi di memorizzazione di massa condivisi; un dispositivo di memorizzazione di massa (*storage*) è una macchina che può essere composta da uno o più dischi per contenere dati.

Un'architettura SAN lavora in modo che tutti i dispositivi di memorizzazione siano disponibili a qualsiasi server della rete LAN o MAN di cui la SAN in questione fa parte; una SAN può essere anche condivisa fra più reti interconnesse, anche di natura diversa: in tal caso uno dei server locali fa da ponte fra i dati memorizzati e gli utenti finali. Il vantaggio di un'architettura

di questo tipo è che tutta la potenza di calcolo dei server è utilizzata per le applicazioni, in quanto i dati non risiedono direttamente in alcuno di questi.

In una rete SAN, le periferiche di storage sono connesse ai server attraverso una topologia costituita essenzialmente da dei canali (solitamente in fibra ottica) e da hub, switch router e bridge che in teoria consente la coesistenza di sistemi e dispositivi di storage di natura eterogenea, sebbene nella pratica gli aspetti di interoperabilità costringano ancora a creare reti SAN omogenee. Questo permette di evitare un sovraccarico della rete dato che tutto il traffico è gestito da questi dispositivi.

Normalmente, una SAN utilizza dischi collegati con una struttura di tipo RAID per migliorare le prestazioni e aumentare l' affidabilità del sistema.

Le aziende devono poter accedere ai dati in modo rapido e sicuro; quindi, la filosofia dell' architettura SAN è quella di poter integrare tutte le caratteristiche dei tradizionali sistemi di memorizzazione: alte prestazioni, alta disponibilità, scalabilità e facilità di gestione. Tutto questo con le caratteristiche di connettività ed accesso distribuito del *network computing*, attraverso un' architettura di rete dedicata alla gestione ed archiviazione dei dati, in grado di non sovraccaricare i server nelle operazioni di scrittura e lettura dei dati, da e verso lo storage.

Le reti SAN forniscono una serie di vantaggi rispetto ai dispositivi di storage connessi direttamente ai server (Direct Attached Storage). Offrono una connettività any-to-any tra server e dispositivi di storage, aprendo in tal modo la strada al trasferimento diretto di dati tra periferiche di memorizzazione (dischi o tape), con conseguenti indubbi miglioramenti dell' efficienza dello spostamento dei dati e di processi, quali il backup o la replica dei dati.





# Capitolo 10

## File e File System

### 10.1 Concetto di file

Un *file* è una raccolta di informazioni, correlate e registrate in memoria secondaria, cui è stato assegnato un nome. Dal punto di vista dell'utente, un file è la più piccola porzione di memoria secondaria logica; cioè, i dati si possono scrivere in memoria secondaria solo all'interno di un file. Comunque, i file rappresentano programmi, sia sorgente sia oggetto, e dati. I file di dati possono essere numerici, alfabetici, alfanumerici o binari, e non possedere un formato specifico, come i file di testo; oppure essere rigidamente formattati. In genere, un file è formato da una sequenza di bit, byte, righe o record il cui significato è definito dal creatore e dall'utente del file stesso.

Le informazioni contenute in un file sono definite dal suo creatore e possono essere di molti tipi: programmi sorgente, programmi oggetto, dati numerici, testo, dati contabili, immagini, registrazioni sonore, e così via. Un file ha una struttura definita secondo il tipo: un file di *testo* è formato da una sequenza di caratteri organizzati in righe e pagine; un file *sorgente* è formato da una sequenza di procedure e funzioni, ciascuna delle quali è a sua volta organizzata in dichiarazioni seguite da istruzioni eseguibili; un file *oggetto* è formato da una sequenza di byte, organizzati in blocchi, comprensibili al modulo di collegamento del sistema; un file *eseguibile* consiste di una serie di sezioni di codice che il calcolatore può caricare in memoria ed eseguire.

#### 10.1.1 Attributi del file

Un file ha attributi che possono variare secondo il sistema operativo, ma che tipicamente comprendono i seguenti:

- Nome

- **Identificazione:** si tratta di un' etichetta unica, di solito un numero, che identifica il file all' interno del File System; è il nome impiegato dal sistema per il file.
- **Tipo**
- **Locazione:** si tratta di un puntatore al dispositivo ed alla locazione del file in tale dispositivo.
- **Dimensione:** si tratta della dimensione corrente del file (in byte, parole o blocchi) ed, eventualmente, della massima dimensione consentita.
- **Protezione:** le informazioni di controllo degli accessi controllano chi può leggere, scrivere o far eseguire il file.
- **Ora, Data ed Identificazione dell' utente:** queste informazioni possono essere relative alla creazione, l' ultima modifica e l' ultimo uso. Questi dati possono essere utili ai fini della protezione e per controllarne l' uso.

Le informazioni sui file sono conservate nella struttura della directory, che risiede a sua volta in memoria secondaria. Di solito, un elemento di directory consiste di un nome di file e di un identificatore unico, che a sua volta individua gli altri attributi del file.

### 10.1.2 Operazioni sui file

Un file è un **tipo di dato astratto**. Per definire adeguatamente un file è necessario considerare le operazioni che si possono eseguire su di esso. Il sistema operativo può offrire chiamate di sistema per creare, scrivere, leggere, spostare, cancellare e troncare un file.

- **Creazione di un file.** Per creare un file è necessario compiere due passaggi. In primo luogo si deve trovare lo spazio per il file nel File System. Secondariamente, per il file si deve creare un nuovo elemento nella directory in cui registrare il nome del file, la sua posizione nel File System ed, eventualmente, altre informazioni.
- **Scrittura di un file.** Per scrivere un file è indispensabile una chiamata di sistema che specifichi il nome del file e le informazioni che si vogliono scrivere. Dato il nome del file, il sistema cerca la sua posizione nella directory. Il File System deve mantenere un puntatore di *scrittura* alla locazione nel file in cui deve avvenire l' operazione di scrittura

successiva. Il puntatore si deve aggiornare ogni volta che si esegue una scrittura.

- **Lettura di un file.** Per leggere da un file è necessario una chiamata di sistema che specifichi il nome del file e la posizione in memoria dove collocare il successivo blocco del file. Anche in questo caso si cerca l'elemento corrispondente nella directory, ed il sistema deve mantenere un puntatore di *lettura* alla locazione nel file in cui deve avvenire la successiva operazione di lettura. Una volta completata la lettura, si aggiorna il puntatore. Di solito, un processo legge o scrive in un file, e la posizione corrente è mantenuta come un **puntatore alla posizione corrente del file**. Sia le operazioni di lettura sia quelle di scrittura adoperano lo stesso puntatore, risparmiando spazio e riducendo la complessità del sistema.
- **Riposizionamento in un file.** Si ricerca l'elemento appropriato nella directory e si assegna un nuovo valore al puntatore alla posizione corrente nel file. Il riposizionamento non richiede alcuna operazione di I/O. Questa operazione è anche nota come *posizionamento* o *ricerca* (*seek*) nel file.
- **Cancellazione di un file.** Per cancellare un file si cerca l'elemento della directory associato al file designato, si rilascia lo spazio associato al file (in modo che possa essere adoperato per altri) e si elimina l'elemento della directory.
- **Troncamento di un file.** Si potrebbe voler cancellare il contenuto di un file, ma mantenere i suoi attributi. Invece di forzare gli utenti a cancellare il file e quindi a ricrearlo, questa funzione consente di mantenere immutati gli attributi (ad esclusione della lunghezza del file) pur azzerando la lunghezza del file e rilasciando lo spazio occupato.

Queste operazioni di base comprendono sicuramente l'insieme minimo delle operazioni richieste per i file. Altre operazioni comuni comprendono l'*aggiunta* (*appending*) di nuove informazioni alla fine di un file esistente e la *ridenominazione* di un file esistente. Queste operazioni primitive si possono combinare per compiere altre operazioni. Ad esempio, per creare una *copia* di un file o per copiare il file in un altro dispositivo di I/O, come una stampante o video, è sufficiente creare un nuovo file, leggere i dati dal file vecchio e scriverli nel nuovo. Sono inoltre necessarie operazioni che consentono ad un utente di leggere ed impostare i vari attributi di un file. Ad esempio, si

potrebbe dover ricorrere ad un'operazione che consenta all'utente di determinare lo spazio di un file, come la lunghezza, e che consenta di definire gli attributi di un file, come il proprietario.

## 10.2 Struttura del File System

I dischi costituiscono la maggior parte della memoria secondaria in cui si conserva il File System. Hanno due caratteristiche importanti che ne fanno un mezzo conveniente per la memorizzazione dei file:

- si possono scrivere localmente; si può leggere un blocco su disco, modificarlo e quindi scriverlo nella stessa posizione;
- è possibile accedere direttamente a qualsiasi blocco di informazione del disco, quindi risulta semplice accedere a qualsiasi file, sia in modo sequenziale sia in modo diretto, e passare da un file all'altro spostando le testine di lettura e scrittura ed attendendo la rotazione del disco.

Anziché trasferire un byte alla volta, per migliorare l'efficienza dell'I/O, i trasferimenti tra memoria centrale e dischi si eseguono per *blocchi*. Ciascun blocco è costituito da uno o più settori. Secondo l'unità a disco, la dimensione dei settori è compresa tra 32 byte e 4096 byte; di solito è pari a 512 byte.

Per fornire un efficiente e conveniente accesso al disco, il sistema operativo fa uso di uno o più **File System** che consentono di memorizzare, individuare e recuperare facilmente i dati. Un File System presenta due problemi di progettazione piuttosto diversi. Il primo riguarda la definizione dell'aspetto del File System agli occhi dell'utente. Questo compito implica la definizione di un file e dei suoi attributi, delle operazioni permesse su un file e della struttura della directory per l'organizzazione dei file. Il secondo riguarda la creazione di algoritmi e strutture dati che permettono di far corrispondere il File System logico ai dispositivi fisici di memoria secondaria.

Lo stesso File System è composto da molti livelli distinti. Ogni livello si serve delle funzioni dei livelli inferiori per crearne di nuove impiegate dai livelli superiori.

Il livello più basso, il *controllo dell'I/O*, costituito dal **driver dei dispositivi** e dai gestori dei segnali d'interruzione, si occupa del trasferimento delle informazioni tra memoria centrale e memoria secondaria. Di solito, un driver di dispositivo scrive specifiche configurazioni in bit in specifiche locazioni della memoria del controllore di I/O per indicare quali azioni il dispositivo di I/O debba compiere, ed in quali locazioni.

Il **File System di base** deve solo inviare dei generici comandi all' appropriato driver di dispositivo per leggere e scrivere blocchi fisici nel disco. Ogni blocco fisico si identifica con il suo indirizzo numerico nel disco.

Il **modulo di organizzazione dei file** è a conoscenza dei file e dei loro blocchi logici, così come dei blocchi fisici dei dischi. Conoscendo il tipo di allocazione dei file usato e la locazione dei file, può tradurre gli indirizzi dei blocchi logici, che il File System di base deve trasferire, negli indirizzi dei blocchi fisici. I blocchi logici di ciascun file sono numerati da 0 (o 1) a  $n$ ; i blocchi fisici contenenti tali dati, di solito, non corrispondono ai numeri dei blocchi logici; quindi, per individuare ciascun blocco è necessaria una traduzione. Il modulo di organizzazione dei file comprende anche il gestore dello spazio libero, che registra i blocchi non assegnati e li mette a disposizione del modulo di organizzazione dei file quando sono richiesti.

Infine, il **File System logico** gestisce i metadati; si tratta di tutte le strutture del File System, eccetto gli effettivi dati (il contenuto dei file). Il File System logico gestisce la struttura della directory per fornire al modulo di organizzazione dei file le informazioni di cui necessita, dato un nome simbolico di file. Mantiene le strutture di file tramite i **blocchi di controllo dei file** (*File Control Block*, FCB), contenenti informazioni sui file, come la proprietà, i permessi, e la posizione del contenuto. Il File System logico è responsabile anche della protezione e della sicurezza.

## 10.3 Realizzazione del File System

Per realizzare un File System si usano parecchie strutture dati, sia nei dischi sia in memoria. Queste strutture variano secondo il sistema operativo ed il File System, ma esistono dei principi generali. Nei dischi, il File System tiene informazioni su come eseguire l' avviamento di un sistema operativo memorizzato nei dischi stessi, il numero totale di blocchi, il numero e la locazione dei blocchi liberi, la struttura delle directory ed i singoli file.

Fra le strutture presenti nei dischi ci sono le seguenti:

- Il **blocco di controllo dell' avviamento** (*Boot Control Block*), contenente le informazioni necessarie al sistema per l' avviamento di un sistema operativo da quel volume; se il disco non contiene un sistema operativo, tale blocco è vuoto. Di solito è il primo blocco di un volume.
- I **blocchi di controllo dei volumi** (*Volume Control Block*); ciascuno di loro contiene i dettagli riguardanti il relativo volume (o partizione), come il numero e la dimensione dei blocchi nel disco, il contatore dei

blocchi liberi ed i relativi puntatori, il contatore dei blocchi di controllo dei file liberi ed i relativi puntatori.

- Le **strutture delle directory** (una per File System) usate per organizzare il file.
- I **blocchi di controllo di file** (FCB), contenenti molti dettagli dei file, compresi i permessi d'accesso ai relativi file, i proprietari, le dimensioni e le locazioni dei blocchi di dati.

