VII. Coerența Cache. Retele de interconectare

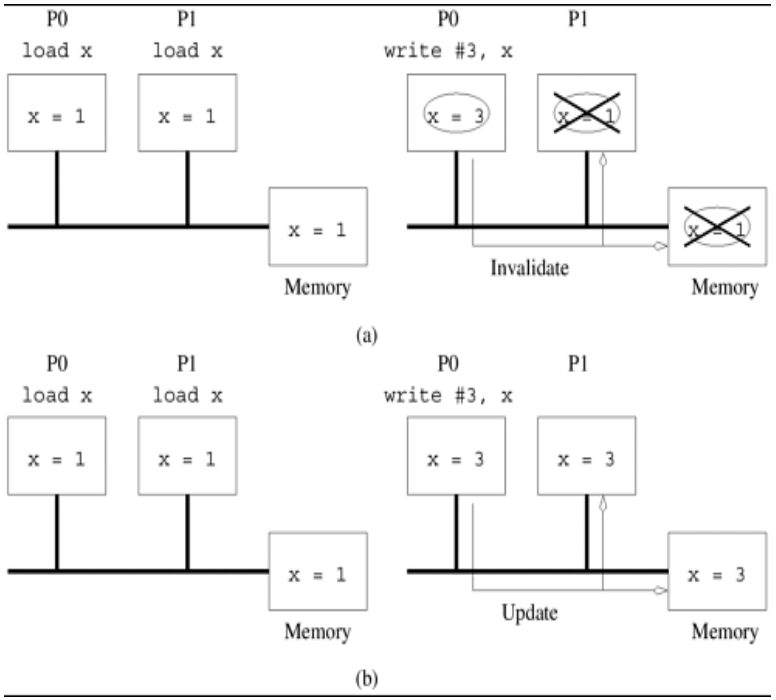
Continut

- coerența cache-ului în sistemele multiprocesoare
- Retelele de interconectare
  - clasificare
  - topologii
  - evaluarea rețelelor de interconectare statică și dinamică

Coerența cache-ului în sistemele multiprocesoare

Problema

- Rețelele de interconectare oferă mecanisme de bază pentru comunicarea mesajelor.
- În cazul calculatoarelor cu spații de adresare partajat, este necesar un hardware suplimentar pentru a păstra mai multe copii ale datelor în concordanță între ele.
- Dacă există două copii ale datelor (în cache-uri / elemente de memorie diferite), cum ne asigurăm că procesoare diferite funcționează pe acestea într-o manieră care urmează semantica predefinită?
- Exemplu:
  - Două procese P0 și P1 sunt conectate printr-o magistrala partajata la o memorie accesibilă la nivel global.
  - Ambele procesoare încarcă aceeași variabilă.
  - Acum există trei copii ale variabilei.
  - Mecanismul de coerență trebuie să se asigure că toate operațiunile executate pe aceste copii pot fi serializate.



Protocol

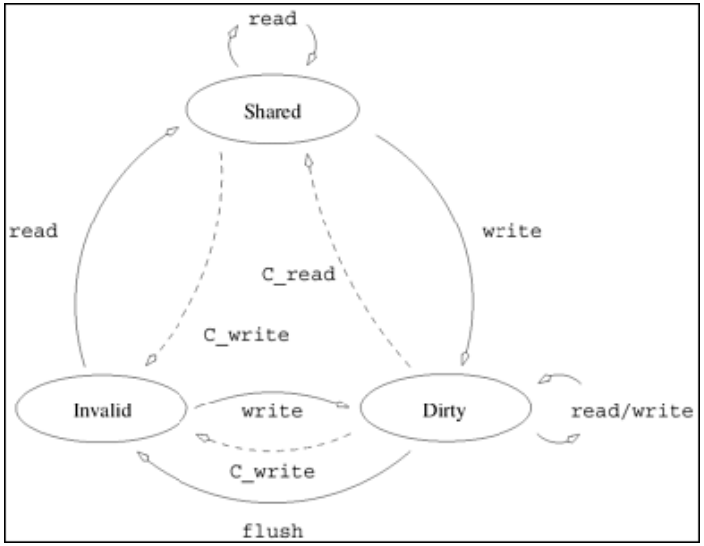
- Protocol de actualizare (b)
  - de fiecare dată când data este scrisa, toate copiile sale din sistem sunt actualizate.
  - Dacă un procesor pur și simplu citește o dată și nu o folosește niciodată, actualizările ulterioare ale acestui articol la alte procesoare provoacă surplus în ceea ce privește latența la sursă și lățimea de bandă în rețea.
- Protocolul de invalidare (a)
  - invalidează data la prima actualizare de catre un procesor la distanță.
  - actualizările ulterioare nu trebuie să fie efectuate pe această copie.

Partajare falsă

- Se referă la situația în care diferite procesoare actualizează diferite părți ale aceleiași linii de memorie cache.
  - deși actualizările nu sunt efectuate pe variabile partajate, sistemul nu detectează acest lucru.
- Într-un protocol de invalidare:
  - când un procesor își actualizează partea din linia cache, celelalte copii ale acestei linii sunt invalidate.
  - când alte procesoare încearcă să își actualizeze părțile din linia cache, linia trebuie să fie preluată de la procesorul de la distanță.
  - partajarea falsă poate face ca o linie de cache să fie in ping-pong între diferitele procesoare.
- Într-un protocol de actualizare,
  - această situație este puțin mai bună: toate citirile pot fi efectuate local, iar scrierea trebuie actualizată.
  - acest lucru salvează o operație de invalidare care este altfel irosită.

Protocol de coerență în trei stări

- Starile etichetate nevalide, murdare și partajate.
- Liniile solide prezintă acțiunile procesorului.
- Linii punctate, acțiuni de coerență.
- De exemplu,
  - Când un procesor execută o citire pe un bloc nevalid,
    - blocul este preluat,
    - o tranziție se face de la invalid la partajat.
  - Dacă un procesor face o scriere pe un bloc partajat,
    - protocolul de coerență propagă un bloc C_write (o scriere de coerență) pe bloc.
    - aceasta declanșează o tranziție de la partajat la invalid la toate celelalte blocuri.



Implementarea protocoalelor de coerență

- poate fi efectuat folosind o varietate de mecanisme hardware:
  - sisteme de spionaj (snoopy),
  - sisteme bazate pe directoare,
  - combinații ale acestora.

