Réseaux Datacenter et HPC

Routage

Elodie Ardoin, Stéphane Mathieu, Damien Gross

11 avril 2018

Bibliographie

Plan du cours

*Rappel :*Les problématiques réseaux liès au calcul parallèle

- L'augmentation de la taille des calculs induit une forte augmentation des phases d'échanges de données
- Pour garantir l'efficacité de la parallélisation, il y est nécessaire de minimiser ces temps de synchronisation :
  - Garantir la performance du réseau (latence minimale et stable, débit garanti)
  - Recouvrir des phases d'échange de données par du calcul (asynchronisme des communications)
  - Optimiser les communications (RDMA, algorithmique des communications collectives, etc)
- Pour garantir l'efficience du réseau, il faut aussi minimiser les coûts :
  - Coût d'aquisition (matériel)
  - Coût d'opération (résilience, consommation électrique)

Rappel : Les choix de design au niveau du réseau d'interconnexion

- Les réseaux massifs connectent tous les noeuds du calcul :
  - La topologie décrit la connexion physique
  - Le routage décrit la route à emprunter d'une source vers une destination.
  - Le mode de commutation décrit comment le message est envoyé sur la route.

Plan du cours

Plan du cours

Glossaire du routage

- **Distance**, longueur du chemin : nombre de *hops* maximum emprunté par la route
- **Latence** : temps de transport d'un paquet entre une source et une destination ; directement proportionnel à la distance
- Profil de trafic ou **traffic pattern** : type de trafic porté par le réseau (point à point, permutation, non-uniforme)
- **Facteur de charge** : nombre de routes porté par un lien
- **Facteur de contention** : nombre de flux distincts {source-destination} portés par un lien
- Boucle de routage ou **livelock** : information de routage qui fait toujours passer le même chemin
- Cul-de-sac ou **deadlock** : information de routage qui conduit à un puits
- **Résilience** du routage : capacité à recalculer une route en cas de défaut d'un lien, d'un switch
- Partage de charge ou **load-balancing** : capacité du routage à distribuer la charge (traffic pattern) sur le réseau physique

Qu'est-ce que le routage ?

- Le routage est le choix d'un chemin entre une source et une destination, parmi plusieurs.
- Si la topologie détermine la performance idéale du réseau, le routage est l'un des deux facteurs clés d'atteinte de cette performance.
- Le routage distribue la charge au sein du réseau.
- Dans les réseaux IP, on distingue le routage **statique** du routage **dynamique**
- Dans les réseaux IP, le routage est **distribué**. Chaque routeur est configuré localement et calcule sa table de forwarding.

Le routage sur les réseaux d'interconnexion de cluster

- **Garantit la connectivité**, donc ne doit générer ni deadlock ni de livelock
- **Minimise la latence**, donc la distance pour l'ensemble des routes
- **Maximise la bande passante** et garantit une bonne répartition de la charge
- **Est résilient**, donc garantit la connectivité même en cas de défaillance (lien, routeur)
- Est calculé de manière globale au réseau par un opérateur centralisé : le **subnet manager**
- Les différentes formes de routage sur les réseaux d'interconnexion sont :
  - Routage déterministe ou adaptatif ou réparti(*oblivious*)
  - Routage agnostique ou spécifique à la topologie
  - Routage minimal ou non-minimal

Routage **déterministe**

- Aussi appelé routage **statique**
- Les routes sont pré-calculées et ne sont modifiées qu'en cas de modification de la topologie (défaillance du réseau)
- Deux messages émis par la même source vers la même destination empruntera exactement le même chemin au sein du réseau au cours du temps
- Inconvénients :
  - Ces algorithmes ne prennent pas en compte l'état courant du réseau (surcharge, dysfonctionnement, etc.)
  - La congestion (qui dépend du profil de charge) n'est pas gérée par le routage
- Avantages
  - Le calcul du routage est fait a priori. Il n'y a pas de latence induite à chaque saut par une décision de routage.
  - On a une connaissance précise et constante du routage.

Routage *adaptatif*

- Aussi appelé routage **dynamique** dans les réseaux IP.
- Les routes principales et optionnelles sont calculées de manière détermiste.
- Le choix du "next-hop" est fait en fonction de l'état du réseau
- Les facteurs de considération sont : **disponibilité** et **charge des liens**
- Le chemin entre une même source et une même destination peut donc changer au cours du temps
- Inconvénients :
  - Perte de la connaissance précise des routes (à tout instant)
  - Difficulté d'estimer la charge globale de la fabrique. La congestion peut être un état transitoire. Remonter l'information de congestion sur toute la fabrique est couteux en temps.
- Avantages
  - Meilleure répartition du trafic en cas de charge uniforme
  - Moins de risque de congestion

Routage *oblivious*, réparti

- Les routes principales et les routes optionnelles sont calculées de manière détermiste.
- Le choix du "next-hop" est choisi de manière**aléatoire** parmi plusieurs routes optionnelles.
- Ce type de routage s'applique à certains types de messages, insensibles à l'ordre des paquets.
- Avantages
  - Meilleure utilisation de la diversité des chemins offerte par la topologie
  - Meilleure répartition de la charge que le routage déterministe
- Inconvénients
  - Désordonne les messages

