Réseaux Datacenter et HPC

Notions spécifiques aux réseaux datacenter/HPC

Elodie Ardoin, Damien Gros

4 mai 2018

Qui sommes-nous ?

- Laboratoire réseau du DSSI
- Réseau d'entreprise classique
- Réseau d'interconnect HPC
- Sécurité réseau
- etc.

Les stages au CEA

- La liste des stages au CEA `http://www.cea.fr/emploi/Pages/stages/offres-stage.aspx`
- Thématiques possibles au sein du laboratoire :
  - L'interconnect HPC
  - Software Defined Network
  - Sécurité des réseaux
  - Vos propres idées !

Les réseaux

- Qu'est-ce qu'un réseau ?
- Où trouve-t-on les réseaux ?
- Pourquoi les réseaux sont-ils importants ?

Les réseaux d'interconnexion

- Qu'est-ce qu'un réseau d'interconnexion ?
- Où trouve-t-on les réseaux d'interconnexion ?
- Pourquoi les réseaux d'interconnexion sont-ils importants ?

Qu'est-ce qu'un réseau d'interconnexion ?

- Système configurable qui transporte des données entre des terminaux :
  - Configurable car il établit différentes connexions entre différents points dans le temps
  - Systeme car composé de différents éléments : buffers mémoire, canaux, commutateur et plan de contrôle
- Dans les systèmes de calcul : connecte les processeurs à la mémoire, ou les systèmes d'I/O aux controlleurs I/O
- Dans les systèmes de commutation : connecte les ports d'entrée aux ports de sortie
- Les interconnexions haute-performance sont réalisées via des interconnexions point-à-point (no bus)
- Elément de limitation de la performance :
  - Bande passante et latence d'accès à la mémoire
  - Vitesse de commutation et nombre de ports du switch
- Centre de coût :
  - 50% de la consommation électrique
  - 33% des investissements matériel

Existe-t'il des technologies réseaux spécifiques ?

- Les réseaux Ethernet
- Les réseaux Infiniband
- Les réseaux avec des technologies spécifiques

Différence entre les réseaux HPC et Datacenter

- Qu'est-ce qu'un DataCenter ?
- Quelles sont les points communs entre ces deux types de réseau ?
- Quelles sont les différences avec le HPC ? En terme de machines ? En terme de locaux ?
- La notion de disponibilité ? Qu'est-ce que cela implique pour ces deux architectures ?
- Quelle notion de performance pour un datacenter ?
- Quels sont les objectifs pour un datacenter ?

Le contenu de l'UE

- Les réseaux d'interconnexion de cluster dédié au calcul massivement parallèle
- Focus sur la technologie Infiniband
- Les architectures réseaux Datacenter
- La configuration au travers d'interface virtuelle (= abstraction)
- L'automatisation et la provision à la demande

Le contenu du cours

- Beaucoup de vocabulaires

1. Les topologies spécifiques des réseaux d'interconnexion
2. Les différents types/algorithmes de routage
3. Le switching, la gestion de la congestion, etc.

Module 1

- Notions génériques et problématiques liées au HPC
- Intervenants : E.Ardoin, D.Gros
- 3(*3h30) cours, 3(*3h30) séances de TPs

Examen et notations

- Pas d'examen sur table
- 1 rapport à la fin du dernier TP : lundi 14 mai, les détails sont sur le sujet du TP
- Participation en cours : exercices à faire
- Supports disponibles à la fin du cours

Les objectifs de ce module

- Cours : Comprendre les problématiques des réseaux d'interconnexion d'un calculateur
  - Connaitre les grands principes d'architecture d'interconnexion d'un calculateur
  - Comprendre les problematiques : performance, efficience, résilience
- TP :
  - Manipuler les outils de simulation
  - Comprendre les principes de topologie, routage et impact du placement/routage
  - Les éléments qui permettent de comprendre des questions de congestion

Bibliographie

Besta, M. and Hoefler, T. (2014). Slim fly : A cost effective low-diameter network topology. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '14, pages 348–359, Piscataway, NJ, USA. IEEE Press.

Dally, W. and Towles, B. (2003). Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Duato, J., Yalamanchili, S., and Lionel, N. (2002). Interconnection Networks : An Engineering Approach. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Plan du cours

Plan du cours

Plan du cours

D'où vient la performance d'un calculateur ?

- Fréquence *brute* du processeur (*ie* : nombre d'opérations par seconde)
- Mémoire : taille de la mémoire, des caches ainsi que le temps d'accès ; les caches L1 et L2 sont d'accès plus rapides que la RAM
- Latence : temps d'acheminement d'un paquet au travers du réseau
- Réseau d'interconnexion :
  - Débit : volume de données acheminées aux terminaux du réseau
  - Topologie du réseau : le placement physique des éléments du réseau
  - Routage : les chemins empruntés (ou possibles) d'un paquet pour aller d'un nœud source au nœud destination
  - Placement des *jobs* : comment sont répartis les *jobs* au sein du cluster

Les problématiques réseaux liés 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 des données par du calcul (asynchronisme des communications)
  - Optimiser les communications (RDMA, algorithmique des communications collectives, etc)
  - Garantir la fiabilité du transfert des messages
  - Garantir l'ordre d'arrivé des messages (important pour MPI)
- Pour garantir l'efficience du réseau, il faut aussi minimiser les coûts :
  - Coût d'acquisition (matériel)
  - Coût d'opération (résilience, consommation électrique)
  - Coût de l'administration
  - Coût/délai de la maintenance

Les choix de design au niveau du réseau d'interconnexion

- Les réseaux massifs connectent tous les nœuds 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. Il décrit comment sera géré la congestion.
- Chaque élément influe de manière significative sur la performance mais aussi sur la résilience du réseau d'interonnexion.
- Il est donc nécessaire de prendre en compte chacun de ces éléments dans le design d'un calculateur.

Plan du cours

Configuration type d'un calculateur

- Système d'exploitation : Linux
- 40 000 cœurs de calculs et traitements Intel® Xeon® de la famille E5 Broadwell, cadencés à 2.4 Ghz
  - 28 cœurs/nœuds
  - 128 Go de mémoire/nœuds
- 4 nœuds grande mémoire à 3 To/nœud
- 18 nœuds hybrides (Intel Boroadwell- Nvidia GPU Pascal)
- La capacité de stockage local des données est de 2.5 Po.
- Le réseau d'interconnexion :
  - Est de technologie InfiniBand EDR.
  - La topologie d'interconnexion est un Fat-Tree à 3 niveaux, pruné à 1/6.
  - Le routage est spécifique à la topologie et non-bloquant pour les permutations.
  - Le routage distingue les nœuds I/O des nœuds de calcul.

Plan du cours

Qu'est-ce que la topologie d'un réseau ?

- C'est la façon d'organiser **physiquement** les différents éléments d'un réseau les uns par rapport aux autres
- Un réseau d'interconnexion est analogue à une carte routière. Les canaux de communication (routes) transportent les paquets (voitures) d'un routeur (intersection) à un autre.
- Quelques topologies connues (sur Ethernet) :
  - En étoile
  - En anneau
  - En bus
- On distingue les topologies :
  - directes : chaque nœud de la topologie est à la fois un host et un switch
  - indirectes : les nœuds de la topologie sont soit un host, soit un switch

Les enjeux de la topologie

- Les choix de design sont le résultat de compromis (trade-off). Les différents points d'attention sont :
  - La performance brute : minimiser les distances physiques et le nombre de sauts, garantir le débit entre chacun des nœuds
  - L'adéquation à la charge : offrir une connectivité suffisamment bonne pour tout type d'algorithmes ayant des patterns divers
  - La scalabilité et l'extensibilité : conserver les performances dans les phases d'expansion du nombre de nœuds
  - La facilité de déploiement et les contraintes physiques : densité de câblage, contrôle de la température, etc.
  - La résilience et la réparabilité : détection et correction automatique des défauts, accès physique aisé
  - Les contraintes de coût de déploiement et d'opération

Vocabulaire spécifique à la topologie

- commutateur/switch : équipement réalisant de la commutation
- routeur : équipement qui porte une fonction de routage en plus de la fonction de commutation.
- arité, radix : nombre de ports d'un switch/routeur.
- pruning : rapport entre le nombre de liens entrants et le nombre de liens sortants
- terminaux/hosts : nœuds de calcul, serveurs de base de données, serveurs webs, parefeux, etc.
- leaf/edge : switch d'accès interconnectant des terminaux/hosts.
- spin/core : switch de cœur de réseau (ou dorsale) interconnectant les switches d'accès (ou leafs)
- topologie : graphe non-orienté composé de sommets (commutateurs ou nœuds) et d'arêtes (liens, câbles, fibres optique).
- *diamètre* : distance max en nombre de sauts entre deux terminaux
- *bissection bandwidth* : bande passante au point milieu du système
- *hop count* ou distance entre les nœuds : nombre de commutateurs traversés
- *path diversity* : nombre de chemins distincts entre deux éléments

High radix router

- De quelle manière le radix du switch influence la topologie ?
- Les avantages d'un switch de haut-radix :
  - Réduit le diamètre du réseau et la latence
  - Réduit le nombre de câbles (qui peuvent être extrêmement chers)
- Les inconvénients :
  - Augmente la longueur des câbles (augmentation de la latence, perte de débit)
  - Augmente la complexité des routeurs/switchs (réalise plus de traitements)

Topologie indirecte : Niveau de switch

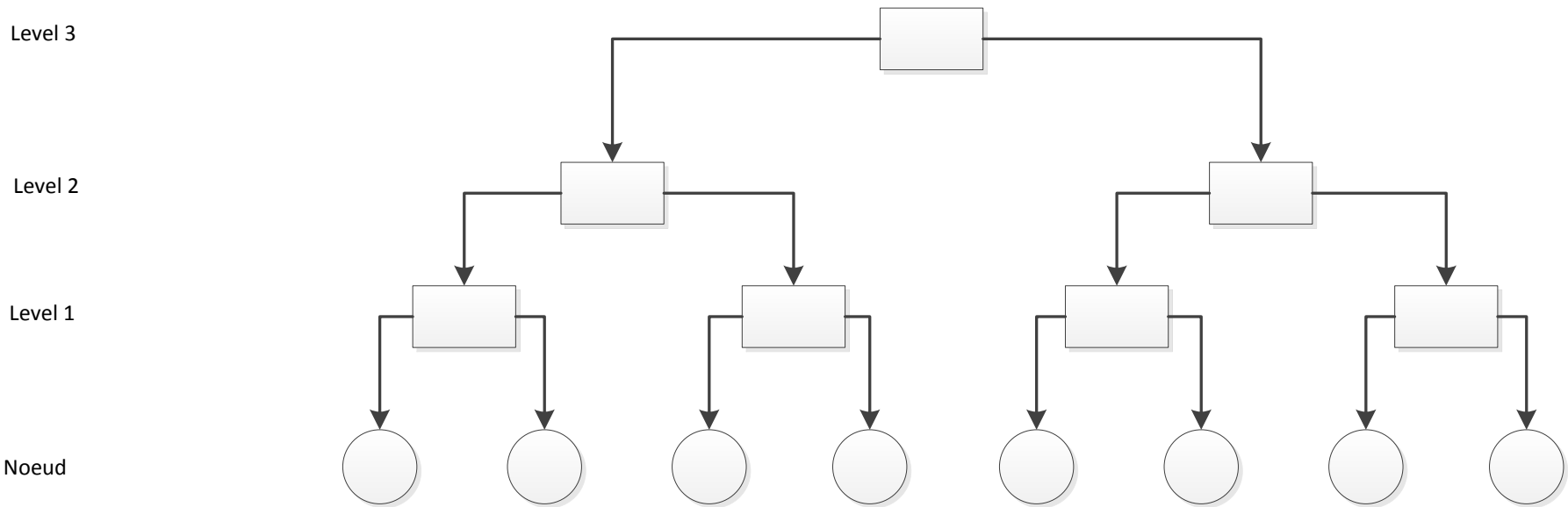
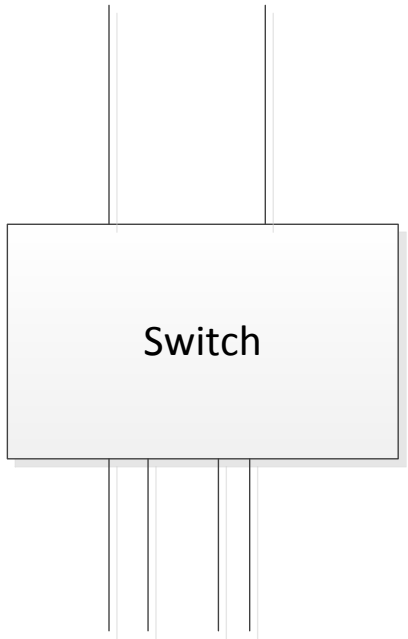


FIGURE – Niveau de switchs

Pruning simple



$$\text{Pruning} = 2/4 \rightarrow 1/2$$

Pruning

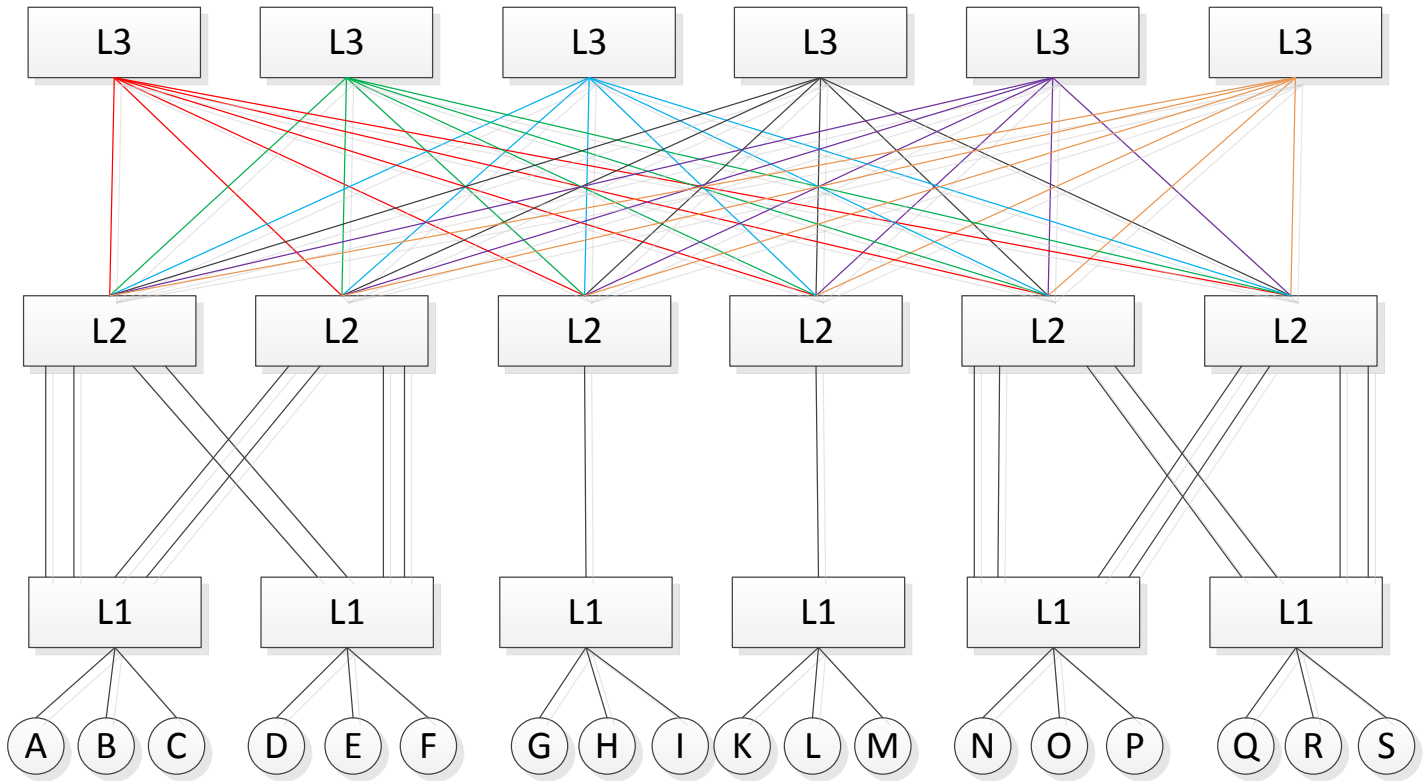


FIGURE – Pruning à l'échelle d'une fabrique

Bande passante de bissection

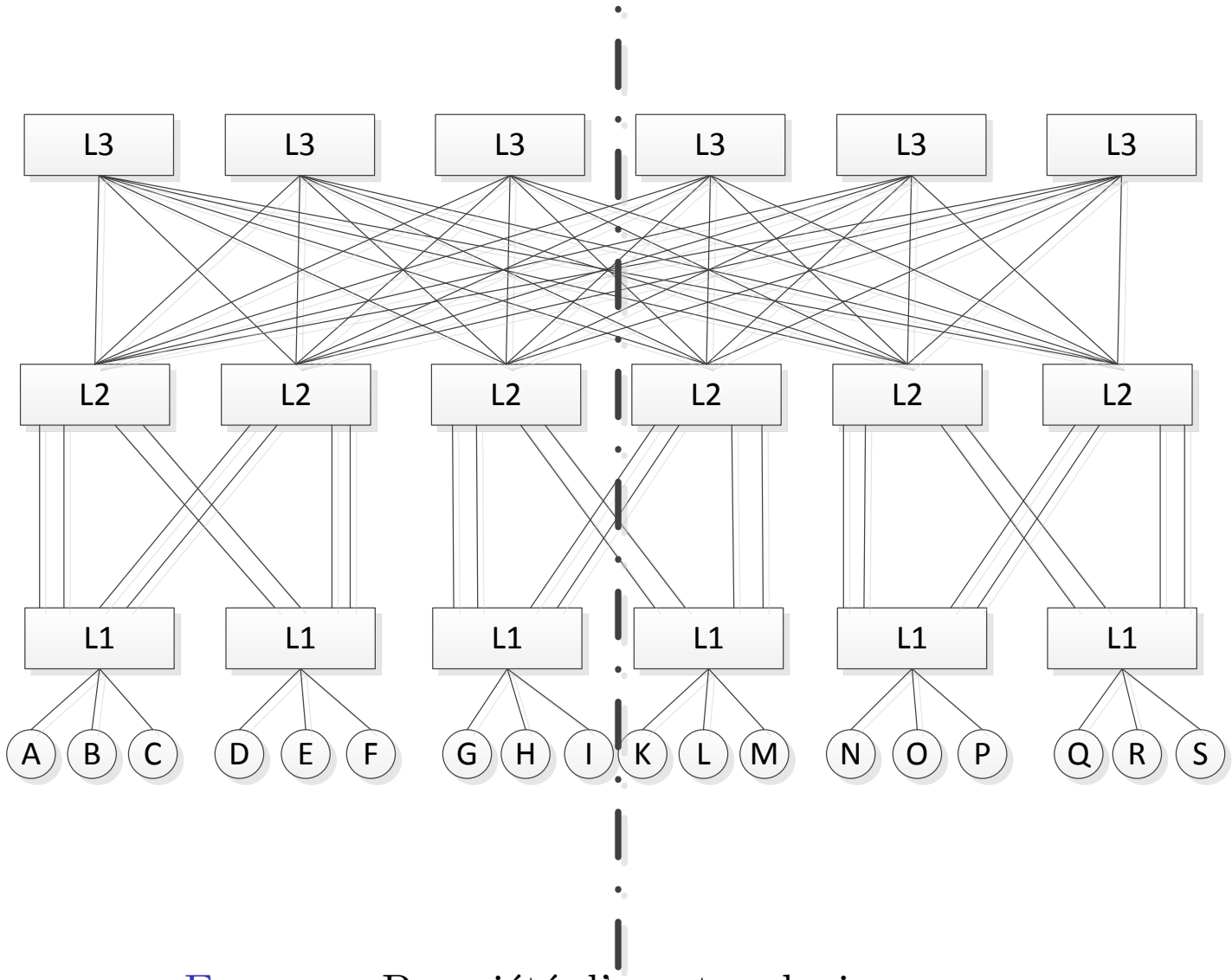


FIGURE – Propriété d'une topologie

Quelques définitions : bande passante de bissection

- directional capacity of a network between two equal-sized partitions of nodes. The cut across the network is taken at the narrowest point in each bisection of the network.
- la bande passante entre deux ensembles disjoints de nœuds connectés au commutateur. Puisque la multitude de combinaisons possibles ne peuvent toutes être vérifiées, on définit l'effective bisection bandwidth par la bande passante moyenne si on mesure la performance de la bissection aléatoire de la communication

Propriétés d'une topologie

- On peut définir une topologie réseau suivant plusieurs critères.
- Ces critères vont influencer les divers choix que l'on pourra faire par la suite (taille des câbles, regroupement en îlots, etc.)
- Régulière ou irrégulière
  - Régulière signifie que c'est un graphe régulier
- Distance moyenne
  - Moyenne du nombre de sauts empruntés entre deux nœuds
- Bloquante ou non-bloquante
  - Si on peut connecter l'ensemble des sources et des destinations, alors la topologie est non bloquante

Tableaux de comparaison des topologies

Topologie	T3D	T5D	HC	LH-HC	FT-3	DLN	FBF-3	DF
Enpoints(N)	10.648	10.648	8.192	8.192	19.876	40.200	20.736	58.806
Routers (N_r)	10.648	10.368	8.192	8.192	2.311	4.020	1.728	5.346
Radix(k)	7	11	14	19	43	43	43	43
Electric cables	31.900	50.688	32.768	53.248	19.414	32.488	9.504	56.133
Fiber cables	0	0	12.288	12.888	40.215	33.842	20.739	29.524
Cost per node ($)	1.682	3.176	4.631	6.481	2.346	1.743	1.570	1.438
Power per node (W)	19,6	30,8	39,2	53,2	14,0	12,04	10,8	10,9

Topologie	FT-3	DLN	FBF-3	DF	DF	SF
Enpoints(N)	10.718	9.702	10.000	9.702	10.890	10830
Routers (N_r)	1.531	1.386	1.000	1.386	990	722
Radix(k)	35	28	33	27	43	43
Electric cables	7.350	6.837	4.500	9.009	6.885	6.669
Fiber cables	24.806	7.716	10.000	4.900	1.012	6.869
Cost per node ($)	2.315	1.566	1.535	1.342	1.365	1.033
Power per node (W)	14,0	11,2	10,8	10,8	10,9	8,02

