Reti

La **topologia di rete** è il modello geometrico ([grafo](https://it.wikipedia.org/wiki/Grafo)) finalizzato a rappresentare le relazioni di [connettività](https://it.wikipedia.org/wiki/Connessione_(informatica)), fisica o logica, tra gli elementi costituenti la rete stessa (detti anche *nodi*).

La rete, dunque rappresenta il mezzo attraverso il quale un dato passa da un processore alla memoria o viceversa.

Le reti di interconnessione sono classificabili come:

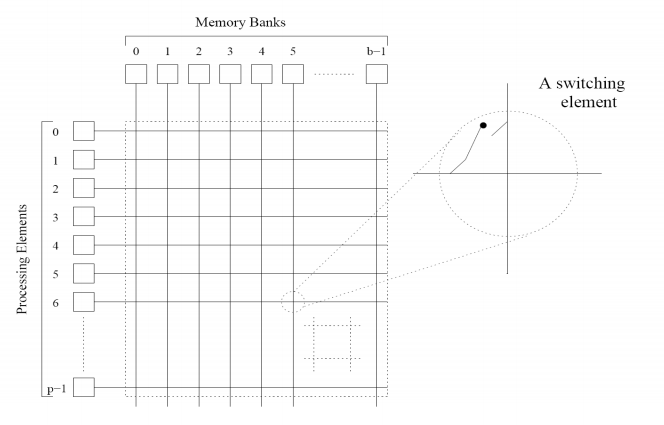
* ***Statiche***
  + Connessione **diretta** punto a punto tra nodi
* ***Dinamiche***
  + Connessione **indiretta** tramite switches e link di comunicazione.

**Reti di bus**

Nelle macchine caratterizzate da questa rete un bus comune fa da mezzo di comunicazione fra i vari processori. Hanno una bassa complessità ma anche una bassa scalabilità, e allo stesso tempo è alto il rischio di **bottleneck.**

## **Crossbar**

E’ una struttura matriciale (p x b) che collega p input con b output in modo non bloccante. Hanno un’alta complessità ma anche un’alta scalabilità, Nessun rischio di **blottleneck** ma è troppo costosa. In questa topologia processori e banchi di memoria sono collegati tramite degli switches.

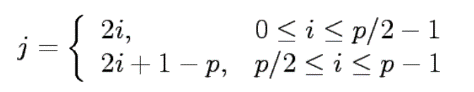


## **Reti multistage**

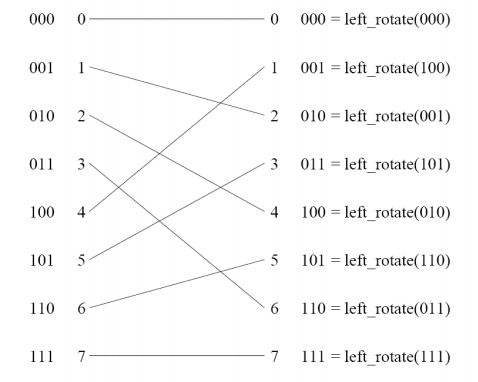
* Rappresentano il giusto compromesso fra performance (scalabilità) e costo.



* La rete **OMEGA** è quella più diffusa:
  + Contiene log (p) stage con p = numero di input/output
  + Per ogni stage l’input i è collegato all’output j se:



(*Left rotation* o *perfect shuffle*): rotazione del numero binario verso sinistra



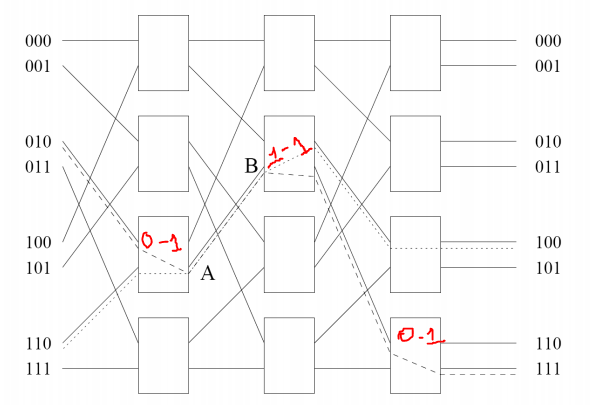
* Ogni stage è composto da più switches ognuno dei quali opera in due modi:
  + **Cross over** (b): segnale deviato
  + **Pass through (a)**: segnale passa dritto



Cos’è uno *switch*?

* In una rete di interconnessione è un dispositivo formato da una serie di porte input output che è responsabile dell’instradamento di un messaggio fra nodi.

**Routing**

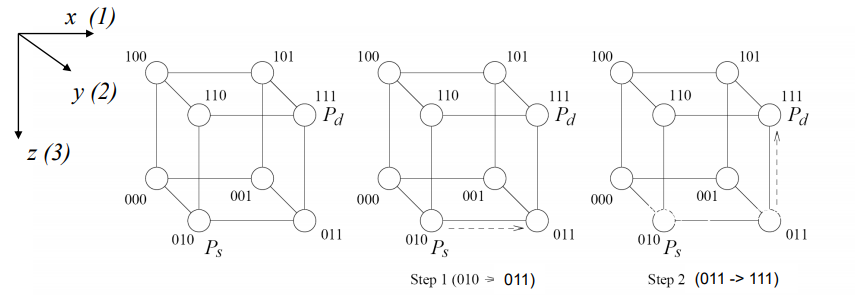
* Consiste nel tracciare la connessione fra due nodi instradando dei pacchetti tramite degli switch.
* Avviene in 3 fasi:
  + Conversione in numeri binari del nodo sorgente e del nodo destinatario
  + Confronto del primo bit più significativo dei due nodi:
    - Se sono gli stessi l’instradamento avviene in modalità pass-through
    - Altrimenti in modalità cross over
  + Si ripete per ogni stage.
  + 