Sisteme de cache snoopy

- Cache-urile sunt de obicei asociate cu sisteme multiprocesoare bazate pe rețele de interconectare de difuzare, cum ar fi o magistrala sau un inel.
- Toate procesoarele snoop (monitorizează) magistrala pentru tranzacții.
  - Acest lucru permite procesorului să facă tranziții de stare pentru blocurile sale de memorie cache.
- Studiat extensiv și utilizat în sisteme comerciale.
  1. Dacă diferite procese operează pe diferite date, acestea pot fi memorate în cache.
     - Odată ce aceste articole sunt etichetate murdare, toate operațiunile ulterioare pot fi efectuate local pe cache fără a genera trafic extern.
  2. Dacă un element de date este citit de mai multe procesoare, acesta trece la starea partajată din cache și toate operațiunile de citire ulterioare devin locale.
- În ambele cazuri, protocolul de coerență nu adaugă niciun fel de surplus.
- Dezavantaje:
  - Dacă mai multe procesoare citesc și actualizează același element de date, acestea generează funcții de coerență între procesoare.
  - Deoarece o magistrala partajata are o lățime de bandă finită, doar un număr constant de astfel de operații de coerență se pot executa într-o unitate de timp.

Sisteme bazate pe directoare

- memoria globală este mărită cu un director care menține o harta de biți care reprezintă blocurile de memorie cache și procesoarele la care sunt memorate în cache.
  - Aceste intrări bitmap sunt uneori denumite biți de prezență.
- Doar procesoarele care dețin un anumit bloc (sau îl citesc) participă la tranzițiile de stare din cauza operațiilor de coerență.
  - Rețineți că pot exista și alte tranziții de stare declanșate de citirea, scrierea sau fluxul procesorului (retragerea unei linii din cache), dar aceste tranziții pot fi gestionate local cu operația reflectată în biții de prezență și starea în director.
- Dacă procesoare diferite operează pe blocuri de date distincte,
  - aceste blocuri se murdăresc în cache-urile respective,
  - toate operațiunile după prima pot fi efectuate local.
- Dacă mai multe procesoare citesc (dar nu actualizează) un singur bloc de date,
  - blocul de date este replicat în cache-urile în starea comună
  - citirile ulterioare pot avea loc fără a declanșa niciun surplus al coerenței.

Scheme de directoare distribuite

- În arhitecturi scalabile, memoria este distribuită fizic pe procesoare.
- Biții de prezență corespunzători ai blocurilor sunt, de asemenea, distribuiti.
- Fiecare procesor este responsabil de menținerea coerenței propriilor blocuri de memorie.
  - Deoarece fiecare bloc de memorie are un proprietar, locația directorului său este implicit cunoscută de toate procesoarele.
  - Când un procesor încearcă să citească un bloc pentru prima dată, acesta solicită proprietarului blocul.
  - Proprietarul direcționează în mod adecvat această solicitare pe baza prezenței și informațiilor de stat disponibile local.
  - Când un procesor scrie într-un bloc de memorie, acesta propagă o invalidare către proprietar, care la rândul său transmite invalidarea tuturor proc-urilor care au o copie în cache a blocului.
- Costurile de comunicare asociate cu mesajele de actualizare a stării nu sunt reduse.
- Directoarele distribuite permit operațiunile de coerență simultană O (p), cu condiția ca rețeaua de bază să poată susține mesajele de actualizare a stării asociate.
  - Din acest punct de vedere, directoarele distribuite sunt, în mod inerent, mai scalabile decât sistemele de snoopy sau sistemele de directoare centralizate.
  - Latența și lățimea de bandă a rețelei devin blocaje fundamentale de performanță pentru astfel de sisteme.

Rețele de interconectare

Rețele de interconectare

- Asigurați mecanisme de transfer de date între:
  - noduri de procesare sau
  - procesoare și module de memorie.
- O cutie-neagră a unei rețele de interconectare constă din n intrări și ieșiri m.
  - Rezultatele pot fi sau nu distincte de intrări.
- Rețelele tipice de interconectare sunt construite folosind legături și comutatoare.
- Legături:
  - O legătură corespunde mediilor fizice, cum ar fi un set de fire sau fibre capabile să transporte informații.
  - Cuplarea capacitivă dintre fire limitează viteza de propagare a semnalului.
  - Această cuplare capacitivă și atenuarea puterii semnalului sunt funcții ale lungimii legăturii.

Comutator

- O rețea de interconectare formată din două seturi de porturi de intrare și ieșire.
- Grad: nr. total a porturilor de pe un comutator
- Comutatoarele oferă o gamă largă de funcționalități:
  - Funcționalitate minimă: o mapare de la intrare la porturile de ieșirefunctionality: a mapping from the input to the output ports.
  - Comutatoarele pot oferi, de asemenea, suport pentru:
    - memorie internă (când portul de ieșire solicitat este ocupat),
    - rutare (pentru a atenua congestionarea în rețea),
    - multicast (aceeași ieșire pe mai multe porturi).
- Maparea este furnizată folosind o varietate de mecanisme bazate pe:
  - traversele (crossbar) fizice,
  - memorii multiportate,
  - multiplexor-demultiplexor,
  - magistrale multiplexate.
- Costul unui comutator este influențat de costul:
  - hardware-ul de mapare: crește ca pătrat al gradului;
  - hardware periferic: liniar;
  - costurile de ambalare: liniar in număr de pini.

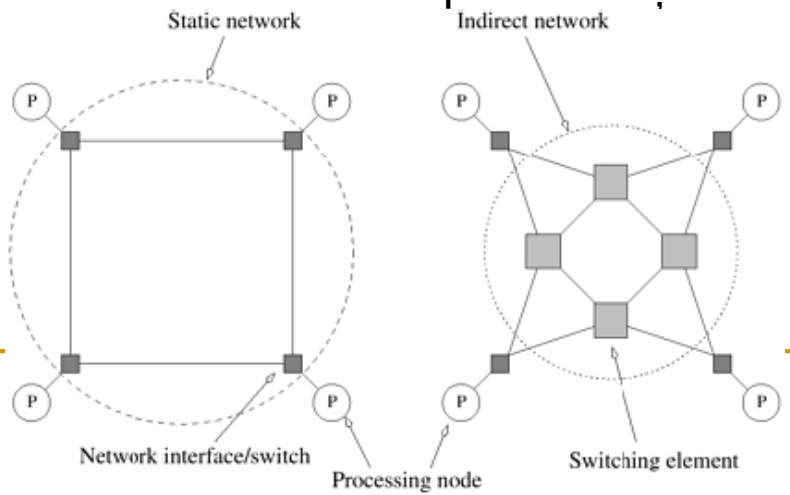
Interfata de retea

- Oferă conectivitatea dintre noduri și rețea.
- Interfața de rețea are porturi de intrare și ieșire care conectează date către și din rețea.
- Responsabilitate pentru:
  - pachetarea datelor,
  - calcularea informațiilor de rutare,
  - tamponarea datelor de intrare și de ieșire pentru potrivirea vitezei rețelei și a elementelor de procesare,
  - verificarea erorilor.
- Interfețele de rețea convenționale sunt legate de magistralele I / O.
- Interfețele din mașinile paralele sunt strâns cuplate la magistrala de memorie.