Plan du cours

Tor

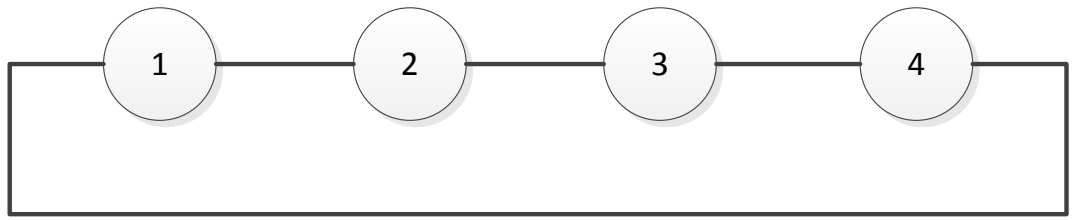


FIGURE – Topologie Tor : 4-ary 1-cube

Discussion sur la topologie k-ary n-cube

- k= nombre d'éléments par dimension
- n = nombre de dimensions
- Degré : 2n : 2
- Diamètre : n*k /2 : 2
- Distance Moyenne n*k /4 : 1
- Bissection : n*k /4 : 1
- Nombre de nœuds : N
- Nombre de liens en fonction du nombre de nœuds : N
- Bande passante : vitesse d'un seul lien, b
- Bande passante totale : N * b

Tor

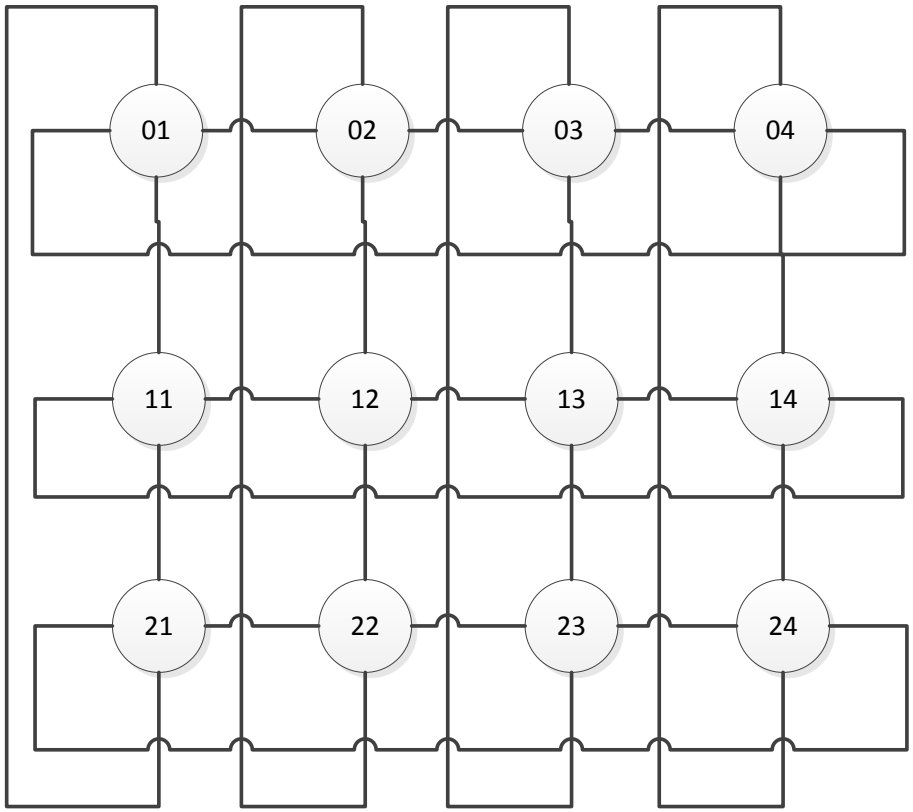


FIGURE – Topologie Tor : 4-ary 2-cube

Discussion sur la topologie k-ary n-cube

- Degré : 2n : 4
- Diamètre : n*k /2 : 8
- Distance Moyenne n*k /4 : 2
- Bissection : n*k /4 : 2
- Nombre de nœuds : N
- Nombre de liens en fonction du nombre de nœuds : N
- Bande passante : vitesse d'un seul lien, b
- Bande passante totale : N * b

Discussion globale sur ces deux topologies

- k-ary : nombre de nœuds par dimension
- 2-cube : dimension du cube
- Path-Diversity : meilleur que le Tor classique
- Résilience : meilleur

Hypercube

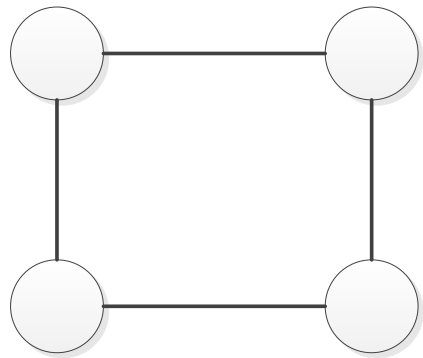


FIGURE – Hypercube : dimension 2

Discussion sur la topologie

- Dimension : d
- Diamètre : d= log N
- Nombre de nœuds :N=2^d
- Nombre de liens en fonction du nombre de nœuds : (N*d/2) = (N*log N) /2
- Bande passante : vitesse d'un seul lien, L
- Bande passante totale : L* (N * log N) /2
- Bissection : N/2
- Distance moyenne : N
- Bissection : N/2

Hypercube

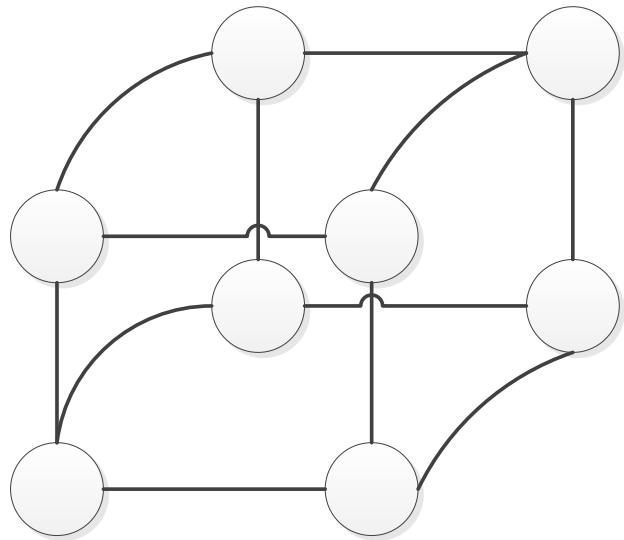


FIGURE – Hypercube : dimension 3

Discussion sur la topologie

- Path-Diversity : assez faible
- Résilience : assez faible

Hypercube

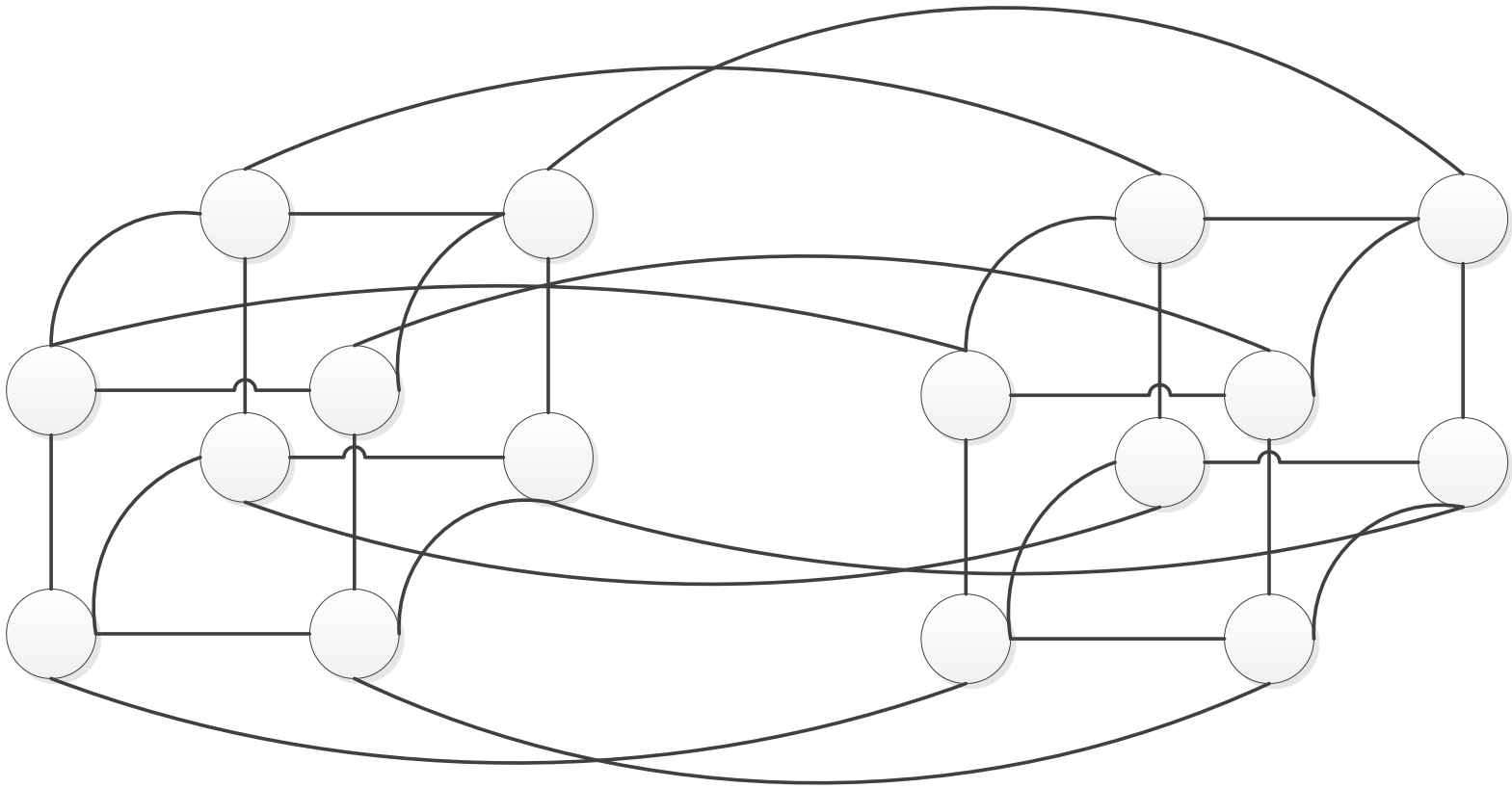


FIGURE – Hypercube : dimension 4

Propriétés de l'hypercube

- Dimension N
- Nombre de nœuds : 2^N
- Nombre de liens par nœud ; N
- Chemin le plus long : N
- Pas dessinable au delà de 4.

Discussion sur la topologie

- Path-Diversity : bonne
- Résilience : bonne

Dragonfly 1/2

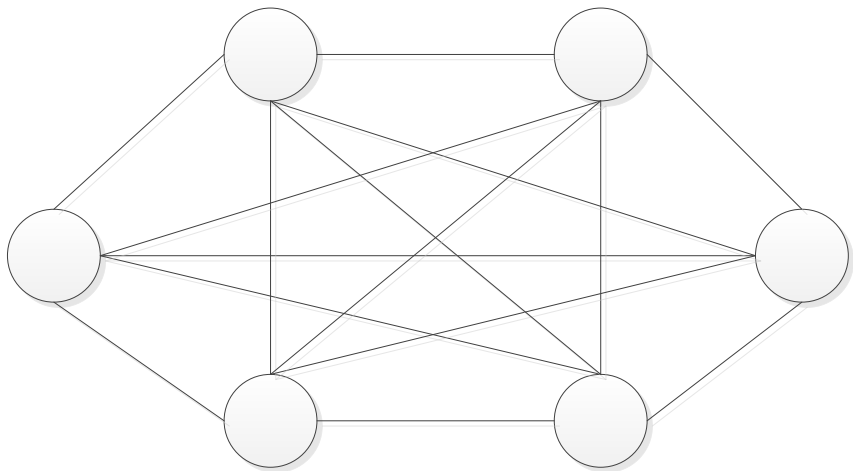
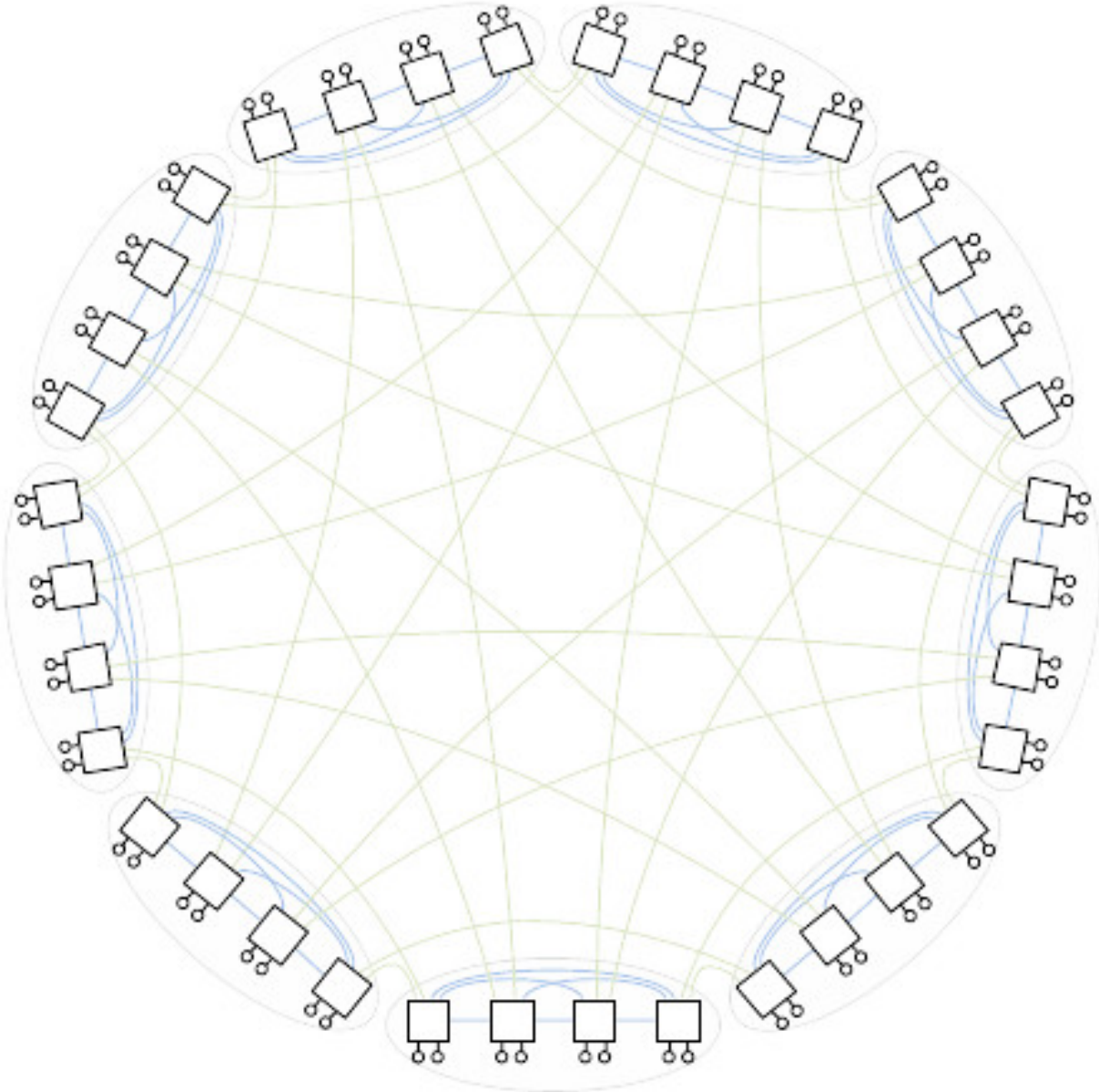


FIGURE – DragonFly : ilôt

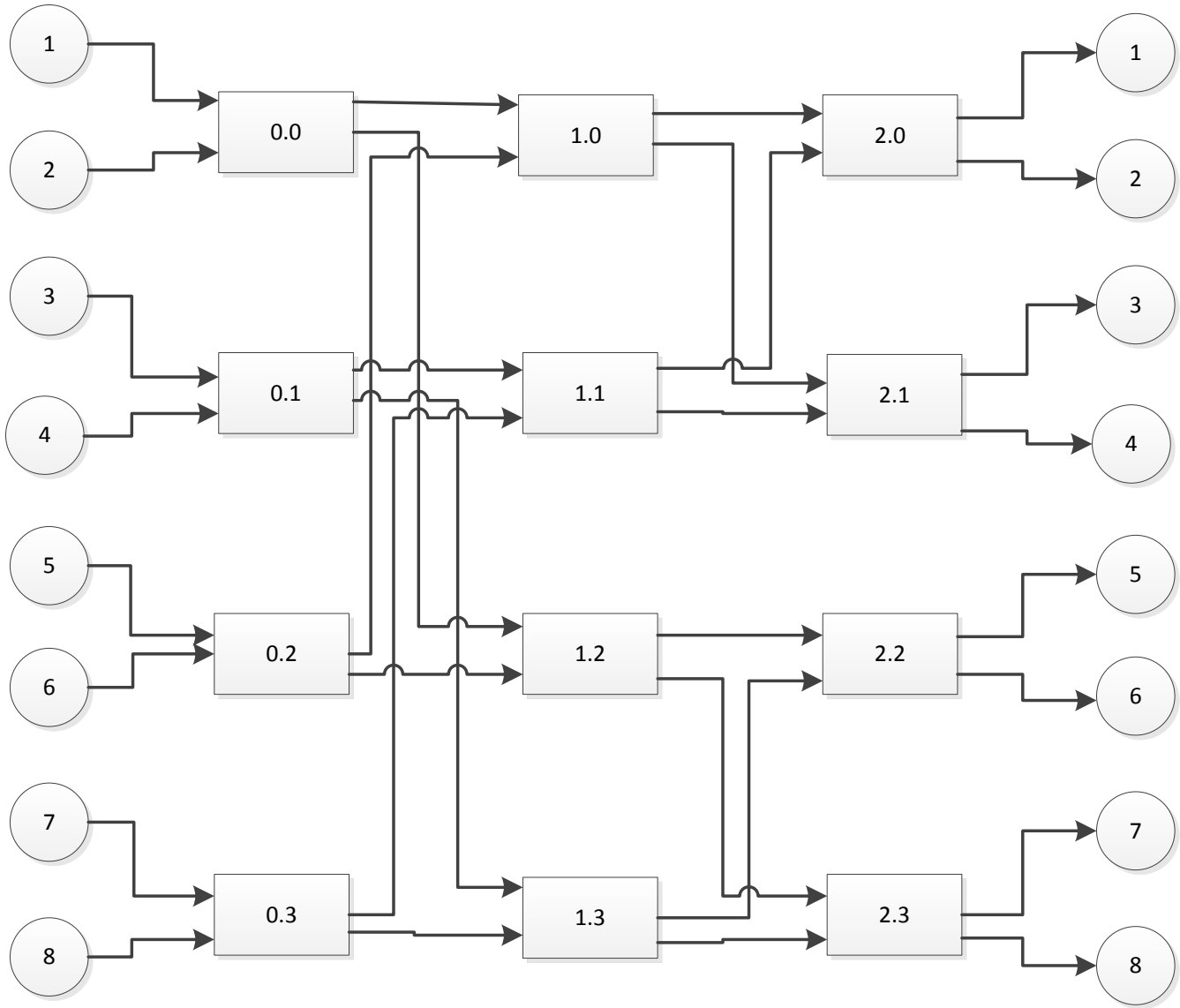
Dragonfly 2/2



Discussion sur la topologie

- Les ilots peuvent avoir une topologie "spécifique" : fat-tree, tor, etc.
- Path-Diversity
- Résilience
- Topologie établie et utilisée par Cray
- Difficile de données des métriques précises
  - Un réseau par (a,p,h) : a=nombre de routeurs, p=nombre de nœuds par routeur, h=nombre de canaux globaux, Balanced : a=2p=2h
  - Nombre de nœuds : (a *p)(a*h +1)
  - Radix des routeur : p+a+h-1
  - K' = a*(p+h) : effective radix of groups

Butterfly 2-ary 3-fly



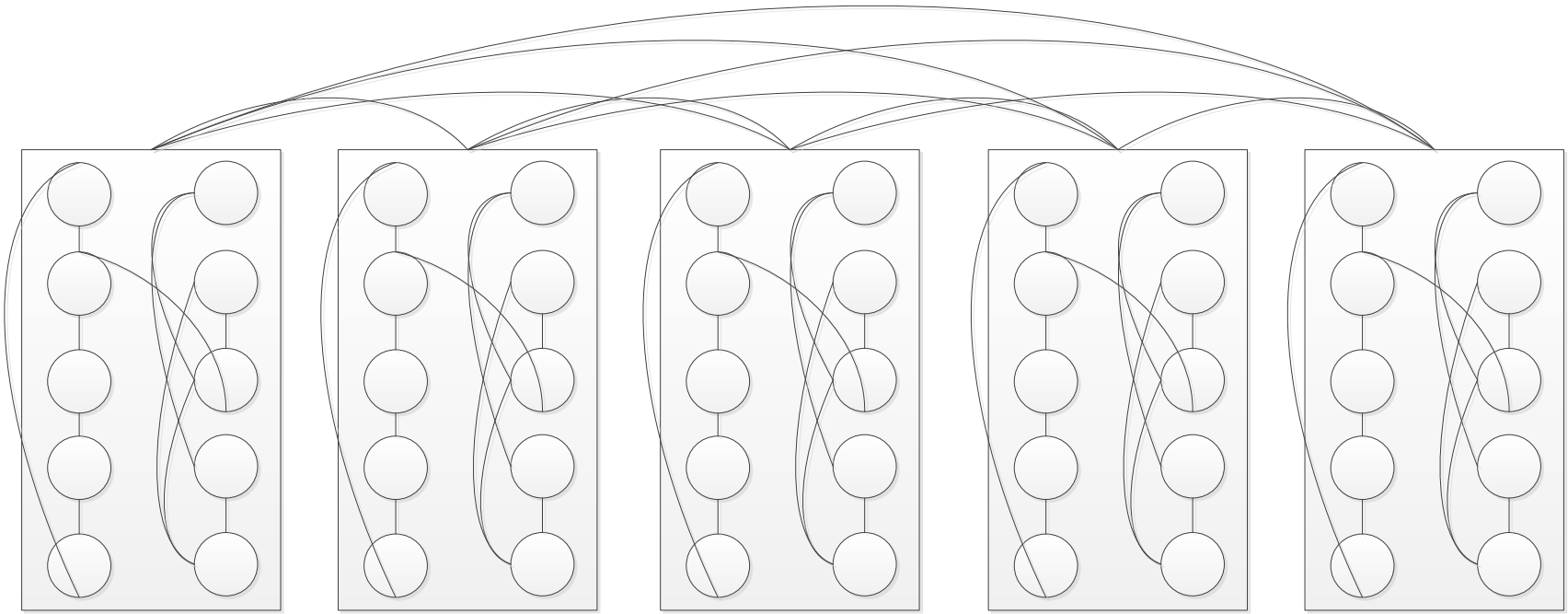
Propriété du butterfly : k-ary n-fly

- Nombre de "stage" : n
- Nombre de nœuds : k^n
- Degré : k
- Diamètre : n+1
- Bisection : $k^n/4$
- Hop Count : n+1
- Aucun Path diversity