Routage en fonction de la destination

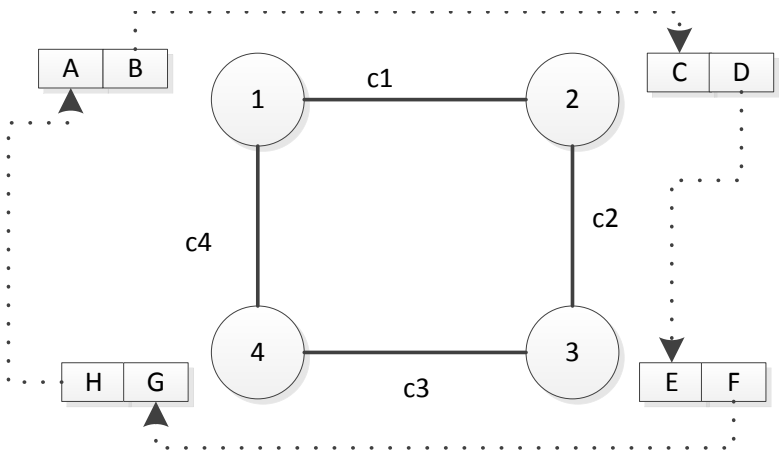
- Chaque paquet possède l'adresse de destination
- Des coordonnées pour des topologies de type Tor, un identifiant pour les nœuds
- Le choix de la route dans la table est fonction de l'adresse de destination

Routage en fonction de la source et de la destination

- Les informations sont stockées dans l'entête du paquet (ce qui augmente naturellement sa taille)
- Ces tables sont pré-calculés par le nœud
- Chaque routeur traversé récupère son information de routage dans le but de l'exploiter

Empêcher les deadlocks

- Qu'est-ce qu'un **deadlock** ?
- Un deadlock est un blocage au sein du réseau parce qu'un message attend la libération d'une ressource (utilisée donc par un autre message) : généralement un buffer ou un virtual channel
- Exemple simple : une dépendance cyclique



Empêcher les livelocks

- Qu'est-ce qu'une boucle de routage ou **livelock** ?
- Un livelock est un problème similaire aux **deadlocks**, c'est-à-dire qu'un paquet ne peut pas être acheminé correctement jusqu'à sa destination finale. Il continue "d'avancer" dans le réseau sans pour autant atteindre sa destination finale
- Se produit généralement avec un routage adaptatif
- Se résout avec un mécanisme de priorité sur les liens

Routage agnostique et topologique

- Quels sont les avantages/inconvénients d'un routage agnostique (c'est-à-dire sans prise en compte des propriétés de la topologie) ?
- Quels sont les avantages/inconvénients d'un routage topologique (c'est-à-dire tirant partie des propriétés de la topologie) ?

Combinaisons topologie/routage

TABLE II. USABILITY OF TOPOLOGY/ROUTING COMBINATIONS; : DEADLOCK-FREE; : ROUTING FAILED; : DEADLOCK DETECTED

	Fat-tree	Up*/Down*	DOR	Torus-2QoS	MinHop	SSSP	DFSSSP	LASH
	artificial topologies							
2D mesh	r	r	o	o	d	d	o	o
3D mesh	r	r	o	o	d	d	o	o
2D torus	r	r	d	o	d	d	o	o
3D torus	r	r	o	o	d	d	o	o
Kautz	r	r	d	r	d	d	o	o
k-ary n-tree	o	o	o	r	o	o	o	o
XGFT	o	o	o	r	o	o	o	o
Dragonfly	r	r	d	r	d	d	o	o
Random	r	r	o	r	d	d	o	o
	real-world HPC systems							
Deimos	r	o	o	r	o	o	o	o
TSUBAME2.0	o	o	o	r	o	o	o	o
	topology-aware				topology-agnostic			

FIG.: Source : J. Domke , T. Hoefler, and S. Matsuoka, "Fail-in-place Network Design : Interaction Between Topology, Routing Algorithm and Failures," in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '14, (Piscataway, NJ, USA), pp. 597-608, IEEE Press, 2014.

Quelle résilience ?