Deoarece magistralele I / O sunt de obicei mai lente decât magistralele de memorie => acestea din urmă pot suporta o lățime de bandă mai mare.

Retele statica vs. cele dinamice

- Retelele statice
  - constau din legături de comunicare punct la punct între nodurile de procesare;
  - sunt denumite și *rețele directe*.
- Retelele dinamice
  - sunt construite folosind switch-uri și legături de comunicare.
  - legăturile de comunicare sunt conectate dinamic de către comutatoare pentru a stabili căi între nodurile de procesare și băncile de memorie.
  - Sunt denumite și *rețele indirecte*.
- Figura:
  - (a) o simplă rețea statică formată din patru elemente sau noduri de procesare.
    - Fiecare nod de procesor conectat printr-o interfață de rețea la 2 noduri într-o configurație de plasă.
  - (b) o rețea dinamică de 4 noduri conectate printr-o rețea de comutatoare la alte noduri.

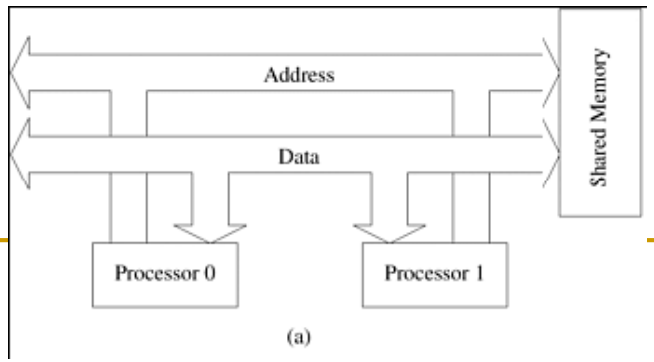


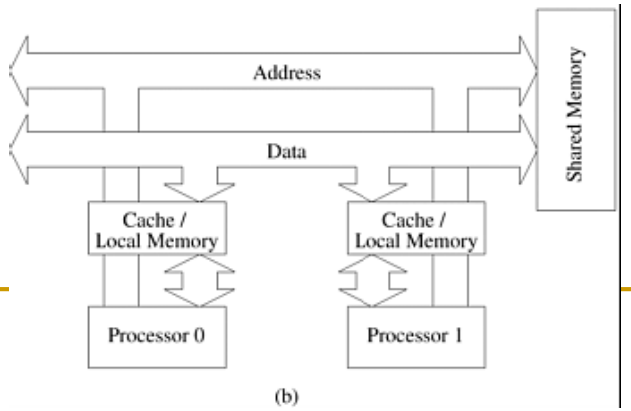
Topologii de rețea: rețele bazate pe magistrale

- În rețelele de interconectare au fost utilizate o mare varietate de topologii de rețea.
  - Aceste topologii încearcă să schimbe costurile și scalabilitatea cu performanța.
- Cea mai simplă rețea: o rețea bazată pe magistrala
  - constând dintr-un mediu comun care este comun tuturor nodurilor.
  - Avantaje:
    - costul rețelei scalează liniar ca numărul de noduri, p.
    - distanța dintre oricare două noduri din rețea este constantă.
    - ideal pentru difuzarea informațiilor printre noduri.
  - Dezavantaje:
    - lățimea de bandă delimitată a unui autobuz plasează limitări la performanța generală a rețelei pe măsură ce numărul de noduri crește.
    - mașinile obișnuite bazate pe autobuz sunt limitate la zeci de noduri.
  - Serverele Sun Enterprise și multiprocesoarele magistrala-comuna bazate pe Intel Pentium sunt exemple de astfel de arhitecturi.

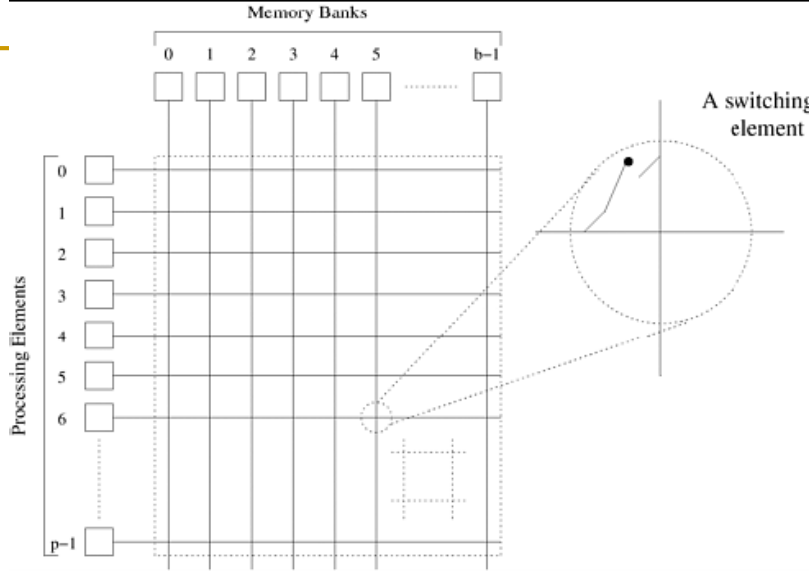
Interconectare pe baza de magistrala, cu cache

- Programe tipice: majoritatea datelor accesate sunt locale pentru nod.
  - Pentru astfel de programe, este posibil să fie furnizat un cache pentru fiecare nod.
  - Datele private sunt memorate în cache la nod și se accesează numai datele la distanță prin intermediul magistralei.
- Figura (a) :
  - *p* procesoare care partajează o magistrala către memorie.
  - Presupunem că fiecare procesor accesează *k* elemente de date,
  - Presupunem că fiecare acces la date necesită timp *tcycle*,
  - Limita de jos a timpul de execuție este *tcycle* x *k p* seconds.
- Figura (b):
  - Presupunem că 50% din accesarile de memorie (0,5 k) sunt făcute la date locale.
  - Presupunem timp de acces la memoria privată identică cu memoria globală, *tcycle*.
  - Limita de jos a timpul de execuție este 0.5 x *tcycle* x *k* + 0.5 x *tcycle* x *k p*.
  - Pentru *p* mare, organizarea figurii (b) are ca rezultat o limită inferioară.
  - Acest timp reprezintă o îmbunătățire de 50% a timpului de execuție în comparație cu organizarea din figura (a).