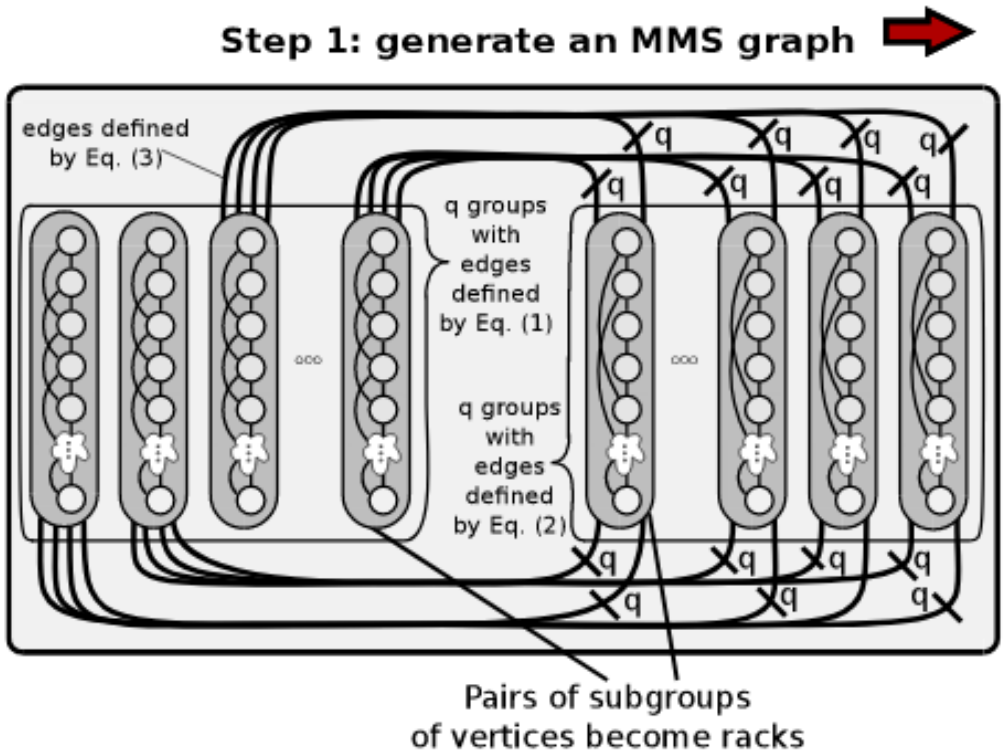
Discussion sur la topologie

- Path-Diversity : inexistant
- Résilience : inexistant

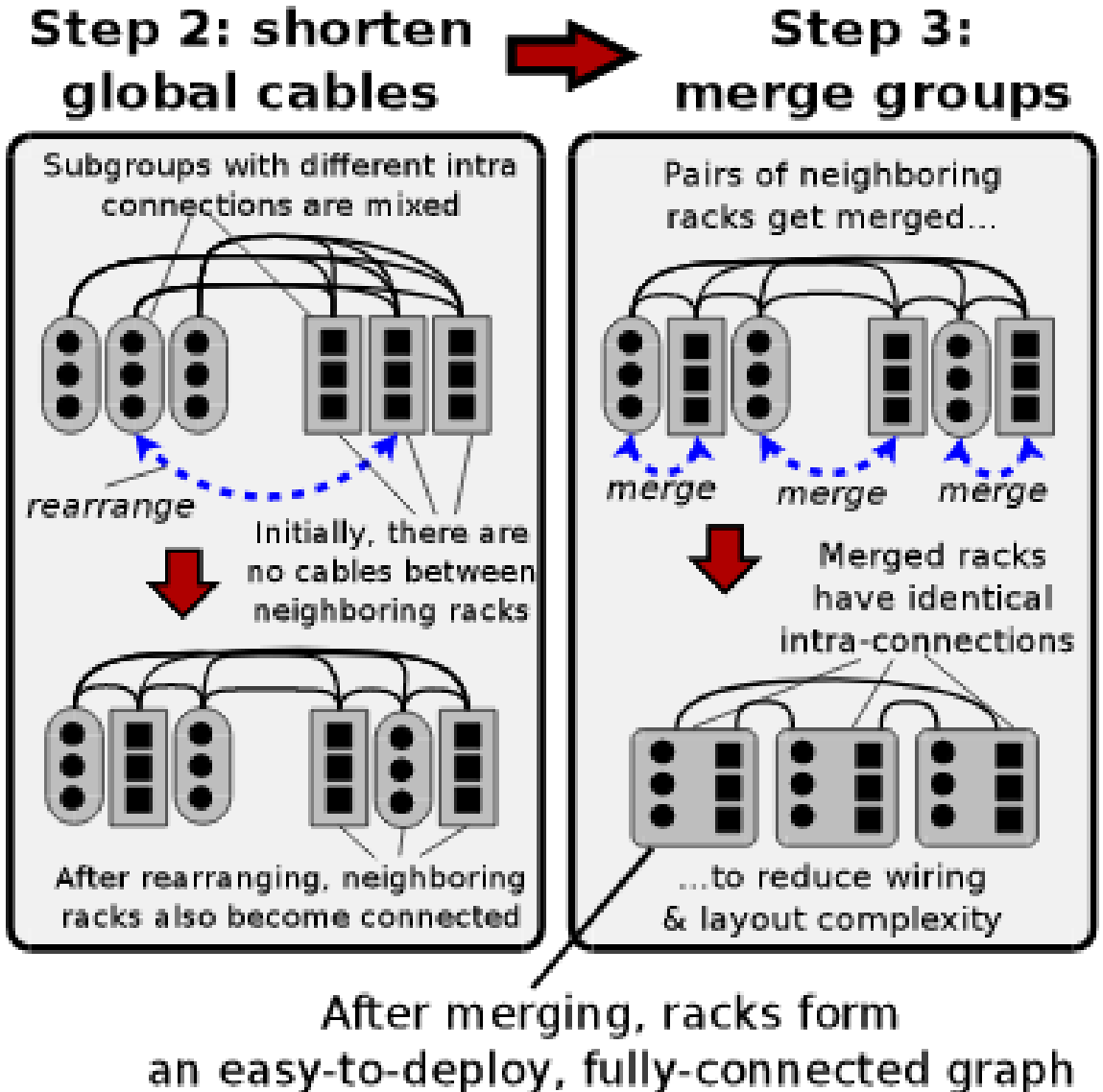
SlimFly



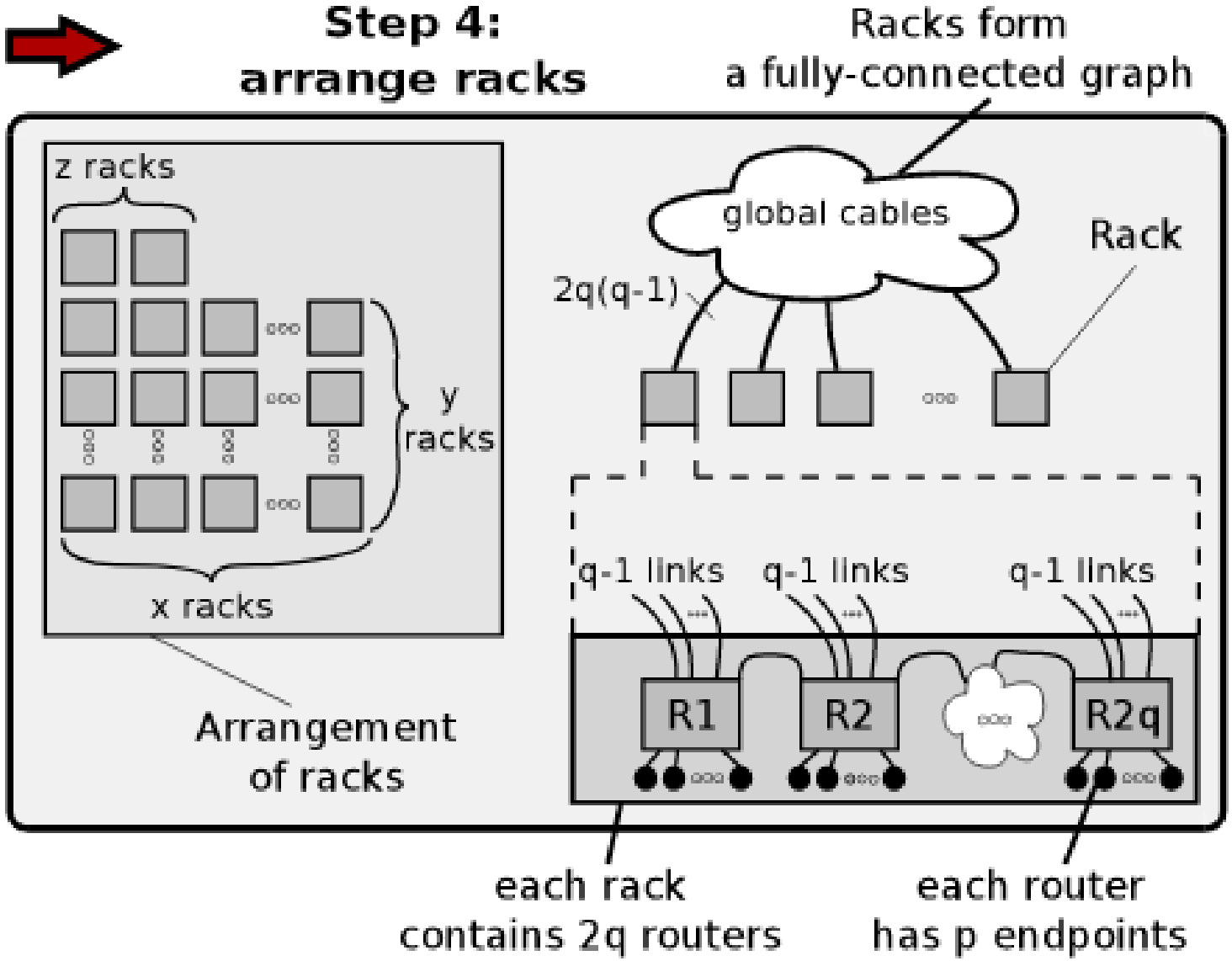
SlimFly : construction



SlimFly : construction



SlimFly : construction



Fat-Tree

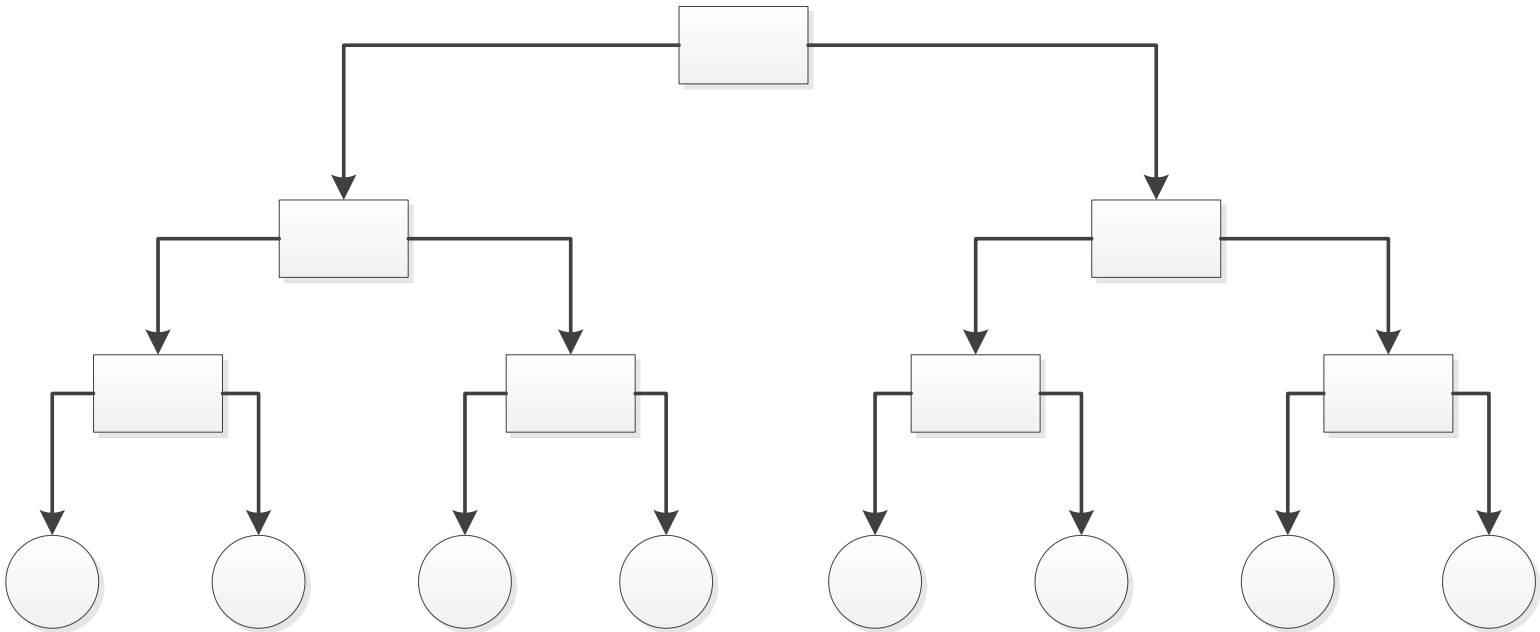


FIGURE – Toplogie Fat-Tree 1

Propriétés des arbres

- k = profondeur de l'arbre
- Nombre de nœuds $N = 2^k - 1$
- Diamètre $2 * k - 2 \equiv 2 \log N$
- Bande passante : L
- Nombre de liens en fonction du nombre de nœuds : 3*N -1
- Bande passante totale : L*(3*N -1)

Discussion sur la topologie

- Path-Diversity : faible
- Résilience : faible
- Inutilisable dans la vraie vie car le radix du nœud root doit être très élevé
- Très faible en cas d'échec du nœud root

XGFT

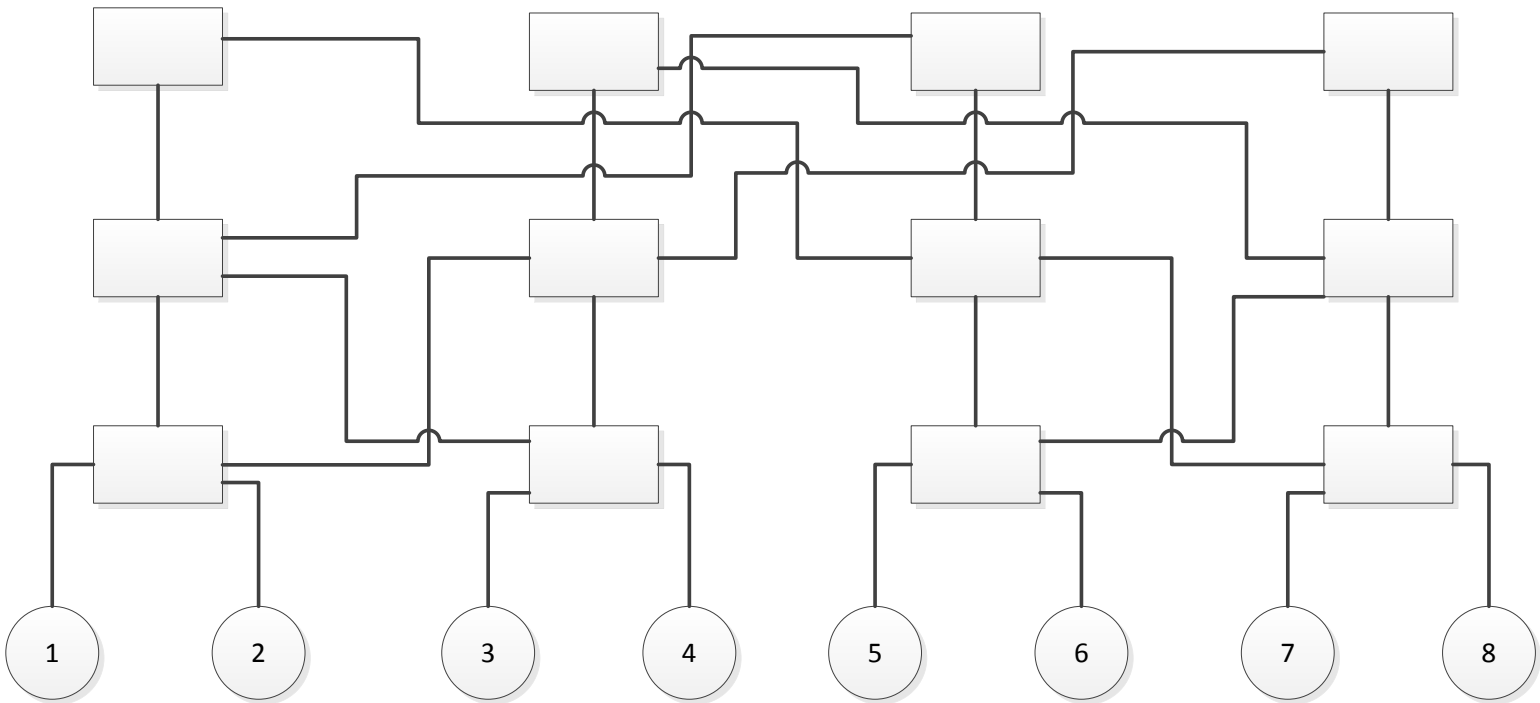


FIGURE – Topologie Fat-Tree 2

Discussion sur la topologie

- Path-Diversity : meilleur
- Résilience : meilleur

Discussion sur k-ary n-tree

- Nombre de nœuds : k^n
- Nombre de switchs : $n * k^{n-1}$

Plan du cours

Comparaison des topologies

- Tableaux extrait de [Duato et al., 2002] : Diamètre pour la comparaison
- T3D : Tor 3D : $3/2\sqrt[3]{N_r}$
- T5D : Tor 5D : $5/2\sqrt[5]{N_r}$
- HC : HyperCube : $\log_2 N_r$
- LH-HC : Long Hop Hyper Cube : 4-6
- FT-3 : fat tree à 3 niveaux : 4
- DLN : Random Topology : 3-10
- FBF-3 : Flattened Butterfly : 3
- DF : Dragon Fly : 3
- SF : Slim Fly : 2

(N.B. : N_r Nombre de routeurs total)

Low-radix topology

Topologie	T3D	T5D	HC	LH-HC
Enpoints(N)	10.648	10.648	8.192	8.192
Routers (N_r)	10.648	10.368	8.192	8.192
Radix(k)	7	11	14	19
Electric cables	31.900	50.688	32.768	53.248
Fiber cables	0	0	12.288	12.888
Cost per node ($)	1.682	3.176	4.631	6.481
Power per node (W)	19,6	30,8	39,2	53,2

High-radix topology 1/4

Topologie	T3D	T5D	HC	LH-HC
Enpoints(N)	10.648	10.648	8.192	8.192
Routers (N_r)	10.648	10.368	8.192	8.192
Radix(k)	7	11	14	19
Electric cables	31.900	50.688	32.768	53.248
Fiber cables	0	0	12.288	12.888
Cost per node ($)	1.682	3.176	4.631	6.481
Power per node (W)	19,6	30,8	39,2	53,2

High-radix topology 2/4

Topologie	FT-3	DLN	FBF-3	DF
Enpoints(N)	19.876	40.200	20.736	58.806
Routers (N_r)	2.311	4.020	1.728	5.346
Radix(k)	43	43	43	43
Electric cables	19.414	32.488	9.504	56.133
Fiber cables	40.215	33.842	20.739	29.524
Cost per node ($)	2.346	1.743	1.570	1.438
Power per node (W)	14,0	12,04	10,8	10,9

High-radix topology 3/4

Topologie	FT-3	DLN	FBF-3	DF
Enpoints(N)	10.718	9.702	10.000	9.702
Routers (N_r)	1.531	1.386	1.000	1.386
Radix(k)	35	28	33	27
Electric cables	7.350	6.837	4.500	9.009
Fiber cables	24.806	7.716	10.000	4.900
Cost per node ($)	2.315	1.566	1.535	1.342
Power per node (W)	14,0	11,2	10,8	10,8

High-radix topology 4/4

Topologie	DF	SF
Enpoints(N)	10.890	10830
Routers (N_r)	990	722
Radix(k)	43	43
Electric cables	6.885	6.669
Fiber cables	1.012	6.869
Cost per node ($)	1.365	1.033
Power per node (W)	10,9	8,02

Synthèse des tableaux de comparaison

Topologie	T3D	T5D	HC	LH-HC	FT-3	DLN	FBF-3	DF
Enpoints(N)	10.648	10.648	8.192	8.192	19.876	40.200	20.736	58.806
Routers (N_r)	10.648	10.368	8.192	8.192	2.311	4.020	1.728	5.346
Radix(k)	7	11	14	19	43	43	43	43
Electric cables	31.900	50.688	32.768	53.248	19.414	32.488	9.504	56.133
Fiber cables	0	0	12.288	12.888	40.215	33.842	20.739	29.524
Cost per node ($)	1.682	3.176	4.631	6.481	2.346	1.743	1.570	1.438
Power per node (W)	19,6	30,8	39,2	53,2	14,0	12,04	10,8	10,9

Topologie	FT-3	DLN	FBF-3	DF	DF	SF
Enpoints(N)	10.718	9.702	10.000	9.702	10.890	10830
Routers (N_r)	1.531	1.386	1.000	1.386	990	722
Radix(k)	35	28	33	27	43	43
Electric cables	7.350	6.837	4.500	9.009	6.885	6.669
Fiber cables	24.806	7.716	10.000	4.900	1.012	6.869
Cost per node ($)	2.315	1.566	1.535	1.342	1.365	1.033
Power per node (W)	14,0	11,2	10,8	10,8	10,9	8,02

Exercice 1

On souhaite interconnecter 16 nœuds grâce à une topologie de type Butterfly. Nous ne disposons que de switch ayant un radix de 4.

1. Quelle serait la meilleur **configuration** du ButterFly ?
2. Dessinez-la.
3. Quelle est le path-diversity de cette solution ? Comment pourriez-vous améliorer le path-diversity ?

Exercice 2

En considérant la notion de `folded tor`, représentez un folded 4-ary 2-cube.

Exercice 3

On dispose de 30 switchs de 24 ports. On souhaite réaliser une topologie de type XGFT. On veut avoir le pruning suivant : entre L1 et L2 : 1 :1, entre L2 et L3 : 1 :2 pruning
Combien peut-on brancher de nœuds ? Dessinez la représentation de cette topologie.