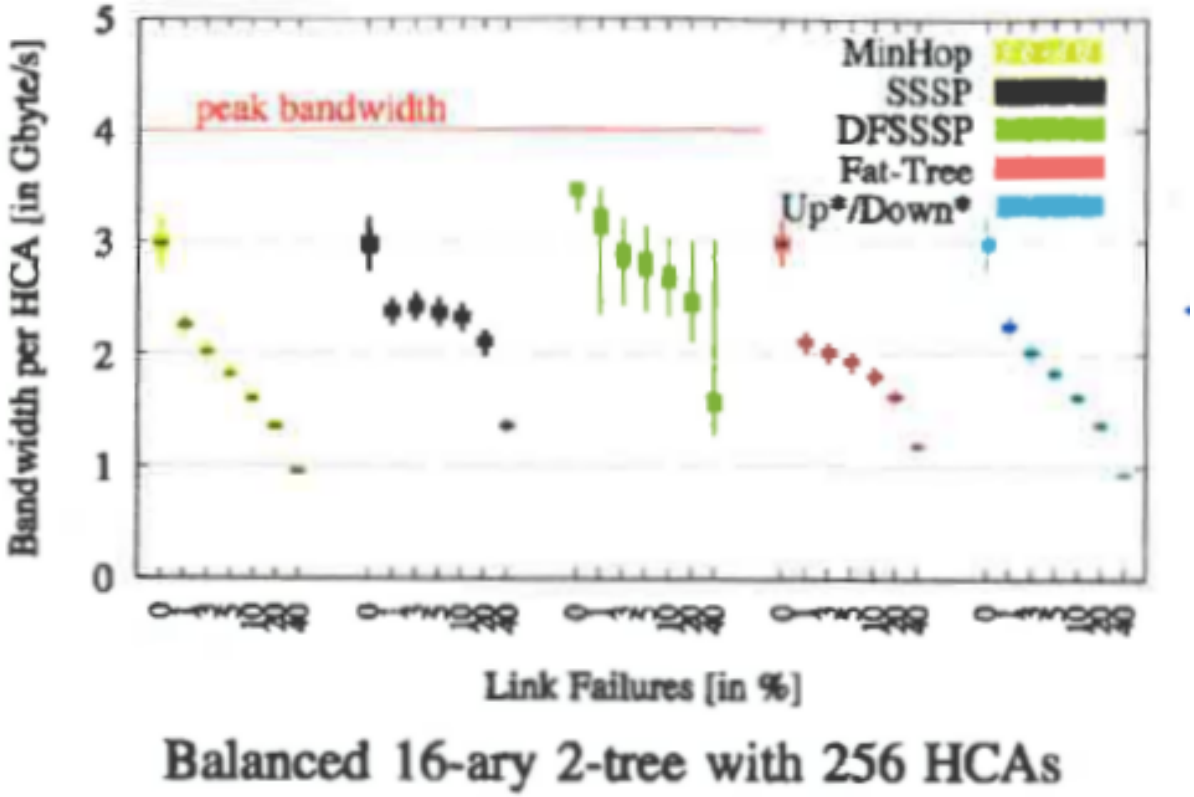
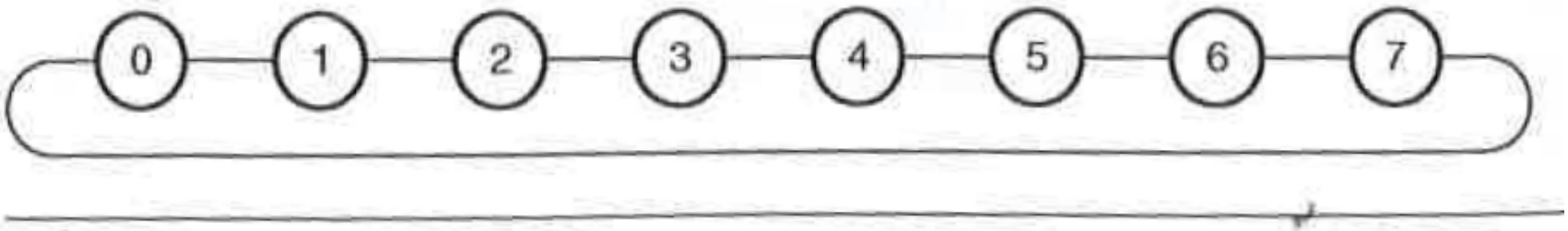


FIG.: Source : J. Domke, refer to previous slide

Un peu de réflexion

- Source :*Principles and practices of Interconnection Networks, Willian Daly and Brian Towles* chez *Elsevier*
- Exercice 8.1 p.171

Soit une topologie simple :



An 8-node ring network.

Soient des algorithmes de routage tel que :

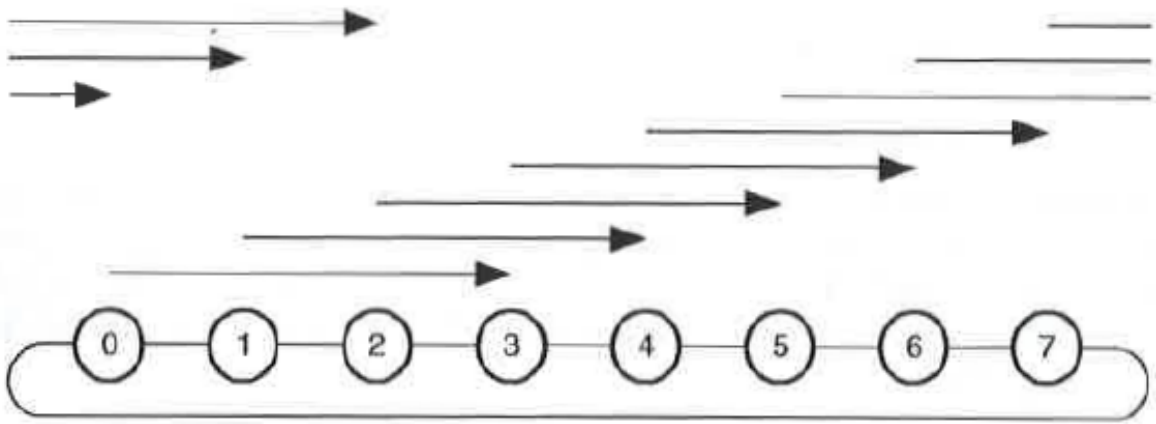
Greedy: Always send the packet in the shortest direction around the ring. For example, always route from 0 to 3 in the clockwise direction and from 0 to 5 in the counterclockwise direction. If the distance is the same in both directions, pick a direction randomly.

