クラスタ

慶應義塾大学理工学部
天野英晴
hunga@am. ics. keio. ac. jp

NORA/NORMA

- 共有メモリを持たない
- 交信はメッセージのやりとりで行う
  - MPIが主に使われる
- 接続はGigabit EthernetやInfiniband
- 最近は多出力のスイッチを用いる
  - ハイラディックスネットワーク

データセンターなどで要求レベル並列性を処理

クラスタコンピューティング

Beowulf クラスタ

1994年NASA T.Sterling

- 安価で簡単に大規模並列計算環境を作ろう
  - コモディティのPCを利用
  - コモディティのネットワーク（Ethernet）を利用
  - コモディティのソフトウェア（Linux)を利用
  - PVMやMPI（来週紹介）などのメッセージパッシング型ライブラリでプログラム

→現在のClusterの元祖となった

現在のClusterは、InfinibandなどのSANを使うものも多いが、基本的に上記の原則を守っている

Infiniband

- System (Storage) Area Network (SAN)用.
- 8b/10b コードを利用.
- 様々な接続形態に対応.
- マルチキャスト可能.

	SDR	DDR	QDR
1X	2Gbit/s	4Gbit/s	8Gbit/s
4X	8Gbit/s	16Gbit/s	32Gbit/s
12X	24Gbit/s	48Gbit/s	96Gbit/s

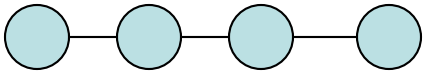

RHiNET-2 cluster

クラスタの接続網

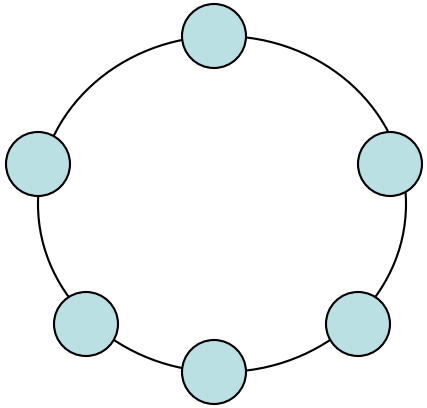
- 直接網（direct/distributed)
  - ノード同士を直接つなぐ
  - k-ary n-cubeが主に利用される
- 間接網（indirect/centralized)
  - スイッチを経由してつなぐ
  - Fat Tree, DragonflyなどHigh-radix系が流行っている
- パケットをどのように転送するかが重要