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

Plan du cours

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

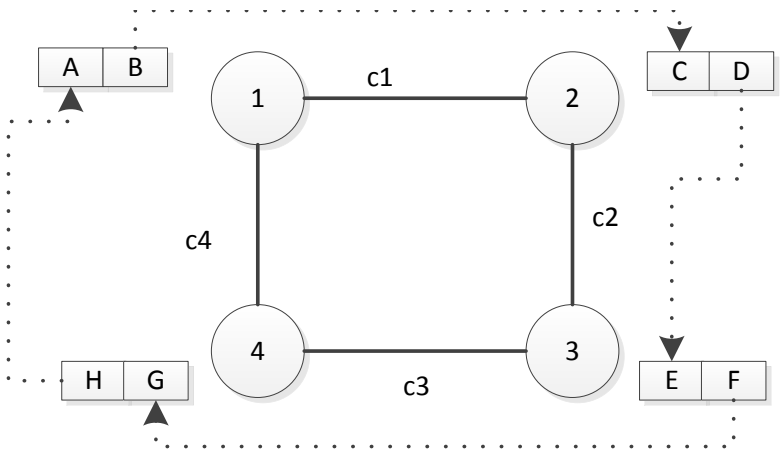
Routage en fonction de la source et de la destination

- 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

Plan du cours

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			

FIGURE – 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 ?

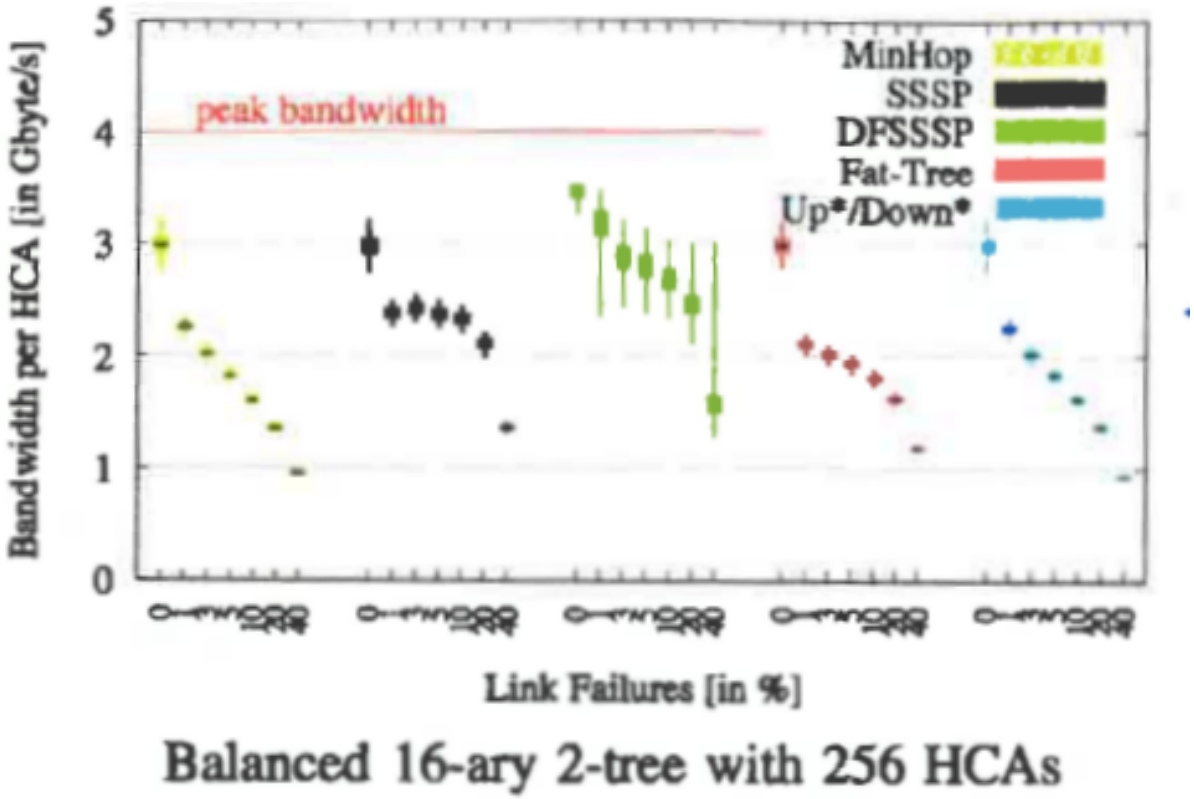
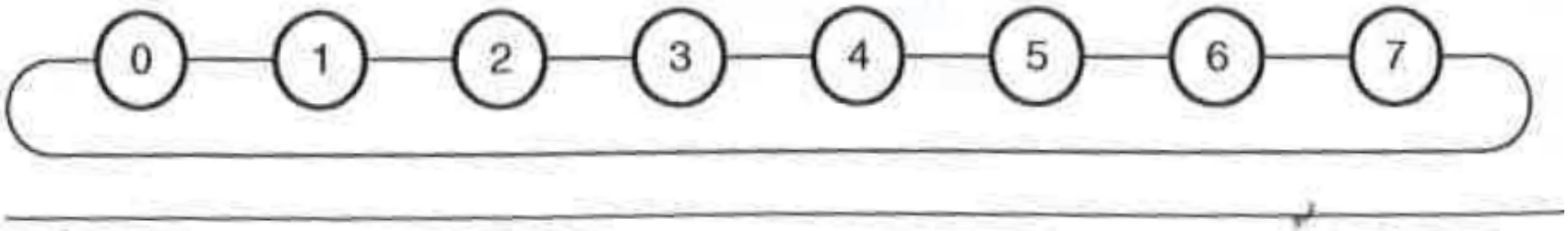


FIGURE – Source : J. Domke, refer to previous slide

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 :

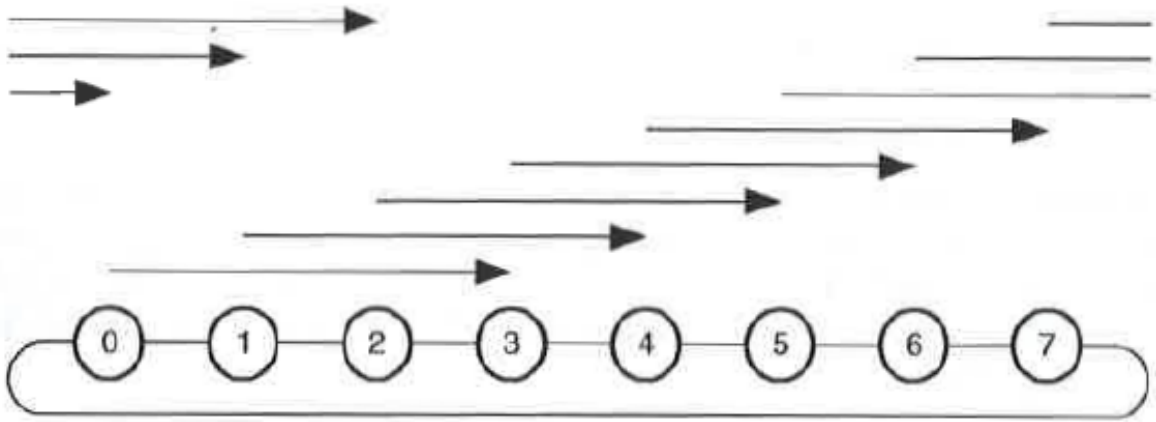
Greedy: Always send the packet in the shortest direction around the ring. For example, always route from 0 to 3 in the clockwise direction and from 0 to 5 in the counterclockwise direction. If the distance is the same in both directions, pick a direction randomly.

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 ?

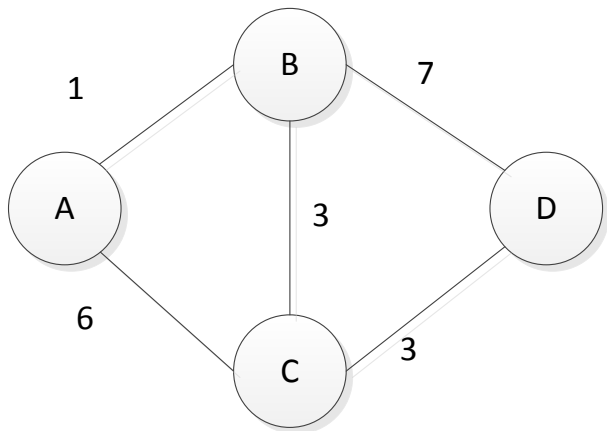
Plan du cours

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

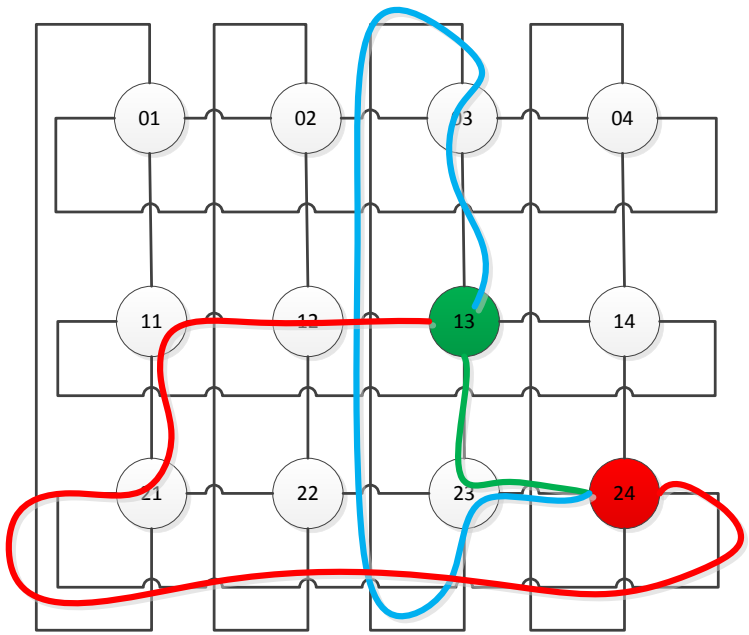


FIGURE – Toplogie Tor : 4-ary 2-cube

Up/Down

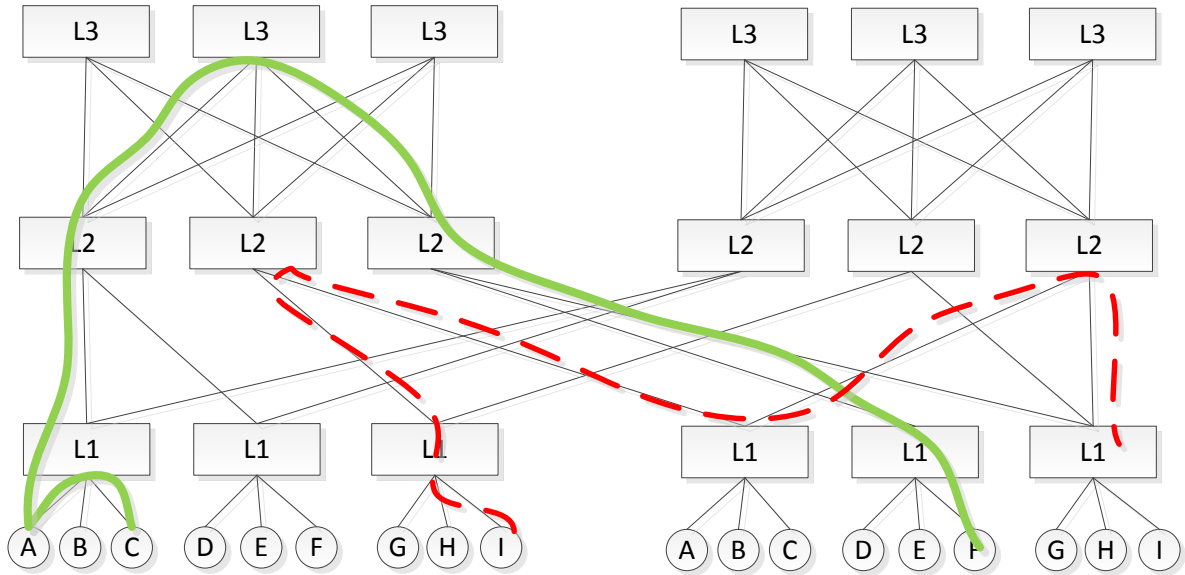


FIGURE – Up/Down

- 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 PGFT.

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.

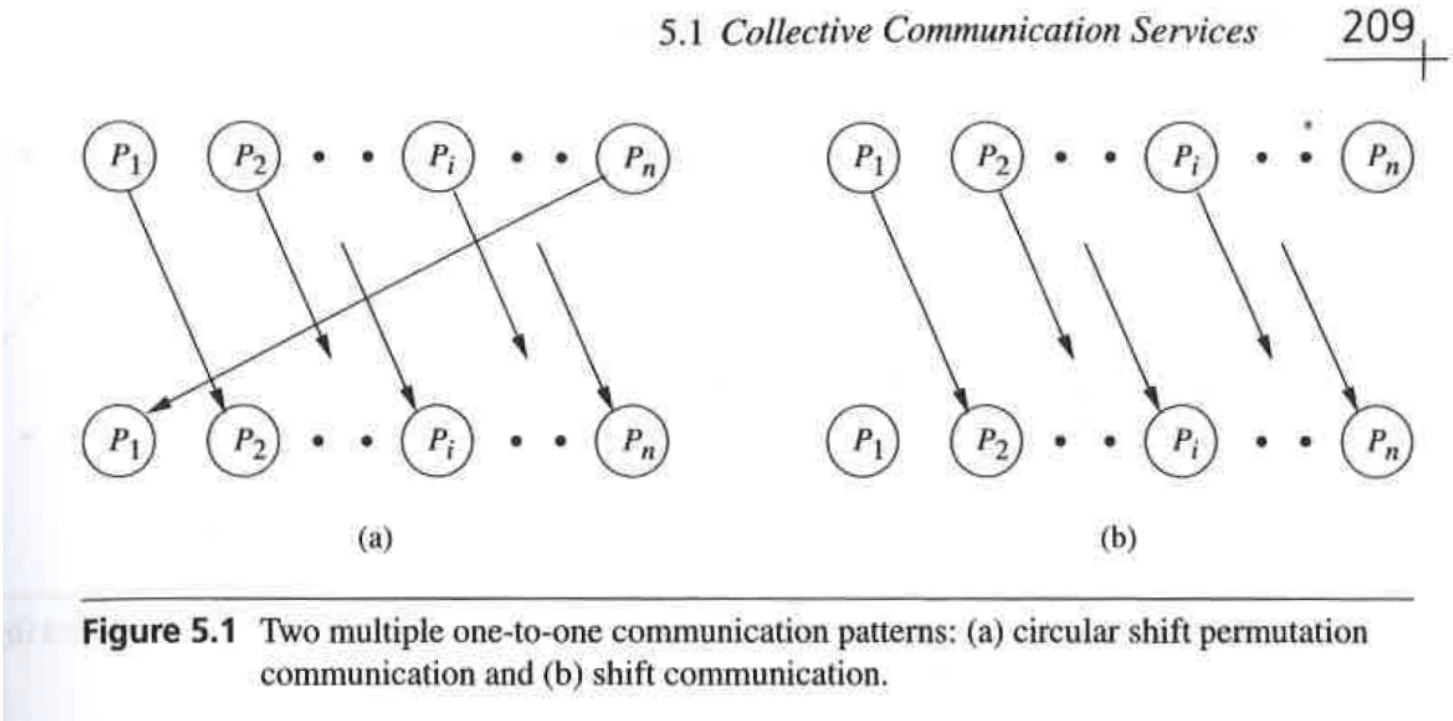


FIGURE – Source : Interconnection networks, José Duato, Sudakhar Yalamanchili, Lionel Li

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

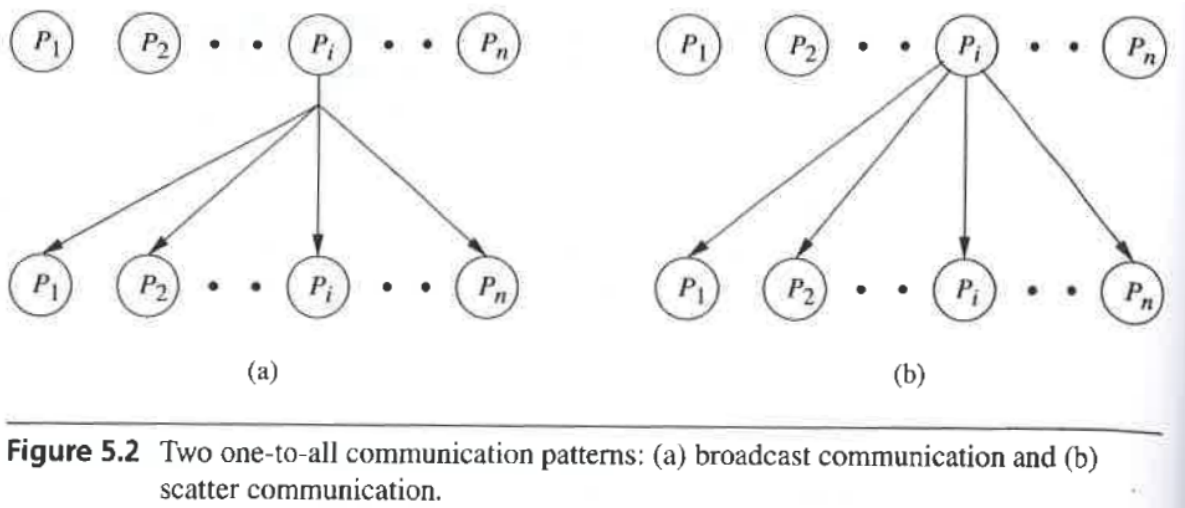


Figure 5.2 Two one-to-all communication patterns: (a) broadcast communication and (b) scatter communication.

FIGURE – 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

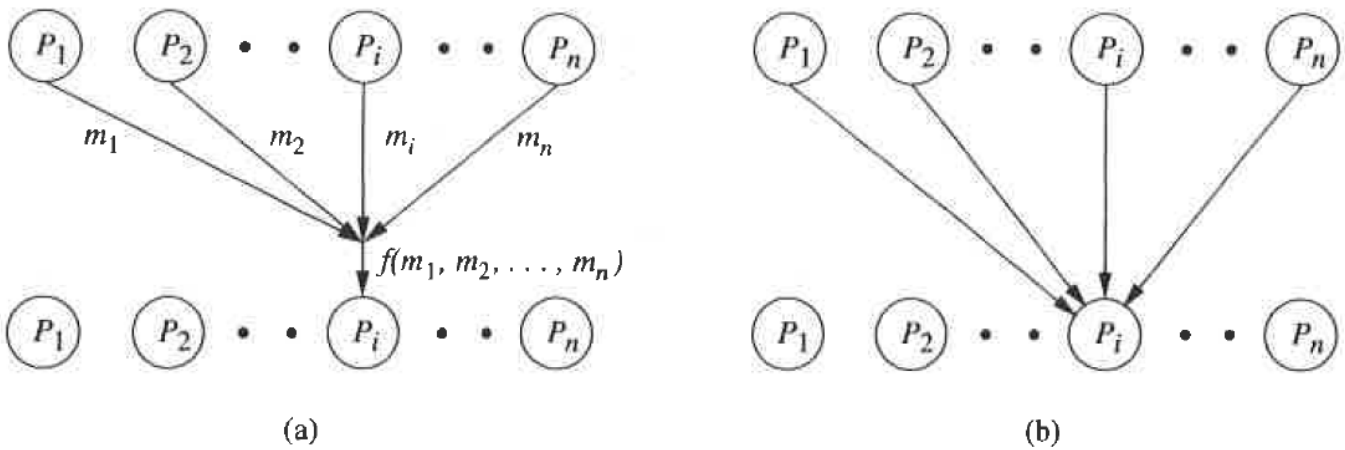


Figure 5.3 Two all-to-one communication patterns: (a) reduce communication and (b) gather communication.

FIGURE – 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

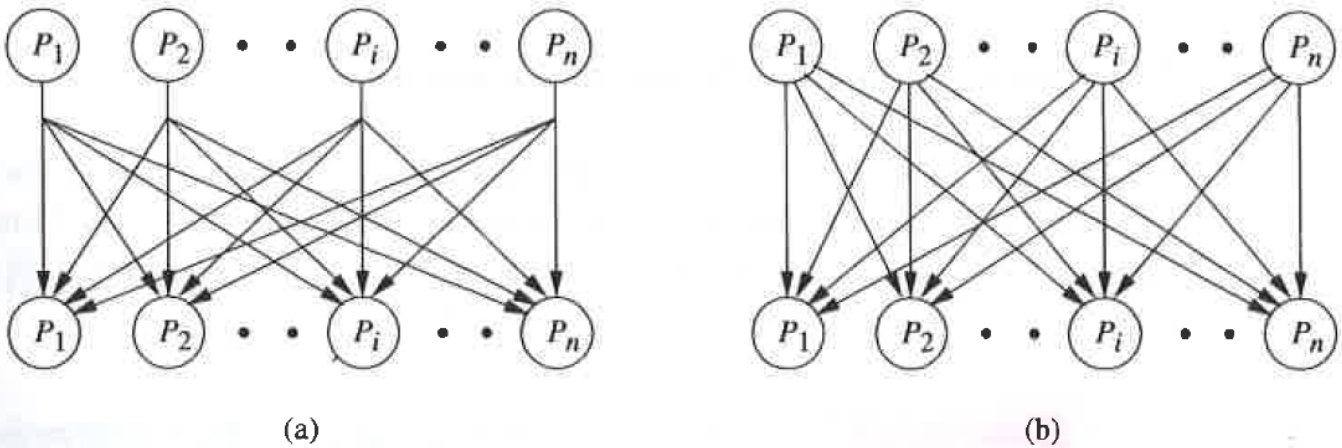


Figure 5.4 Two all-to-all communication patterns: (a) all-broadcast communication and (b) all-scatter communication.

FIGURE – 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*.