Uniform random: Randomly pick a direction for each packet, with equal probability of picking either direction.

Weighted random: Randomly pick a direction for each packet, but weight the short direction with probability $1 - \Delta/8$ and the long direction with $\Delta/8$, where Δ is the (minimum) distance between the source and destination.

Adaptive: Send the packet in the direction for which the local channel has the lowest *load*. We may approximate load by either measuring the length of the queue serving this channel or recording how many packets it has transmitted over the last T slots. Note that this decision is applied once at the source because we have disallowed backtracking.

Soit un motif de communication :



Tornado traffic on an 8-node ring network. With greedy routing, all traffic moves in a clockwise direction around the ring, leaving the counterclockwise channels idle.

Questions :

Quel algorithme choisiriez-vous pour :

- minimiser la latence d'un trafic de type Tornado ?
- maximiser la bande passante d'un trafic de type Tornado ?

Se limiter au choix d'un seul algorithme pour chaque critère et justifier le choix.

Plus de questions

En reprenant l'exercice du *Daly* :

- quel algorithme minimise la latence d'une communication multiple 1 to 1 ?
- quel algorithme minimise la latence d'une communication All-to-1 ?

Analyse

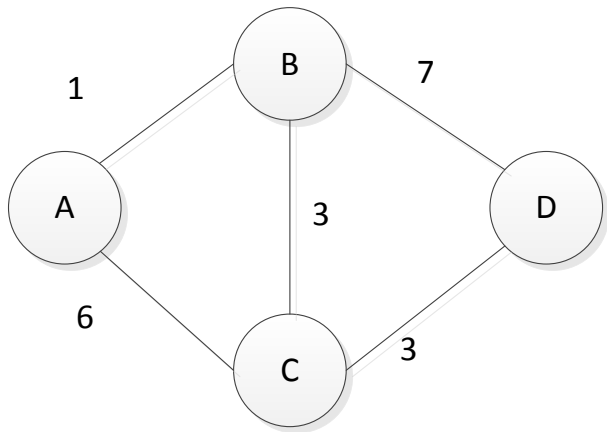
Déductions ?

Algorithme MinHop

- Repose sur l'algorithme de Dijkstra qui résout le problème du plus court chemin
- Chaque arête possède un **poids**
- On pré-calcule l'ensemble des routes avec pour objectifs de réduire au maximum le poids total de la route
- Si il existe deux chemins possibles (donc avec le même nombre de sauts), le port le moins assigné est utilisé
- Routage agnostique, par défaut dans certaines topologies et par certains *Subnet Manager*

Single-Source Shortest Path

- Min-hop calculé pour chaque source vers l'ensemble des destinations.
- Les routes sont distribuées sur les liens par source. Permet une autre distribution des routes



Layered Shortest Path

- Idée : virtualiser un lien dans le but d'exploiter plusieurs niveaux
- Utilisation des *virtual channels* pour créer des plans de routage virtuels
- Distribution des routes sur ces plans de routage

Dimension Order

- Utilisé pour les topologies de type hypercube ou tor
- Utilisation des coordonnées de la destination pour choisir un chemin
- Plusieurs chemins possibles suivant le nombre de dimension de la topologie