1. 直接網

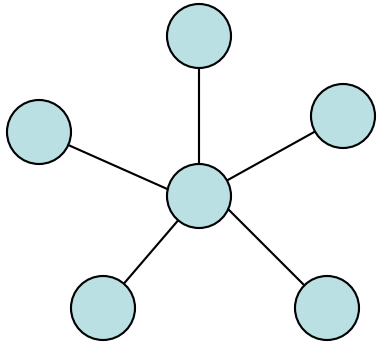
まずは基本的なもの



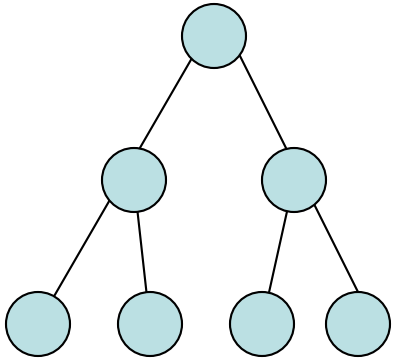
Linear



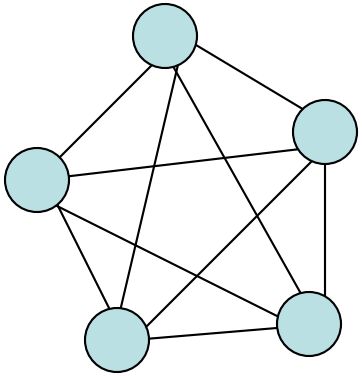
Ring



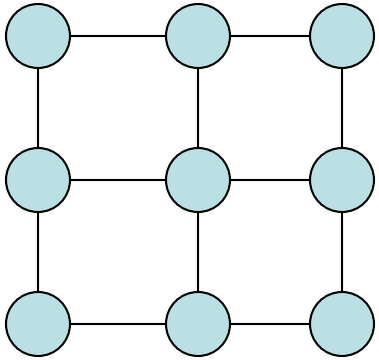
Central concentration



Tree



Complete connection

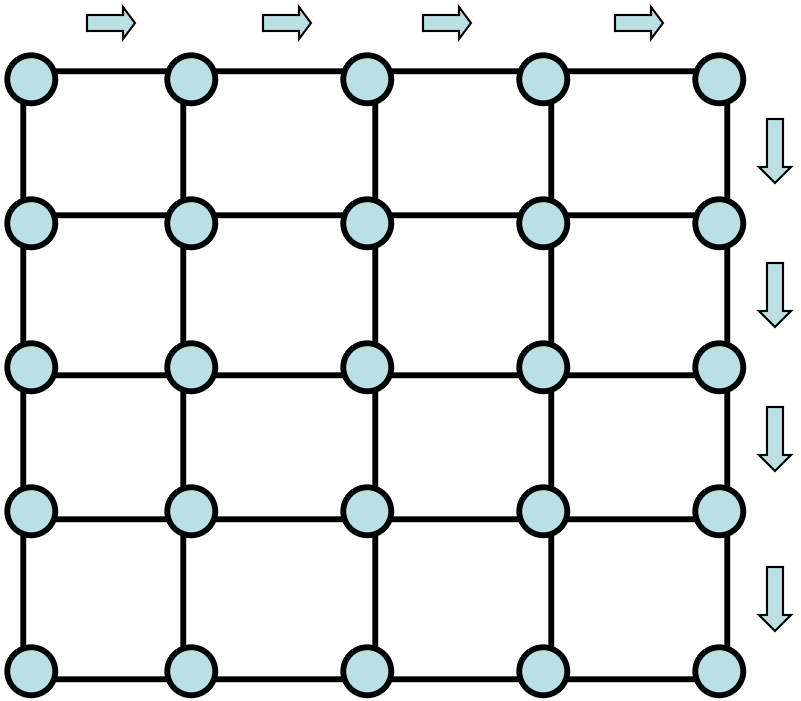


Mesh

直接網の評価基準（D and d）

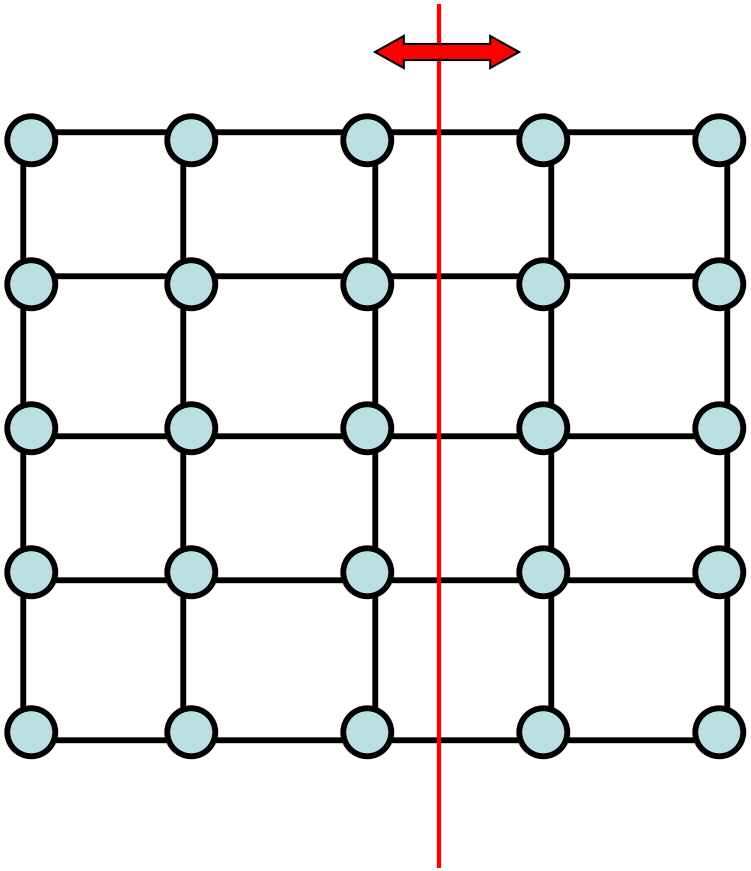
- 直径（Diameter）：D
  - ネットワーク中の最も遠い2ノード間の最短ホップ数
- 次数（degree）: d
  - ノードに繋がるリンクの最大数
- ASPL (Average Shortest Parth Length)
  - 平均距離