Retele crossbar (comutare a transverselor)



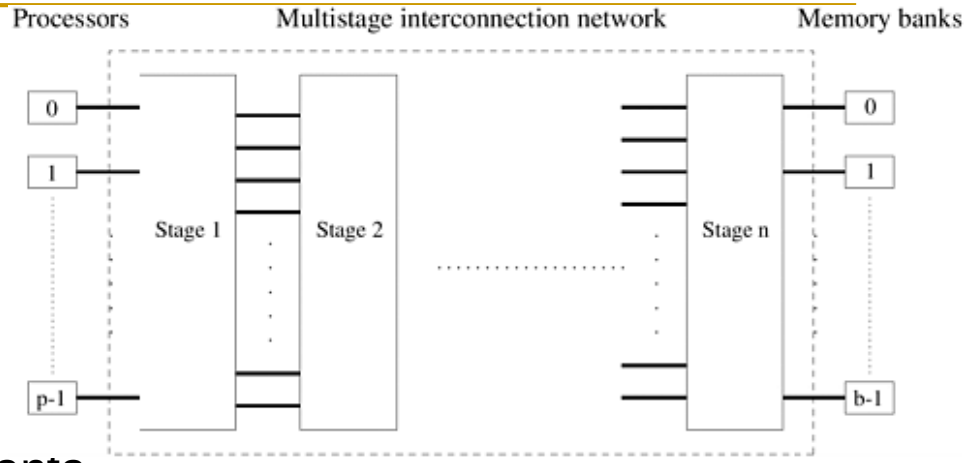
- O modalitate simplă de conectare a p procesoare la b bănci de memorie
- Utilizează o grilă de întrerupătoare sau noduri de comutare.
- Rețea care nu blochează;
  - în sensul că conexiunea unui nod de procesare la o bancă de memorie nu blochează conexiunea niciunui alt nod de procesare la alte bănci de memorie.
- Numărul total de noduri de comutare necesare pentru a implementa o astfel de rețea esteO(pb).
- Uzual $b < p$; în caz contrar, la un moment dat, vor exista unele noduri de procesare care nu vor putea accesa bănci de memorie.

⇒ Pe măsură ce valoarea p este crescută, complexitatea (numărul de componente) al rețelei de comutare crește la felO(p^2).

- Cand numărul nodurilor de procesare devine mare, această complexitate a comutatorilor este dificil de realizat la rate mari de date.

⇒ rețelele transversale nu sunt foarte scalabile din punct de vedere al costurilor.

Rețele multi-etape



- Observatii:
  - Retelele crossbar
    - Scalabile in termeni de performanta
    - Nescalabile in termeni de cost.
  - Rețeaua cu magistrala partajată este
    - Scalabile in termeni de cost
    - Nescalabile in termeni de performanta.
- Între aceste două extreme se află o clasă intermediară de rețele (rețele de interconectare multi-etape).
  - Mai scalabil decât magistrala din punct de vedere al performanței
  - Mai scalabil decât crossbar din punct de vedere al costurilor.
  - Schema generală a unei rețele multi-etape constând din p noduri de procesare și b bănci de memorie este prezentată în figura.

Reteaua Omega

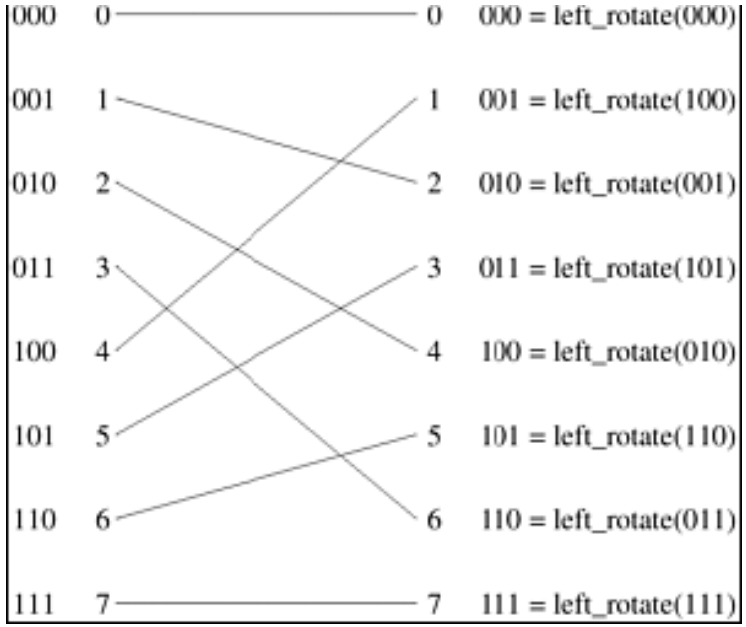
- O rețea de conexiune multi-etape utilizată frecvent.
- Constă în log p etape, în care p este numărul de intrări (noduri de procesare) și, de asemenea, numărul de ieșiri (bănci de memorie).
- Fiecare etapă constă dintr-un model de interconectare care conectează intrările p și ieșirile p.
- Există o legătură între intrarea i și ieșirea j dacă următoarele sunt adevărate:

$$j = \begin{cases} 2i, & 0 \le i \le p/2 - 1 \\ 2i + 1 - p, & p/2 \le i \le p - 1 \end{cases}$$

  - Această ecuație reprezintă o operație de rotație stângă pe reprezentarea binară a lui i pentru a obține j.
  - Acest model de interconectare este denumit un amestec perfect.
- În fiecare etapă a unei rețele omega, un model de interconectare perfectă se introduce într-un set de comutatoare p / 2.

Exemplu de amestecare perfecta și mod de comutare

- Amestecare perfecta la p=8



- Fiecare comutator al rețelelor Omega opereaza:

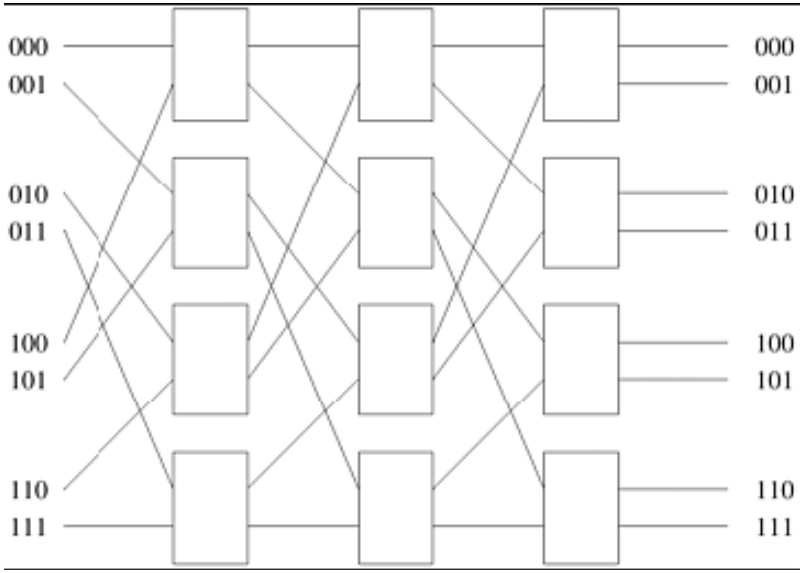
1. intrările sunt trimise direct la ieșiri - conexiune trecere.
2. intrările la nodul de comutare sunt traversate și apoi trimise - conexiune încrucișată.