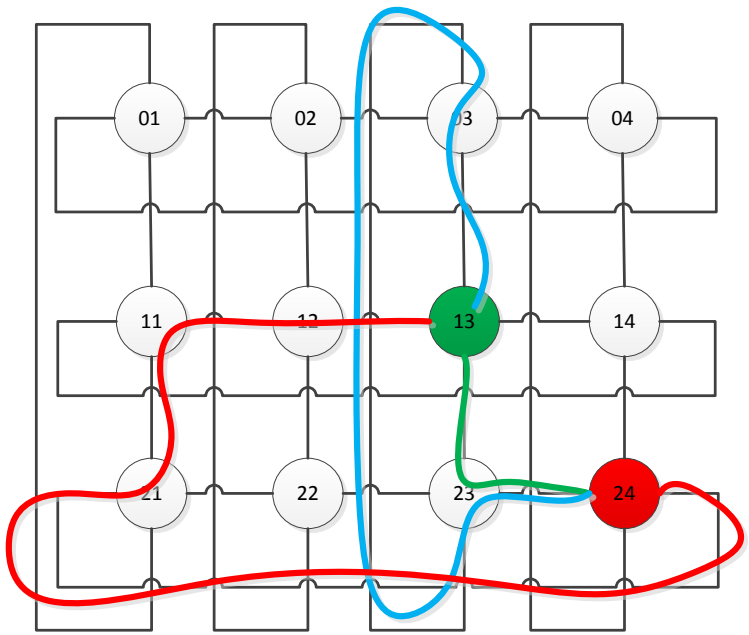


FIG.: Toplogie Tor : 4-ary 2-cube

Up/Down

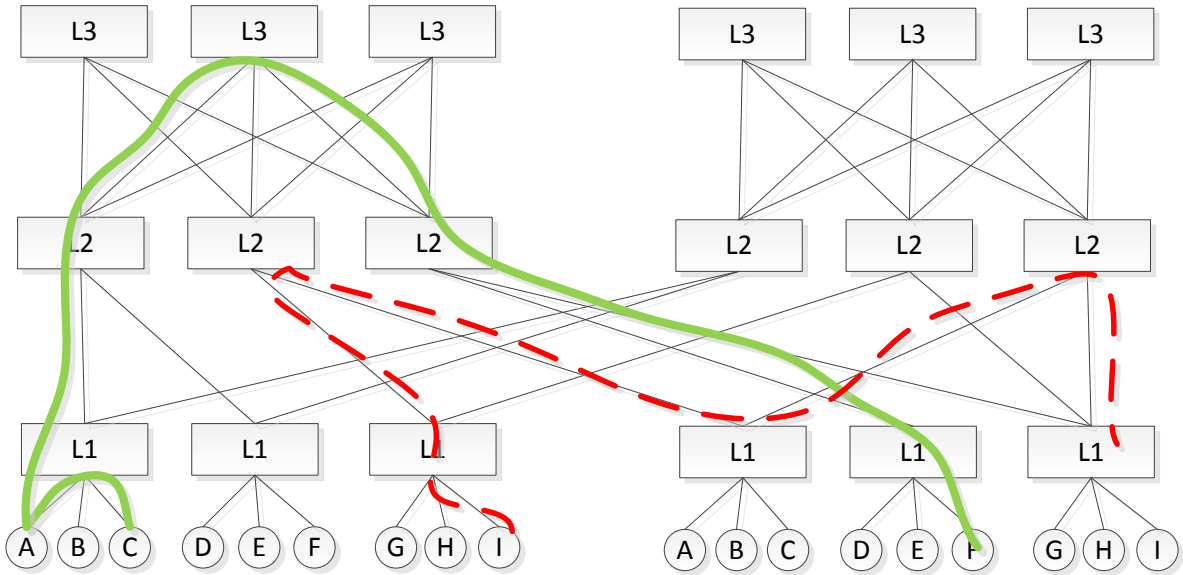


FIG.: Up/Down

- Ne fonctionne que pour des topologies indirectes et hiérarchiques, de type Fat-tree
- Le choix du next-hop se base sur la charge des liens montants jusqu'au *root switches* commun entre la source et la destination. Puis sur la charge des liens descendants jusqu'aux *leaf switches*.
- L'intérêt par rapport au Min-hop réside dans la simplicité, et donc à un temps de calcul plus court sur des grandes topologies

Ftree ou d-mod-k

- Ne fonctionne que pour des topologies hiérarchiques, de type Fat-tree (PGFT)
- Rappel sur les topologies PGFT : la bande passante de bissection vaut 1 donc le Fat-tree est *non- bloquant* ; C'est à dire que pour N/2 paires distinctes de noeuds (où N est le nombre de noeuds), *il existe* un routage non-bloquant.
- **il existe** signifie qu'il est possible de distribuer les routes sur les liens pour n'avoir qu'une route par lien.
- *Up/down* distribue la charge à chaque niveau sur les liens de manière aléatoire.
- Mais quelle distribution choisir ? Quelles sont les communications pour lesquelles il est intéressant d'être non-bloquant ?

Les motifs de communications

Programmer sur des processeurs à mémoire ditribuée nécessite une librairie d'échange de messages = MPI ! Il existe une grande variété de mouvements de données et d'opérations de contrôle des processus, comme :

- Echange de données **point-à-point**
- Transposition de matrice = permutation
- Distribution de l'élément pivot d'une matrice = réplication d'une même donnée entre plusieurs processus
- Somme de matrice, calcul du min/max d'un vecteur = réduction, contraire de la réplication

Aparté sur les opérations collectives 1/4

- **Multiple 1-to-1**
  - Permutation circulaire : chaque processus envoie un message au processus Pi+1.