Le informazioni tenute in memoria servono sia per la gestione del File System sia per migliorare le prestazioni attraverso l'uso di cache. I dati si caricano al momento del montaggio e si eliminano allo smontaggio. Le strutture contenenti queste informazioni comprendono:

- la tabella di montaggio interna alla memoria che contiene informazioni relative a ciascun volume montato;
- la struttura della directory, tenuta in memoria, contenente le informazioni relative a tutte le directory cui i processi hanno avuto accesso di recente (per le directory che costituiscono dei punti di montaggio, può essere presente un puntatore alla tabella dei volumi);
- la tabella generale dei file aperti, contenente una copia del blocco di controllo del file per ciascun file aperto, insieme con altre informazioni;
- la tabella dei file aperti per ciascun processo, contenente un puntatore all'appropriato elemento della tabella generale dei file aperti, insieme con altre informazioni.

Le applicazioni, per creare un nuovo file, eseguono una chiamata al File System logico, il quale conosce il formato della struttura della directory. Esso crea ed alloca un nuovo FCB. (In alternativa, nel caso dei File System che creano tutti i blocchi di controllo al momento della loro installazione, esso alloca semplicemente un blocco di controllo libero). Il sistema carica, quindi, la directory appropriata in memoria, la aggiorna con il nome del nuovo file e con il blocco di controllo associato, e le scrive nuovamente sul disco.

## 10.4 Metodi di allocazione

La natura ad accesso diretto dei dischi permette una certa flessibilità nella realizzazione dei file. In quasi tutti i casi, molti file si memorizzano nello stesso disco. Dunque, il problema principale consiste nell'allocare lo spazio

per questi file in modo che lo spazio nel disco sia usato efficientemente e l'accesso ai file sia rapido. Esistono tre modi principali per l'allocazione dello spazio di un disco; infatti, può essere contigua, concatenata o indicizzata. Ciascuno di questi metodi presenta vantaggi e svantaggi.

### 10.4.1 Allocazione contigua

Per usare il metodo di **allocazione contigua**, ogni file deve occupare un insieme di blocchi contigui del disco. Gli indirizzi del disco definiscono un ordinamento lineare nel disco stesso. Con questo ordinamento, l'accesso al blocco  $n + 1$ , dopo il blocco  $n$ , non chiede normalmente alcuno spostamento della testina. Il numero dei posizionamenti (*seek*) richiesti per accedere a file il cui spazio è allocato in modo contiguo è trascurabile, così come è trascurabile il tempo di ricerca (*seek time*), quando quest'ultimo è necessario.

L'allocazione contigua dello spazio per un file è definita dall'indirizzo del primo blocco (inteso come numero di blocco) e dalla lunghezza (espressa in blocchi). Se il file è lungo  $m$  blocchi e comincia dalla locazione  $n$ , allora occupa i blocchi  $n, n + 1, n + 2, \dots, n + m - 1$ . L'elemento di directory per ciascun file indica l'indirizzo del blocco d'inizio e la lunghezza dell'area assegnata per questo file.

Accedere ad un file il cui spazio è assegnato in modo contiguo è facile. Quando si usa un accesso sequenziale, il File System memorizza l'indirizzo dell'ultimo blocco cui è stato fatto riferimento e, se è necessario, legge il blocco successivo. Nel caso di un accesso diretto al blocco  $i$  di un file che comincia al blocco  $n$  si può accedere immediatamente al blocco  $n + i$ . Quindi, sia l'accesso sequenziale sia quello diretto si possono gestire con l'allocazione contigua.

L'allocazione contigua presenta, però, alcuni problemi; una difficoltà riguarda l'individuazione dello spazio per un nuovo file. La realizzazione del sistema di gestione dello spazio libero determina il modo in cui tale compito è eseguito. Si può usare ogni sistema di gestione, anche se alcuni sono più lenti di altri.

Il problema dell'allocazione contigua dello spazio dei dischi si può considerare un'applicazione particolare del problema generale dell'**allocazione dinamica della memoria**; il problema generale è, infatti, quello di soddisfare una richiesta di dimensione  $m$  data una lista di buchi liberi. I più comuni criteri di scelta di un buco libero da un insieme di buchi disponibili sono quelli del primo buco abbastanza grande (*first-fit*) e del più piccolo tra i buchi abbastanza grandi (*best-fit*). Questi due criteri sono più efficienti di quello di scelta del buco più grande (*worst-fit*) sia in termini di tempo sia d'uso della memoria.

Questi algoritmi soffrono della **frammentazione esterna**: assegnando e liberando lo spazio per i file, lo spazio libero dei dischi è frammentato in tanti piccoli pezzi. La frammentazione esterna si ha ogniqualvolta lo spazio libero è suddiviso in pezzi, e diviene un problema quando il più grande di tali pezzi contigui non è sufficiente a soddisfare una richiesta; la memoria è frammentata in tanti buchi, nessuno dei quali è abbastanza grande da contenere dati. Secondo la capacità dei dischi e la dimensione media dei file, la frammentazione esterna può essere un problema più o meno grave.

Un altro problema che riguarda l'allocazione contigua è la determinazione della quantità di spazio necessaria per un file. Quando si crea un file, occorre trovare ed allocare lo spazio di cui necessita. Esiste il problema di conoscere la dimensione del file da creare; in alcuni casi questa dimensione si può stabilire in modo abbastanza semplice, ad esempio quando si copia un file semplice; in generale, tuttavia, non è facile stimare la dimensione di un file che deve contenere dati emessi da un programma.

Se un file riceve poco spazio, può essere impossibile estenderlo: soprattutto nel caso in cui si adoperi il criterio di allocazione del più piccolo tra i buchi abbastanza grandi, lo spazio oltre le due estremità del file può essere già in uso, quindi non è possibile ampliare il file in modo contiguo. Esistono allora due possibilità. La prima è che il programma utente si possa terminare con un idoneo messaggio d'errore. L'utente deve allora allocare più spazio ed eseguire di nuovo il programma. Queste esecuzioni ripetute possono essere onerose; per prevenire tale circostanza, normalmente l'utente sovrastima la quantità di spazio necessaria, sprecandone parecchio. L'altra possibilità consiste nel trovare un buco più grande, copiare il contenuto del file nel nuovo spazio e rilasciare lo spazio precedente. Queste operazioni si possono ripetere finché esiste spazio, anche se ciò può far perdere tempo.

Anche se si conosce in anticipo la quantità di spazio necessaria per un file, l'allocazione preventiva può in ogni modo essere inefficiente. Ad un file che cresce lentamente in un periodo di tempo lungo si deve allocare spazio sufficiente per la sua dimensione finale, anche se molto di quello spazio può rimanere inutilizzato per molto tempo. Il file ha perchiò un'estesa frammentazione interna.

Per ridurre al minimo questi inconvenienti, alcuni sistemi operativi fanno uso di uno schema di allocazione contigua modificato: inizialmente si assegna una porzione di spazio contiguo, e se questo non è abbastanza grande si aggiunge un'altra porzione di spazio, un'**estensione**. La locazione dei blocchi dei file si registra come una locazione ed un numero dei blocchi, insieme con l'indirizzo del primo blocco dell'estensione seguente.



### 10.4.2 Allocazione concatenata

L' **allocazione concatenata** risolve tutti i problemi dell' allocazione contigua. Con questo tipo di allocazione, infatti, ogni file è composto da una lista concatenata di blocchi del disco i quali possono essere sparsi in qualsiasi punto del disco stesso. La directory contiene un puntatore al primo ed all' ultimo blocco del file. Ogni blocco contiene un puntatore al blocco successivo.

Per creare un file si crea semplicemente un nuovo elemento nella directory. Con l' allocazione concatenata, ogni elemento della directory ha un puntatore al primo blocco del file. Questo puntatore s' inizializza con il valore del puntatore di fine lista, ad indicare un file vuoto. Un' operazione di scrittura nel file determina la ricerca di un blocco libero attraverso il sistema di gestione dello spazio libero, la scrittura in tale blocco, e la concatenazione di tale blocco alla fine del file. Per leggere un file occorre semplicemente leggere i blocchi seguendo i puntatori da un blocco all' altro. Con l' allocazione concatenata non esiste frammentazione esterna e per soddisfare una richiesta si può usare qualsiasi blocco libero della lista. Inoltre, non è necessario dichiarare la dimensione di un file al momento della sua creazione. Un file può continuare a crescere finché sono disponibili blocchi liberi, di conseguenza non è mai necessario compattare lo spazio del disco.

L' allocazione concatenata presenta comunque alcuni svantaggi. Il problema principale riguarda il fatto che può essere usata in modo efficiente solo per i file ad accesso sequenziale. Per trovare l'  $i$ -esimo blocco di un file occorre partire dell' inizio del file e seguire i puntatori finché non si raggiunge l'  $i$ -esimo blocco. Ogni accesso ad un puntatore implica una lettura del disco, e talvolta un posizionamento della testina. Di conseguenza, per file il cui spazio è assegnato in modo concatenato, la funzione d' accesso diretto è inefficiente.

Un altro svantaggio dell' allocazione concatenata riguarda lo spazio richiesto per i puntatori.

La soluzione più comune a questo problema consiste nel riunire un certo numero di blocchi contigui in **cluster** (gruppi di blocchi), e nell' allocare i cluster anziché i blocchi. Così i puntatori usano una quantità di spazio di disco che si riduce in modo proporzionale al numero di cluster. Questo metodo permette che la corrispondenza tra blocchi logici e blocchi fisici rimanga semplice, ma migliora la produttività del disco: si hanno meno posizionamenti della testina del disco e diminuisce lo spazio necessario per l' allocazione dei blocchi e la gestione della lista dei blocchi liberi. Il costo di questo metodo è dato da un incremento della frammentazione interna, poiché se un cluster è parzialmente pieno si spreca più spazio di quanto se ne sprecherebbe con un solo blocco parzialmente pieno.

Un altro problema riguarda l' affidabilità. Poiché i file sono tenuti insieme da puntatori sparsi per tutto il disco, s' immagina che cosa accadrebbe se un puntatore andasse perduto o danneggiato. Un errore di programmazione del sistema operativo oppure un errore di un' unità del disco potrebbero causare il prelevamento del puntatore errato. Questo errore, a sua volta, potrebbe causare il collegamento alla lista dei blocchi liberi oppure ad un altro file. Una soluzione parziale a tale problema consiste nell' usare liste doppiamente concatenate oppure nel memorizzare il nome del file ed il relativo numero di blocco in ogni blocco; questi schemi però sono ancora più onerosi per ogni file.

### 10.4.3 Allocazione indicizzata

L' allocazione concatenata risolve il problema della frammentazione esterna e quello della dichiarazione delle dimensioni dei file, entrambi presenti nell' allocazione contigua. Tuttavia, l' allocazione concatenata non è in grado di sostenere un efficiente accesso diretto, poiché i puntatori ai blocchi sono sparsi, con i blocchi stessi, per tutto il disco e si devono recuperare in ordine. L' **allocazione indicizzata** risolve questo problema, raggruppando tutti i puntatori in una sola locazione: il **blocco indice**.

Ogni file ha il proprio blocco indice: si tratta di un array d' indirizzi di blocchi del disco. L' *i*-esimo elemento del blocco indice punta all' *i*-esimo blocco del file. La directory contiene l' indirizzo del blocco indice. Per individuare e leggere l' *i*-esimo blocco occorre usare il puntatore che si trova nell' *i*-esimo elemento del blocco indice, per poi localizzare e leggere il blocco desiderato.

Una volta creato il file, tutti i puntatori del blocco indice sono inizializzati al valore del puntatore di fine lista. Quando si scrive l' *i*-esimo blocco per la prima volta, il gestore dei blocchi liberi fornisce un blocco; l' indirizzo di questo blocco è inserito nell' *i*-esimo elemento del blocco indice. Poiché ogni blocco libero del disco può soddisfare una richiesta di maggiore spazio, l' allocazione indicizzata consente l' accesso diretto senza soffrire di frammentazione esterna.

Lo spazio aggiuntivo richiesto dai puntatori del blocco indice è generalmente maggiore dello spazio aggiuntivo necessario per l' allocazione concatenata.

Gli schemi d' allocazione indicizzata soffrono di alcuni dei problemi di prestazioni dell' allocazione concatenata. In particolare, i blocchi indice si possono caricare in memoria, ma i blocchi dei dati possono essere sparsi per un intero volume.

### 10.4.4 Prestazioni

I metodi d' allocazione hanno diversi livelli di efficienza di memorizzazione e differenti tempi d' accesso ai blocchi di dati; entrambi i fattori sono importanti nella scelta del metodo o dei modi d' allocazione più adatti da impiegare in un sistema operativo.

Per qualsiasi tipo d' accesso, l' allocazione contigua richiede un solo accesso per ottenere un blocco. Poiché è facile tenere l' indirizzo iniziale del file in memoria, si può calcolare immediatamente l' indirizzo del disco dell'  $i$ -esimo blocco, oppure del blocco successivo, e leggerlo direttamente.

Con l' allocazione concatenata si può tenere in memoria anche l' indirizzo del blocco successivo e leggerlo direttamente. Questo metodo è valido per l' accesso sequenziale mentre, per quel che riguarda l' accesso diretto, un accesso all'  $i$ -esimo blocco può richiedere  $i$  letture del disco.

Da tutto ciò segue che alcuni sistemi gestiscono i file ad accesso diretto usando l' allocazione contigua, ed i file ad accesso sequenziale tramite l' allocazione concatenata. Per questi sistemi, il tipo d' accesso si deve dichiarare al momento della creazione del file. Un file creato per l' accesso sequenziale è un file concatenato e non si può usare per l' accesso diretto. Un file creato per l' accesso diretto è contiguo e consente entrambi i tipi di accesso, purché se ne dichiari la lunghezza massima al momento della sua creazione. In questo caso, il sistema operativo deve avere strutture dati idonee ad algoritmi capaci di gestire *entrambi* i metodi di allocazione.