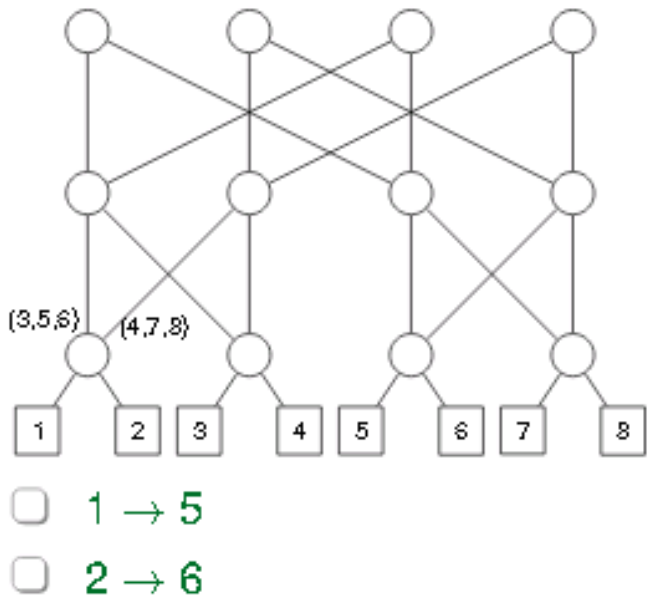


FIGURE – Source : Evaluation pour les routages haute-performance, Matthieu Perrotin, BULL, 2013

Routage bloquant

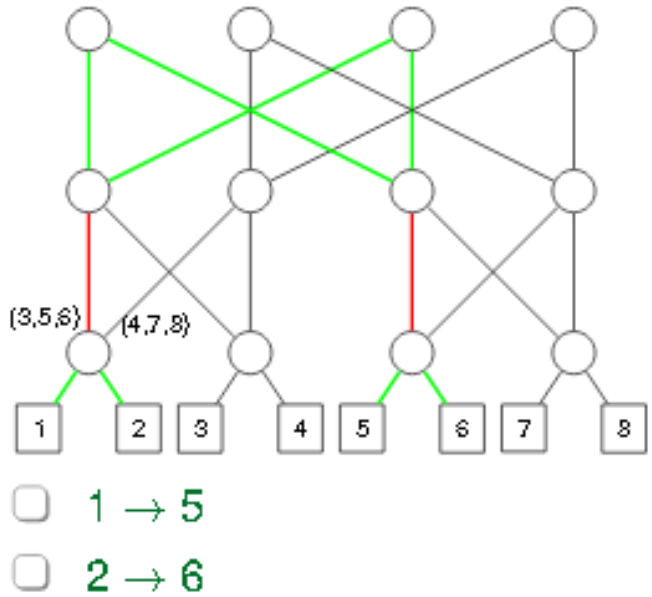


FIGURE – Source : Evaluation pour les routages haute-performance, Matthieu Perrotin, BULL, 2013

Routage non bloquant

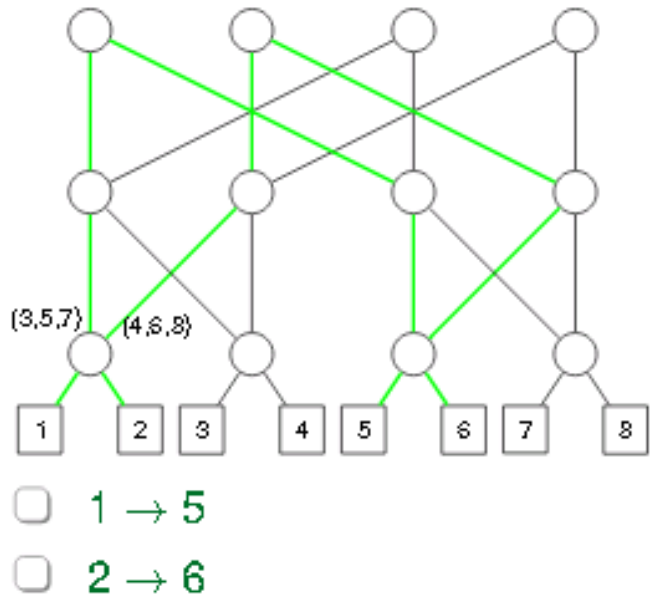


FIGURE – Source : Evaluation pour les routages haute-performance, Matthieu Perrotin, BULL, 2013

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 ?

Plan du cours

Glossaire Commutation/Contrôle de flux

- flux : séquence de paquets entre une paire source-destination ; c'est l'unité sur laquelle s'applique la QoS
- contrôle de flux : synchronisation des ressources du réseau telle que la bande passante, espace buffers etc, garantissant la fiabilité du transport
- message : unité de transfert d'information pour le client du réseau, i.e. l'application (message MPI, message Lustre) ; de taille variable
- paquet : unité de routage dans le réseau ; un message est découpé en un ou plusieurs paquets de taille variable mais limitée ; il y a trois type de paquets (*header, body, tail*)
- flit : *FLow control digIT* : plus petite unité d'allocation de ressources dans un router. Un paquet est divisé en flit de taille fixe pour simplifier l'allocation des ressources dans un routeur (buffer, pipeline)
- phit : *PHysical digIT* : unité de transmission sur le canal
- canaux : *channels*, transportent les paquets entre les différents noeuds du réseau
- *buffers* : ressources mémoire qui stockent temporairement les paquets

Plan du cours

Exemple de paramètres réseau : Interconnect processeur/mémoire

Parameters of processor-memory interconnection networks.

Parameter	*Value*
Processor ports	1–2,048
Memory ports	0–4,096
Peak bandwidth	8 Gbytes/s
Average bandwidth	400 Mbytes/s
Message latency	100 ns
Message size	64 or 576 bits
Traffic patterns	arbitrary
Quality of service	none
Reliability	no message loss
Availability	0.999 to 0.99999

Exemple de paramètres réseau : Interconnect I/O

Parameters of I/O interconnection networks.

Parameter	*Value*
Device ports	1–4,096
Host ports	1–64
Peak bandwidth	200 Mbytes/s
Average bandwidth	1 Mbytes/s (devices) 64 Mbytes/s (hosts)
Message latency	10 μs
Message size	32 bytes or 4 Kbytes
Traffic patterns	arbitrary
Reliability	no message loss[a]
Availability	0.999 to 0.99999

[a] A small amount of loss is acceptable, as the error recovery for a failed I/O operation is much more graceful than for a failed memory reference.

Exemple de paramètres d'un réseau : Fabrique Ethernet-TCP/IP

Parameters of a packet switching fabric.

Parameter	*Value*
Ports	4–512
Peak Bandwidth	10 Gbits/s
Average Bandwidth	7 Gbits/s
Message Latency	10 μs
Packet Payload Size	40–64 Kbytes
Traffic Patterns	arbitrary
Reliability	$< 10^{-15}$ loss rate
Quality of Service	needed
Availability	0.999 to 0.99999

Les couches OSI

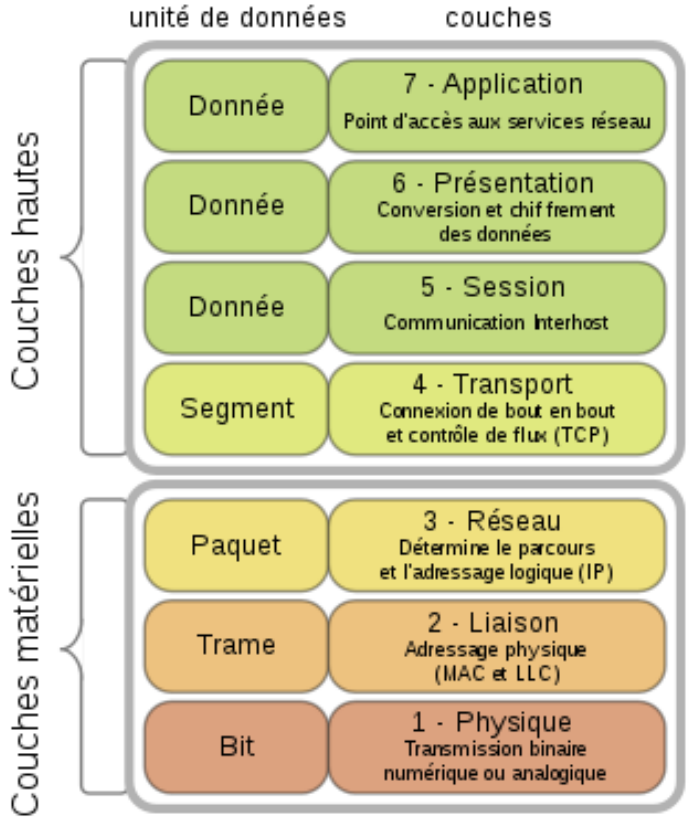
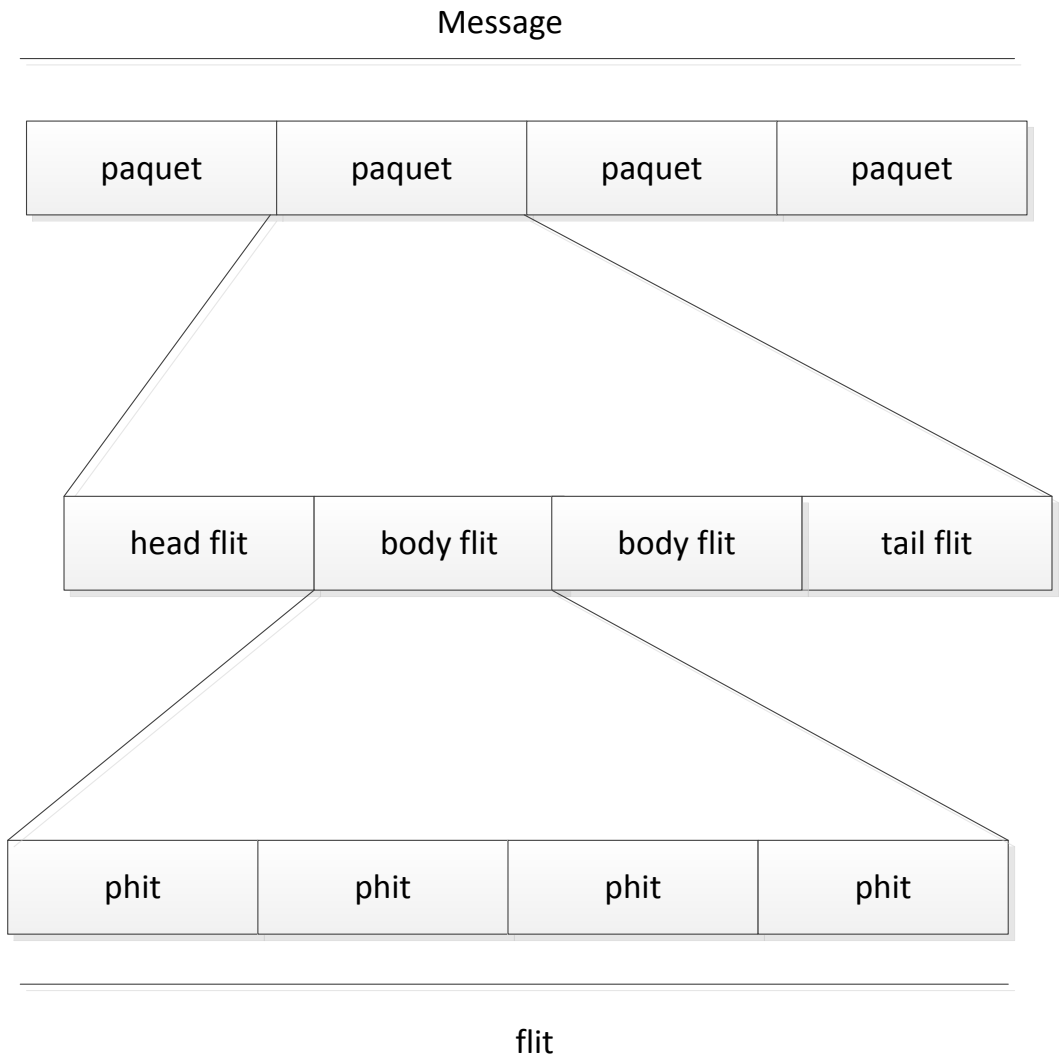


FIGURE – Souce : *Wikipédia, Modèle OSI*

Les unités d'allocation



Les unités d'allocation

- Quelles informations contient le message ? données applicatives utiles ou instructions (MPI send, MPI WAIT, ACK protocolaire, ...)
- Quel découpage ? paquet/flit : encapsulation protocolaire et découpage permettant de limiter la taille et le temps d'allocation des ressources sur le réseau ;
  - Body flit : morceau (*chunk*) de données
  - Head flit : entête de routage
    - entête IP : @IP DST, @IP SRC, Port Src, Port Dst, taille du paquet, Service, TTL, checksum,...
    - entête routage local IB : LID DST, numéro de VL, Service Level, LID SRC, taille du paquet,...
    - entête BXI : @DST, #VC, bit de routage adaptatif ou non, ...
  - Tail flit : flit de queue, fin du paquet ; indique la fin de validité des infos de routage et contient un E2E-CRC pour assurer la validité des données transmises voire permettre l'application d'un code correcteur d'erreur
- Quelles informations contient le phit ? découpage et encodage pour transmission sur le canal physique ;

Fine-grained latence

- La latence est le temps mis par un **paquet** pour traverser le réseau, du *Head Flit* au *Tail flit*
- Elle se décompose donc en deux parties :
  - Th : temps mis par le *header* pour traverser le réseau = temps de traversée des routeurs Tr et temps de transfert sur les canaux Tw
    - Tr = nombre de *hops* x temps unitaire de traversée d'un switch
    - Tw = Distance du câble / vitesse de propagation
  - Ts : temps de sérialisation, ou temps mis par un paquet de taille L pour traverser un réseau de bande passante b

$$T_0 = \underbrace{H_{\min} t_r}_{\text{router}} + \underbrace{\frac{L}{b}}_{\text{serialization}} + \underbrace{\frac{D_{\min}}{v}}_{\text{wire}}$$

Un peu de réflexion

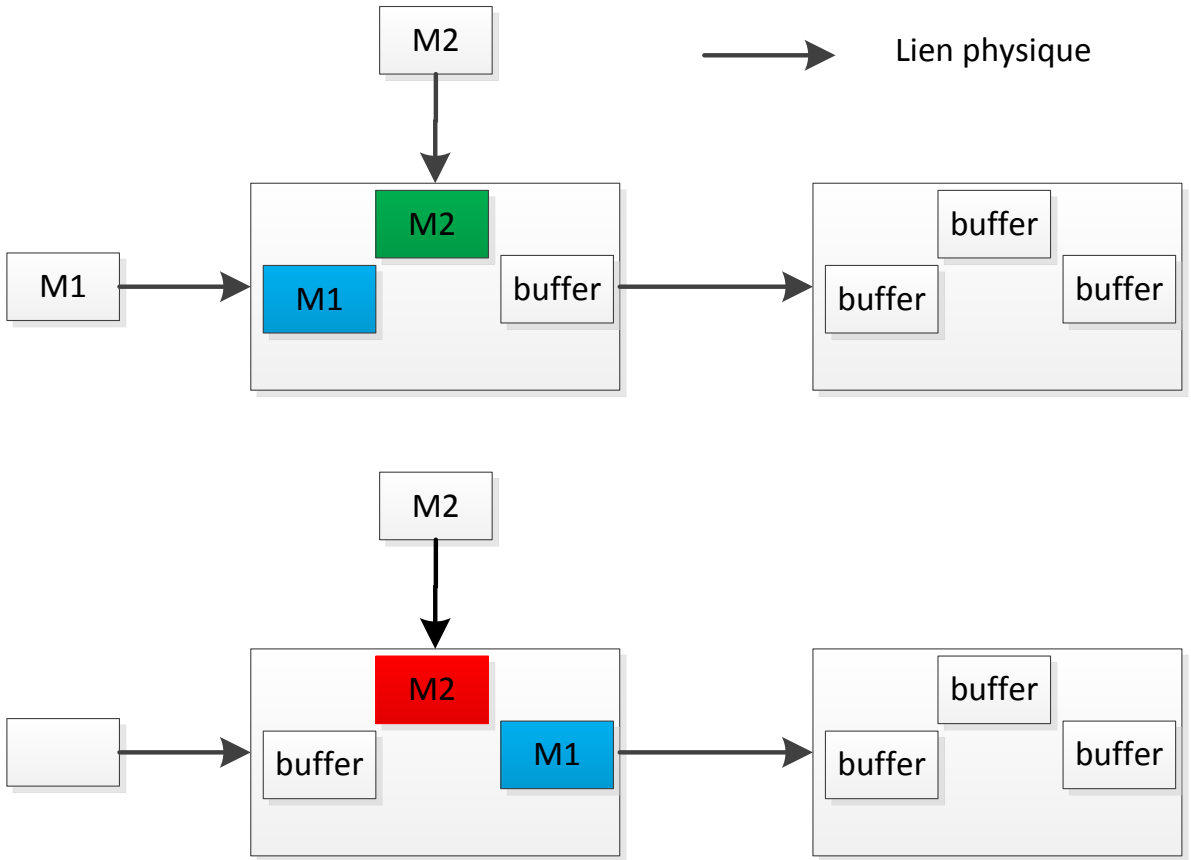
- Soit une technologie réseau transmettant sur 4 canaux physiques à 25Gb/s par lien, avec un encodage 64b/66b
- Soit une unité de routage (*flit*) de 32 octets, l'encapsulation du routage (*header*, *tail*) étant de 1 flit chacun
- La taille maximale d'un paquet étant de 1Ko (1024 octets), calculez :
  - le débit utile maximum atteignable ?
  - la latence d'un paquet sans *payload* ?
  - la latence d'un message de 512 o ?
  - la latence d'un message de 64 Ko ?

Plan du cours

Rappels

- La **topologie** décrit le motif d'interconnexion des noeuds. Selon quels critéres ?
- Le **routage** déterminent quels chemins sont empruntés par les messages. Selon quels critères ?
- Le **mode de commutation ou contrôle de flux** détermine quels messages a accès à quelle resssource réseau. Ce sont les feux de circulation. Pour atteindre les performances potentielles apportées par la topologie et le réseau, le controle de flux doit éviter les conflits d'accès aux ressources. Un message bloqué sur un canal ne doit pas empéché un message d'accèder à un autre canal (*Head Of Line blocking*)

Allocation des ressources 1/2



Allocation des ressources 2/2

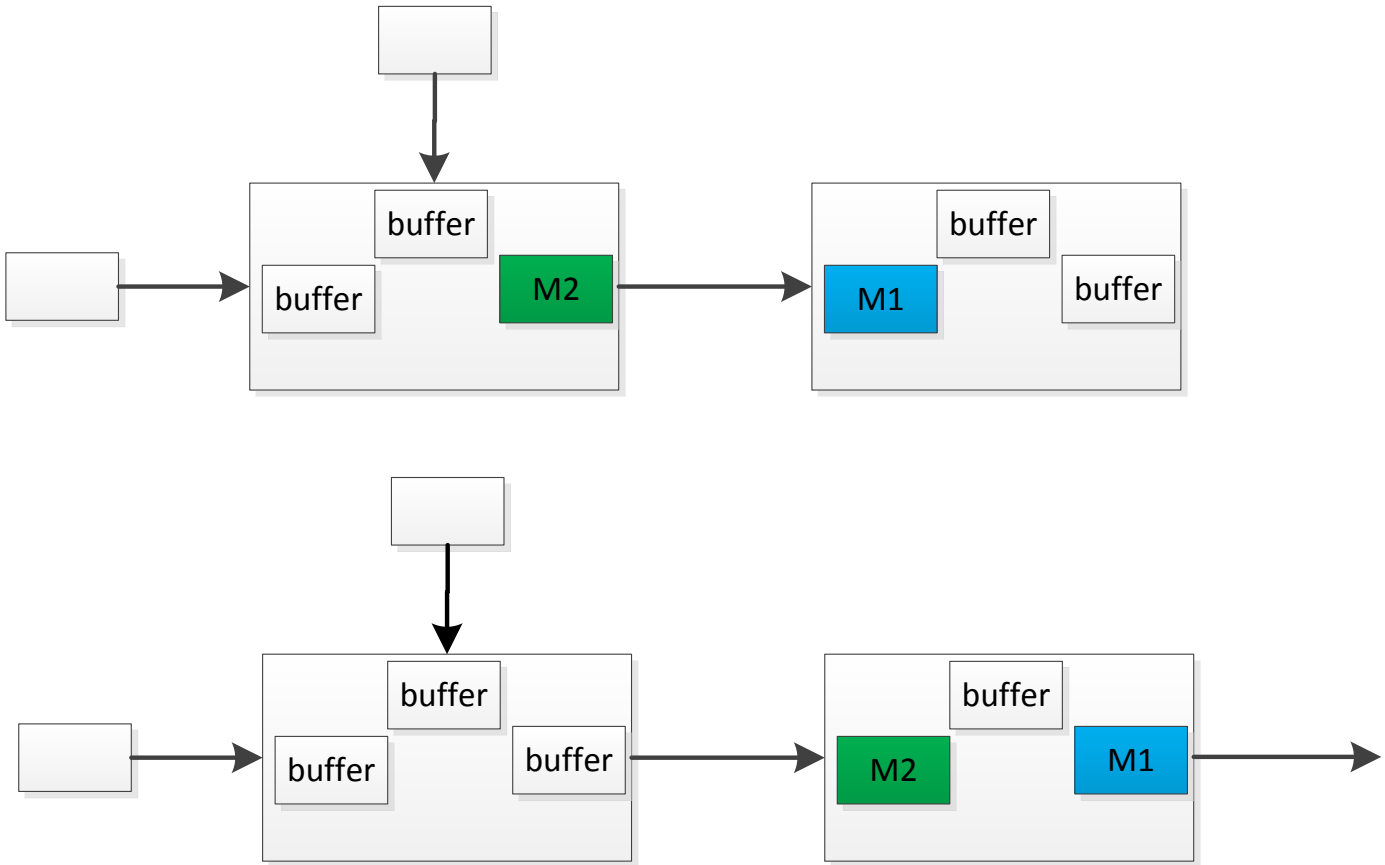
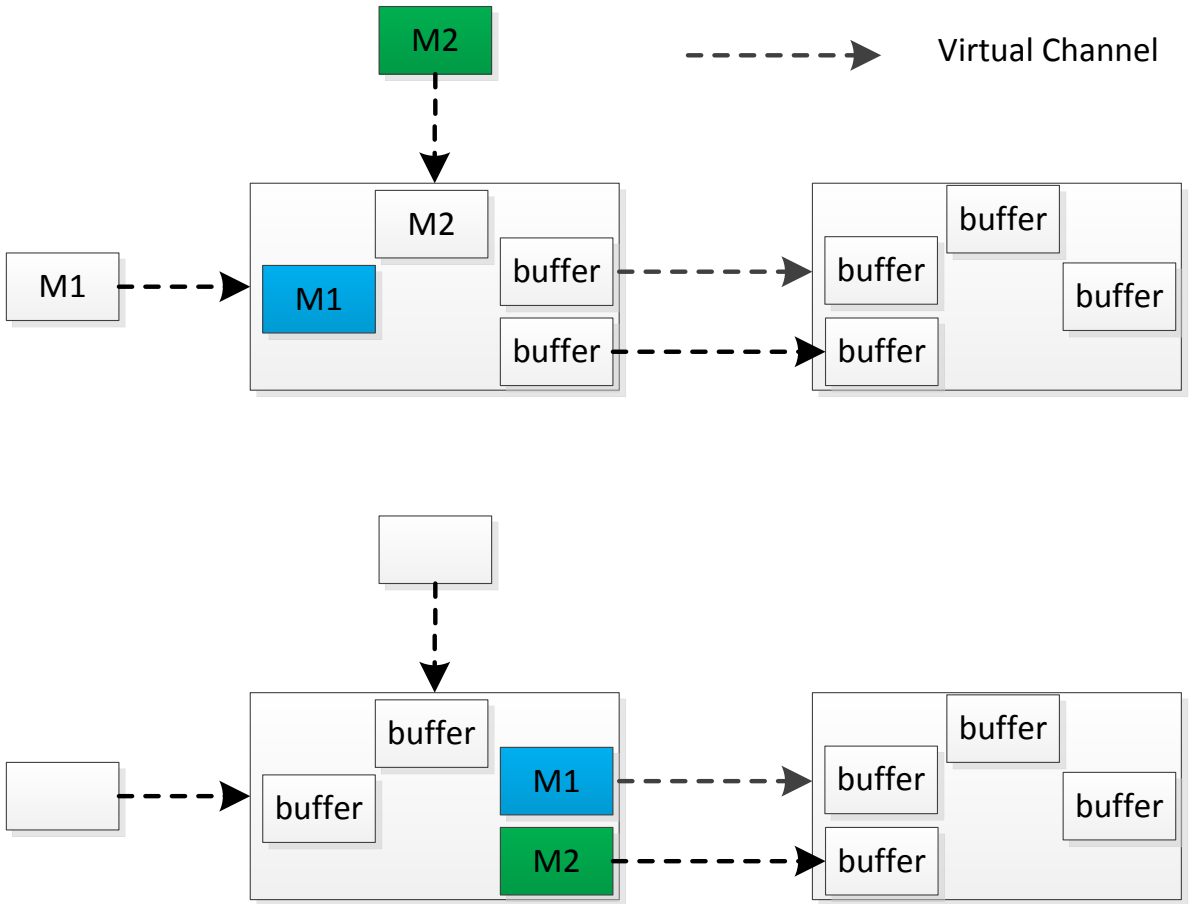