直径の例



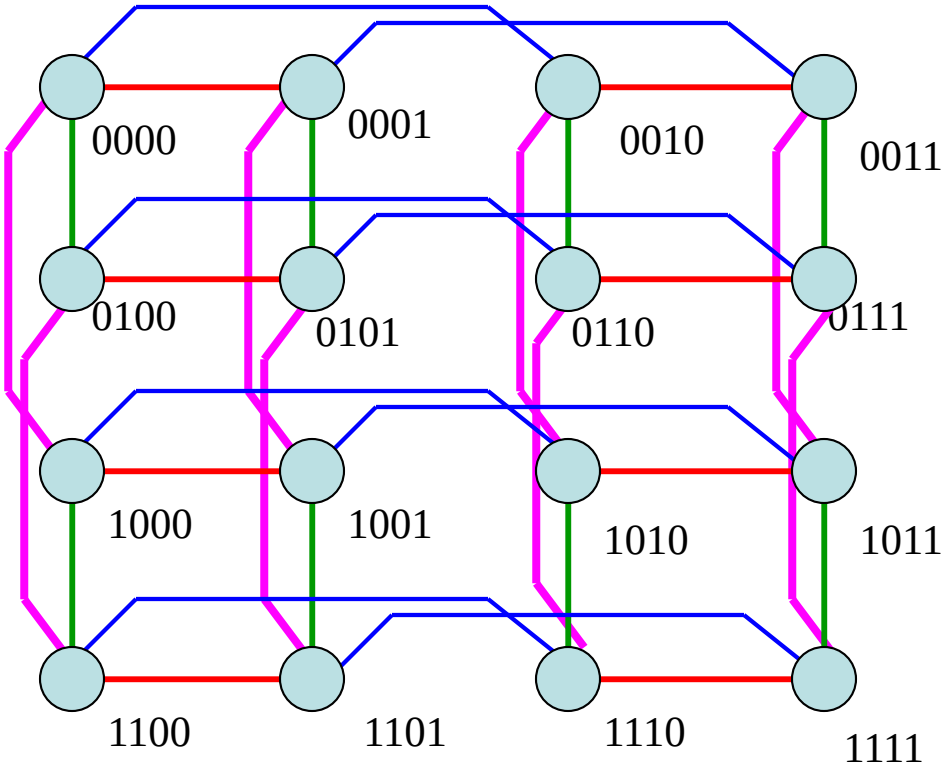
２（ｎ－１）

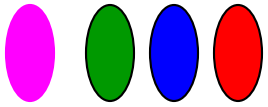
二分バンド幅
bi-section bandwidth



ネットワークを等分した際の
交信量→もっとも小さいものを
取る

Hypercube

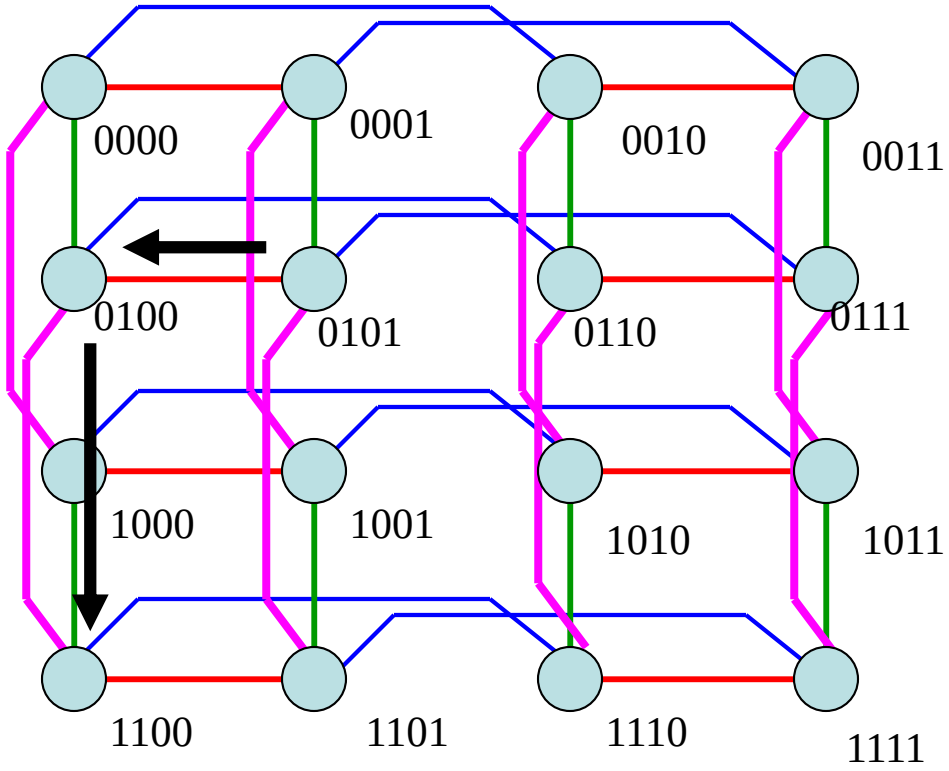

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

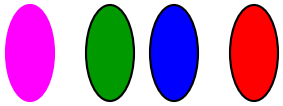


Routing on hypercube

0101→1100

Different bits