L' allocazione indicizzata è più complessa. Se il blocco indice è già in memoria, l' accesso può essere diretto. Tuttavia, per tenere il blocco indice in memoria occorre una quantità di spazio considerevole. Se questo spazio di memoria non è disponibile, occorre leggere prima il blocco indice e quindi il blocco di dati desiderato. Per un indice a due livelli possono essere necessarie due letture del blocco indice. Se un file è estremamente grande, per compiere l' accesso ad un blocco che si trovi vicino alla fine del file, prima di leggere il blocco dei dati occorre leggere tutti i blocchi indice per seguire la catena dei puntatori. Quindi, le prestazioni dell' allocazione indicizzata dipendono dalla struttura dell' indice, dalla dimensione del file e dalla posizione del blocco desiderato.

Alcuni sistemi combinano l' allocazione contigua con l' allocazione indicizzata, usando quella contigua per i file piccoli (fino a tre o quattro blocchi) e passando automaticamente a quella indicizzata per i file grandi. Poiché generalmente i file sono piccoli, ed in questo caso l' allocazione contigua è efficiente, le prestazioni media possono risultare abbastanza buone.

## 10.5 Gestione dello spazio libero

Poiché la quantità di spazio libero dei dischi è limitata, è necessario riutilizzare lo spazio lasciato dai file cancellati per scrivere, se è possibile, nuovi file. Per tenere traccia dello spazio libero in un disco, il sistema conserva una **lista dello spazio libero**; vi sono registrati tutti gli spazi *liberi*, cioè non allocati ad alcun file o directory. Per creare un file occorre cercare nella lista dello spazio libero la quantità di spazio necessaria ed assegnarla al nuovo file, quindi rimuovere questo spazio dalla lista. Quando si cancella un file, si aggiungono alla lista dello spazio libero i blocchi di disco ad esso assegnati.

### 10.5.1 Vettore di bit

Spesso la lista dello spazio libero si realizza come una **mappa di bit**, o **vettore di bit**. Ogni blocco è rappresentato da un bit: se il blocco è libero, il bit è 1, se il blocco è assegnato, il bit è 0.

I vantaggi principali che derivano da questo metodo sono la sua relativa semplicità ed efficienza nel trovare il primo blocco libero o  $n$  blocchi liberi consecutivi nel disco; in effetti, molti calcolatori forniscono istruzioni di manipolazione di bit utilizzabili con efficacia a tale scopo.

Sfortunatamente, i vettori di bit sono efficaci solo se tutto il vettore è mantenuto in memoria centrale, ed è di tanto in tanto scritto in memoria secondaria allo scopo di consentire eventuali operazioni di ripristino; è possibile tenere il vettore in memoria centrale se i dischi sono piccoli.

### 10.5.2 Lista concatenata

Un altro metodo di gestione degli spazi liberi consiste nel collegarli tutti, tenere un puntatore al primo di questi in una speciale locazione del disco e caricarlo in memoria. Questo primo blocco contiene un puntatore al successivo blocco libero, e così via. Questo schema non è tuttavia efficiente; per attraversare la lista occorre leggere ogni blocco, e l'operazione richiede un notevole tempo di I/O. Fortunatamente, l'attraversamento della lista dello spazio libero non è un'operazione frequente. Di solito il sistema operativo ha semplicemente bisogno di un blocco libero perché possa assegnarlo ad un file, quindi si usa il primo blocco della lista.

### 10.5.3 Raggruppamento

Una possibile modifica del metodo della lista dello spazio libero prevede la memorizzazione degli indirizzi di  $n$  blocchi liberi nel primo di questi. I primi

$n - 1$  di questi blocchi sono effettivamente liberi; l'ultimo blocco contiene gli indirizzi di altri  $n$  blocchi liberi, e così via. L'importanza di questo metodo, diversamente dall'ordinaria lista concatenata, è data dalla possibilità di trovare rapidamente gli indirizzi di un gran numero di blocchi liberi.

#### 10.5.4 Conteggio

Un altro orientamento sfrutta il fatto che, generalmente, più blocchi contigui si possono allocare o liberare contemporaneamente, soprattutto quando lo spazio è allocato usando l'algoritmo di allocazione contigua o attraverso l'uso di cluster. Quindi, anziché tenere una lista di  $n$  indirizzi liberi, è sufficiente tenere l'indirizzo del primo blocco libero ed il numero  $n$  di blocchi liberi contigui che seguono il primo blocco. Ogni elemento della lista dello spazio libero è formato da un indirizzo del disco ed un contatore. Anche se ogni elemento richiede più spazio di quanto ne richieda un semplice indirizzo del disco, se il contatore è generalmente maggiore di 1 la lista complessiva è più corta.

### 10.6 Efficienza e prestazioni

Dopo aver descritto le opzioni di allocazione dei blocchi e di gestione delle directory, è possibile considerare i loro effetti sulle prestazioni e l'efficienza d'uso dei dischi. I dischi tendono, di solito, ad essere il principale collo di bottiglia per le prestazioni di un sistema, essendo i più lenti tra i componenti più rilevanti di un calcolatore.

L'uso efficiente di un disco dipende fortemente dagli algoritmi usati per l'allocazione del disco e la gestione delle directory.

Dopo aver scelto gli algoritmi fondamentali del File System le prestazioni possono essere migliorate in diversi modi. Alcuni controllori di unità a disco contengono una quantità di memoria locale sufficiente per la creazione di una **cache** interna al controllore sufficientemente grande da memorizzare un'intera traccia del disco alla volta. Eseguito il posizionamento della testina, si legge la traccia nella cache del controllore del disco a partire dal settore sotto cui si viene a trovare la testina (riducendo il tempo di latenza). Il controllore trasferisce quindi al sistema operativo tutte le richieste di settori. Quando i blocchi sono trasferiti dal controllore del disco alla memoria centrale, il sistema operativo ha la possibilità di inserirli in una propria cache nella memoria centrale.

Alcuni sistemi riservano una sezione separata della memoria centrale come **cache del disco**, dove tenere i blocchi in previsione di un loro utilizzo entro

breve tempo. Altri sistemi impiegano una **cache delle pagine** per i file; si tratta di una soluzione che impiega tecniche di memoria virtuale per la gestione dei dati dei file come pagine anziché come blocchi di File System; l'uso degli indirizzi virtuali è molto più efficiente dell'uso dei blocchi fisici di disco. Diversi sistemi usano le cache delle pagine, sia per le pagine relative ai processi sia per i dati dei file. Questo metodo è noto come **memoria virtuale uniformata**.

La cache delle pagine, il File System ed i driver del disco interagiscono in modi interessanti. Quando i dati sono scritti su un file del disco, le pagine sono memorizzate nella cache, che qui funge da buffer, mentre il drive del disco ordina la propria coda di dati in uscita in base all'indirizzo sul disco. Queste due azioni consentono al driver del disco di minimizzare gli spostamenti della testina del disco e fanno sì che la scrittura dei dati rispetti i tempi ottimali per la rotazione del disco.

## 10.7 Cache del disco

La **cache** è un insieme di dati che è memorizzato in una posizione temporanea, dalla quale possa essere recuperato velocemente su richiesta. Questo significa che non c'è nessuna certezza che i dati si trovino nella cache, ma che convenga comunque fare un tentativo per verificarne l'eventuale esistenza.

Una cache è associata ad una memoria principale, in cui risiedono i dati. Essa è tipicamente di capienza inferiore rispetto alla memoria principale, ma il suo utilizzo è più conveniente in termini di tempo di accesso e/o carico sul sistema.

Quando è necessario l'accesso ad un dato, questo dato è prima cercato nella cache. Se è presente e valido, è utilizzata la copia presente. Viceversa, è recuperato dalla memoria principale, e memorizzato nella cache, nel caso possa servire successivamente.

La memoria principale può essere qualcosa di semplice come un disco rigido, ma anche un complesso database distribuito, come il DNS o il web. In questi casi, la memoria principale può essere modificata senza passare dalla cache, il che comporta problemi di coerenza tra i dati "originali" e quelli nella cache.

In alcuni casi, è possibile validare i dati contenuti nella cache interrogando la memoria principale per verificare se sono ancora attuali. Questo è quello che fanno i *server proxy*: chiedono al server HTTP se la pagina che posseggono è stata modificata dopo la sua memorizzazione, e se non lo è evitano di trasferirla e la ripropongono direttamente al client. In altri casi, si utilizza un meccanismo di scadenza a tempo dei dati memorizzati, e fino a

quando un dato presente nella cache non è scaduto questo è utilizzato, anche se non corrisponde a quanto presente nella memoria principale. Questo è il meccanismo adottato dal DNS.

Una cache riduce il carico di richieste che deve essere smaltito dalla memoria principale, e dal collegamento tra questa e l'utente dei dati. Anche questo può contribuire a migliorare le prestazioni del sistema. Si pensi, per esempio, ad un server proxy utilizzato da molti utenti: quando un utente richiede una pagina che era già stata richiesta da un altro, il proxy potrà rispondere senza doversi collegare al sito originale, ed eviterà, così, di caricare sia il sito originale che la rete, migliorando le prestazioni del sistema anche per le richieste che devono essere inoltrate ai siti originali.

Una cache utilizza un algoritmo per decidere quali dati mantenere e quali scartare, e tiene conto delle pagine utilizzate più di recente, della contiguità delle pagine, o di diversi altri fattori. Una cache può indicizzare i dati memorizzati sulla base del loro indirizzo (un blocco di memoria o di dati su disco fisso) o del loro "nome" (cache associativa, ad esempio una pagina web o un nome DNS). In alcuni casi, la memoria cache supporta anche la modifica dei dati. Questo è di implementazione semplice se la cache è l'unico percorso di accesso alla memoria principale, come nel caso della cache della memoria RAM presente nei processori: la cache "accetta" una operazione di scrittura verso la RAM, permettendo al processore di proseguire l'elaborazione, presenta da subito al processore i dati aggiornati se questo ne chiede nuovamente la lettura, e si prende carico di scriverli sulla RAM prima di eliminare la pagina. In questo modo, se un dato in memoria è modificato frequentemente dal processore, è possibile mantenere le modifiche nella cache ed evitare continui trasferimenti verso la RAM.

### 10.7.1 Problematiche

Avere una memoria cache può complicare notevolmente la progettazione di un dispositivo elettronico o del software di gestione, dato che bisogna mantenere la coerenza tra le memorie cache e la memoria principale. Ciò si vede, specialmente, nelle macchine multiprocessore dove lo sviluppo dei processori e delle schede madri deve tenere conto dei potenziali problemi di coerenza delle cache e deve provvedere adeguati meccanismi che evitino corruzioni dei dati per problemi di mancata sincronizzazione tra cache e memoria principale. Il software di gestione di una memoria cache, inoltre, deve essere altamente sofisticato sia nella realizzazione sia nella strategia di recupero dei dati, in modo da evitare il cosiddetto *overhead* da recupero (ovvero, le risorse accessorie richieste in sovrappiù rispetto a quelle strettamente necessarie, per ottenere un determinato scopo in seguito all'introduzione di un metodo o di

un processo più evoluto o più generale). In sostanza, non deve mai capitare che la ricerca o la memorizzazione di un dato di cache memory impieghi risorse (in special modo il tempo) tali da appesantire troppo il recupero stesso che, paradossalmente, diventerebbe più conveniente da memoria RAM. In generale, gli algoritmi di gestione dei dati di cache sono legati a due concetti fondamentali:

- località del codice;
- eliminazione dalla cache dei dati usati meno recentemente.

Il primo aspetto è legato direttamente alla redazione di codice che interessi pagine di memoria ben definite in ogni fase di elaborazione: una volta terminata l'elaborazione di una certa parte di codice, le pagine di memoria utilizzate non dovrebbero più essere referenziate in modo tale da cambiare la località del codice, ma anche dei dati che si utilizzeranno da quel momento in poi. Il secondo aspetto prevede che un dato presente in cache ed appartenente ad una pagina di memoria referenziata, quando la località del codice era diversa, può essere eliminato dalla cache per far posto ai dati relativi alla corrente località del codice.



# Capitolo 11

## Alcuni formati di memorizzazione

### 11.1 Grafica raster

La **grafica raster**, o **grafica raster** è una tecnica utilizzata in computer grafica per descrivere un' immagine.

In computer grafica, il termine *raster* indica la griglia ortogonale di punti che costituisce un' immagine raster. Nella grafica raster, l' immagine è vista come una scacchiera ed ad ogni elemento della scacchiera, chiamato **pixel**, è associato uno specifico colore. Il colore può essere definito con due tecniche:

- se l' immagine contiene pochi colori (massimo 256) si crea un elenco dei colori da utilizzare, e nella scacchiera è inserito l' indice che punta allo specifico colore del pixel;
- nel caso si vogliano utilizzare molti più colori il singolo pixel non definisce più l' indice a una tavolozza di colori, ma definisce il colore direttamente.

Il colore è definito come un' unione delle componenti blu, rossa e verde. Questo non è l' unico modo di definire un colore: esistono molti modi che sono chiamati spazi di colore, ma nel caso delle immagini generate al computer il sistema RGB (RED Rosso, GREEN verde BLUE Blu) è il più diffuso dato che le schede grafiche lo utilizzano nativamente per generare il segnale da visualizzare con il monitor.

#### 11.1.1 Proprietà della grafica raster

La grafica bitmap è caratterizzata da due proprietà:

- risoluzione;
- profondità di colore.

La prima è determinata dal numero di pixel contenuti nell' unità di misura considerata ed è ottenuta moltiplicando il numero di pixel orizzontali per quello dei pixel verticali; si misura in PPI (*Points Per Inch*). La seconda è definita dalla memoria che si dedica ad ogni pixel, ovvero dal numero di bit dedicati ad ogni pixel per descrivere il colore; si misura in BPP (*Bit Per Pixel*). Maggiore è il numero di bit, maggiore è il numero di colori che è possibile descrivere.