Costs and routing in Omega netw

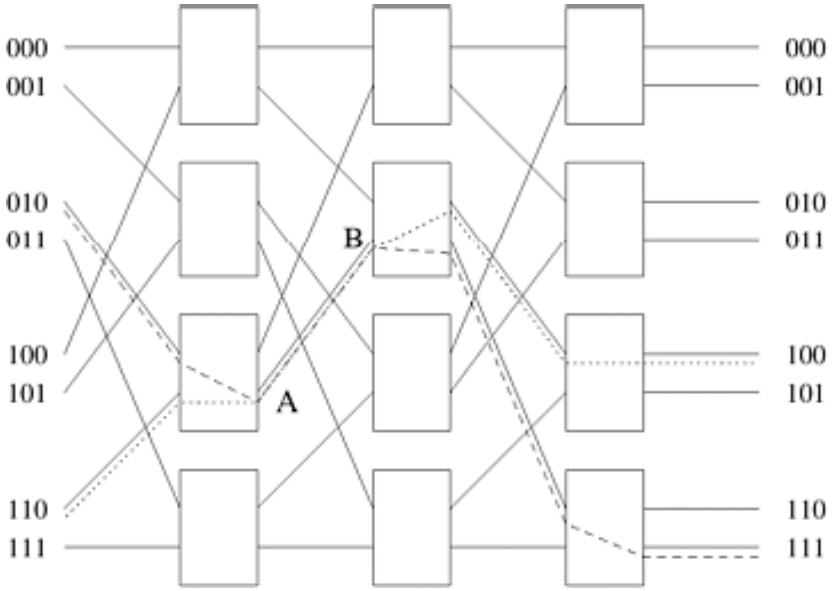
- O rețea omega are noduri de comutare *p* / 2 x log *p*, iar costul unei astfel de rețele crește cu O(*p* log *p*).
  - acest cost este mai mic decât costul O(p^2) al unei rețele crossbar.
- Datele de rutare într-o rețea omega se realizează după cum urmează:
  - Fie *s* reprezentarea binară a unui procesor care trebuie să scrie unele date în banca de memorie *t.*
  - Datele traversează legătura către primul nod de comutare.
  - Dacă cei mai importanți biți *s* și *t* sunt identici, atunci datele sunt dirijate în modul de trecere prin comutator.
  - Dacă acești biți sunt diferiți, datele sunt trimise în modul de încrucișare.
  - Această schemă se repetă la următoarea etapă de comutare folosind următorul bit mai semnificativ.
  - Traversarea a log *p* etape folosește toti log p biți din reprezentările binare ale lui *s* și *t*.

Exemple

- O rețea omega completă care conectează 8 intrări și 8 ieșiri.

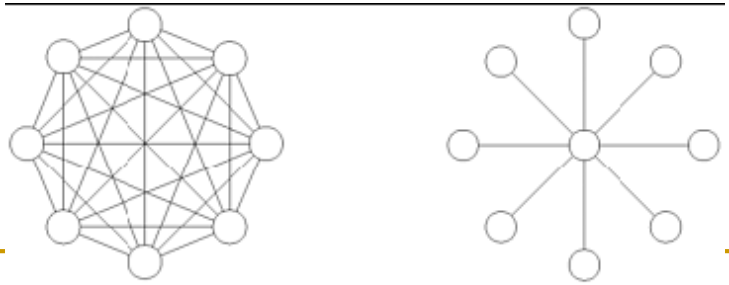


- Accesul la o bancă de memorie de către un procesor poate interzice accesul la o altă bancă de memorie de către un alt procesor.
  - Rețelele cu această proprietate sunt denumite rețele cu blocare.
- Exemplu de blocare în omega netw: unul dintre mesajele (010 până la 111 sau 110 până la 100) este blocat la link-ul AB.



Retele de conectare complete si stea

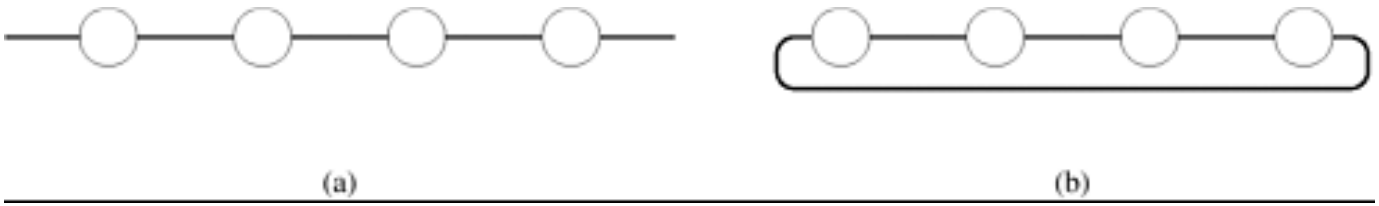
- Retele complete:
  - fiecare nod are o legătură de comunicare directă cu orice alt nod din rețea.
  - ideal în sensul că un nod poate trimite un mesaj altui nod într-o singură etapă.
  - omologul static ale rețelelor de comutare a traverselor,
    - comunicarea dintre oricare pereche I / O nu blochează comunicarea între nicio altă pereche.
- Retele stea,
  - un procesor acționează ca procesor central.
  - Fiecare alt procesor are o legătură de comunicare care îl conectează la acest procesor.
  - Similar cu rețeaua bazată pe magistrala
    - Comunicarea între orice pereche de procesoare este dirijată prin procesorul central.
  - Procesorul central: gatuire.