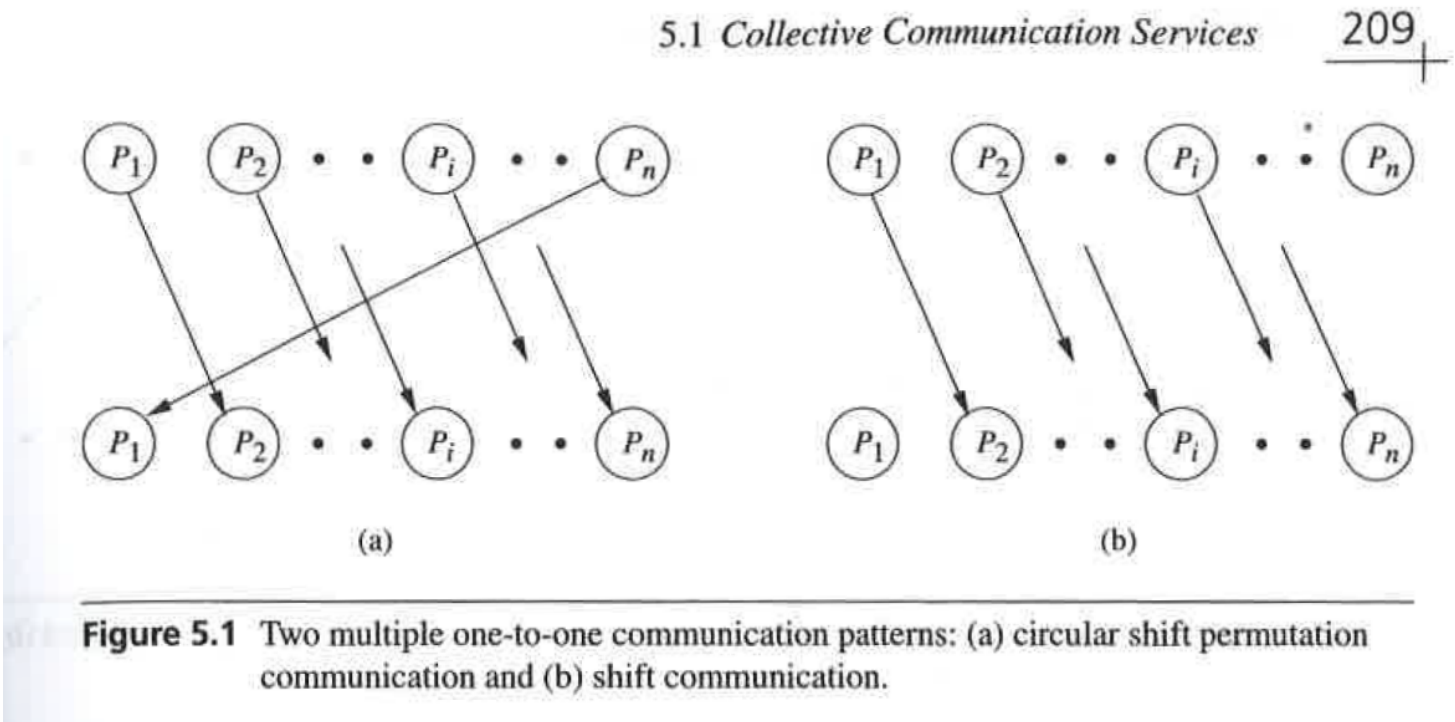


FIG.: Source : Interconnection networks, José Duato, Sudakhar Yalamanchili, Lionel Li

Aparté sur les opérations collectives 2/4

- **1 -to- all**
  - **Broadcast** : un émetteur envoie un même message à plusieurs récepteurs
  - **Scatter** : un émetteur envoie des messages différents à plusieurs récepteurs

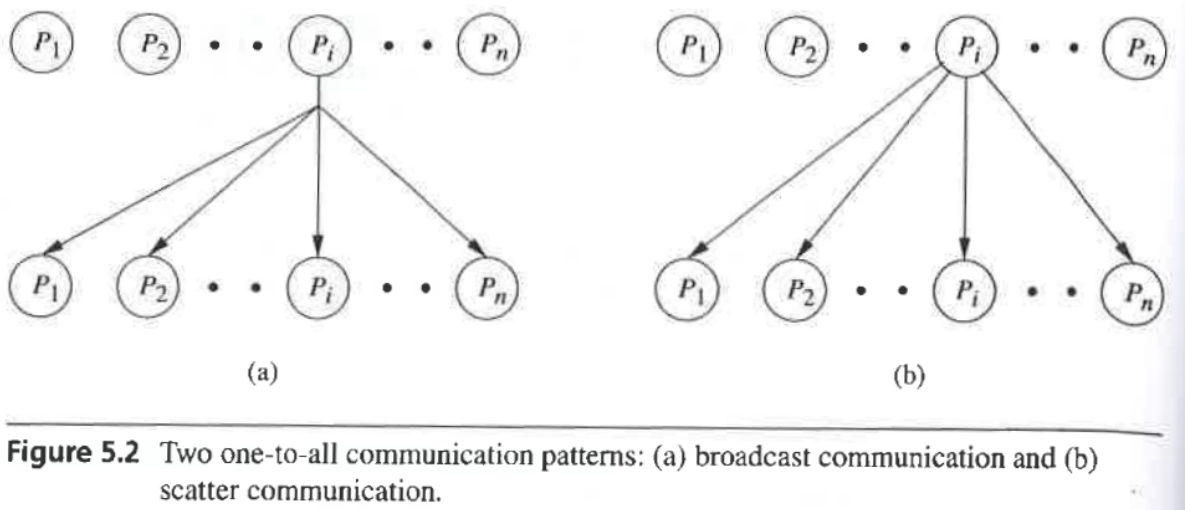


Figure 5.2 Two one-to-all communication patterns: (a) broadcast communication and (b) scatter communication.