La grafica bitmap non è vantaggiosa se l' utente necessita di apportare modifiche all' immagine, perché nel caso, ad esempio, di uno zoom, la risoluzione diventa bassissima e, quindi, la qualità dell' immagine peggiora notevolmente. I software grafici, per ridurre il problema, sono in grado di ripristinare la risoluzione inserendo nuovi pixel che sono calcolati facendo una interpolazione di punti: il processo inserisce, perciò, deliberatamente una quantità di informazioni presunte.

La grafica bitmap è, invece, ideale per rappresentare immagini della realtà, per modificare contrasti e luminosità di queste, per applicare filtri di colore.

### 11.1.2 Formati di immagini raster

I dati raster possono essere memorizzati attraverso tipologie di file che sfruttando algoritmi di compressione diversi, gravando in modo differente sul supporto di memorizzazione. I formati raster più comuni sono i seguenti:

#### Non compressi

Questi formati di file hanno richieste di elaborazione minima, non essendo necessari algoritmi di compressione (in fase di scrittura) e decompressione (in fase di lettura). Tuttavia, mancando di compressione, risultano particolarmente voluminosi, in termini di spazio occupato su disco (o altro dispositivo di memorizzazione), rispetto agli altri formati:

- raw
- bmp

### Con compressione lossless

Le immagini salvate con un algoritmo di compressione dati di tipo lossless occupano meno spazio nei dispositivi di memorizzazione, mantenendo inalterata tutta l'informazione originale:

- png
- tga
- tiff (sebbene questo sia l'uso più comune, questo formato permette diversi tipi di compressione)
- gif (per immagini fino a 256 colori)

### Con compressione lossy

Le immagini memorizzate con un algoritmo di compressione lossy, subiscono una perdita di informazione; pertanto, questa tecnica non è adatta per salvare le immagini che sono rielaborate con programmi di fotoritocco (le continue modifiche comporterebbero un progressivo degrado dell'immagine ad ogni salvataggio e riapertura). Invece, in virtù delle ridotte dimensioni del file, sono particolarmente indicate per la trasmissione di immagini o per ridurre le dimensioni di un'applicazione o di un prodotto da distribuire.

- jpeg
- gif (per immagini con più di 256 colori si ottiene una compressione lossy, poiché sono eliminate la maggior parte delle sfumature di colore)

## 11.2 Grafica vettoriale

Anche la **grafica vettoriale** è una tecnica utilizzata in computer grafica per descrivere un'immagine. Nella grafica vettoriale, l'immagine è descritta mediante un insieme di primitive geometriche che evidenziano punti, linee, curve e poligoni ai quali possono essere attribuiti colori e anche sfumature. È radicalmente diversa dalla grafica raster, in quanto nella grafica raster le immagini sono descritte come una griglia di pixel opportunamente colorati.

I dati vettoriali possono essere memorizzati attraverso le seguenti tipologie di file:

- svg
- pdf

- swf
- ...

### 11.2.1 Vantaggi

I principali vantaggi della grafica vettoriale rispetto alla grafica raster sono la qualità, la maggiore compressione dei dati e la più facile gestione delle eventuali modifiche.

La grafica vettoriale, essendo definita attraverso equazioni matematiche, è indipendente dalla risoluzione, mentre la grafica raster, se è ingrandita o visualizzata su un dispositivo dotato di una risoluzione maggiore di quella del monitor, perde di definizione. Tale sistema di descrizione delle informazioni grafiche presenta, inoltre, l'indubbio vantaggio di una maggiore compressione dei dati: in pratica, un'immagine vettoriale occupa molto meno spazio rispetto ad una corrispondente raster, con una riduzione dell'occupazione di RAM e memoria di massa, principalmente nelle forme geometriche o nei riempimenti a tinta piatta. Risulta, inoltre, più facile da gestire e da modificare, essendo minore la quantità di dati coinvolti in ogni singola operazione di aggiornamento. Questo rende il vettoriale particolarmente adatto per gestire grandi quantità di dati. Infine, l'ingrandimento o la riduzione delle misure e proporzioni del soggetto prodotto in vettoriale non incide in maniera significativa sul peso dell'immagine stessa: il riempimento di forme con tinte piatte è generato da semplici funzioni matematiche e risulta, quindi, estremamente leggero in termini di memoria utilizzata.

### 11.2.2 Svantaggi

Il principale svantaggio della grafica vettoriale rispetto alla grafica raster è che la realizzazione di immagini vettoriali non è una attività intuitiva come nel caso delle immagini raster. I programmi vettoriali dispongono di molti strumenti che, per essere sfruttati pienamente, richiedono svariate conoscenze. Un altro difetto è legato alle risorse richieste per trattare le immagini vettoriali: una immagine vettoriale molto complessa può essere molto corposa e richiedere l'impiego di un computer molto potente per essere elaborata. Inoltre, le risorse richieste per trattare l'immagine non sono definibili a priori, e quindi ci si potrebbe trovare nell'impossibilità di elaborare un'immagine per la mancanza di risorse sufficienti. Nel caso di una immagine raster, invece, una volta definita la risoluzione ed il numero di colori, è abbastanza semplice definire le risorse massime necessarie per trattare l'immagine stes-

sa; al contrario di quanto accade con le tinte piatte, i riempimenti sfumati o complessi generati in vettoriale comportano un alto impiego di risorse.

## 11.3 Joint Photographic Experts Group (JPEG)

<b>Sviluppatore</b>	Joint Photographic Experts Group
<b>Tipo</b>	Compressione dell' immagine
<b>Lossy/Lossless</b>	lossy (solitamente) e lossless
<b>Licenza</b>	Royalty free
<b>Sito web</b>	<a href="http://www.jpeg.org">http://www.jpeg.org</a>

Tabella 11.1: Joint Photographic Experts Group (JPEG)

**JPEG** è l' acronimo di **Joint Photographic Experts Group**, un comitato ISO/CCITT che ha definito il primo standard internazionale di compressione per immagini a tono continuo, sia a livelli di grigio che a colori. È un formato gratuito e open-source.

Attualmente JPEG è lo standard di compressione delle immagini fotografiche più utilizzato. Le estensioni più comuni per questo formato sono `.jpeg`, `.jpg`, `.jfif`, `.JPG` e `.JPE`, anche se il più comune in tutte le piattaforme è `.jpg`.

JPEG specifica solamente come una immagine può essere trasformata in uno stream di byte, ma non come questo può essere incapsulato in supporti di memorizzazione. Un ulteriore standard, chiamato JFIF (JPEG File Interchange Format), creato da Independent JPEG Group, specifica come produrre un file appropriato per la memorizzazione su computer di uno stream JPEG. Nell' uso comune, quando qualcuno parla di "file JPEG" generalmente intende un file JFIF o alcune volte un file Exif JPEG. Ci sono, comunque, altri formati di file basati su JPEG, come ad esempio JNG. Indice

Essenzialmente, il JPEG opera in tre passi fondamentali per trasformare un' immagine raster in una JPEG e viceversa. Tali passi sono:

1. Rappresentazione in ambito frequenziale tramite DCT (trasformata discreta del coseno) se opera in modalità lossy; uso dei predittori in modalità lossless;
2. Quantizzazione effettuata tramite opportune matrici che, solitamente, pesano i coefficienti di ordine più basso (rappresentano le basse frequenze spaziali) in maniera più decisa, in quanto, per le proprietà della DCT, sono più importanti ai fini della sintesi dell' immagine. Questo

perché il sistema visivo umano percepisce maggiormente le basse frequenze spaziali rispetto alle alte frequenze. Risulta, quindi, necessario dare maggior importanza alle basse frequenze spaziali.

3. Codifica entropica ed eliminazione delle ridondanze di tipo statistico tramite codifica RLE e codici di Huffman. La componente continua della DCT, invece, è codificata in DPCM.

Il fattore di compressione che si può raggiungere è determinato essenzialmente da un parametro di scalature per le matrici di quantizzazione: tanto più piccolo è questo parametro, tanto peggiore è la qualità. Si può ottenere un fattore di compressione 15:1 senza alterare visibilmente la qualità dell'immagine.

Lo standard JPEG definisce due metodi di compressione di base, uno basato sull'uso della trasformata discreta del coseno (DCT) con compressione di tipo lossy (cioè, con perdita di informazione), l'altro sull'uso di un metodo predittivo con compressione di tipo lossless (cioè, senza perdita di informazione). L'algoritmo base dello JPEG di tipo lossy è detto *baseline*; inoltre, sono state definite delle estensioni opzionali del metodo lossy per la compressione di tipo gerarchico e progressivo.

JPEG/JFIF è il formato più utilizzato per la memorizzazione di fotografie. È, inoltre, il formato più comune sul World Wide Web. Non è, invece, adatto per disegni geometrici, testo o icone, per cui sono utilizzati, comunemente, i formati PNG e GIF.

## 11.4 Tagged Image File Format (TIFF)

**TIFF** è l'acronimo di **Tagged Image File Format**, ed è un formato immagine di tipo raster piuttosto diffuso. Le specifiche del formato TIFF permettono una notevole flessibilità; questo, di per sé, è un vantaggio, ma rende difficile scrivere un interprete pienamente conforme alle specifiche. Il TIFF è largamente utilizzato per lo scambio di immagini raster fra stampanti e scanner, perché permette di specificare numerose indicazioni aggiuntive (come le tabelle di gamut o informazioni sulla calibratura del colore).

Lo standard TIFF definisce un metodo di compressione di tipo lossless.

Permette di rappresentare immagini con diversi spazi di colore: scale di grigio, RGB, CMYK e CIELab.

Le immagini possono essere memorizzate, oltre come linee di scansione, anche in riquadri: questo permette di avere un rapido accesso ad immagini di grosse dimensioni.

Un file TIFF può contenere immagini divise su più “pagine” (videate), ad esempio si possono inserire in un unico file tutte le pagine che compongono un fax.

## 11.5 Graphics Interchange Format (GIF)

---

<b>Sviluppatore</b>	CompuServe
<b>Tipo</b>	Compressione dell' immagine
<b>Lossy/Lossless</b>	lossless
<b>Licenza</b>	Conteneva algoritmi il cui brevetto è scaduto
<b>Sito web</b>	<a href="http://www.compuserve.org">http://www.compuserve.org</a>

---

Tabella 11.2: Graphics Interchange Format (GIF)

**GIF** è l' acronimo di **Graphics Interchange Format**, ed è un formato per immagini di tipo bitmap molto utilizzato nel World Wide Web, sia per immagini fisse che per le animazioni.

Il formato GIF prevede l' utilizzo di un numero massimo di 256 colori, essendo basato sull' uso della **Palette VGA**. Ogni colore all' interno della Palette è definito da una terna di valori (RGB) delle dimensioni di un Byte (quindi con valore da 0 a 255), consentendo quindi di definire, per ogni colore,  $256 \times 256 \times 256$  sfumature. Una volta definita la Palette (quindi 256 terne di bytes) dell' immagine, per definire il colore di un pixel, è sufficiente indicare per esso il numero (posizione) del colore nella palette.

Esiste una tecnica per simulare un numero maggiore di colori, utilizzando pixel di colore simile a quello desiderato, ma in questo modo si perdono dei dettagli e, siccome non esiste un unico algoritmo di scelta dei colori, la qualità dell' immagine è molto variabile. Questa tecnica, inoltre, diminuisce la comprimibilità dell' immagine stessa.

Uno dei colori della tavolozza può essere definito come trasparente e quindi, in fase di visualizzazione, è sostituito con lo sfondo. Questa caratteristica (canale alfa) permette di simulare immagini non rettangolari.

## 11.6 Portable Network Graphics (PNG)

PNG è l' acronimo di **Portable Network Graphics**. È un formato di file per memorizzare immagini.

Il formato PNG è superficialmente simile al GIF, in quanto è capace di immagazzinare immagini in modo lossless, ossia senza perdere alcuna infor-

<b>Sviluppatore</b>	Thomas Boutell (ideatore originario)
<b>Tipo</b>	Compressione dell' immagine
<b>Lossy/Lossless</b>	lossless (solitamente) e lossy
<b>Licenza</b>	BSD-Like/GPL
<b>Sito web</b>	<a href="http://www.libpng.org">http://www.libpng.org</a>

Tabella 11.3: Portable Network Graphics (PNG)

mazione, ed è più efficiente con immagini non fotorealistiche (ovvero, immagini che non contengono troppi dettagli per essere compresse in poco spazio). Essendo stato sviluppato molto tempo dopo, non ha molte delle limitazioni tecniche del formato GIF: può memorizzare immagini in colori reali (mentre il GIF è limitato a 256 colori) e ha un canale dedicato per la trasparenza (canale alfa). Esiste, inoltre, un formato derivato, **Multiple-image Network Graphics** o **MNG**, che è simile al GIF animato.

Il formato PNG supporta: gestione dei colori classica tipo bitmap oppure indicizzata, possibilità di trasmettere l' immagine lungo un canale di comunicazione seriale (serializzazione dell' immagine), visualizzazione progressiva dell' immagine (grazie all' interlacciamento della medesima), supporto alla trasparenza mediante un canale alfa dedicato, informazioni ausiliare di qualsiasi natura accluse al file, completa indipendenza dall' hardware e dalla piattaforma in uso, compressione dei dati di tipo lossless, immagini truecolor fino a 48 bpp, immagini in scala di grigio sino a 16 bpp, filtro dei dati prima della (eventuale) compressione, correzione della gamma dei colori presenti nell' immagine (per riprodurre esattamente quanto visualizzato all' atto della creazione dell' immagine), (debole) verifica dell' integrità dei file, rapida presentazione iniziale dell' immagine (grazie alla visualizzazione progressiva).