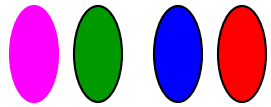


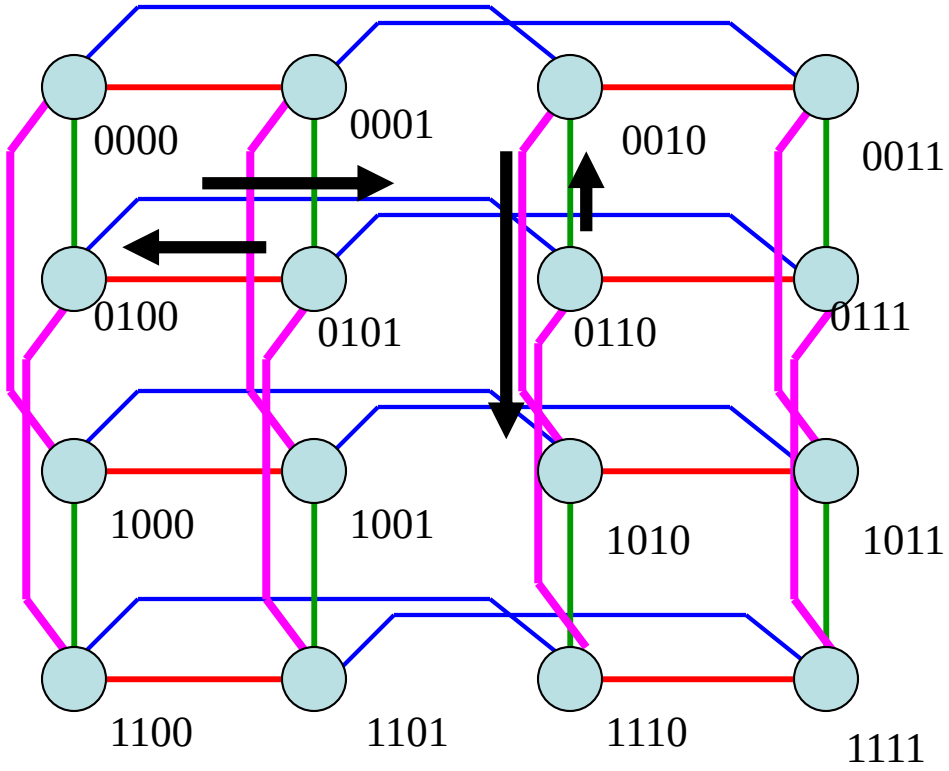


hypercubeの直径

0101→1010

All bits are different
→ the largest distance

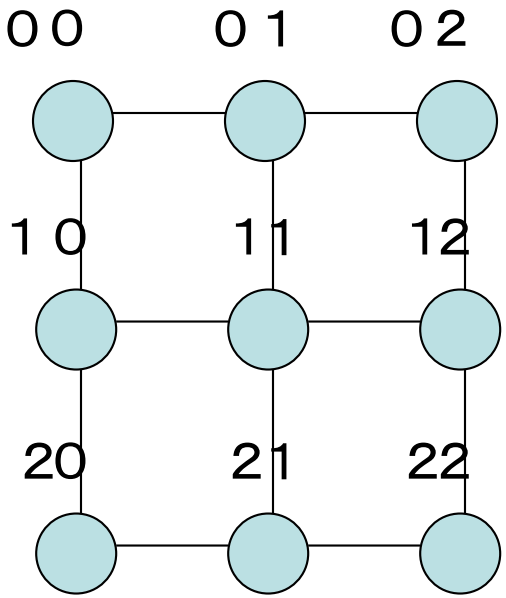




k-ary n-cube

- メッシュ、トーラスの一般化
- n桁のK進数を各ノードに割り当てる。
- それぞれの次元（桁）方向にリンクを順に設ける。
- 巡回するリンク(n-1→0)を持てば トーラス、そうでなければメッシュ
- 2次元、3次元メッシュ、トーラス、リング、直線状、hypercubeを含むファミリー