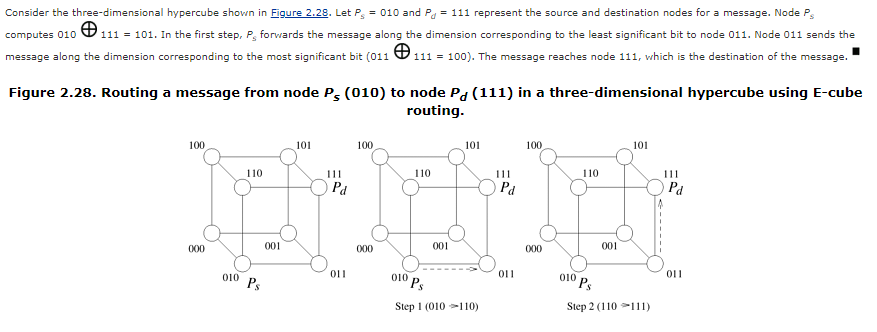
I principali tipi di rete sono:

* ***Reti a stella***: ogni nodo comunica con un altro attraverso un nodo centrale
* ***Reti completamente interconnesse***: ogni nodo comunica con tutti gli altri
* ***Mashes****:* reti molto particolari che possono avere più dimensioni: (2 STEPS ROUTING = passaggio messaggio per nodo intermedio)
  + *Linear array:* mesh a una dimensione (1 D) in cui ogni nodo ha due vicini (uno a destra e uno a sinistra)
  + *Anello 1D:* linear array chiuso in quanto i due nodi terminali sono connessi
  + *Mesh k-*D: hanno più dimensioni in quanto sono costituiti dall’unione di più meshes 1 D. Ogni nodo ha 4 vicini (N, S, E, O)
* ***Ipercubo****:* caso particolare di mesh in cui il numero di dimensioni dipendono necessariamente dal numero di nodi
  + Con n nodi si hanno necessariamente log n dimensioni [**DIFFERENZA MESH-IPERCUBO**]
  + Proprietà fondamentali di un ipercubo:
    - La distanza tra due nodi è al massimo **log(p).**
      * P=numero di nodi
    - Ogni nodo ha esattamente **log(p)** nodi vicini.
    - La distanza tra due nodi è data dal numero di bit che differiscono (1111-0000 = 4 bit)
* ***Reti ad albero***: si ispessiscono man mano che si scende verso il basso *(fat-tree)*

**E-cube Routing**

* Tecnica di Routing applicata agli ipercubi
* Il funzionamento è il seguente:
  + Rappresentazione binaria del nodo sorgente e destinatario
  + XOR tra le rappresentazioni binarie
  + Si prende il bit meno significativo diverso da 0 del risultato dello XOR e si segue la direzione specificata per instradare il messaggio



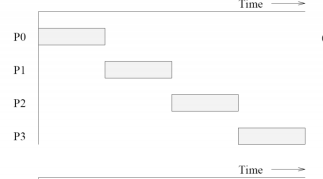


**XOR**

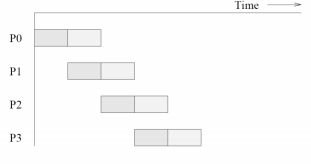
* Confronta due bit dando come risultato un terzo bit che sarà 1 nel caso in cui i due bit sono diversi, 0 in tutti gli altri casi.

# MODALITA’ ROUTING

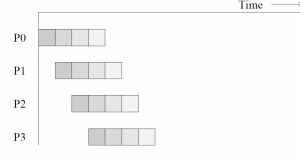
## STORE AND FORWARD

E’ il più costoso perché ogni switch attende l’arrivo completo del messaggio prima di inviarlo.

## PACKET ROUTING

Utilizza una metodologia **pipeline**, prende un messaggio e lo divide in pacchetti, ogni pacchetto viene inviato appena pronto.I pacchetti possono prendere strade diverse.

## CUT-THROUGH ROUTING

Utilizza il concetto di packet Routing estremizzandolo, ogni dato viene suddiviso in piccolissimi pacchetti chiamati **flit (4-32 byte)**. Ogni flit prende la stessa strada degli altri.

SOLITAMENTE E’ PIU’ EFFICIENTE AGGREGARE UN MESSAGGIO ED INVIARLO CHE INVIARNE MOLTI PICCOLI.

Ogni comunicatore è caratterizzato da:

* Gruppo di processi
* Contesto (oggetto definito dal sistema per identificarlo)
* **Topologia virtuale** (topologia di rete di interconnessione creata dal programmatore per rispondere alle esigenze del proprio algoritmo e che viene mappata sulla rete fisica già esistente sulla macchina.

La topologia virtuale più gettonata è la topologia **cartesiana** (tipo una matrice) caratterizzata da:

* K dimensioni
* Toroidalità

Poiché in MPI le funzioni di comunicazione richiedono ranghi unidimensionali (quindi non caratterizzati da 2 coordinate), esistono funzioni apposite che date due coordinate restituiscono il rango unidimensionale. [MPI\_Cart\_rank, MPI\_Cart\_coord]