## 11.7 Scalable Vector Graphics (SVG)

**SVG** è l' acronimo di **Scalable Vector Graphics** ed indica una tecnologia in grado di visualizzare oggetti di grafica vettoriale e, pertanto, di gestire immagini scalabili dimensionalmente. Più specificamente, si tratta di un linguaggio derivato dall' XML.

SVG permette di trattare tre tipi di oggetti grafici: forme geometriche (cioè, linee costituite da segmenti di retta e curve e aree delimitate da linee chiuse), immagini della grafica raster ed immagini digitali, testi esplicativi (eventualmente cliccabili). Gli oggetti grafici possono essere raggruppati in oggetti più comprensivi, muniti di attributi di stile e aggiunti ad oggetti grafici precedentemente costruiti e visualizzati. Un testo può far parte di un



qualsiasi namespace XML sottoponibile ad una applicazione; questa possibilità consente di aumentare la ricercabilità e l'accessibilità delle immagini SVG. Il repertorio delle operazioni attuabili include trasformazioni annidate, percorsi di clipping, canale alpha, effetti di filtro, oggetti template ed estensibilità.



## Parte III

### Famiglie di CPU e Multiprocessori



# Capitolo 12

## La CPU

L' **unità centrale di elaborazione** (CPU), anche chiamata nella sua implementazione fisica *processore*, è uno dei due componenti principali della macchina di von Neumann, il modello su cui sono basati la maggior parte dei moderni computer. Il compito della CPU è quello di leggere i dati dalla memoria ed eseguirne le istruzioni; il risultato dell' esecuzione dipende dal dato su cui opera e dallo stato interno della CPU stessa, che tiene traccia delle passate operazioni.

In base all' organizzazione della memoria si possono distinguere due famiglie di CPU:

- con architettura di von Neumann classica, in cui dati ed istruzioni risiedono nella stessa memoria (è dunque possibile avere codice automodificante). Questa architettura è la più comune, perché è più semplice e flessibile.
- con architettura Harvard, in cui dati ed istruzioni risiedono in due memorie separate. Questa architettura garantisce migliori prestazioni, poiché le due memorie possono lavorare in parallelo, ma è più complessa da gestire.

### 12.1 Struttura della CPU

Una generica CPU contiene:

- una **ALU** (Arithmetic Logic Unit) che si occupa di eseguire le operazioni logiche e aritmetiche;
- una **Unità di Controllo** o **CU** (Control Unit) che legge dalla memoria le istruzioni (se occorre, legge anche i dati per l' istruzione letta), esegue

l'istruzione e memorizza il risultato (se c'è), scrivendolo in memoria o in un registro della CPU.

- dei **registri**, cioè speciali locazioni di memoria interne alla CPU, molto veloci, a cui è possibile accedere molto più rapidamente che alla memoria: il valore complessivo di tutti i registri della CPU costituisce lo stato in cui essa si trova attualmente. Due registri sempre presenti sono:

- il registro IP (Instruction Pointer) o PC (Program Counter), che contiene l'indirizzo in memoria della prossima istruzione da eseguire;
- il registro dei flag, che non contiene valori numerici convenzionali, ma è piuttosto un insieme di bit, detti appunto *flag*, che segnalano stati particolari della CPU ed alcune informazioni sul risultato dell'ultima operazione eseguita. I flag più importanti sono:

\* Flag di stato:

- Overflow: indica se il risultato dell'operazione precedente era troppo grande per il campo risultato: 0 assenza di overflow, 1 overflow
- Zero: vale 1 se l'ultima operazione ha avuto risultato zero, altrimenti vale 0.
- Carry: vale 1 se l'ultima operazione ha ecceduto la capacità del registro che contiene il risultato, altrimenti vale 0.
- Segno indica il segno del risultato dell'operazione precedente: 0 risultato positivo, 1 risultato negativo.

\* Flag di controllo:

- Interrupt: se a questo flag è assegnato valore 1, la CPU smette di rispondere alle richieste di servizio esterne delle periferiche finché non è ripristinato al valore 0, o finché non arriva dall'esterno un segnale di RESET.

Oltre a queste unità, possono esserne presenti altre, per esempio:

- una FPU (Floating Point Unit) che si occupa di eseguire calcoli in virgola mobile;
- una MMU (Memory Management Unit) che si occupa di tradurre gli indirizzi di memoria logici in indirizzi fisici, supportando la protezione della memoria e/o uno o più meccanismi di memoria virtuale.

Una generica CPU deve eseguire i suoi compiti sincronizzandoli con il resto del sistema: perciò, è dotata, oltre a quanto sopra elencato, anche di uno o più bus interni che si occupano di collegare registri, ALU, unità di controllo e memoria. Inoltre, all'unità di controllo interna della CPU fanno capo una serie di segnali elettrici esterni che si occupano di tenere la CPU al corrente dello stato del resto del sistema e di agire su di esso. Il tipo e il numero di segnali esterni gestiti possono variare, ma alcuni, come il RESET, le linee di IRQ ed il CLOCK sono sempre presenti.

Per quanto riguarda i registri, le CPU possono gestirli in molti modi: i più comuni sono registri nominativi (CPU CISC classiche), file di registri (RISC) e stack di registri (Transputer e simili).

- Stack di registri: i registri sono organizzati in una struttura a stack (pila); questa architettura ha il vantaggio di non dover specificare su quale registro interno operare (è sempre quello in cima allo stack) ottenendo istruzioni più corte e più semplici da decodificare. Nel caso sia necessario un dato in fondo allo stack, il suo recupero è un'operazione molto lenta.
- Registri nominativi: ogni registro è identificato singolarmente, e le istruzioni che usano registri specificano di volta in volta quale registro devono usare. Spesso alcuni registri sono dedicati a scopi particolari (registri indice, accumulatori, registri di segmento ecc.), imponendo la non ortogonalità del set di istruzioni.
- File di registri: i registri sono organizzati come una memoria interna della CPU e indicizzati. La CPU alloca un certo numero di registri per ogni processo e/o subroutine in esecuzione, eliminando la necessità di accedere alla RAM per salvare gli stack di chiamata delle funzioni ed i dati di task switching nei sistemi multitask.

Una CPU è un circuito digitale sincrono: vale a dire che il suo stato cambia ogni volta che riceve un impulso da un segnale di sincronismo detto *clock*, che ne determina, di conseguenza, la velocità operativa, detta *velocità di clock*. Quindi, il tempo di esecuzione di una istruzione si misura in cicli di clock, cioè in quanti impulsi di clock sono necessari perché la CPU la completi. In effetti, una parte importante e delicata di ogni CPU è il sistema di distribuzione che porta il segnale di clock alle varie unità e sottounità di cui è composta, per fare in modo che siano sempre in sincronia: tale sistema si dirama in una struttura ad albero con divisori e ripetitori che giunge ovunque nella CPU. La velocità di questa distribuzione determina in maniera diretta la massima frequenza operativa di una CPU: nessuna CPU può essere più veloce del suo

*critical path*, cioè del tempo che impiega il clock per percorrere il tratto più lungo in tutto l' albero di distribuzione del clock.

## 12.2 Architetture CISC ed architetture RISC

Quando i transistor disponibili su un solo chip erano pochi e i calcolatori erano spesso programmati in assembly, era naturale sfruttarli in modo tale da avere CPU con istruzioni potenti, evolute e complesse: più queste erano vicine alle istruzioni dei linguaggi di programmazione ad alto livello più il computer sarebbe stato facile da programmare, ed i programmi avrebbero occupato poco spazio in memoria (anch' essa poca e preziosa). Le CPU progettate secondo questo approccio sono dette *CISC* ed avevano unità di controllo complesse capaci di sfruttare al meglio pochi registri e i cui programmi erano di dimensioni relativamente piccole. A cavallo fra gli anni Settanta e gli anni Ottanta la situazione cambiò rapidamente: la RAM divenne più economica e comparvero i primi compilatori moderni, ottimizzanti, in grado di generare linguaggio macchina molto efficiente: per questo si iniziò a pensare ad un nuovo modo di progettare le CPU, prendendo in esame la possibilità di usare i transistor disponibili per avere, invece, molti registri ed un set d' istruzioni elementare, molto ridotto, che delegasse al compilatore il lavoro di tradurre le istruzioni complesse in serie di istruzioni più semplici, permettendo così di avere unità di controllo particolarmente semplici e veloci. Attualmente, la distinzione fra queste due classi di architetture è venuta in gran parte meno: il numero di transistor disponibili su un solo chip è aumentato tanto da poter gestire molti registri ed anche set di istruzioni complesse.

### 12.2.1 Complex Instruction Set Computer (CISC)

**CISC** è l' acronimo di **Complex Instruction Set Computer**. Un processore di questo tipo implementa un numero relativamente scarso (una decina) di registri di uso generale, ed ha una unità di controllo microprogrammata: la logica del programma è memorizzata in una memoria veloce situata nella parte di controllo, invece di essere espressa tramite una rete combinatoria.

Il set di istruzioni associato a CPU di tipo CISC è molto esteso e composto in genere di alcune centinaia di codici operativi diversi che svolgono funzioni anche molto complesse, fra cui sono caratteristici i trasferimenti memoria-memoria, assenti nei RISC; le istruzioni hanno lunghezza variabile e possono presentarsi in formati diversi, e sono necessari due o più (a volte molti di più) cicli di clock per completare una istruzione; è possibile specificare la



posizione dei dati necessari alle istruzioni usando molti metodi di indirizzamento diversi. Il ridotto numero di registri interni obbliga questi processori a scrivere in memoria ogni volta che si verifica una chiamata di funzione, che si verifica un context switch o che è salvato un registro nello stack.

### 12.2.2 Reduced Instruction Set Computer (RISC)

**RISC** è l' acronimo di **Reduced Instruction Set Computer**. Il tipico set di istruzioni RISC è molto piccolo, circa sessanta o settanta istruzioni molto elementari (logiche, aritmetiche ed istruzioni di trasferimento memoria-registro e registro-registro): hanno tutte lo stesso formato e la stessa lunghezza, e molte sono eseguite in un solo ciclo di clock. La diretta conseguenza di tale scelta progettuale è che i processori RISC posseggono una unità di controllo semplice ed a bassa latenza, riservando invece molto spazio per i registri interni: una CPU RISC ha di solito da un minimo di un centinaio ad alcune migliaia di registri interni generici, organizzati in un file di registri. Il fatto di avere un formato unico di istruzione permette di strutturare l' unità di controllo come una pipeline, cioè una catena di montaggio a più stadi: questa innovazione ha il grosso vantaggio di ridurre il critical path interno alla CPU e consente ai RISC di raggiungere frequenze di clock più alte rispetto agli analoghi CISC.

Nel caso di context-switch o di chiamata a subroutine o di uso dello stack, i RISC, invece di accedere alla memoria di sistema, usano un meccanismo chiamato *register renaming*, che consiste nel rinominare i registri in modo da usare per la nuova esecuzione una diversa zona del file di registri, senza dover accedere alla memoria ogni volta.

### 12.2.3 RISC vs CISC

La semplicità dei RISC si traduce in una minore espressività del linguaggio assembler: il numero di word necessarie per esprimere una computazione in una macchina RISC è maggiore/uguale alla controparte CISC. Questo si traduce in programmi più grossi, penalità molto alta in altre epoche, in cui la RAM era una componente costosa e di bassa capacità. L' architettura CISC dipende dal compilatore più di quanto non dipenda il RISC: dato che le istruzioni prevedono più metodi di indirizzamento, e che sono presenti istruzioni dalla semantica complessa, al compilatore è prospettato un ampio ventaglio di scelte per quanto concerne la traduzione di una istruzione. Come spesso accade nei problemi di ottimizzazione, la scelta della configurazione migliore è un compito NP completo, e non è pensabile utilizzare un compilatore che, per ogni istruzione, valuti la scelta migliore in base al contesto.

## 12.3 Intel 80286I/80386/80486

L' **Intel 80286** è un microprocessore a 16 bit della famiglia x86. Inizialmente rilasciato in versioni da 6 e 8 MHz, in seguito fu accelerato fino a 20 MHz. Il processore 80286 offre prestazioni doppie per ciclo di clock rispetto al suo predecessore 8086 e può sfruttare fino a 16 MB di RAM, al contrario del singolo MB indirizzabile dall' 8086. Nei computer basati sul sistema operativo DOS, questa capacità addizionale di RAM poteva essere utilizzata solo attraverso una emulazione di memoria estesa. In ogni caso, pochi computer basati sul 286 ebbero più di 1 MB di RAM.

Il processore 80286 è stato progettato per applicazioni multitasking, relative alle comunicazioni (ad es. centralini telefonici PBX), processi di controllo in tempo reale e sistemi multi-utente.

Una caratteristica interessante di questo processore è che fu il primo tra quelli in architettura x86 con la possibilità di passare da modalità reale a modalità protetta, permettendo l' uso di tutta la memoria di sistema come un unico blocco e offrendo un certo grado di protezione delle zone di memoria usate dalle applicazioni. Il passaggio a tale modalità non è però reversibile: il ritorno alla modalità normale richiede il reset del processore. Tale limitazione fece sì che la modalità protetta non trovasse grande impiego fino all' arrivo del processore 80386, che è in grado di passare da una modalità all' altra indifferentemente.

Il successore dell' 80286 fu il processore a 32 bit: **Intel 80386**. L' 80386 fu il primo microprocessore di Intel con architettura a 32 bit e modalità protetta con supporto hardware alla memoria virtuale paginata.