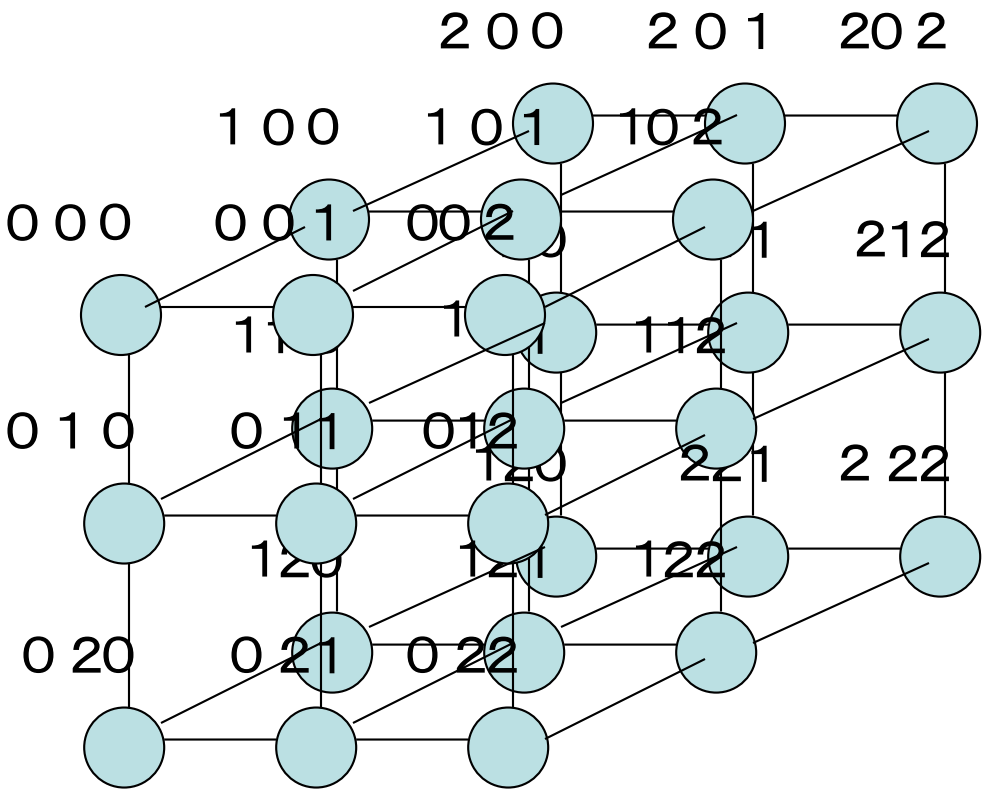
k-ary n-cube



3-ary 1-cube

3-ary 2-cube

k-ary n-cube

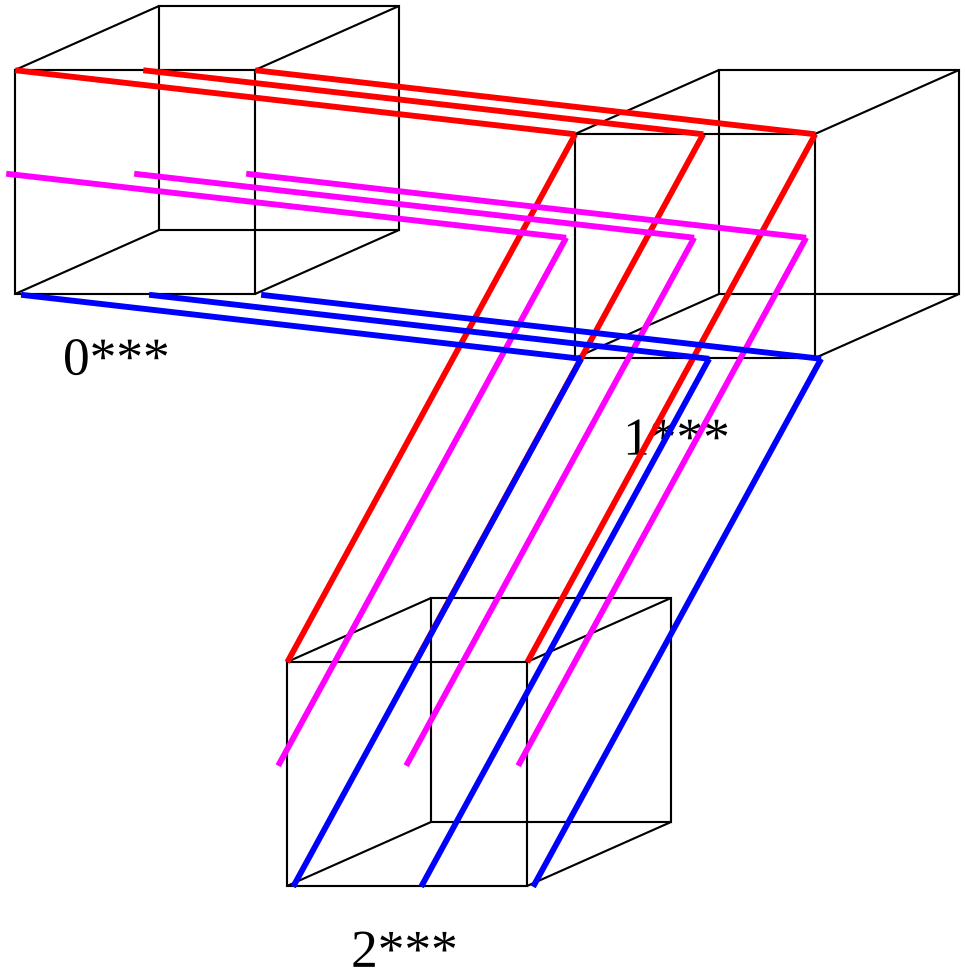


3-ary 1-cube