FIGURE – Latence de traversée pour M1 : 1 cycle, pour M2 : 2 cycles

Découplage des ressources 1/2, *virtual channel*



Découplage des ressources 2/2, *Virtual channel*

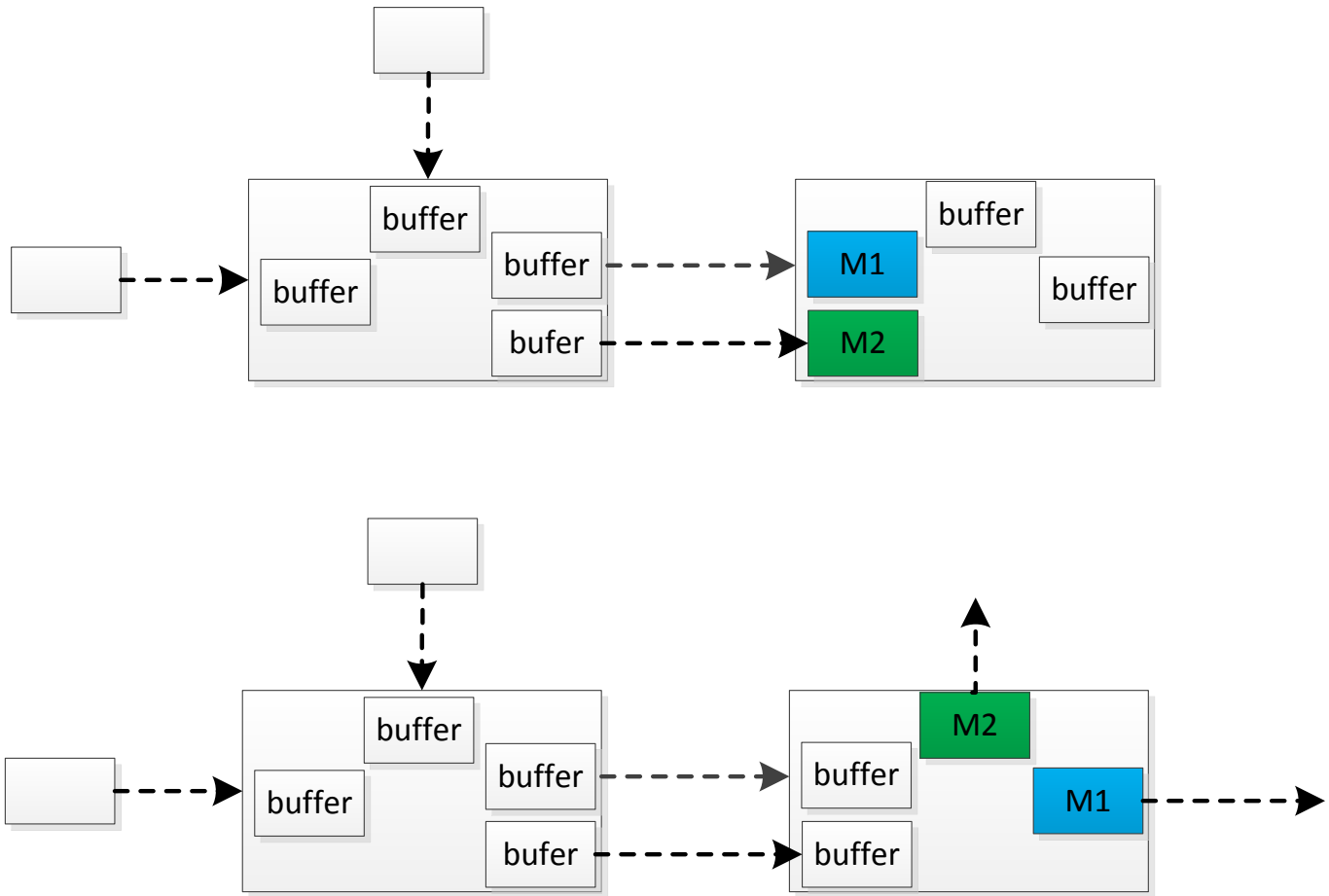


FIGURE – Virtual channel = multiplexage des flux

Head of Line blocking

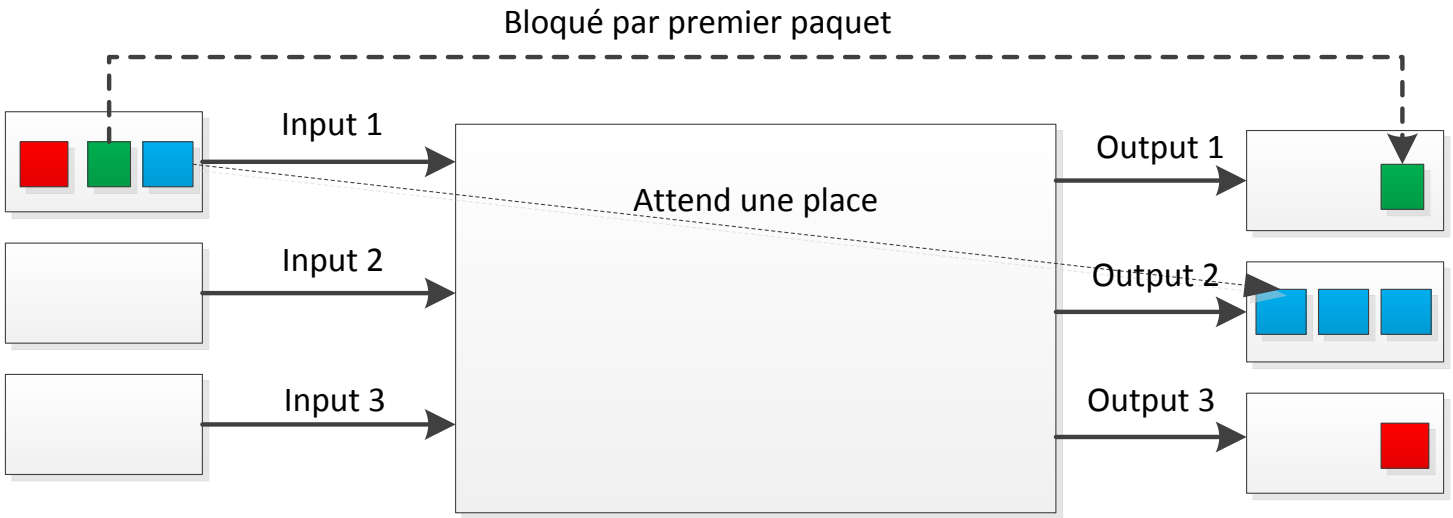


FIGURE – Head of line blocking

Head of Line blocking mitigation

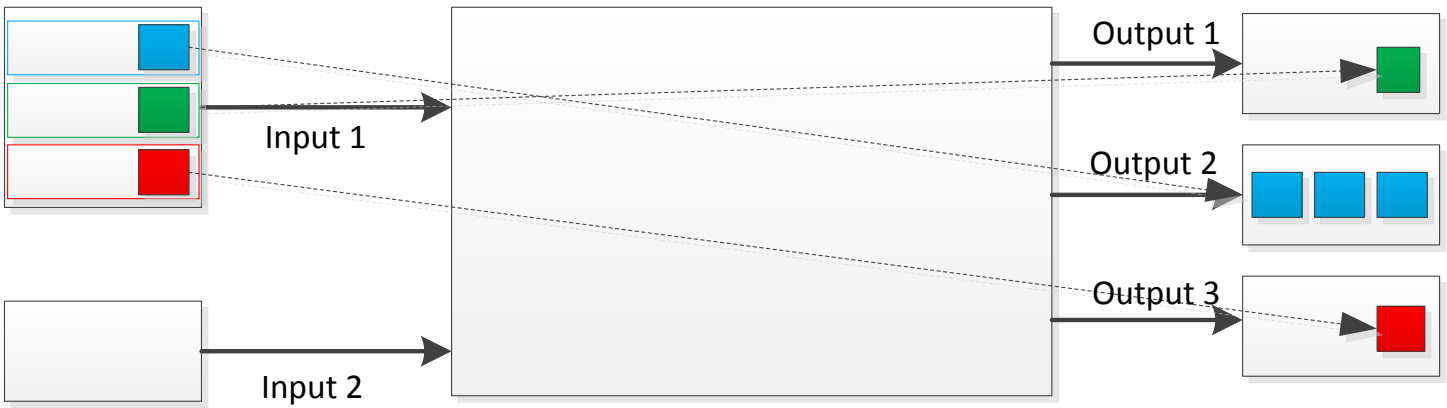


FIGURE – Utilisation de multiples *Virtual Channels*

Commutation *store-and-forward* 1/2

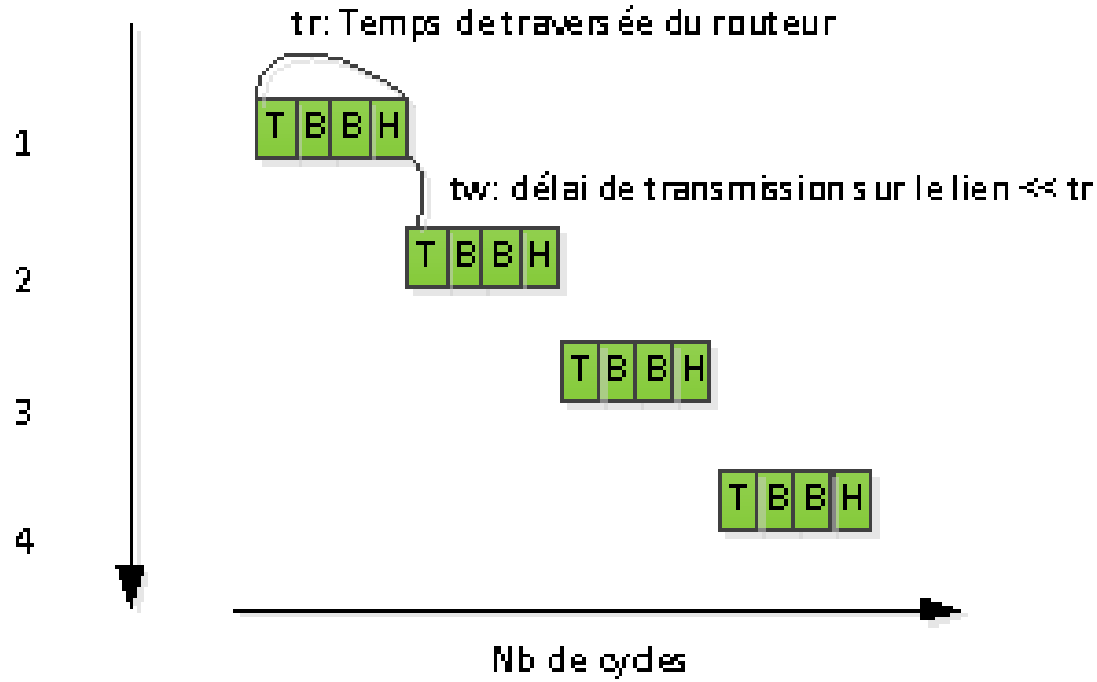


FIGURE – Diagramme de progression : SAF avec tr = 1 par flit et tw=0

Commutation *store-and-forward* 2/2

- En mode *SAF*, l'allocation des ressources (canal et buffer) se fait par paquet. Chaque routeur sur la route attend que l'ensemble des flits d'un paquet soit arrivé, avant de le transmettre sur le canal suivant, une fois celui-ci disponible.
- Aucun canal n'est bloqué par le paquet mais ce mode de commutation induit une forte latence car le temps de sérialisation se cumule à chaque saut.
- La latence s'exprime donc ainsi :
  - To = #hop x (tr + L/b)

Commutation *Cut-through* 1/2

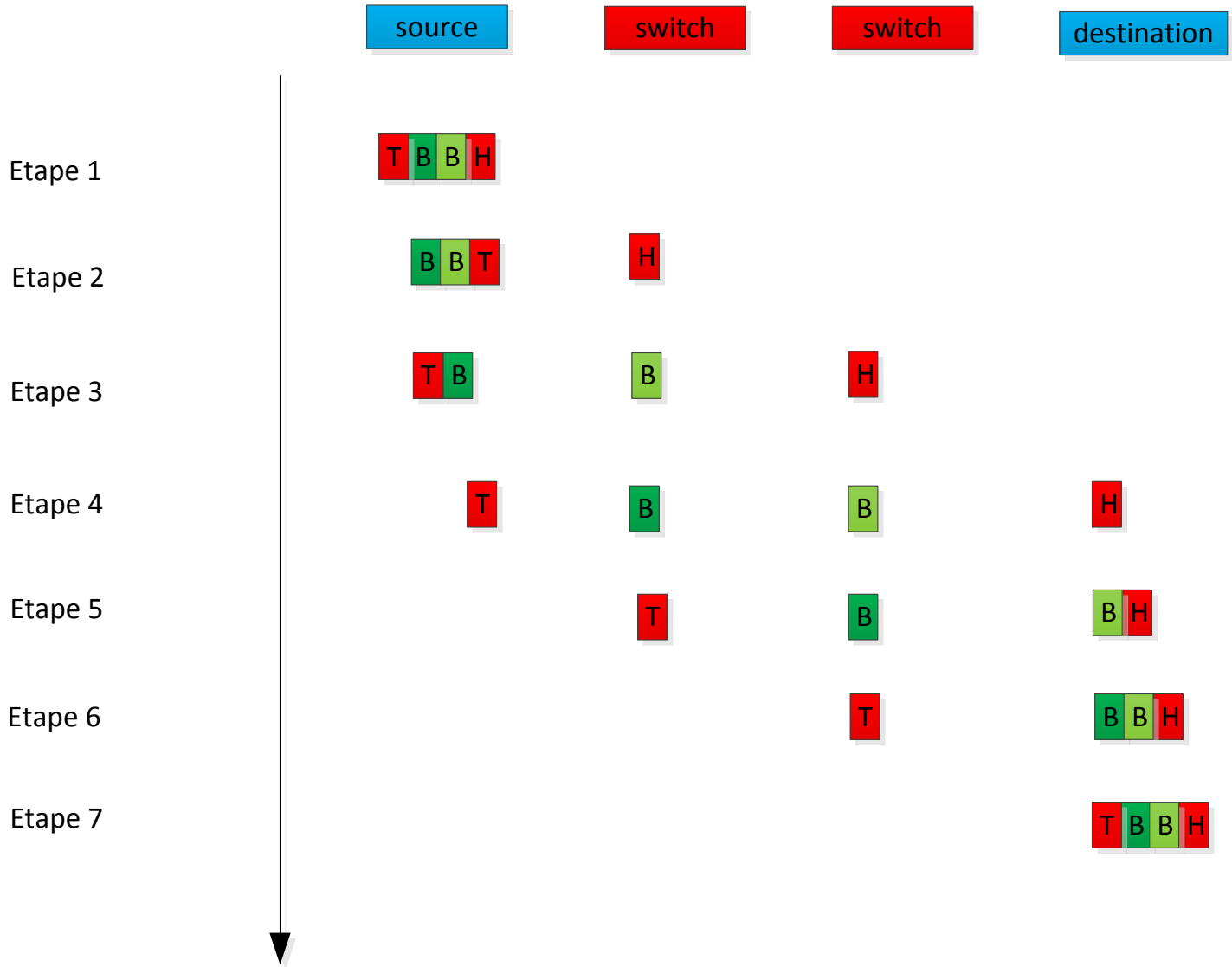


FIGURE – Diagramme de progression *cut-through*

Commutation *Cut-through* 2/2

- En mode *cut-through*, l'allocation des ressources (canal et buffer) se fait aussi pour chaque paquet.
- Cependant, le routeur transmet un paquet dès que le *header flit]* est reçu et que les ressources (buffer et canal) sont disponibles.
- L'avantage est que le temps de sérialisation ne se cumule plus à chaque étape.
- La latence s'exprime donc ainsi :
  - To = #hop x tr + L/b

Commutation en mode *wormhole*

- Le mode *wormhole* fonctionne comme le mode *cut-through*, car le routeur transmet chaque flit dès qu'il est reçu, mais l'allocation des buffer et canal est faite pour un flit et non tout un paquet.
- La latence s'exprime aussi ainsi :
  - To = #hop x tr + L/b
- L'avantage d'une latence minimale est conservé, tout en optimisant les ressources à chaque hop. Cela permet de concevoir des routeurs simples, rapides et peu chers.
- L'inconvénient majeur est de cette technique est sa sensibilité aux problèmes de *Head-Of-Line blocking*
- Dans un buffer d'entrée (soit un canal physique), on ne mixe pas les flits de deux messages différents car un body flit ne contient d'information d'identité.
- Il est donc nécessaire de diviser un canal physique en plusieurs canaux logiques pour augmenter les plans de transmission.
- Chaque *Virtual Channel* utilise un buffer d'entrée propre, et les paquets sont mutlipléxés lors de la transmission sur le lien physique.
- Pour coordonner l'envoi des paquets sur les ports de sortie, le routeur doit mémoriser l'état d'allocation des *Virtual Channel*.

Commutation en mode *wormhole*

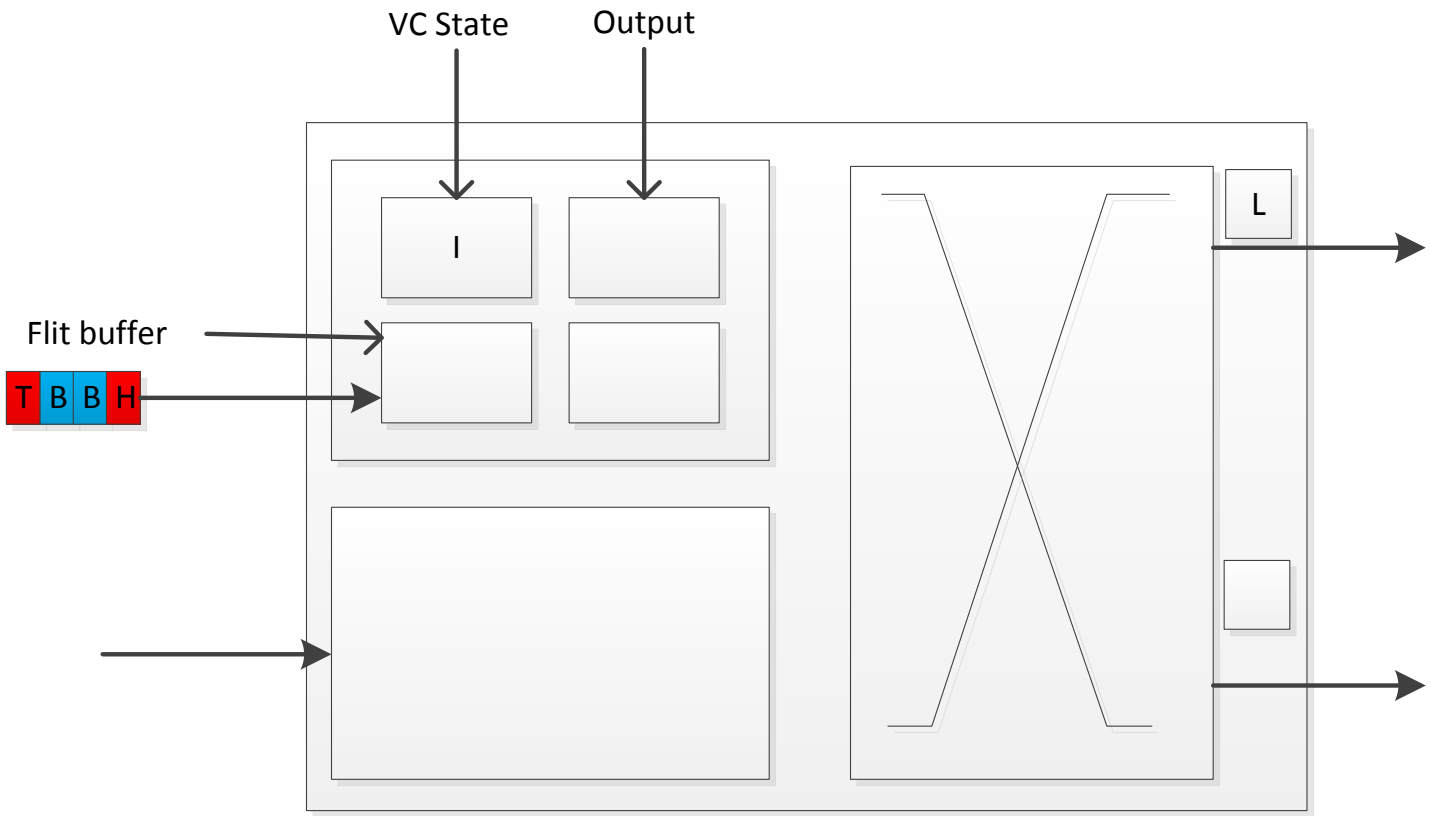


FIGURE – Le VC est dans l'état *idle* quand le paquet arrive

Commutation en mode *wormhole*

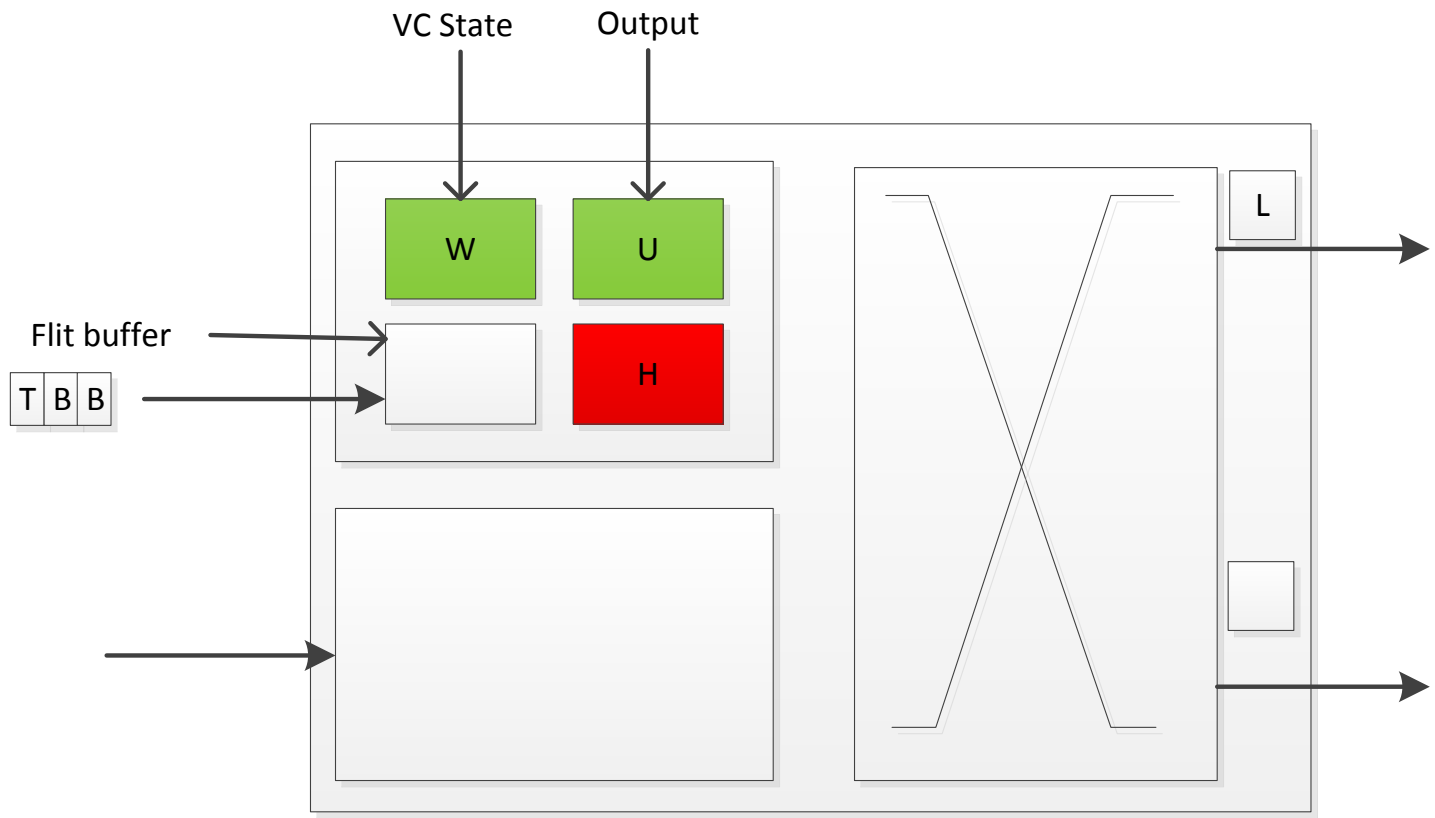


FIGURE – Le header indique une route vers le port de sortie UP. Celui-ci est alloué au port de sortie LOW. Le VC passe dans l'état WAITING, en attente de disponibilité de la ressource sur le port de sortie.