L' **Intel i486** (o anche **80486**) è una gamma di microprocessori con architettura CISC e x86 prodotti da Intel. Da un punto di vista software, l' i486 è molto simile al predecessore i386, se non per l' aggiunta di alcune istruzioni. Da un punto di vista hardware, invece, questo processore è molto innovativo: possiede una memoria cache di 8 kb unificata per dati ed istruzioni, un' ulteriore unità di calcolo in virgola mobile (*FPU*) (opzionale, inclusa solo nella versione DX, DX2 e DX4), una *bus interface unit* migliorata, e le caratteristiche di *Power management* e l' *SMM* (*System Management Mode*) che divennero standard nel processore. In condizioni ottimali, questo processore può eseguire un' istruzione per ciclo di clock. Questi miglioramenti permettono all' i486 di offrire prestazioni quasi doppie a quelle del predecessore, a parità di clock.

## 12.4 Intel Pentium

Il **Pentium** è un microprocessore appartenente alla quinta generazione di processori con architettura x86 prodotti da Intel, ed è il successore dell' Intel 80486. Le maggiori differenze tra il Pentium e l' 80486 sono:

- *Architettura superscalare*: il Pentium possiede due pipeline che gli permettono di completare più di una operazione per ciclo di clock. Una pipeline, chiamata *pipeline U*, può eseguire qualunque istruzione, mentre l' altra, chiamata *pipeline V*, è in grado di eseguire solo quelle più semplici e comuni. L' utilizzo di più pipeline è una caratteristica delle architetture RISC; una delle tante caratteristiche che, nel tempo, sono state implementate sulle architetture x86, dimostrando la possibilità di unire le due tecnologie e creare dei processori che possano essere definiti “ibridi”.
- Data path a 64 bit: questa caratteristica raddoppia la quantità di informazioni prelevate dalla memoria in ogni operazione di fetch. È importante sottolineare, però, che questo aspetto non consente assolutamente al Pentium di poter eseguire codice a 64 bit, dato che i suoi registri continuano ad essere a 32 bit.
- Supporto per le istruzioni MMX (solo i modelli più recenti).
- Introduzione dell' istruzione CPUID: si trattava di una funzionalità che consente il riconoscimento della CPU da parte del software.

I Pentium, potendo eseguire più istruzioni per singolo ciclo di clock, offrono prestazioni di poco inferiori al doppio a quelle di un 486 di pari frequenza.

I primi Pentium furono rilasciati a velocità di 60 e 66 MHz, la stessa del BUS di sistema; più tardi divennero disponibili versioni con frequenza di clock maggiore: 75 MHz con BUS a 50 MHz e moltiplicatore a 1,5×, dalle prestazioni praticamente pari a quelle della versione a 66 MHz per via del rallentamento del BUS di sistema.

I *Pentium OverDrive* furono rilasciati, a velocità di 63 MHz e 83 MHz, per poter aggiornare computer basati sui vecchi 486 senza stravolgimenti hardware.

Il Pentium originale era un microprocessore superscalare che eseguiva le istruzioni in ordine attraverso una pipeline dati. Inizialmente, fu prodotto con la tecnologia a 0,8  $\mu\text{m}$ , ovvero 800 nm, integrando 3,1 milioni di transistor. In seguito, cominciò la produzione del P54, un adattamento del P5 alla tecnologia a 0,6  $\mu\text{m}$ , che era predisposto a lavorare in coppia ed aveva una velocità di clock superiore a quella del front side bus (è molto più difficile

aumentare la velocità di quest' ultimo). Il P54 fu sostituito a sua volta dal P54C, prodotto con tecnologia a  $0,35\ \mu\text{m}$ . In seguito fu diffuso il P55C, o *Pentium MMX*, che riprendeva il core P5 ed il processo di produzione a  $0,35\ \mu\text{m}$ , e che inoltre includeva l' instruction set MMX, costituito da 57 nuove istruzioni, che permettevano maggiori prestazioni nel trattamento dei dati multimediali. Il software doveva, però, essere ottimizzato per poter sfruttare queste istruzioni, ed il miglioramento prestazionale del P55C fu dovuto in gran parte al raddoppiamento della memoria Cache, da 16 a 32 KB.

L' Intel Pentium consente la **predizione delle diramazioni** (*branch prediction*), che è il compito della **BPU** (**Branch Prediction Unit**), una componente della CPU che cerca di prevedere l' esito di un' operazione su cui si basa l' accettazione di una istruzione di salto condizionato, evitando rallentamenti che possono essere molto evidenti in una architettura con pipeline. L' importanza di questa operazione è evidente soprattutto per i microprocessori moderni, superscalari e con lunghe pipeline, che per ogni errore di previsione devono sprecare molti cicli di clock di lavoro prezioso. La predizione delle diramazioni non va confusa con la predizione dell' indirizzo di arrivo della diramazione. Questa predizione è svolta dall' unità *branch target predictor* che, data la predizione dell' unità di predizione delle diramazioni, cerca di predire l' indirizzo di arrivo del salto e di caricare le istruzioni corrispondenti prima che il salto sia svolto in modo da evitare rallentamenti dovuti al caricamento delle istruzioni dopo il salto.

### 12.4.1 Pentium MMX

Il **Pentium MMX** (sigla di progetto P55C) è una variante dell' architettura Pentium che Intel ha introdotto nel 1996 per aumentare le prestazioni su talune applicazioni, nella fattispecie quelle multimediali.

La sigla *MMX* (MultiMedia eXtension), infatti, indica la presenza di alcune istruzioni macchina aggiuntive di tipo SIMD (Single Instruction Multiple Data). Le istruzioni SIMD applicano la stessa operazione a più dati simultaneamente.

Per semplificare la progettazione, gli ingegneri Intel decisero di preservare la vecchia architettura alla quale venne aggiunta l' unità MMX che operava tramite un cambio di contesto del processore. L' unità MMX utilizza i registri IA-32 della FPU. Questa infelice scelta progettuale impediva di utilizzare istruzioni in virgola mobile e istruzioni MMX contemporaneamente. Quando il processore incontrava istruzioni per la FPU doveva salvare i dati MMX per eseguire le istruzioni dell' FPU. Per massimizzare le prestazioni, i programmatori utilizzavano il processore in un solo dei due modi possibili,

isolando le parti MMX che venivano eseguite in modo esclusivo in modo da evitare il più possibile i lenti cambi di contesto.

Un altro difetto delle istruzioni MMX che erano istruzioni in grado di operare solo su dati interi.

Assieme a queste istruzioni aggiuntive, il Pentium MMX si distingue per una memoria cache di primo livello duplicata (32 kBytes invece di 16) rispetto al Pentium tradizionale. L' aumento della cache offre un guadagno prestazionale, in applicazioni non ottimizzate MMX, di circa il 7%. Il software scritto per trarre vantaggio dalle istruzioni MMX offre, invece, un guadagno prestazionale che va mediamente dal 10% al 30%.

### 12.4.2 Intel P6

L'**architettura P6** per i processori x86 è stata una pietra miliare nella storia della Intel. Si tratta di un' architettura di sesta generazione, derivata dalla P5, che nelle intenzioni iniziali doveva essere alla base di una CPU che avrebbe sostituito il processore Pentium. In realtà, questo non avvenne mai ed il processore *Pentium Pro* (basato su P6) fu indirizzato esclusivamente al settore server e workstation prima di essere sostituito dalla serie Xeon. L' importanza dell' architettura P6 diventa chiara se si considera che da essa sono derivati i successivi Pentium II, Pentium III e Pentium M.

Il Pentium Pro utilizza il socket 8 per collegarsi alla motherboard, ed è molto differente dal Pentium. L' architettura P6, infatti, supporta l' esecuzione out-of-order e l' esecuzione speculativa delle istruzioni, la rinominazione dei registri e la possibilità di possedere una pipeline aggiuntiva per istruzioni semplici. Il Pentium Pro aveva in dotazione una memoria cache di 256 KB e 512 KB, e più tardi fu rilasciata una versione con 1 MB. Tutte le versioni del processore sono molto costose, soprattutto quelle che ne implementano più di 256 KB. Infatti, tale cache era prodotta con un metodo innovativo: il processore e la cache erano su die differenti, integrati insieme nello stesso involucro, e connessi da un BUS cloccato alla velocità del processore. I due die, entrambi molto grandi per gli standard di allora, dovevano essere uniti in uno stadio precoce del processo di produzione, prima che fosse possibile testarne il funzionamento. Questo significava che un singolo e piccolo difetto in una delle due parti rendeva necessario scartare l' intero insieme. Questa è una delle ragioni della bassa efficienza del processo di produzione del Pentium Pro.

L' esecuzione speculativa di una parte di programma (branch) in attesa dell' esecuzione dell' operazione di selezione comportava un maggiore spreco nel caso che la previsione sulla selezione fosse errata: quindi, il Pentium Pro possedeva un algoritmo di branch prediction molto più sofisticato rispetto

al predecessore. Per la stessa ragione, col Pentium Pro venne introdotta l'istruzione di *conditional move* (*cmov*) che in certi casi poteva risparmiare l'utilizzo di una istruzione di selezione. Le prestazioni relative al codice a 32 bit erano eccellenti; tuttavia, il Pentium Pro spesso era più lento di un normale Pentium nel far girare codice a 16 bit ed i sistemi operativi.

## 12.5 Hyper-Threading

**Hyper-Threading** è il nome commerciale dato da Intel alla sua prima implementazione della tecnologia *Simultaneous Multi-Threading* utilizzata per migliorare le prestazioni dei propri processori.

L'idea alla base della tecnologia Hyper-Threading era quella di duplicare alcune unità di elaborazione all'interno dei microprocessori al fine di poter eseguire simultaneamente alcune operazioni, grazie a tecniche di multithreading. Si trattava, in un certo senso, di un tentativo di creare un processore di transizione tra i tradizionali single core ed i successivi dual core, non inserendo due interi core all'interno di un unico package, ma duplicando solo alcune aree "sensibili" del singolo core. Grazie all'Hyper-Threading, un singolo core era comunque in grado di gestire più thread in contemporanea. Quando le istruzioni di un thread rimanevano bloccate nella pipeline, il processore procedeva ad elaborare un secondo thread al fine di mantenere le unità di elaborazione sempre attive. I singoli thread possono, infatti, essere bloccati nella loro esecuzione da molteplici fattori, quali ad esempio problemi di recupero dei dati da elaborare (per esempio un cache miss) o per problemi di dipendenza dai dati che si trovano in elaborazione presso altre istruzioni in esecuzione.

La replicazione solo di alcune unità di elaborazione non poteva competere, come prestazioni, con un processore dual core vero e proprio, dove l'intero core era duplicato. Il punto veramente debole della tecnologia Hyper-Threading, quando confrontata con un processore dual core vero e proprio, risiedeva nella capacità di tale tecnologia di gestire in parallelo solo thread e non processi; in altre parole, in un sistema dotato di processore con Hyper-Threading traggono vantaggio solo le applicazioni dotate di più thread, mentre in un sistema dotato di un processore dual core, possono essere eseguiti in parallelo sia i thread di un singolo programma (come avviene con la tecnologia Hyper-Threading), sia i thread appartenenti a programmi diversi, che possono quindi essere eseguiti simultaneamente solo in quest'ultimo tipo di sistemi.

Dal punto di vista del sistema operativo, un processore single core, ma dotato di tecnologia Hyper-Threading, era indistinguibile da un normale pro-

cessore dual core. Intel aveva infatti sviluppato tale nuova soluzione in modo che il sistema operativo “vedesse”, comunque, la presenza di due core di elaborazione. Tali core non erano in effetti due core “fisici” (come nei processori dual core), ma erano due core “logici”, vale a dire che un’ applicazione non doveva essere pensata espressamente per la tecnologia Hyper-Threading; se essa era in grado di sfruttare due core di elaborazione, era anche in grado di sfruttare le potenzialità della nuova tecnologia (con vantaggi prestazionali inferiori). Il BIOS della scheda madre, che doveva a sua volta supportare la nuova tecnologia, faceva “credere” al sistema operativo ed alle applicazioni di trovarsi di fronte ad un vero e proprio sistema biprocessore.





# Capitolo 13

## Il multiprocessore

Con il termine **multiprocessore** si intende un sistema equipaggiato con più di due processori operanti in parallelo in cui le elaborazioni di un processore sono replicate e controllate da un processore gemello, per garantire l'integrità e l'esattezza dei dati. Inizialmente, questo avveniva solo sui sistemi che chiedevano un'elevata potenza di calcolo, come quelli che dovevano operare su database di grandi dimensioni, o eseguire calcoli molto complessi di simulazione o in ambienti dove le informazioni sono di importanza vitale.

Per sfruttare i vantaggi di un sistema multiprocessore, al pari di uno biprocessore, è necessario che anche il sistema operativo ed i programmi siano realizzati in maniera da utilizzare le maggiori risorse offerte. In pratica, in fase di programmazione è necessario “parallelizzare” il codice in modo che non si abbiano situazioni in cui l'applicazione “vede” solo uno dei processori saturandolo, lasciando inutilizzati tutti gli altri.

Spesso è necessario utilizzare più processori, gruppi di processori e cluster. Nel caso di basi di utenti molto ampie e diffuse, i gruppi di più processori e le varie istanze del database possono anche essere geograficamente dispersi.

Nei multiprocessori diverse CPU condividono una memoria comune:

- le CPU devono coordinarsi per accedere alla memoria
- esistono diversi schemi di collegamento tra CPU e memoria. Quello più semplice prevede un bus condiviso.

### 13.1 Tassonomia dei calcolatori

Si definisce **tassonomia dei multiprocessori** la classificazione dei calcolatori che analizza le caratteristiche di una macchina attraverso elementi funzionali che la compongono e le relative interconnessioni.