(a) (b)

Tablouri liniare

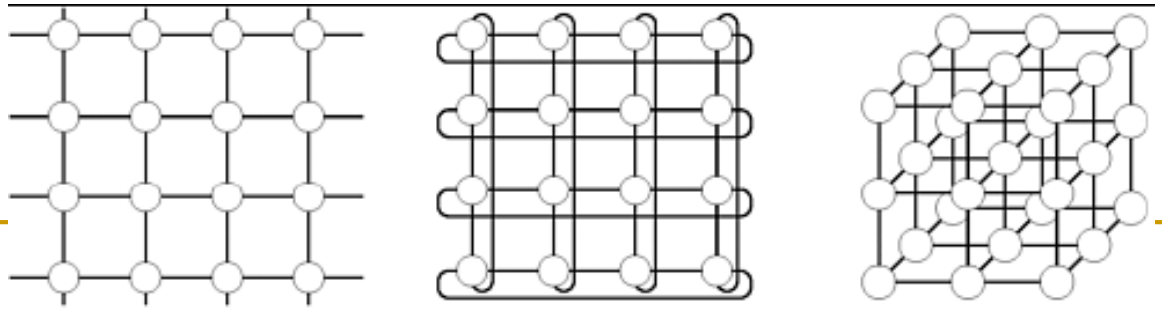
- Datorită numărului mare de legături în rețele complet conectate, rețelele mai reduse sunt utilizate de obicei pentru a construi computere paralele.
- Un tablou liniar este o rețea statică în care fiecare nod (cu excepția celor două noduri la capete) are doi vecini, unul la stânga și la dreapta sa.
- O simplă extensie a tabloului liniar este inelul sau un tor 1- D:
  - Inelul are o conexiune înfășurată între extremitățile tabloului liniar.
  - În acest caz, fiecare nod are doi vecini.



Grile 2D, tor 2D si grila 3D

- Grila bi-dimensionala:
  - este o extensie a tabloului liniar la două dimensiuni.
  - fiecare dimensiune are noduri sqrt (*p*) cu un nod identificat printr-un perechea (*i, j*).
  - fiecare nod (excepție periferia) conectat la 4 noduri (indicii diferă în orice dimensiune cu 1)
  - Poate fi amplasat în spațiul 2-D, ceea ce îl face atractiv din punct de vedere al cablurilor.
  - O varietate de calcule structurate în mod regulat se mapează foarte natural la o grila 2-D.
  - ⇒ Grilele 2-D meshes au fost folosite adesea ca interconectări în mașini paralele.
- Grilele 2D Poate fi mărită cu link-uri pentru a forma tor 2D (b).
- Grila 3D:
  - Este o generalizare a grilei 2-D la trei dimensiuni, așa cum este ilustrat în (c).
  - Fiecare element nod dintr-un cub 3-D (excepție periferia), este conectat la alte șase noduri, două de-a lungul fiecăreia dintre cele trei dimensiuni.
  - De exemplu in Cray T3E
  - O varietate de simulări fizice (de exemplu, modelarea vremii în 3 D, modelarea structurală etc.) pot fi mapate în mod natural la topologii de rețea 3-D.

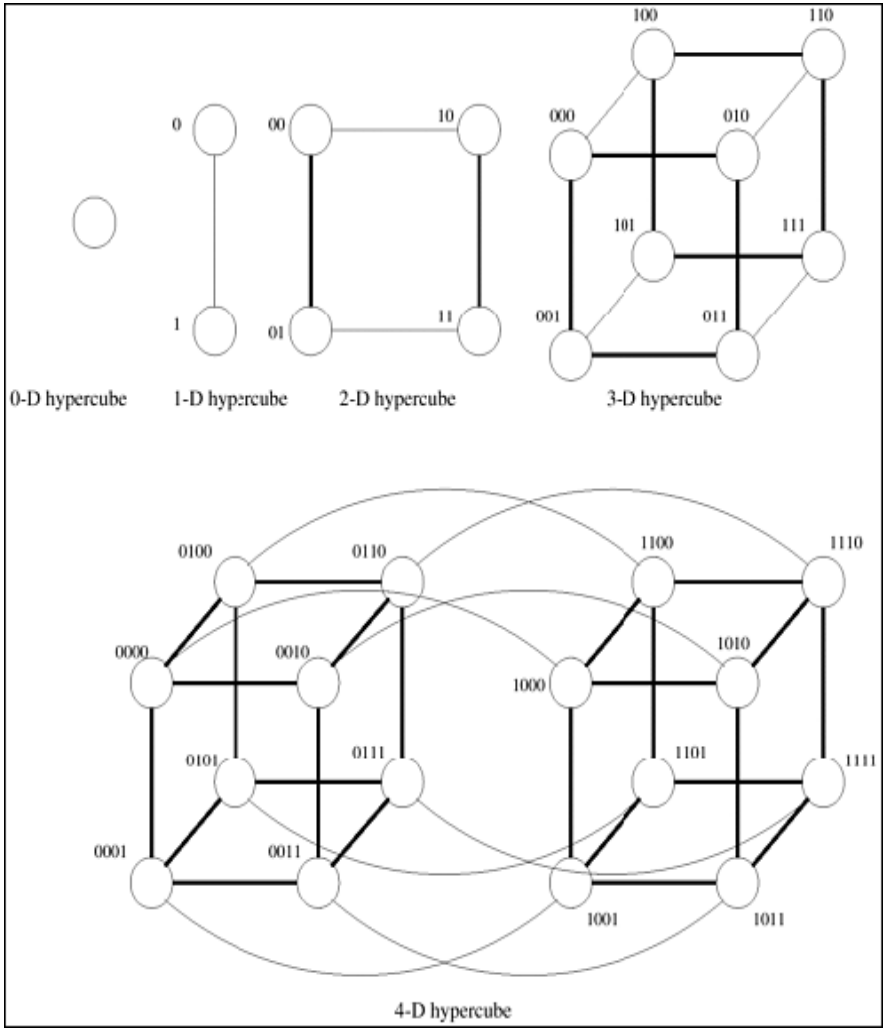
  => Grila 3-D sunt utilizate frecvent în rețelele de interconectare pentru computere paralele



Grile *k-d* mesh si hipercub

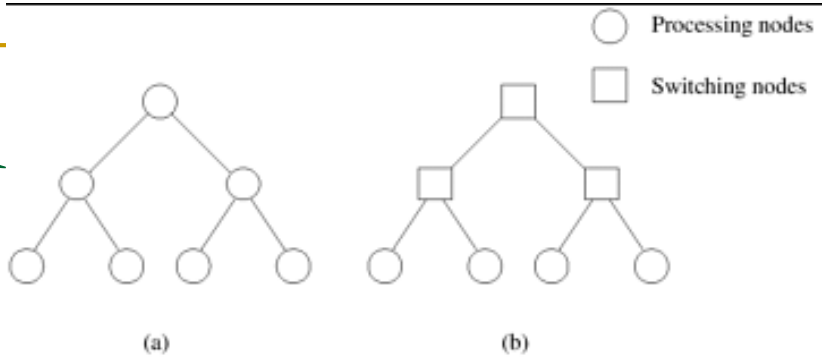
- Grila *k-d* : topologie constând din *d* dimensiuni cu noduri *k* de-a lungul fiecărei dimensiuni.
  - *d*=1: cazul liniar
  - *d*=2: grila
  - *d*=3: cubul 3D
  - *k*=2: hipercub
- Topologia hipercub: două noduri de-a lungul fiecărei dimensiuni și log *p* dimensiuni.
- Constructie:
  - Un hipercub de dimensiune zero este format din 2^0, adica un nod.
  - Un hipercub unidimensional este construit din două hipercuburi de dimensiune zero prin conectarea lor.
  - Un hipercub bidimensional format din patru noduri este construit din două hipercuburi unidimensionale prin conectarea nodurilor corespunzătoare.
  - În general, un hipercub *d* dimensional este construit prin conectarea nodurilor corespunzătoare a două hipercuburi dimensionale (*d* - 1).