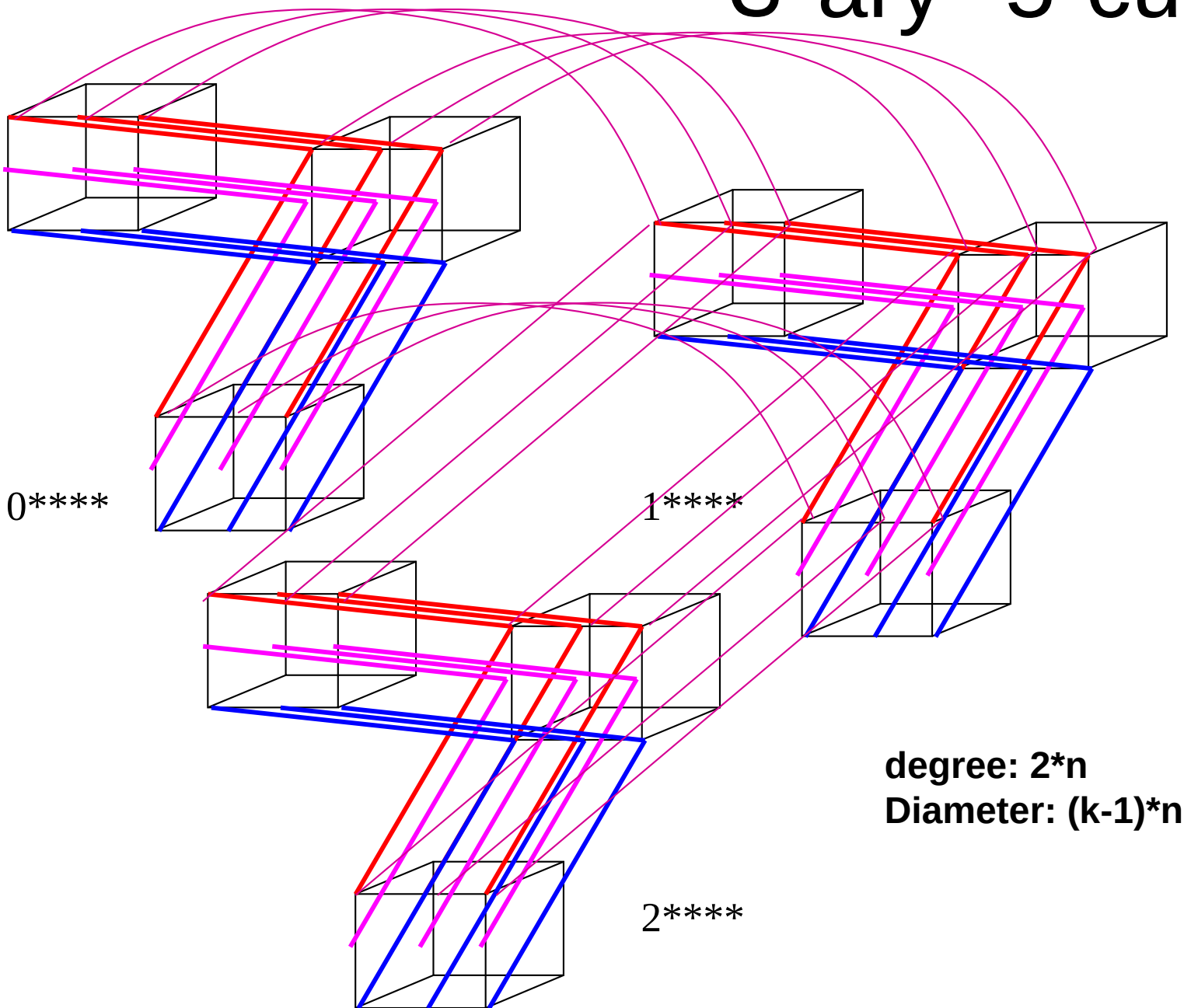
3-ary 2-cube

3-ary 3-cube

3-ary 4-cube



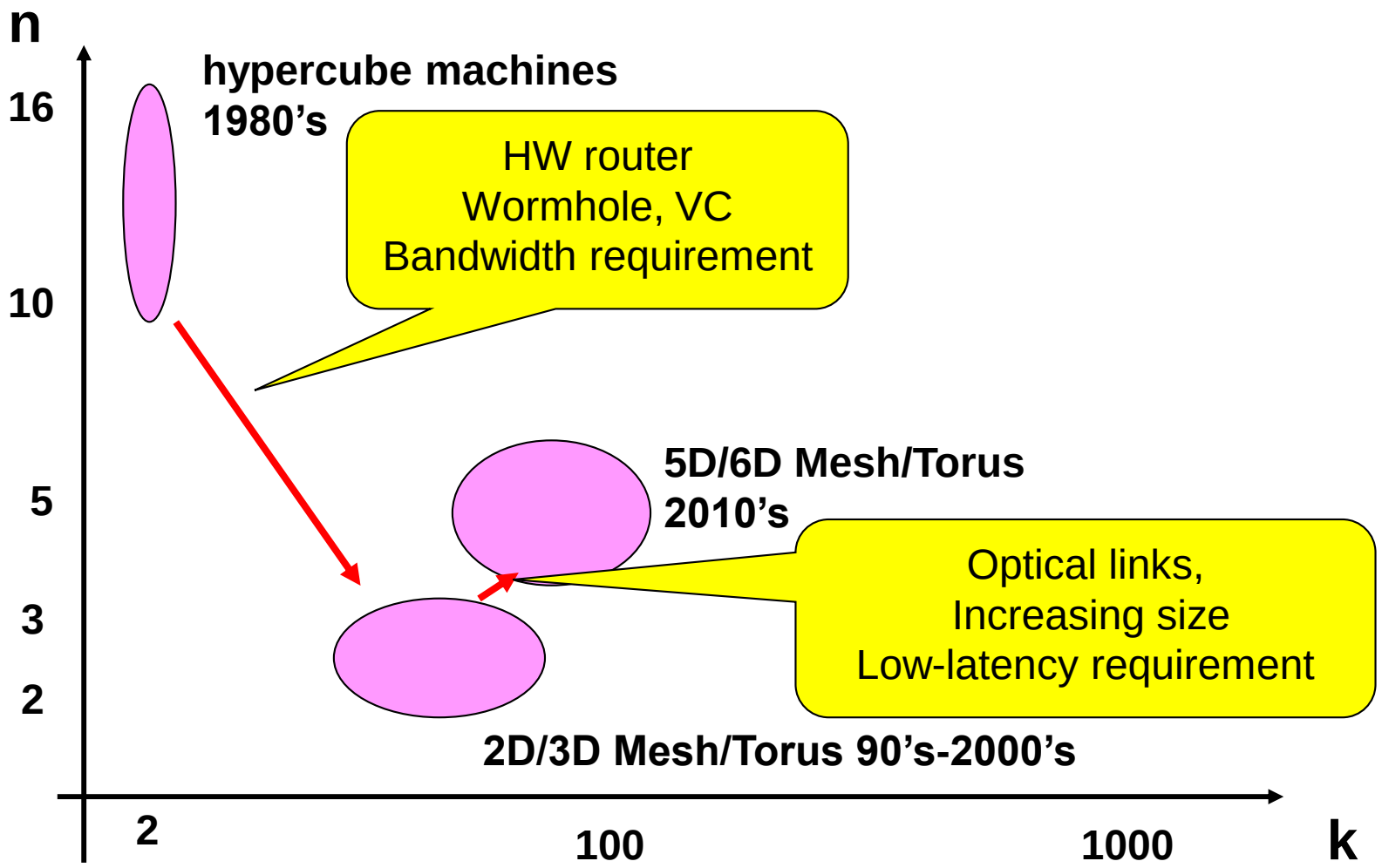
3-ary 5-cube



degree: 2*n