Commutation en mode *wormhole*

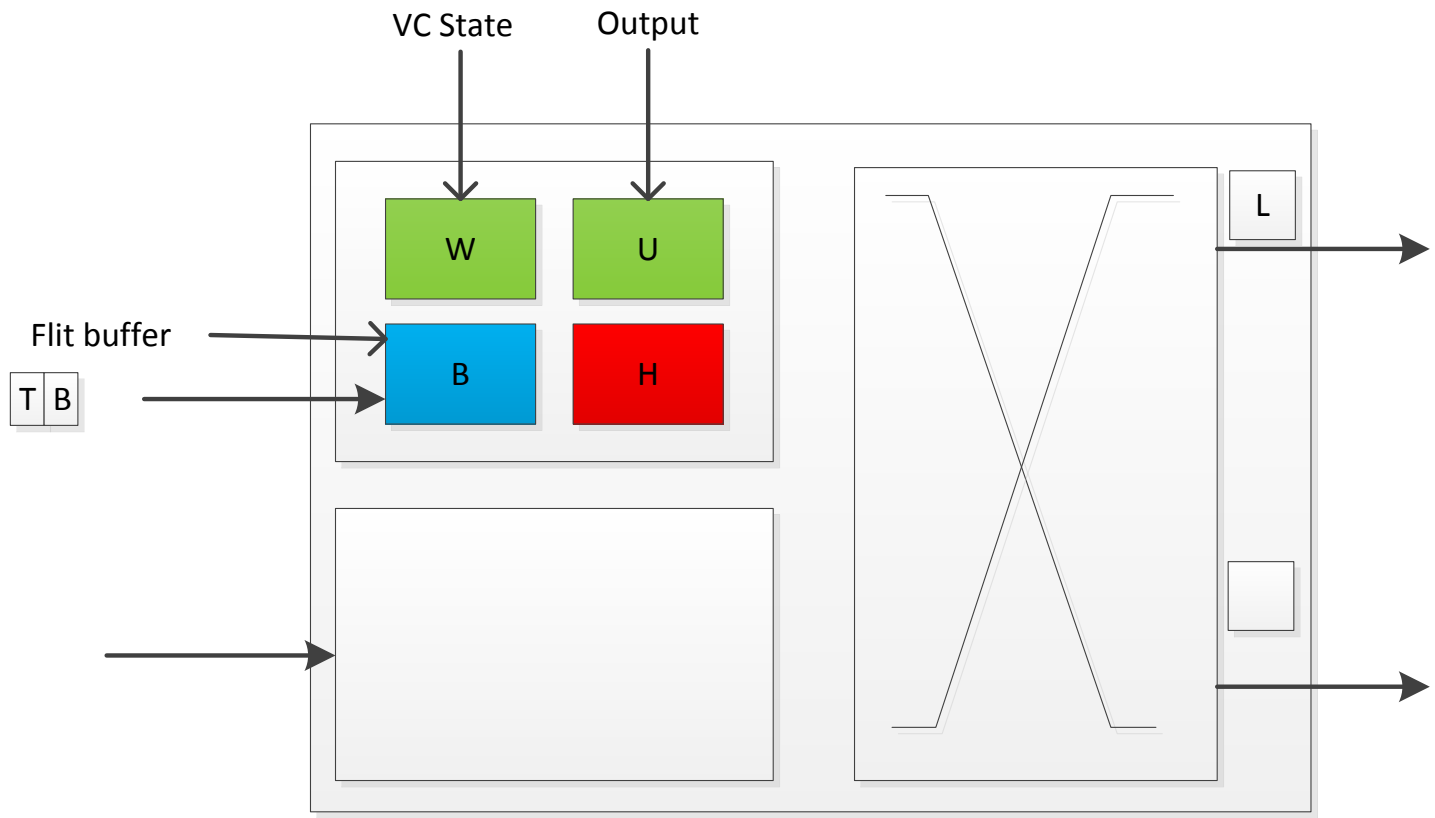


FIGURE – La profondeur du buffer d'entrée est de deux flits. Un premier Body flit peut encore être stocké dans le buffer. Puis la transmission est stoppée jusqu'à libération du VC sur le port de sortie UP.

Commutation en mode *wormhole*

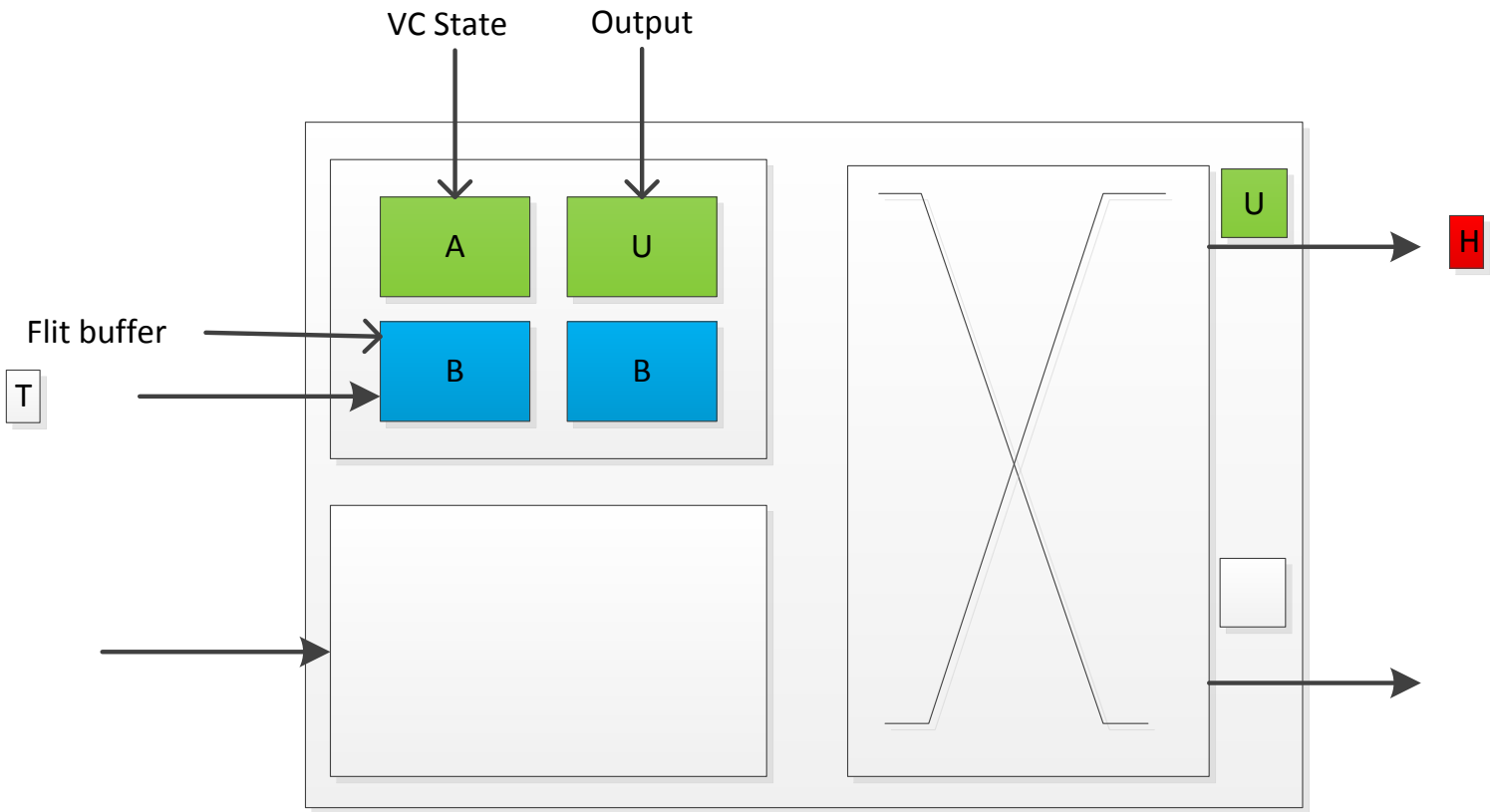


FIGURE – Une fois la ressource disponible, le VC passe dans l'état Actif. Le port de sortie UP est alloué au port d'entrée UP. Le header flit est transmis sur le canal de sortie libérant la ressource dans le buffer d'entrée.

Commutation en mode *wormhole*

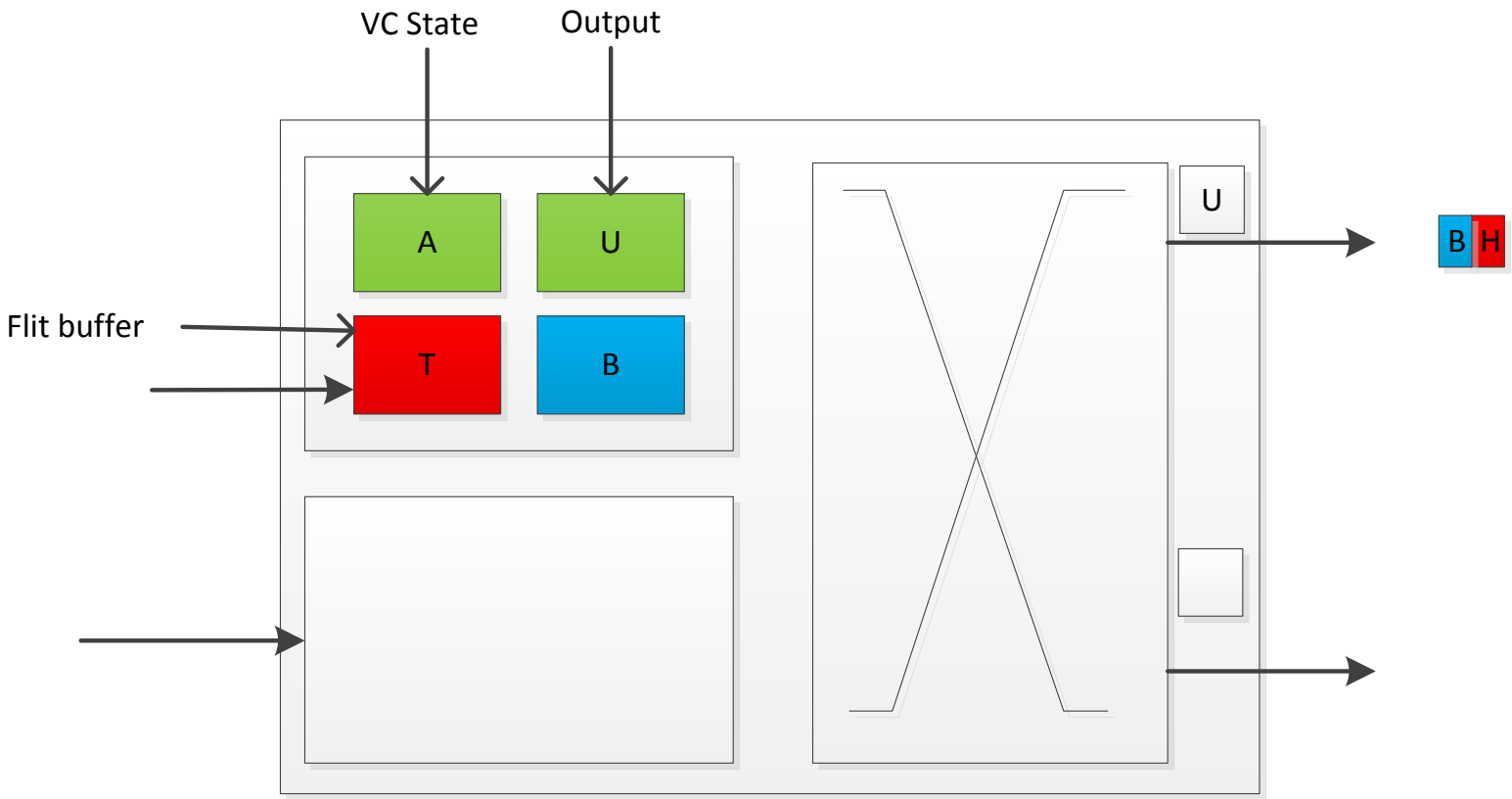


FIGURE – La transmission des flits du paquet progresse.

Commutation en mode *wormhole*

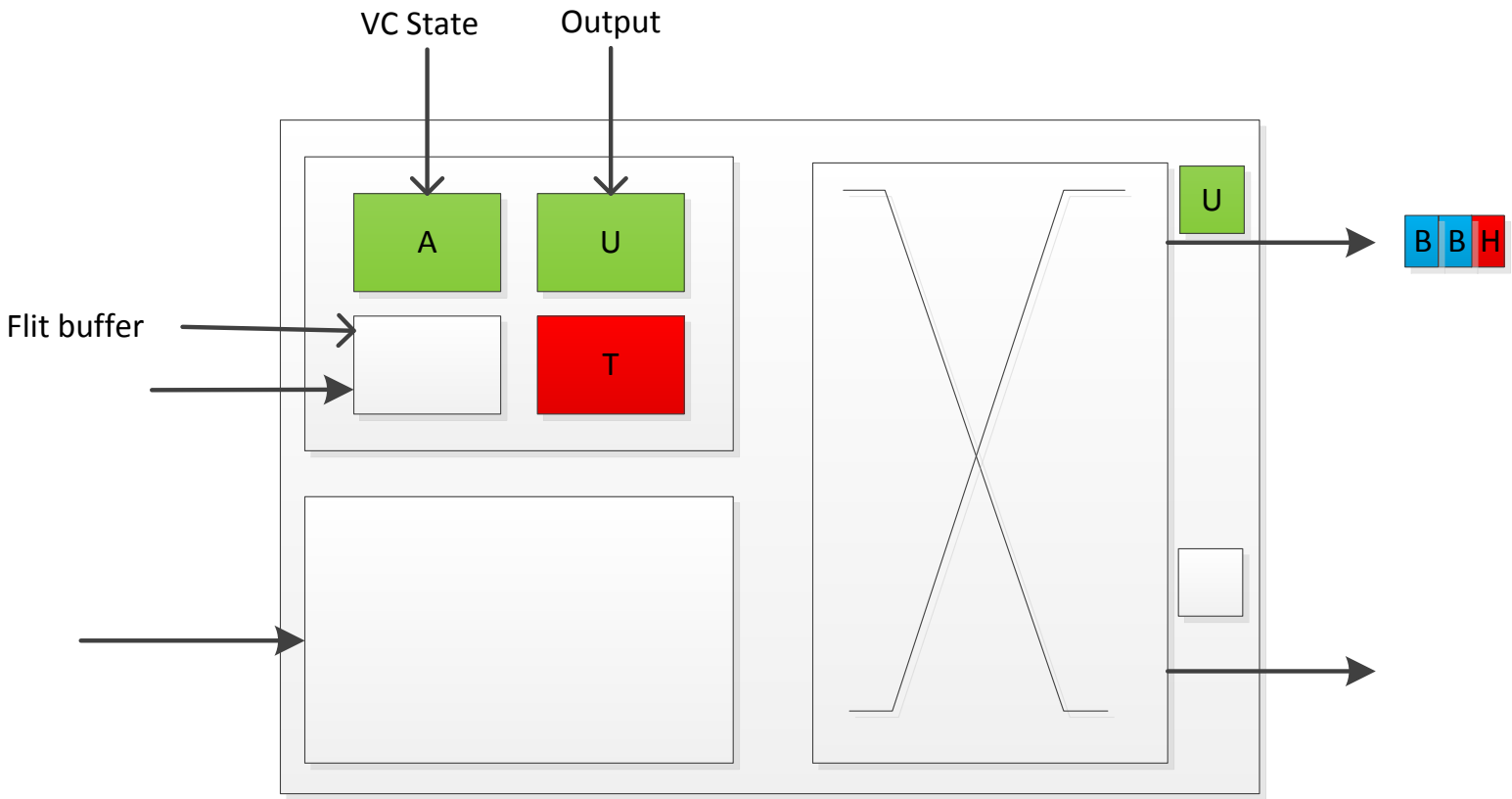


FIGURE – Jusqu'à transmission du *tail flit*.

Commutation en mode *wormhole*

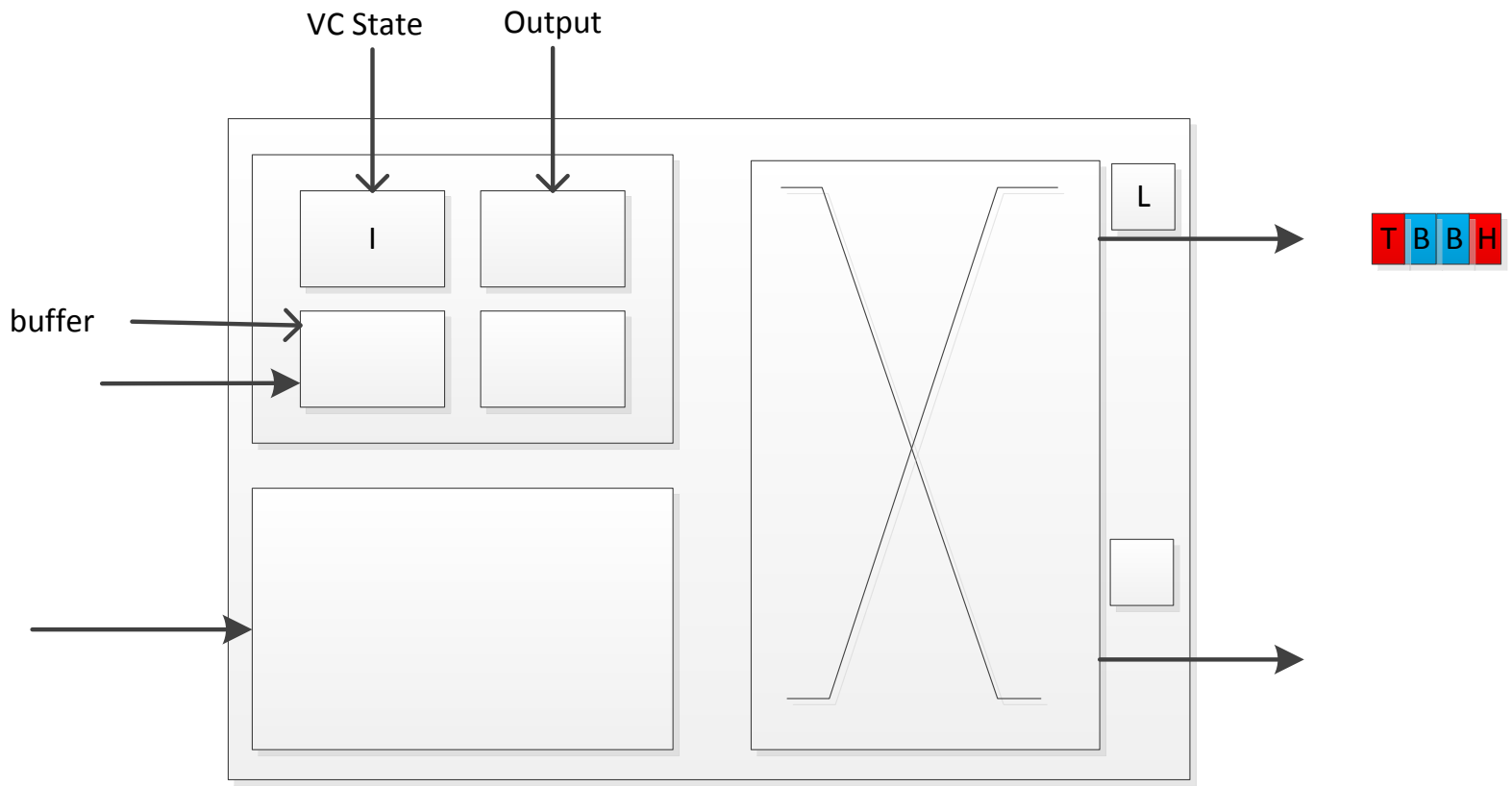


FIGURE – Qui libère les ressources, buffer d'entrée du VC et canal de transmission en sortie

Gestion des buffers

- L'utilisation de buffer nécéssite de remonter l'information de disponibilité de cette ressource aux noeud/routeur en amont, qui ne doivent transmettre des flits qu'en cas de buffer disponible sur le noeud/routeur aval.
- Il existe trois mécanismes de contrôle de flux : ACK/NACK, ON/OFF, Credit-based.