Construcţia hipercubului şi schema de numerotare



- Schema de numerotare:
  - O numerotare a două subcuburi cu *p*/2 noduri => derivă o schemă de numerotare pentru cubul nodurilor p prin prefixarea etichetelor unuia dintre subcuburile cu "0" şi a etichetelor celuilalt cu un "1".
- Proprietate utilă că distanţa minimă între două noduri este dată de nr. de biţi care diferă în cele două etichete.
  - Exemplu: nodurile etichetate 0110 şi 0101 sunt la două legături între ele deoarece diferă la doua poziţii de biţi.
  - Proprietate utilă pentru derivarea unui număr de algoritmi paraleli pentru arhitectura hipercubului.

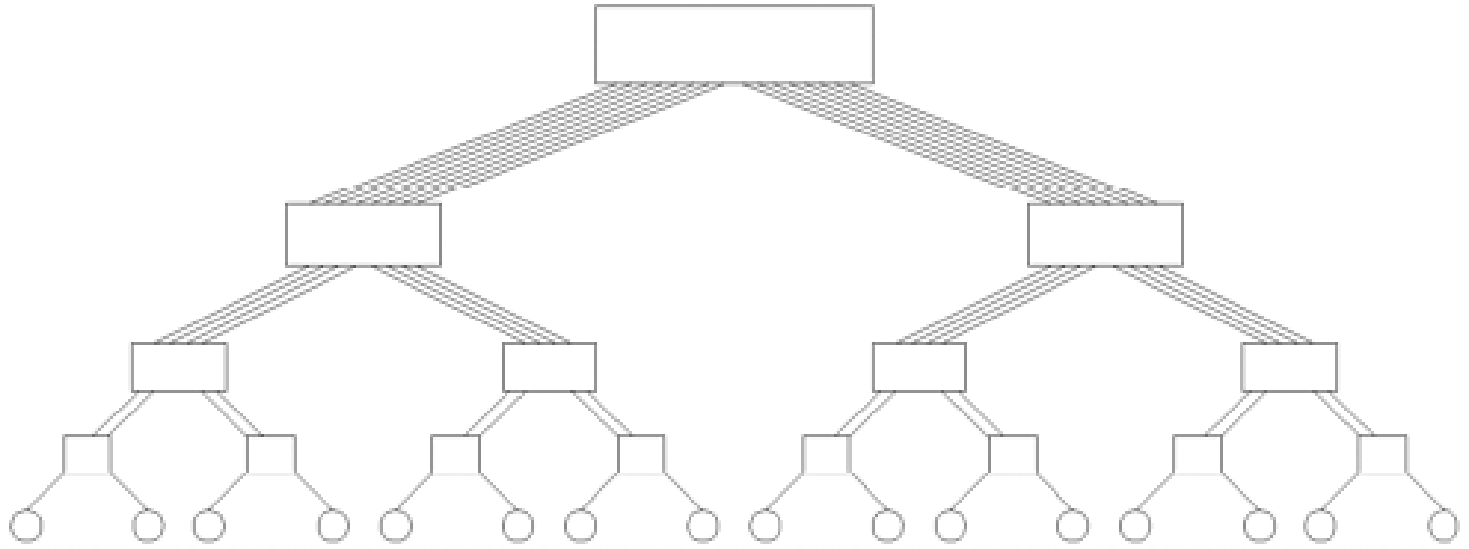
Retele bazate pe arbori



- În care există o singură cale între orice pereche de noduri.
- Tablourile liniare și rețelele in stea sunt cazuri speciale ale rețelelor de arbori.
- Figura: rețele bazate pe arbori binari complete.
- Rețelele statice bazate pe arbori au un element de procesare la fiecare nod al arborelui (in (a)).
- Într-o rețea dinamică bazata pe arbori, nodurile la niveluri intermediare sunt noduri de comutare, iar nodurile frunze sunt elemente de procesare (in (b)).
- Pentru a orienta un mesaj într-un arbore:
  - nodul sursă trimite mesajul în sus, până când ajunge la nodul de la rădăcina celui mai mic subarbore care conține atât nodurile sursă cât și cele de destinație.
  - Apoi, mesajul este dirijat în arbore in jos către nodul de destinație.

Arbore gras

- Rețelele bazate pe arbori suferă de un blocaj de comunicare la niveluri mai înalte ale arborelui.
  - De exemplu, atunci când multe noduri din subtree stânga a unui nod comunică cu nodurile din subtree dreapta, nodul rădăcină trebuie să gestioneze toate mesajele.
- Această problemă poate fi atenuată în rețelele dinamice de arbori prin creșterea numărului nr. legături de comunicare și noduri de comutare mai aproape de rădăcină (arbore gras)



Criterii pentru evaluarea rețelelor de interconectare statice

1. Diametru
2. Conectivitate
3. Lățimea bisecării
4. Capacitatea canalului
5. Lățimea de bandă a bisecării
6. Cost

1. Diametru

- Diametrul unei rețele este distanța maximă dintre oricare două noduri de procesare din rețea.
- Distanța dintre două noduri de procesare este definită ca cea mai scurtă cale (în termeni de număr de legături) între ele.
- Exemple:
  - 1 pentru reteaua complet conectata,
  - 2 pentru reteaua stea.
  - floor(p/2) pentru inel.
  - 2(sqrt(p)-1) pentru grila 2D
  - 2floor(sqrt(p)/2) pentru tor
  - log p pentru hipercub
  - 2 log((p + 1)/2) pentru arbore binar complet

2. Conectivitate

- Conectivitatea unei rețele este o măsură a multiplicității căilor dintre orice două noduri de procesare.
- O rețea cu conectivitate ridicată este de dorit, deoarece scade disputa pentru resursele de comunicare.
- Masura conectivitatii:
  - Numărul minim de arcuri care trebuie eliminate din rețea pentru a-l rupe în două rețele deconectate.
  - Aceasta se numește *conectivitatea arc* a rețelei.
  - Exemple:
    - 1 pentru tablou linear, arbore si stea.
    - 2 pentru inel si grila 2-D,
    - 4 pentru tor 2-D
    - *d* pentru hipercub *d*-dimensional.