Nel 1966, **Michael J. Flynn** classifica i sistemi di calcolo a seconda della molteplicità del flusso di istruzioni e del flusso dei dati che possono gestire; in seguito questa classificazione è stata estesa con una sottoclassificazione per considerare anche il tipo di architettura della memoria. In base a questa classificazione ogni sistema di calcolo rientra in una di queste categorie:

- **SISD** (Single Instruction Single Data): è un'architettura in cui una singola unità esegue un singolo flusso di dati. Nell'architettura SISD è possibile eseguire un'unica istruzione per volta su un unico dato, eseguendo le istruzioni una dopo l'altra come previsto dal paradigma imperativo. Prevede i seguenti cicli di istruzione: fase di Fetch (preleva dalla memoria centrale l'istruzione da eseguire utilizzando l'indirizzo contenuto nel program counter, incrementandolo in maniera tale che possa contenere l'indirizzo della prossima istruzione da eseguire), fase di Decode (la control unit interpreta l'istruzione e determina le operazioni da eseguire) e fase di Execute (L'unità aritmetico logica esegue le operazioni e la control unit controlla l'andamento dell'esecuzione delle operazioni). Un importante limite della architettura SISD è la singola connessione tra il processore e la memoria. Siccome l'accesso alla memoria è limitato soltanto ad un'operazione alla volta, insorge il classico problema definito come "collo di bottiglia", generando un rallentamento generalizzato. Alcune soluzioni individuate sono: allargamento del BUS Dati, introduzione della cache e parallelismo temporale per le istruzioni, facendo in modo che la macchina riesca ad elaborare il più alto numero possibile di istruzioni al secondo.
- **SIMD** (Single Instruction Multiple Data): è un'architettura in cui più unità elaborano dati diversi in parallelo. Questa è utilizzata da processori vettoriali o da processori che funzionano in parallelo. Il modello SIMD è composto da un'unica unità di controllo che esegue una istruzione alla volta, controllando più ALU che operano in maniera sincrona. Ad ogni passo, tutti gli elementi eseguono la stessa istruzione scalare, ma ciascuno su un dato differente. Un elaboratore basato su questo modello è anche detto *Array Processor*. Nell'elaborazioni di dati multimediali spesso si incontrano algoritmi che possono avvantaggiarsi di un'architettura SIMD. Per esempio, per cambiare la luminosità di un'immagine, un microprocessore dovrebbe caricare ogni pixel che compone l'immagine nei suoi registri, effettuare la modifica della luminosità e poi salvare i risultati in memoria. Un processore SIMD eseguirebbe prima un'istruzione che caricherebbe, con un'unica operazione, un certo numero di pixel (il numero preciso dipende dall'architettura), poi il processore modificherebbe tutti i dati in

parallelo, ed in seguito li salverebbe tutti contemporaneamente in memoria. Eseguire le operazioni a blocchi invece che agire sui singoli pixel rende le operazioni molto più efficienti. Un altro vantaggio deriva dal fatto che le istruzioni SIMD sono sempre in grado di manipolare tutti i dati caricati contemporaneamente: quindi, se un processore SIMD è in grado di caricare otto dati, questo sarà anche in grado di processarli tutti contemporaneamente. Le architetture basate su SIMD richiedono un numero elevato di registri. Quindi, a volte i progettisti, per ridurre i costi, decidono di utilizzare i registri della FPU. Questa scelta rende impossibile utilizzare istruzioni SIMD e FPU contemporaneamente a meno di lenti cambi di contesto. Questo era l'approccio scelto da Intel per le istruzioni MMX che, infatti, sono lente se associate ad operazioni in virgola mobile.

- Processori vettoriali
  - Array processor
  - Array sistolici
- **MISD** (Multiple Instruction Single Data)
- **MIMD** (Multiple Instruction Multiple Data): è un'architettura parallela in cui diverse unità effettuano diverse elaborazioni su dati diversi. Nel modello di calcolo MIMD ci sono più processori che operano in parallelo in modo asincrono. Ciascun processore esegue un programma (o istruzione) diverso su dati diversi. In un singolo (ma grande) problema, ogni processore risolve un sotto-problema. La comunicazione tra i vari processori avviene sia mediante la memoria condivisa (si parla di *processori tightly coupled* altamente legati), sia per mezzo di una rete di interconnessione (multi computer debolmente legati). Nell'architettura MIMD multiprocessore, i processori condividono le variabili (o strutture di dati) in una memoria comune. La comunicazione avviene mediante la condivisione della memoria. Per questo motivo, i processori necessitano di alcuni meccanismi di controllo che risolvano il probabile conflitto di scrittura in memoria da parte di due o più processori contemporaneamente. In questa architettura, il calcolo è molto veloce in quanto tutti i processori accedono simultaneamente alla memoria condivisa. Però, c'è da dire che tale architettura è anche molto complessa, in quanto devono essere presenti numerosi circuiti di decodifica tra i processori, ma soprattutto ci vuole una memoria molto grande e quindi molto costosa. Nell'architettura MIMD multi computer, ogni processore ha una memoria propria. I vari processori comunicano tra

loro mediante uno scambio di messaggi definito *message passing*. La comunicazione tra i vari processori è affidata ad una rete di interconnessione che consente a ciascun processore di trasmettere dati a qualunque altro processore presente nella stessa rete. I processori possono anche essere fisicamente lontani (come capita nei sistemi distribuiti). Però, siccome le prestazioni calano in funzione della qualità di comunicazione tra i processori, non è consigliato creare una rete che copra enormi distanze, pena la velocità di scambio dei dati. È molto complicato progettare una rete per molti processori. La rete è progettata come una *macchina special-purpose* dove la sua topologia intrinseca è una caratteristica distintiva di un tipo di multi computer. La complessità della rete è valutata in *gradi* (numero di connessioni che si dipartono da ogni nodo) e in *diametro* (distanza minima, espressa in numero di nodi, tra due nodi più lontani). Un grado elevato ed un piccolo diametro implicano, rispettivamente, un elevato parallelismo nella comunicazione e velocità di comunicazione più elevata.

– Sistemi a memoria distribuita

- \* MPP (Massively Parallel Processing): le macchine MPP sono composte da centinaia di processori (che possono diventare anche centinaia di migliaia in alcune macchine) collegati da una rete di comunicazione. Le macchine più veloci del pianeta sono basati su queste architetture.
- \* COW (Cluster Of Workstations): le architetture COW sono sistemi di elaborazione basati su classici computer collegati da reti di comunicazione. I cluster di calcolo ricadono in questa classificazione.

– Sistemi a memoria condivisa

- \* UMA (Uniform Memory Access): questi sistemi di elaborazione sono dotati di una memoria centralizzata che i processori utilizzano tramite un sistema a bus. La caratteristica fondamentale di questo sistema è il tempo di accesso alla memoria che è costante per ogni processore e per qualsiasi zona di memoria. Questo sistema è relativamente semplice da implementare ma non è molto scalabile. Questi sistemi al massimo gestiscono una dozzina di processori. Sono macchine che richiedono lo stesso tempo per accedere alla memoria principale, a prescindere dal processore che emette la richiesta e dalla parola richiesta.
- \* NUMA (NonUniform Memory Access): è un' architettura di

memoria sviluppata per i sistemi multiprocessore dove i tempi di accesso dipendono dalla posizione della memoria rispetto al processore. Nelle architetture NUMA, un processore può accedere rapidamente alla propria memoria locale, più lentamente alle memorie degli altri processori o alla memoria condivisa. L'architettura NUMA fornisce ad ogni processore una piccola zona di memoria ad accesso esclusivo e veloce, in modo da evitare la creazione di "colli di bottiglia". Nel caso di applicazioni che richiedono la condivisione di dati (come nel caso di server e simili), l'architettura NUMA migliora le prestazioni se si suddivide la memoria centrale in diversi banchi e si assegna ad ogni banco un numero ridotto di processori. Naturalmente i dati non sono realmente separati nelle memorie dei singoli processori, e se dei dati devono essere elaborati da più processori questo è possibile. In questo caso, l'architettura NUMA prevede che il software o dei dispositivi hardware provvedano a spostare i dati da un banco ad un altro. Questa copia dei dati rallenta i processori e, quindi, l'efficienza delle architetture NUMA dipende molto dai compiti svolti dal sistema. L'architettura NUMA può essere vista come una versione ridotta di cluster di computer. L'aggiunta della paginazione della memoria virtuale ad un sistema a cluster permette di emulare un'architettura NUMA anche se questa non esiste in hardware. Tuttavia, questa soluzione è molto più lenta: la comunicazione interprocesso in un'architettura NUMA software è diversi ordini di grandezza più lenta di una comunicazione interprocesso in un'architettura NUMA hardware.

- \* NORMA (NO Remote Memory Access)
  - \* COMA (Cache only memory Access): questa tipologia di elaboratori sono dotati solamente di memorie cache. Analizzando le architetture NUMA, si è notato che queste mantenevano delle copie locali dei dati nelle cache e che questi dati erano memorizzati come doppioni anche nella memoria principale. Questa architettura elimina i doppioni mantenendo solo le memorie cache.
- Macchine dataflow: le macchine a dataflow utilizzano un approccio innovativo nella programmazione. Il programma non è composto da una serie di istruzioni da eseguire sequenzialmente, ma da un *approccio data-driven*. In sostanza, le operazioni sono ese-

guita solamente quando i dati per le elaborazioni sono disponibili. Non esiste un program counter che tiene traccia dello stato del programma.

- Macchine a riduzione: le macchine a riduzione utilizzano un approccio simile a quello delle macchine dataflow, ma utilizzano un punto di vista diverso. Invece di utilizzare un approccio data-driven, utilizzano un *approccio demand-driven*. Questo approccio prevede che le computazioni siano eseguite solamente quando vi è una richiesta dei risultati da elaborare. Anche in questa tipologia di macchina non vi è un program counter.

### 13.1.1 Processore vettoriale

Un **processore vettoriale** o **array processor** è una CPU progettata per svolgere operazioni matematiche su più dati elementari contemporaneamente. Questo in contrasto con l'architettura classica di un processore scalare che prevede l'elaborazione di un singolo dato per volta.

In genere, le CPU sono capaci di manipolare uno o due frazioni di dati alla volta. Per esempio, la maggior parte delle CPU hanno un'istruzione **ch** che, essenzialmente, dice “somma A a B e memorizza il risultato in C”, mentre altre necessitano di due o tre istruzioni per eseguire questo genere di operazioni. I dati per A, B e C possono, teoricamente, essere contenuti nell'istruzione stessa; tuttavia, le cose sono raramente così semplici. In genere, il dato non è inviato in forma grezza, ma è “puntato” passando un indirizzo alla locazione di memoria che contiene il dato. Decodificare questo indirizzo e tirare fuori il dato dalla memoria occupa del tempo. Con l'incremento della velocità delle CPU, questa latenza di memoria è diventata un impedimento sempre più grande per le prestazioni finali. Per ottimizzare i tempi, le più moderne CPU utilizzano una tecnica conosciuta come *pipeline di istruzioni*, nella quale l'istruzione passa attraverso differenti sotto-unità a turno. La prima sotto-unità legge l'indirizzo e lo decodifica, la successiva reperisce i valori situati a quell'indirizzo e la successiva ancora elabora il calcolo matematico. Il “trucco” della pipeline consiste nel cominciare la decodifica della successiva istruzione prima che la precedente sia stata effettivamente elaborata dalla CPU, cosicché la decodifica degli indirizzi sia costante. Ogni istruzione necessita la stessa quantità di tempo per essere completata, ma la CPU può processare un'intera serie di operazioni molto più velocemente rispetto a quanto farebbe calcolandone una alla volta. I processori vettoriali portano questo concetto un passo più avanti. Anziché utilizzare la tecnica di pipeline solo con le istruzioni, la utilizzano anche sui dati stessi; sono quindi

capaci di applicare una medesima istruzione ad un grosso lotto di dati in parallelo, senza decodificare ogni singola istruzione dalla memoria. Questo spesso consente di risparmiare grosse quantità di tempo.

Un processore vettoriale contiene al suo interno una memoria condivisa, varie unità di calcolo, e un' unità di controllo che gestisce il parallelismo tra le unità di calcolo. Il parallelismo è gestito a livello hardware e non è visibile a livello di programmazione.

Esistono due modi in cui un processore vettoriale accede ai suoi operandi, a seconda della sua progettazione interna. Nell' organizzazione da memoria a memoria (*memory to memory*), gli operandi sono recuperati dalla memoria, elaborati, e salvati direttamente nella memoria stessa. Nell' organizzazione da registro a registro (*register to register*) gli operandi sono prima segmentati, e poi caricati in registri vettore; durante l' elaborazione, i risultati sono poi salvati in un terzo registro vettore. Solo in seguito sono immagazzinati nella memoria. Il vantaggio della gestione da memoria a memoria è dato dal fatto che permette al processore di calcolare vettori molto lunghi senza mai fermarsi per “spezzarli” in sotto-vettori, ma i tempi di latenza della memoria sono maggiori rispetto a quelli dei registri. Questo porta ad un rallentamento nei tempi di recupero dei dati dalla memoria. In seguito, questa tecnica fu abbandonata a causa della latenza di memoria troppo elevata, che rendeva questi processori troppo lenti in operazioni su vettori piccoli. I processori vettoriali da registro a registro sono, invece, decisamente più efficienti nelle operazioni su vettori relativamente piccoli.

Non tutti i problemi possono essere ottimizzati con questo tipo di soluzione. Innanzi tutto, aggiungere questo genere di istruzioni rende più complessa l' architettura della CPU, la cui complessità potrebbe rendere altre istruzioni più lente. Le istruzioni più complesse aggiungono, inoltre, ulteriore complessità ai decodificatori, che potrebbero essere più lenti nella decodifica delle istruzioni più comuni. Inoltre, i processori vettoriali lavorano meglio solo quando ci sono grosse quantità di dati su cui lavorare. Per queste ragioni, questo tipo di CPU è utilizzato nei supercomputer, e non nelle applicazioni di utilizzo comune, dato che i supercomputer stessi sono situati dove si lavora spesso con grossi quantitativi di dati. Infine, la realizzazione di questi processori è costosa; essendo utilizzati solo per scopi ben specifici, i costi di progettazione sono una parte consistente del costo totale, e questo tipo di architettura necessita di grossi quantitativi di memoria cache ad alta velocità, anch' essa costosa.