FIG.: Source : Interconnection networks, José Duato, Sudakhar Yalamanchili, Lionel Li

Aparté sur les opérations collectives 3/4

- **all -to- 1**
  - **Reduce** : différents messages de différents émétteurs sont combinés via une opération atomique (OR, AND, XOR) pour un récepteur
  - **Gather** : différents messages de différents émetteurs sont concaténés pour un récepteur ; l'ordre de concaténation dépend de l'ID de l'émtteur

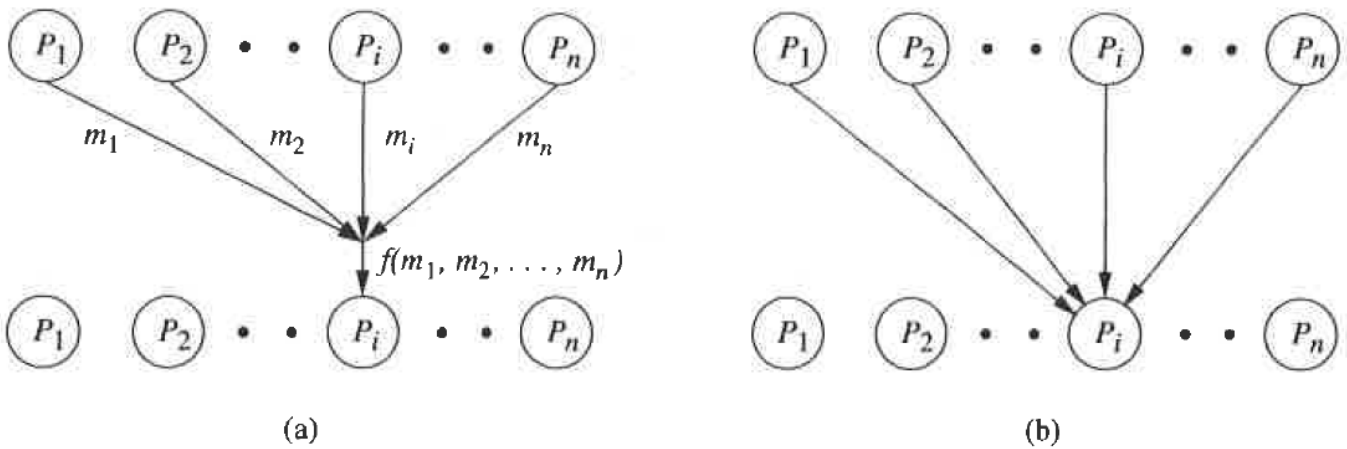


Figure 5.3 Two all-to-one communication patterns: (a) reduce communication and (b) gather communication.

FIG.: Source : Interconnection networks, José Duato, Sudakhar Yalamanchili, Lionel Li

Aparté sur les opérations collectives 4/4

- **all -to- all**
  - **All-broadcast** : tous les processus réalisent leur propre broadcast
  - **All-scatter** : tous les processus réalisent leur propre scatter

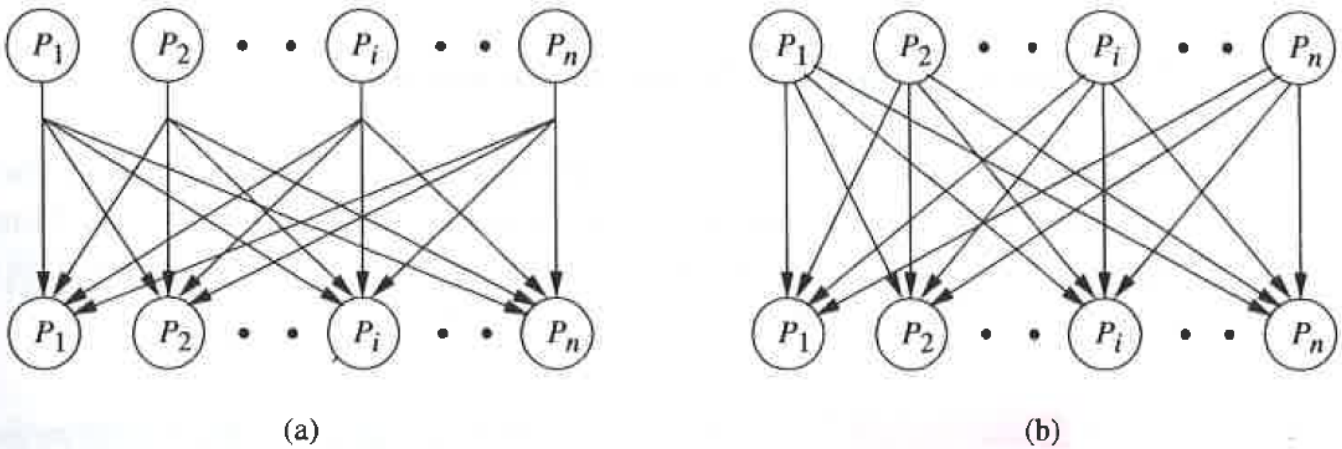


Figure 5.4 Two all-to-all communication patterns: (a) all-broadcast communication and (b) all-scatter communication.

FIG.: Source : Interconnection networks, José Duato, Sudakhar Yalamanchili, Lionel Li

Ftree ou d-mod-k

L'algorithme Fat-Tree ou D-mod-k distribue les routes de manière à maximiser la bande passante des *permutations circulaires*. C'est à dire les communications entre chaque noeud N et son Xième voisin.
Chaque lien porte les routes vers les destinations d modulo k, où k est le nombre de noeuds connectés sur les switches *leaf*.