3. Lățimea bisecării

- Definit ca numărul minim de legături de comunicare care trebuie eliminate pentru a partiționa rețeaua în două jumătăți egale.
- Exemple:
  - 2 pentru inel,
  - sqrt(p) pentru grid 2D cu p noduri
  - 2sqrt(p) pentru tor.
  - 1 pentru arbore si stea,
  - $p^2/4$ pentru graf complet
  - $p/2$ pentru hipercub (cf. constructiei)

4. Lățimea canalului, viteza și lățimea de bandă

- Lățimea canalului:
  - Definește numărul de biți care pot fi comunicați simultan printr-o legătură care conectează două noduri.
  - Egal cu numărul de fire fizice din fiecare legătură de comunicare.
- Viteza:
  - Viteza maximă la care un singur fir fizic poate livra biți.
- Lățimea de bandă:
  - Rata de vârf la care datele pot fi comunicate între capetele unei legături de comunicare.
  - Produsul ratei și lățimii canalului.

5. Lățimea de bandă a bisecării

- definit ca volumul minim de comunicare permis între cele două jumătăți ale rețelei.
- produsul lățimii bisecării și lățimea de bandă a canalului.
- uneori denumită lățime de bandă în secțiune transversală.
- poate fi utilizat ca măsură a costului:
  - asigură o delimitare inferioară a zonei dintr-un ambalaj 2D sau volumul într-un ambalaj 3D.
  - Dacă este *w*, limita inferioară pe zonă:
    - într-un ambalaj 2D este $O(w^2)$,
    - într-un ambalaj 3D este $O(w^{3/2})$.

6. Cost

- poate fi definit în termeni de număr de legături de comunicare sau număr de fire solicitate de rețea.
- Exemple:
  - Tablourile liniare și arborii folosesc doar legături p - 1 pentru conectarea nodurilor p.
  - O grila d-dimensionala are dp legături.
  - Un hipercun are (p log p)/2 legături.

Un rezumat al topologiilor statice de rețea

Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Completely-connected	1	$p^2/4$	$p - 1$	$p(p - 1)/2$
Star	2	1	1	$p - 1$
Complete binary tree	$2 \log((p + 1)/2)$	1	1	$p - 1$
Linear array	$p - 1$	1	1	$p - 1$
2-D mesh, no wraparound	$2(\text{sqrt}(p)-1)$	$\text{sqrt}(p)$	2	$2(p-\text{sqrt}(p))$
2-D wraparound mesh	$2\text{floor}(\text{sqrt}(p)/2)$	$2\text{sqrt}(p)$	4	$2p$
Hypercube	$\log p$	$p/2$	$\log p$	$(p \log p)/2$
Wraparound k-ary d-cube	d floor$(k/2)$	$2k^{d-1}$	$2d$	dp

Evaluarea rețelelor de interconectare dinamică

- Metricele de evaluare pentru rețelele dinamice urmează din valorile corespunzătoare pentru rețelele statice.
- Deoarece un mesaj care traversează un commutator aduce un surplus, este logic să gândim fiecare comutator ca un nod din rețea, pe lângă nodurile de procesare.

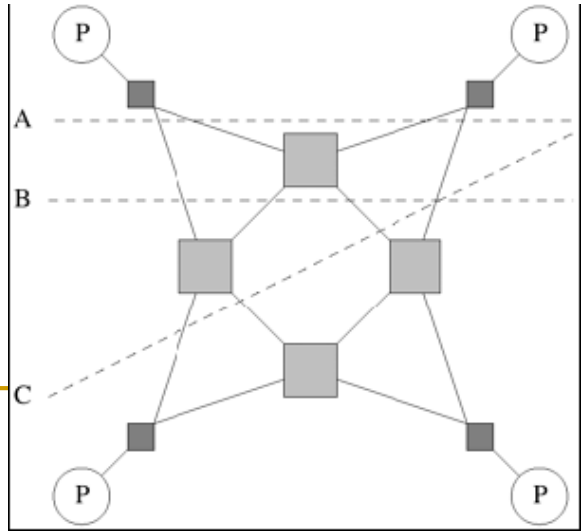
1. Diametrul retelei:
   - poate fi acum definită ca distanța maximă dintre oricare două noduri din rețea (procesare sau comutare).
2. Conectivitatea unei rețele dinamice poate fi definită în termeni de conectivitate cu noduri sau muchii.
   - Conectivitatea nodului este codul min.no.node care trebuie să eșueze (să fie eliminat din rețea) pentru a fragmenta rețeaua în două părți.
   - Ar trebui să luăm în considerare doar trecerea nodurilor (spre deosebire de toate nodurile).
   - Conectivitatea arc a rețelei poate fi definită în mod similar ca numărul minim de muchii care trebuie să eșueze (să fie eliminate din rețea) pentru a fragmenta rețeaua în două părți.

Evaluarea rețelelor de interconectare dinamică - lățimea bisecției

- Considerăm orice posibilă împărțire a nodurilor de procesare *p* în două părți egale.
  - Pentru fiecare astfel de partiție, selectăm o partiționare indusă a nodurilor de comutare astfel încât numărul de muchii care traversează această partiție să fie minimizat.
- Numărul minim de margini pentru o astfel de partiție este lățimea bisecției din rețeaua dinamică.
- Un alt mod intuitiv de a gândi lățimea bisecției:
  - în ceea ce privește numărul minim de muchii care trebuie eliminate din rețea, astfel încât să se imparta rețeaua în două jumătăți cu numar identic de noduri de procesare.

- Exemplu:
  - trei bisecții, A, B și C, fiecare partiționând rețeaua în 2 grupuri de 2 noduri de procesare fiecare.
  - Fiecare partiție are ca rezultat o tăiere de patru muchii.

=> Se conclude că lățimea bisecției acestui grafic este de patru.



Costul topologiilor dinamice de rețea

- Determinat de costul legăturii și costul comutatorului.
- Rețele dinamice tipice: gradul de comutare este constant.
- Numărul de legături și comutatoare este identic asimptotic.
- Costul de comutare depășește costul legăturii => costul rețelelor dinamice este deseori determinat de numărul de noduri de comutare din rețea.

Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Crossbar	1	p	1	p^2
Omega Network	log p	p/2	2	p/2
Dynamic Tree	2 log p	1	2	p - 1