# Parte IV

## Tracciabilità



Si definisce **tracciabilità** o **AIDC** (Automatica Identification & Data Capture) la determinazione della posizione e dello stato passato e corrente di un oggetto. È meglio definita come identificazione automatica e raccolta dai di un oggetto, ed immissione automatica in un sistema di elaborazione.

L' AIDC presenta diversi vantaggi:

- affidabilità: si evitano errori di battitura;
- efficienza: i dati sono acquisiti con velocità maggiore rispetto all' immissione manuale;
- pervasività: il tracciamento non interferisce con le attività principali (guida, manipolazione, ecc. . . ).

In laboratorio, la tracciabilità è utilizzata per l' identificazione di reagenti, di campioni da esaminare, e delle persone che accedono al laboratorio. Inoltre, è impiegata nel campo della localizzazione, quindi risolvere problemi di furti di materiale (o apparecchiature) e verificare l' incompatibilità tra sostanze. Infine, è adoperata per effettuare l' inventario di reagenti e di campioni.

Alcune problematiche relative alla tracciabilità riguardano la compatibilità tra il trattamento automatico ed il metodo di lettura/scrittura utilizzato, ed il formato dei codici numerici.



# Capitolo 14

## I codici a barre

I **codici a barre** sono un insieme di elementi grafici a contrasto in modo da poter essere letti da un sensore a scansione e decodificati per restituire l'informazione contenuta.

I codici a barre lineari sono definiti come una serie di barre di vario spessore separate da spazi e delimitate da spazi più ampi all'inizio ed alla fine. Esiste una corrispondenza tra i gruppi di linee ed i caratteri alfanumerici.

I codici a barre sono letti mediante scansione ottica: un diodo (o un laser) emette un raggio di luce, ed un fotorilevatore misura l'intensità della luce riflessa e la trasforma in segnale elettrico.

### 14.1 Tipi di codici a barre

Tra i tipi più diffusi in Italia, senz'altro troviamo il **codice EAN** (European Article Number), che è utilizzato nella grande distribuzione, seguito dal **Farmacode** o *codice 32* (una rielaborazione matematica del codice 39), adottato per l'identificazione dei farmaci e delle specialità vendibili al banco nelle farmacie. Nell'ambito industriale, hanno trovato grande diffusione il **codice 128**, il **codice 39** (alfanumerico) ed il **2/5** (si legge 2 di 5) **interleaved**.

La maggior parte dei codici ha un codice di controllo (*check digit*) che l'unità di lettura è in grado di ricalcolare e verificare per assicurare la corretta lettura e l'integrità dei dati.

### 14.2 Lettura dei codici a barre

La tipologia dei lettori di codici a barre è andata ampliandosi con l'avvento di nuove tecnologie e con la miniaturizzazione della componentistica elettronica.

### 14.2.1 Tipo di collegamento

Oltre ai lettori collegati ad un personal computer o ad un registratore di cassa, troviamo dei lettori dotati di memoria e, quindi, in grado di immagazzinare un certo numero di letture prima che vi sia la necessità di scaricarle utilizzando un' unit base (detta *calamaio* nel caso delle penne ottiche). Altri lettori sono dotati di un trasmettitore di piccola potenza per comunicare in tempo reale ad un' unità ricevente i dati che sono letti.

### 14.2.2 Tecnologia di lettura

La tecnologia prevalente e più affidabile impiega uno o più raggi laser, abbinato di solito ad una testina oscillante ed, in taluni casi, ad un sistema di specchi, al fine di moltiplicare le probabilità che qualsiasi codice stampato su un oggetto sia letto al primo tentativo. Esistono anche dei lettori più economici che utilizzano una barra di LED per illuminare i codici a barre e un sensore CCD (Charged Coupled Device). Si ottengono, così, dispositivi più leggeri e più resistenti, adatti per scanner da impugnare, che però devono essere portati quasi a contatto con i codici a barre da leggere. Inoltre, in ambito industriale, le ultime tecnologie permettono la lettura del codice a barre tramite l' acquisizione di un' immagine fornita da un sistema video. Questo, tramite l' apposito software, permettono di "fotografare" l' oggetto, riconoscere nella fotografia il codice a barre da leggere e successivamente interpretarlo.

## Capitolo 15

# Radio Frequency Identification (RFID)

Il **Radio Frequency IDentification** (RFID) è una tecnologia per la identificazione automatica di oggetti, animali o persone basata sulla capacità di memorizzare ed accedere a dati usando etichette RFID o transponders o tags. Il sistema si basa sulla lettura a distanza di informazioni contenute in un tag RFID usando dei lettori RFID.

I Tag RFID sono utilizzati nei laboratori di analisi, nell' ambito dell' industria farmaceutica, nella logistica, ecc.

Un *tag RFID* è costituito da:

- processore;
- memoria;
- una antenna (la cui geometria dipende dalla frequenza usata). Possono esserci due antenne: una per trasmettere e l' altra per ricevere;
- package esterno (flessibile e robusto per resistere in ambienti ostili)
- può essere dotato o meno di una batteria.

Un tag è in grado di ricevere e di trasmettere via radiofrequenza le informazioni contenute nel chip ad un transceiver RFID.

Con il termine Tag RFID si intendono, di solito, Tag passivi, ma ci sono anche i Tag RFID attivi. Il *Tag RFID passivo* è costituito da un chip, con processore e memoria, collegato ad una antenna a spirale di dimensioni molto grandi rispetto al chip. In questo tipo di Tag non c' è batteria. Per capire il funzionamento di un Tag passivo, si osservi la figura 15.1. Il lettore, attraverso l' antenna A, genera un campo elettromagnetico che induce una

corrente elettrica nell' antenna B del Tag. La corrente, seppur debole, è sufficiente ad alimentare il processore che legge il dato dalla sua memoria interna e lo trasmette attraverso la stessa antenna B. In questo modo, l' antenna A (quindi, il lettore) riceve il dato.

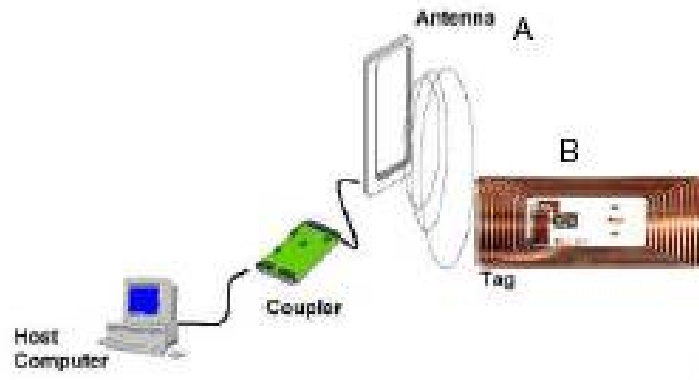


Figura 15.1: Funzionamento di un Tag RFID passivo

Esistono diversi tipi di tag RFID passivi, alcuni dei quali normati da standard ISO: 125/134 kHz (*Tag Low Frequency*, LF), 13,56 MHz (*Tag HF*), 868/915 MHz (*Tag Ultra HF*, UHF) e  $>2,4$  GHz. Invece, i Tag RFID attivi sono alimentati da batterie. Questi Tag (che hanno la stessa tecnologia delle reti di sensori) presentano chip complessi (con maggiore possibilità di calcolo), permettono maggiori distanze di lettura, ed utilizzano protocolli di trasmissione più complessi:

- IEEE 802.15.4 (parente del WLAN e Bluetooth);
- IEEE 802.15.4 + ZigBee;
- IEEE 802.15.4 + 6lowpan (protocollo IP in versione leggera);
- Zware.

La tecnologia RFID ha alcuni vantaggi semplici rispetto alle tradizionali tecnologie dei codici a barre e delle bande magnetiche:

- non deve essere a contatto per essere letto come le bande magnetiche;
- non deve essere visibile per essere letto come per i codici a barre;
- si possono anche aggiungere informazioni sui chip in funzione della tipologia del chip (*Read Only*: si possono solo leggere le informazioni



contenute; *Write Once, Read Many*: si possono scrivere nel chip le informazioni una sola volta, ma leggerle un numero illimitato di volte; *Read and Write*: si possono leggere e memorizzare informazioni per un numero limitato ma grande di volte);

- l' identificazione e la verifica avviene in 1/10 di secondo;
- la comunicazione può essere in chiaro o cifrata.

## 15.1 ZigBee

**ZigBee** è il nome di una specifica per un insieme di protocolli di comunicazione ad alto livello che utilizzano piccole antenne digitali a bassa potenza e basato sullo standard IEEE 802.15.4 per Wireless Personal Area Networks (WPAN). La relazione esistente fra ZigBee e IEEE 802.15.4-2003 è simile a quella esistente tra IEEE 802.11 e la Wi-Fi Alliance.

### 15.1.1 Introduzione

ZigBee opera nelle frequenze radio assegnate per scopi industriali, scientifici e medici (ISM); 868 MHz in Europa, 915 MHz negli Stati Uniti e 2,4 GHz nella maggior parte del resto del mondo. Questa tecnologia ha lo scopo di essere più semplice e più economica di altre WPAN come, ad esempio, Bluetooth. Il nodo ZigBee del tipo più complesso si dice che richieda solamente il 10% del codice necessario per un tipico nodo Bluetooth o Wi-Fi, mentre il più semplice dovrebbe richiedere intorno al 2%. Tuttavia, attualmente le dimensioni reali sono più alte e si aggirano intorno al 50% del codice necessario per Bluetooth. I produttori di chip ZigBee prevedono dispositivi da 128 kB.

### 15.1.2 Usi

I protocolli ZigBee sono progettati per l' uso in applicazioni embedded che richiedano un basso transfer rate e bassi consumi. L' obiettivo attuale di ZigBee è di definire una *Wireless Mesh Network* non mirata, economica e autogestita che possa essere utilizzata per scopi quali il controllo industriale, le reti di sensori, domotica, le telecomunicazioni, ecc. La rete risultante avrà un consumo energetico talmente basso da poter funzionare per uno o due anni sfruttando la batteria incorporata nei singoli nodi.

### 15.1.3 Tipi di dispositivo

Ci sono tre differenti tipi di dispositivo ZigBee:

- **ZigBee Coordinator (ZC)**: è il dispositivo più “intelligente” tra quelli disponibili, costituisce la radice di una rete ZigBee e può operare da ponte tra più reti. Ci può essere un solo Coordinator in ogni rete. Esso è inoltre in grado di memorizzare informazioni riguardo alla sua rete e può agire come deposito per le chiavi di sicurezza.
- **ZigBee Router (ZR)**: questi dispositivi agiscono come router intermedi passando i dati da e verso altri dispositivi.
- **ZigBee End Device (ZED)**: includono solo le funzionalità minime per dialogare con il suo nodo parente (Coordinator o Router), non possono trasmettere dati provenienti da altri dispositivi; sono i nodi che richiedono il minor quantitativo di memoria e quindi risultano spesso più economici rispetto ai ZR o ai ZC.

# Capitolo 16

## EPCglobal

L' **Electronic Product Code** (EPC) è una famiglia di codici creati come un eventuale successore del codice a barre. L' EPC è stato creato con un metodo per poter usare bene la tecnologia RFID. Il sistema EPC è attualmente utilizzato dalla società EPCglobal.

L' **EPCglobal** è una combinazione tra il GS1 (conosciuto come EAN International) ed il GS1 US (conosciuto come Uniform Code Council, Inc.). È una organizzazione di standardizzazione della tecnologia Electronic Product Code. Si tratta di una standardizzazione di:

- eventi applicabili all' oggetto tracciato;
- API di lettura ed interrogazione;
- meccanismi di sicurezza.

### 16.1 EPCIS

L' **EPC Information Services** (EPCIS) è un EPCglobal standard designed utilizzato per l' identificazione univoca tramite codici elettronici. È indipendente dal produttore dei Tag e dell' hardware e software impiegati per utilizzarli; inoltre, è indipendente dall' ente utilizzatore. Un' altra sua caratteristica è quella di presentare profili specifici per specifiche tipologie di utenti.

EPCIS definisce un' interfaccia applicativa standard API per leggere codici, scrivere codici ed eliminare codici; inoltre, definisce un canale di comunicazione.

I produttori di *reader* dovrebbero uniformarsi a tale standard per garantire interoperabilità.

### 16.1.1 Interfaccia di programmazione di un' applicazione

Le **Application Programming Interface** (API – Interfaccia di Programmazione di un' Applicazione) sono ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per un determinato compito. È un metodo per ottenere un' astrazione, di solito tra l' hardware ed il programmatore, o tra software a basso ed alto livello. Le API permettono di evitare ai programmatori di scrivere tutte le funzioni dal nulla. Le API stesse sono un' astrazione: il software che fornisce una certa API è detto *implementazione dell' API*.

Una buona API fornisce una “scatola nera” o un livello di astrazione che evita al programmatore di sapere come funzionano le API ad un livello più basso. Questo permette di riprogettare o migliorare le funzioni all' interno dell' API senza cambiare il codice che si affida ad essa.

# Bibliografia

- [1] Sito Internet <http://www.wikipedia.com>
- [2] Dispense del corso di *Architetture Hardware di Laboratorio* (a.a. 2008/2009)
- [3] Abraham Silberschatz, Galvin P. Baer, Gagne Greg (2006), *Sistemi operativi - Concetti ed esempi (Settima edizione)*, Pearson Education Italia S.r.l., Milano.
- [4] Vincenzo Manca (2003), *Metodi Informazionali*, Bollati Boringhieri, Torino.