Credit-based flow control

- Le routeur amont tient, par canal virtuel, un décompte du nombre de *flit buffer* disponibles sur le routeur aval.
- A chaque envoi, le routeur amont décrémente son compteur de flit. Quand le compteur atteint 0, plus aucun flit n'est envoyé sur le canal.
- Dès que le routeur aval libère un flit-buffer, un crédit est renvoyé au routeur amont qui ré-incrémente son compteur de flit.

Credit-based flow control

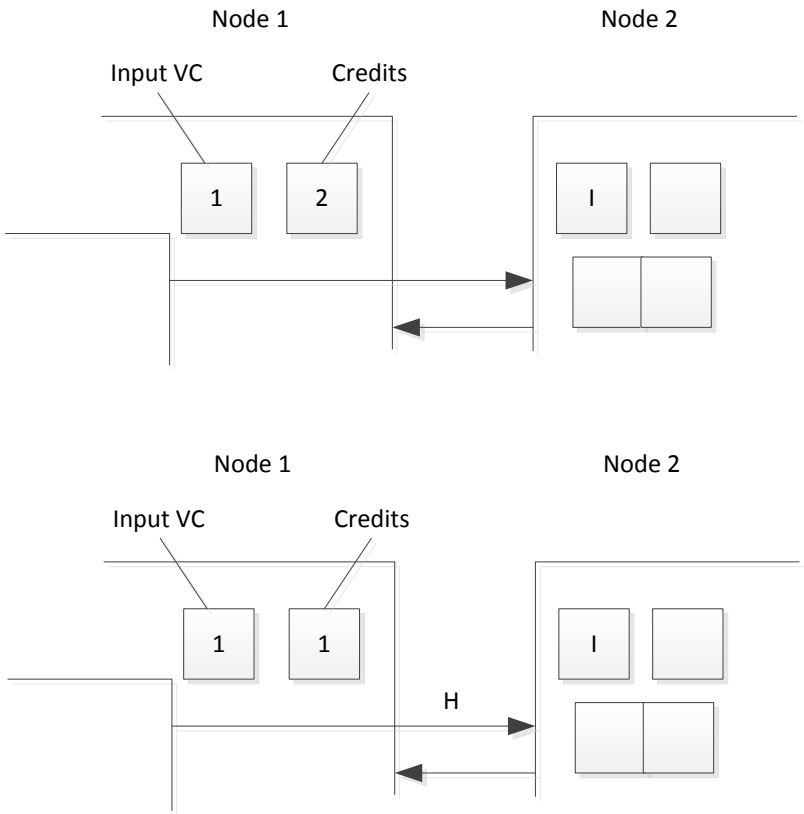


FIGURE – Le noeud/routeur 1 a 2 crédits d'émission sur son canal vers le noeud/routeur 2. La transmission du *header flit* décrémente le compteur de crédit. L'input VC du noeud 2 est en mode *idle*

Credit-based flow control

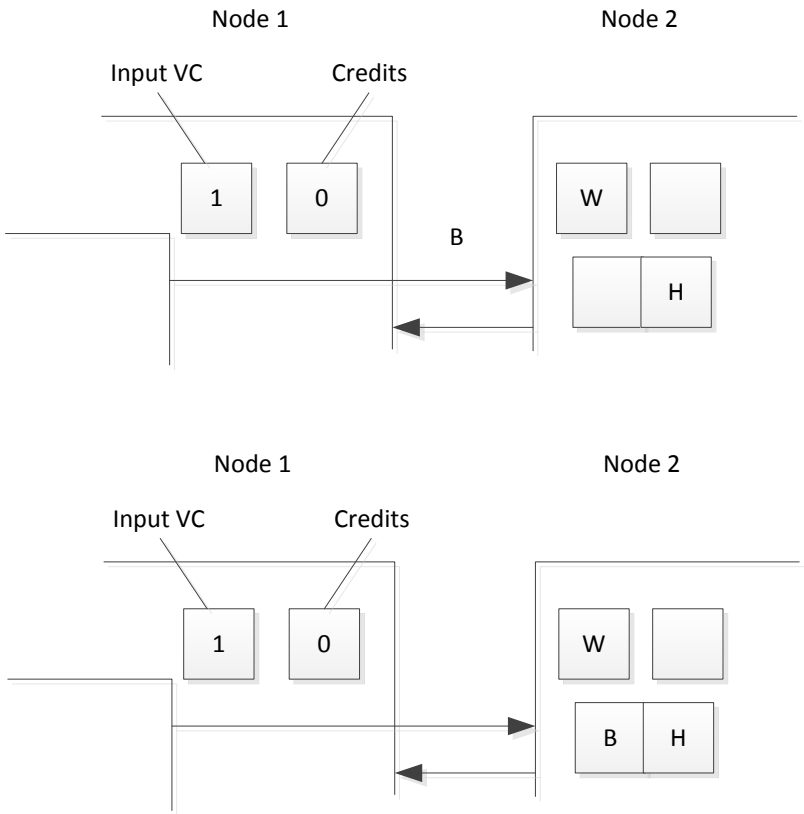


FIGURE – La transmission du *body flit* décrémente le compteur de crédit qui passe à 0. Le noeud étant dans l'état WAITING, les flits ne sont pas consommés et restent stockés dans le *2-flits buffer* du noeud 2.

Credit-based flow control

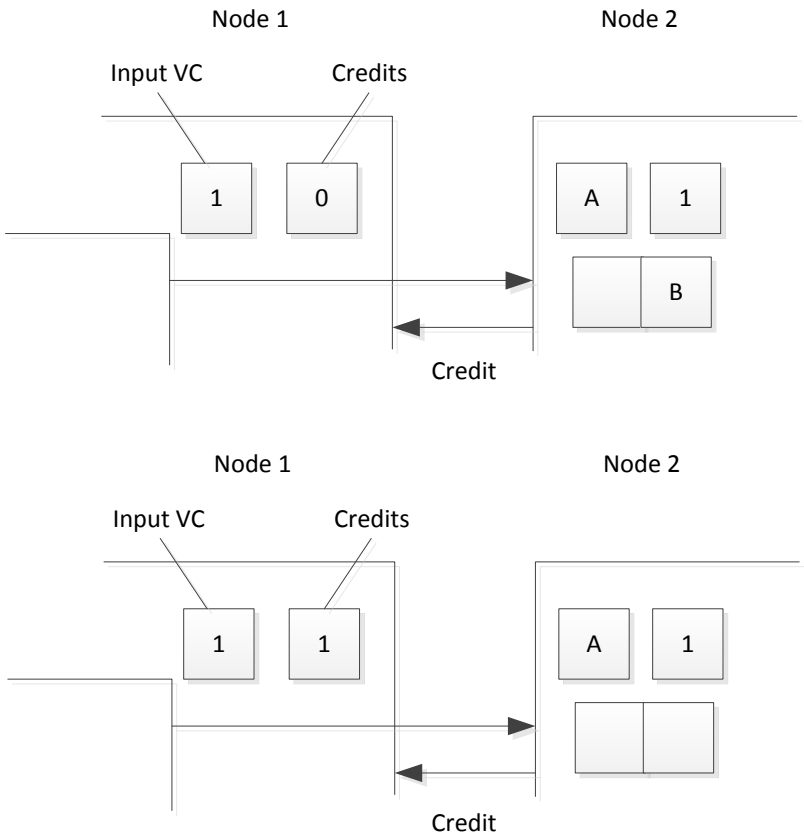


FIGURE – Le VC du noeud 2 passe à l'état Actif. Le *header flit* est consommé. Un *1-flit buffer* se libére. Un crédit est renvoyé au noeud 1, qui incrémente son compteur de crédit. Le *body flit* est consommé. Un second crédit est renvoyé au noeud 1.

Credit-based flow control

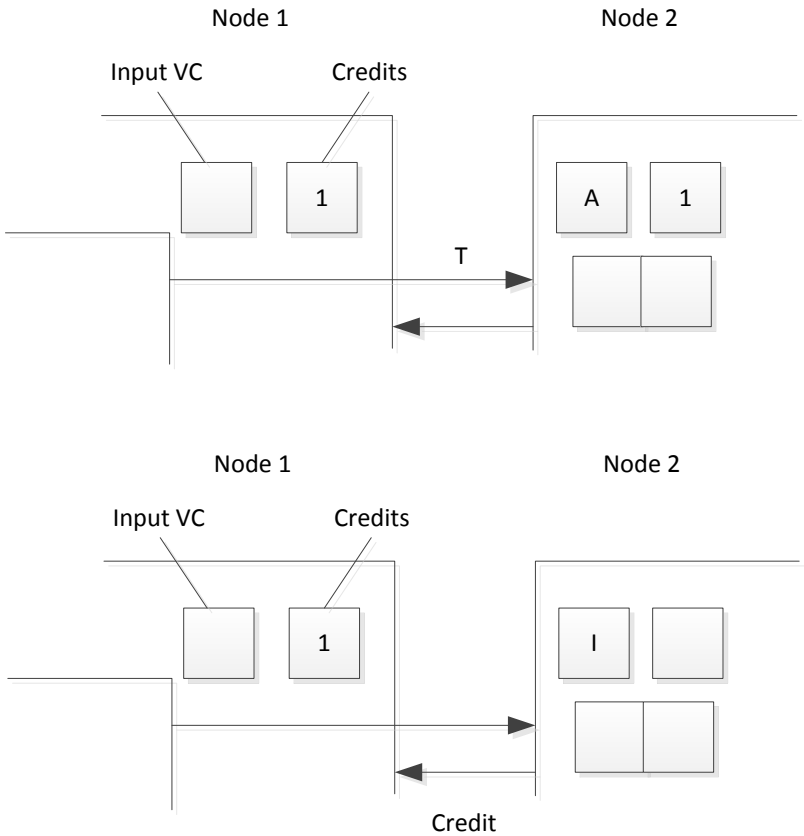
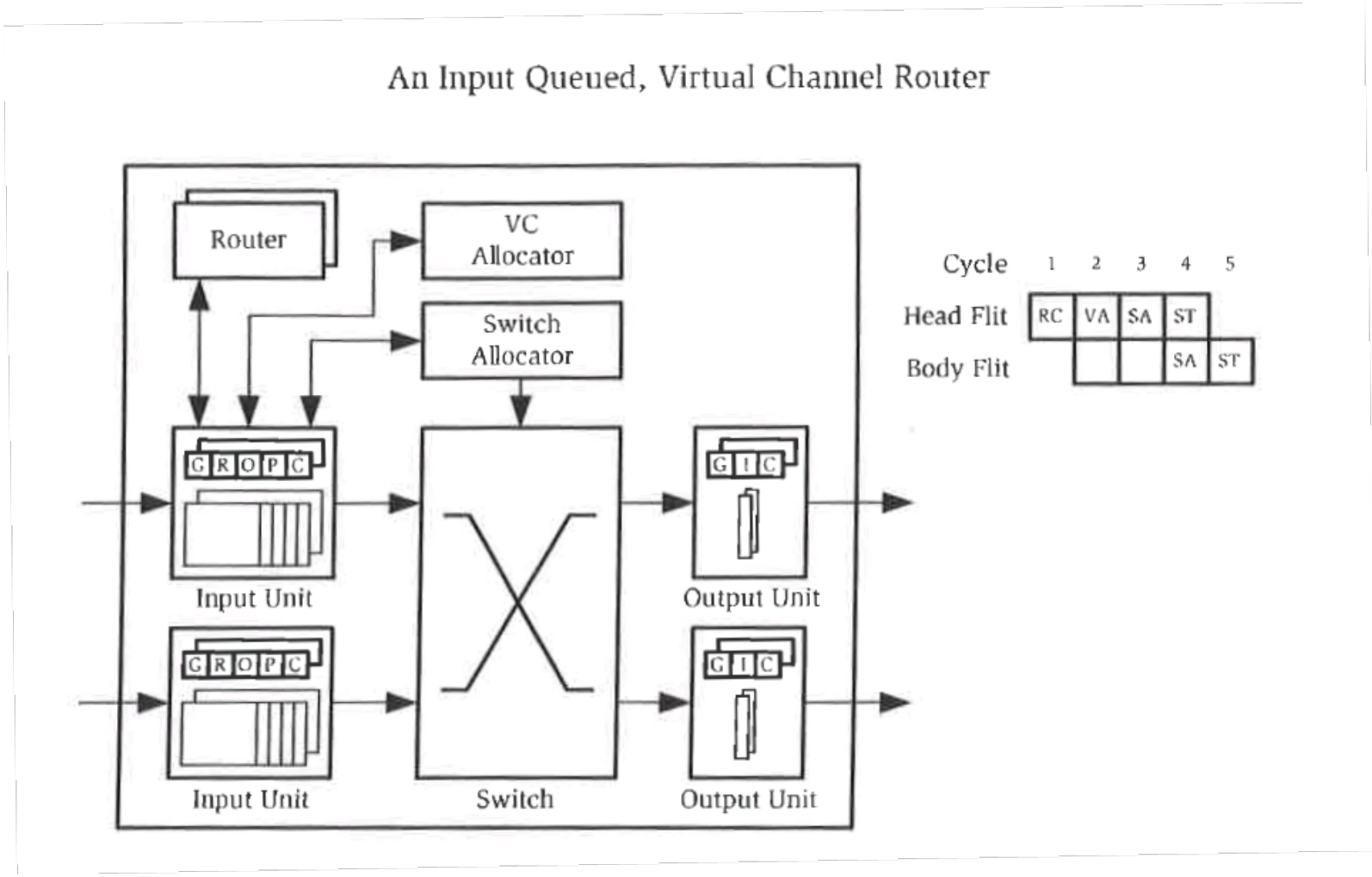


FIGURE – Les crédits sont suffisants pour émettre le *tail flit* vers le noeud 1. Le VC est libéré et repasse à l'état *idle*

Plan du cours

Architecture routeur avec buffers d'entrée et VC 1/3



Architecture routeur avec buffers d'entrée et VC 2/3

- Etape 1 : Routage. Le calcul du routage commence sur réception du *header flit* sur un Input Port. Le paquet est envoyé vers un VC particulier. L'état du VC passe dans létat Routing. La décision de routage renvoit l'Output port vers lequel sera commuté le paquet. L'information est tracée dans les registres d'état de l'Input Port pour ce VC particulier. Le flit est alors stocké dans le buffer du VC concerné.
- Etape 2 : Choix du VC. Le résultat du routage fait passer l'Ouput VC sélectionné dans l'état Actif. Le résultat de l'allocation indique quel VC de sortie est alloué au paquet. L'information des crédits de l'output est reportée dans les registres d'état de l'Input VC.
- Etape 3 : Commutation. Tous les Input VC dans l'état A pointant vers l'Output Port concerné et contenant des flits sont en concurrence pour l'accès au canal physique de sortie. Si les crédits sont supérieurs à 0, le header flit est transmis. le compteur de crédit de l'Output Port est décrémenté. Un crédit est renvoyé à l'émetteur en amont du routeur.

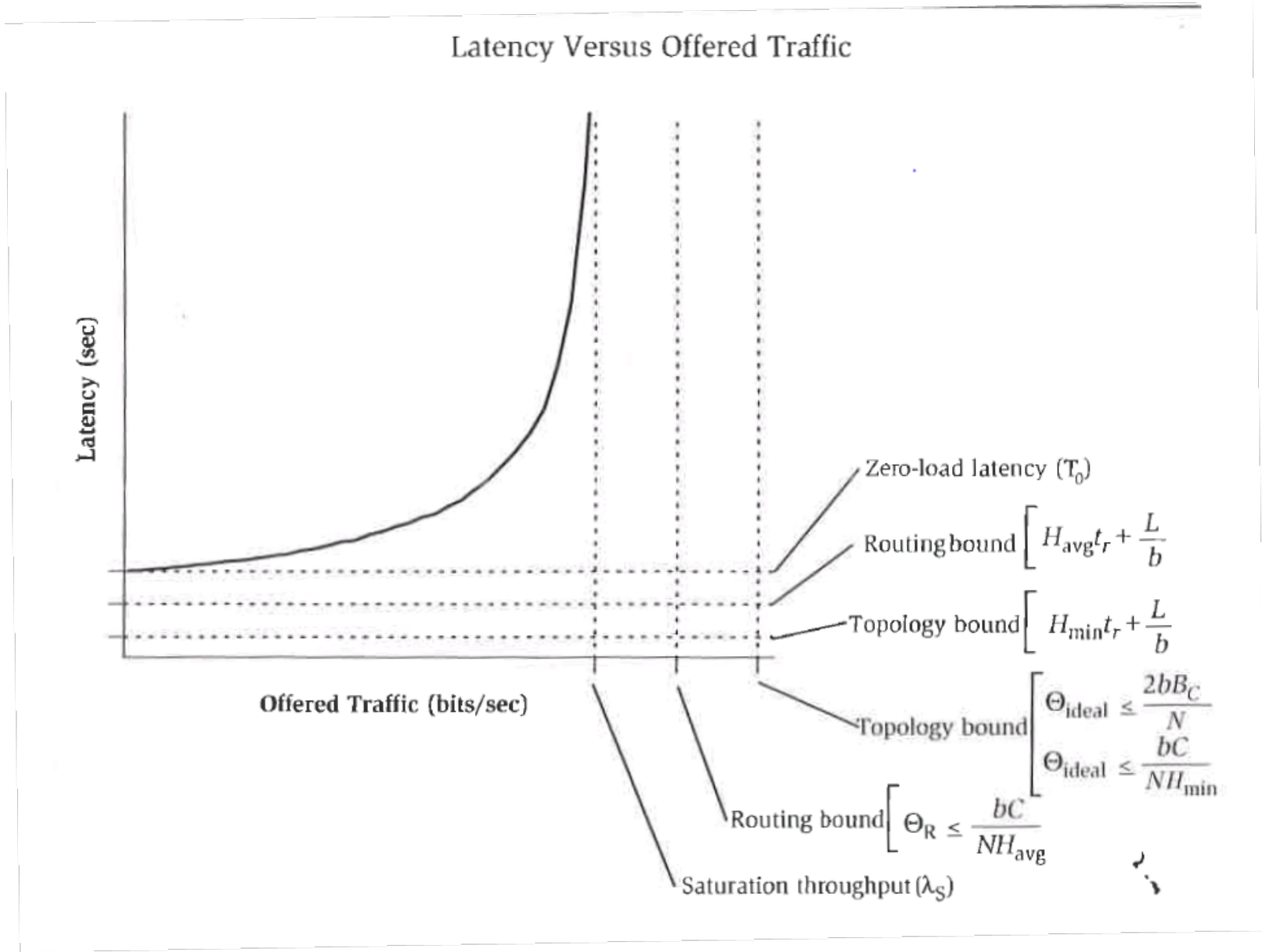
Architecture routeur avec buffers d'entrée et VC 3/3

- Etape 4 : Transmission. Le *head flit* traverse le switch, puis le port de sortie et le canal physique, jusqu'à être stocké dans le buffer d'entrée du routeur suivant. Le *body flit* suivant passe directement à l'étape 3, commutation puis 4, transmission si des crédits sont disponibles.
- Etape finale : Libération des ressources. La réception du *tail flit* réinitialise les registres d'état du VC d'entrée et l'allocation du VC de sortie.

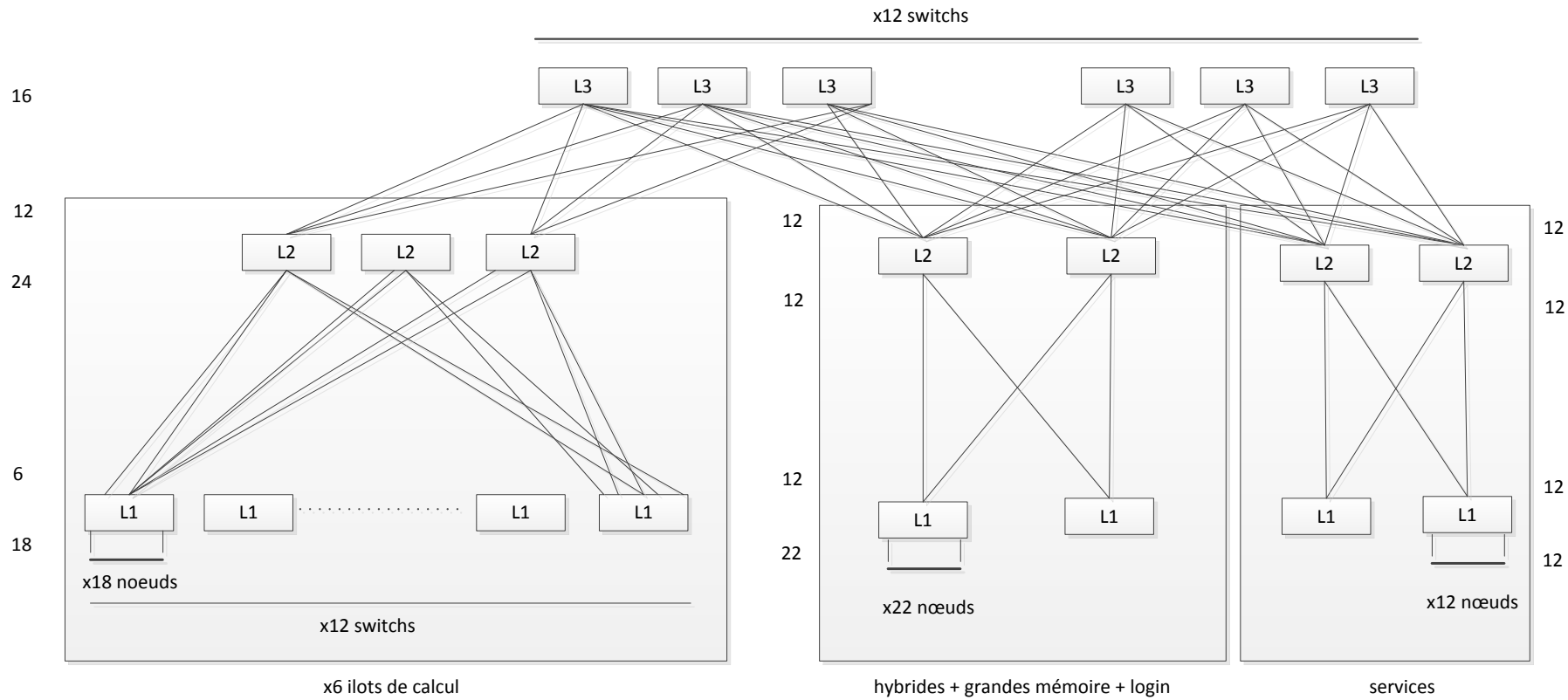
Plan du cours

Plan du cours

Choix d'architecture et performance atteignable



Exemple d'architecture



Merci de votre attention !

Questions ?