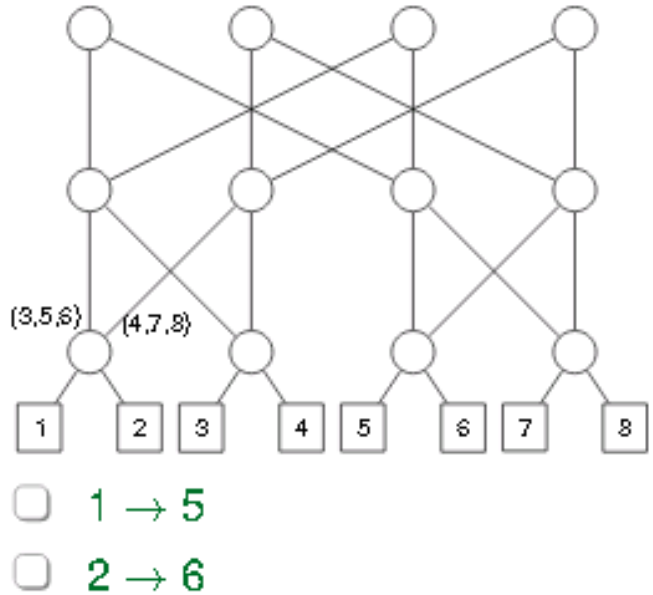


FIG.: Source : Evaluation pour les routages haute-performance, Matthieu Perrotin, BULL, 2013

Routage bloquant

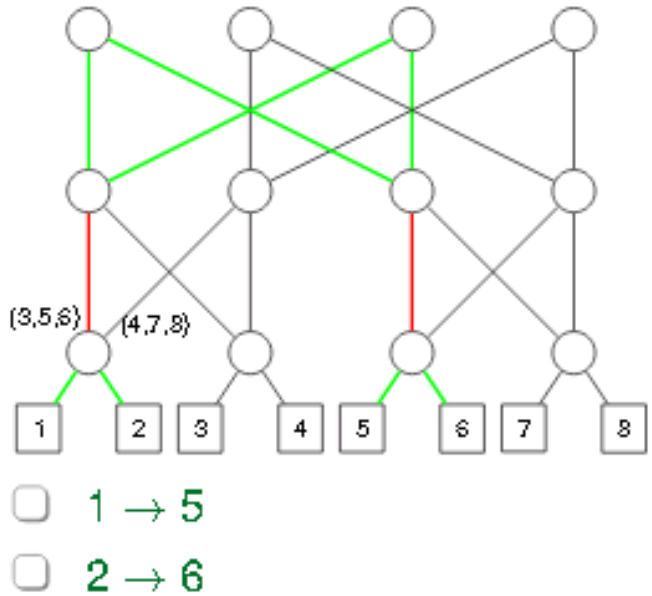


FIG.: Source : Evaluation pour les routages haute-performance, Matthieu Perrotin, BULL, 2013

Routage non bloquant

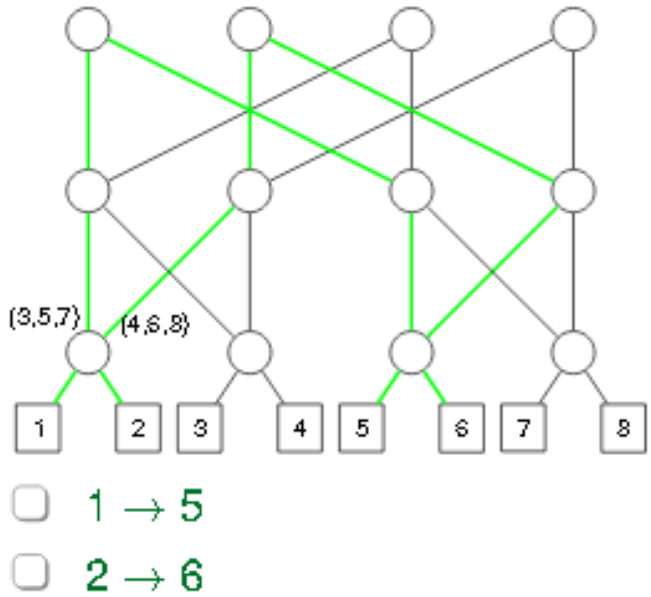


FIG.: Source : Evaluation pour les routages haute-performance, Matthieu Perrotin, BULL, 2013

Pour continuer...

Pour cette topologie PGFT (8 noeuds, 4 leafs, 4 spines) :

- Terminer le routage au sein de cette fabrique.
- Que peut-on en déduire quant à la congestion au sein de ce réseau :
  - Quand chaque noeud communique avec le noeud n+3(modulo 8) ?
  - Quand chaque noeud communique avec le noeud n-2(modulo 8) ?
  - Quand les noeuds pairs communiquent avec le noeud n-2(modulo 8) et que les noeuds impairs communiquent avec le noeud n+3(modulo 8) ?

Analyse

Déductions ?

Merci de votre attention !

Questions ?