Diameter: (k-1)*n


6-次元 Torus
Tofu

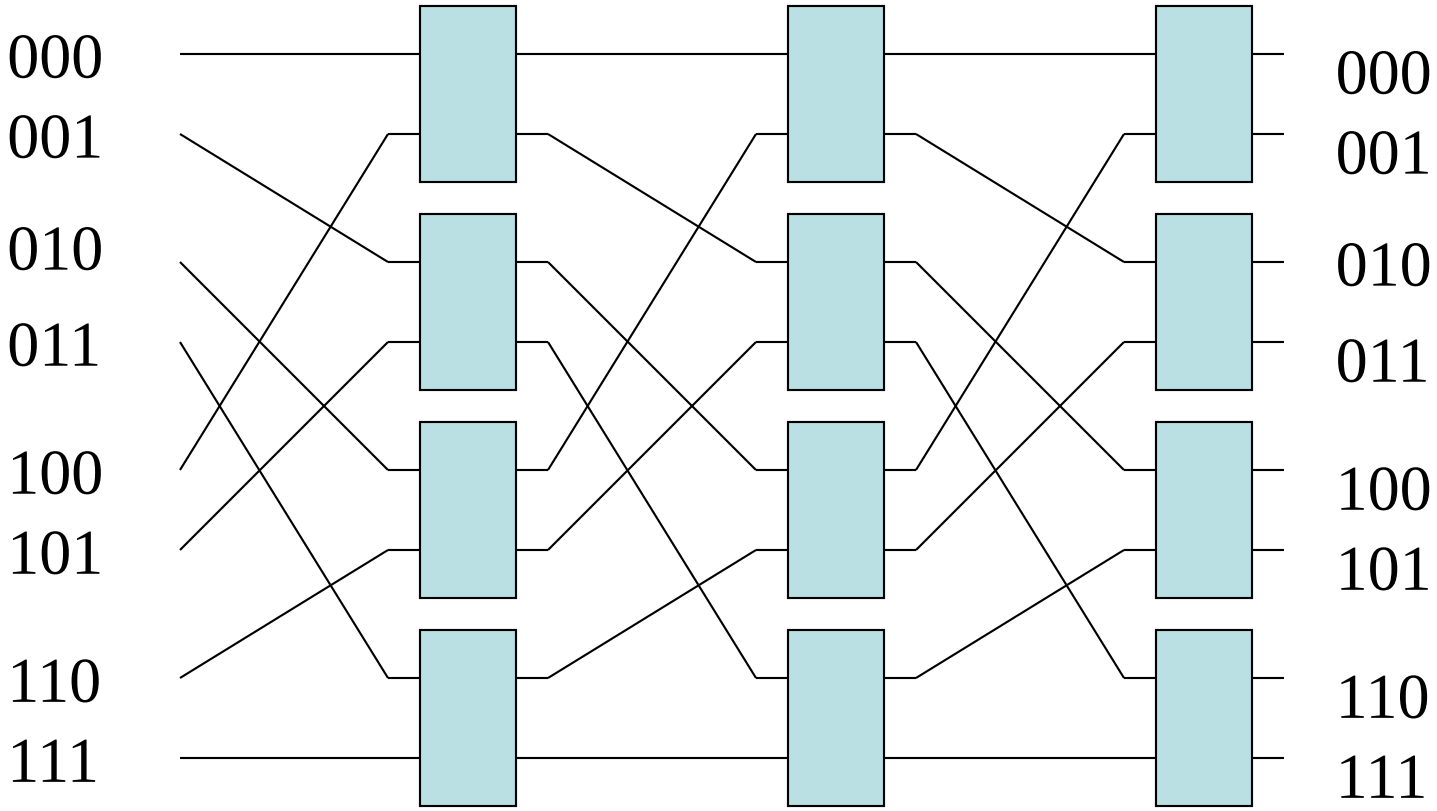
トレンドの移り変わり


n
16
10
5
3
2
hypercube machines
1980's
HW router
Wormhole, VC
Bandwidth requirement
5D/6D Mesh/Torus
2010's
Optical links,
Increasing size
Low-latency requirement
2D/3D Mesh/Torus 90's-2000's
2
100
1000
k

2. 間接網

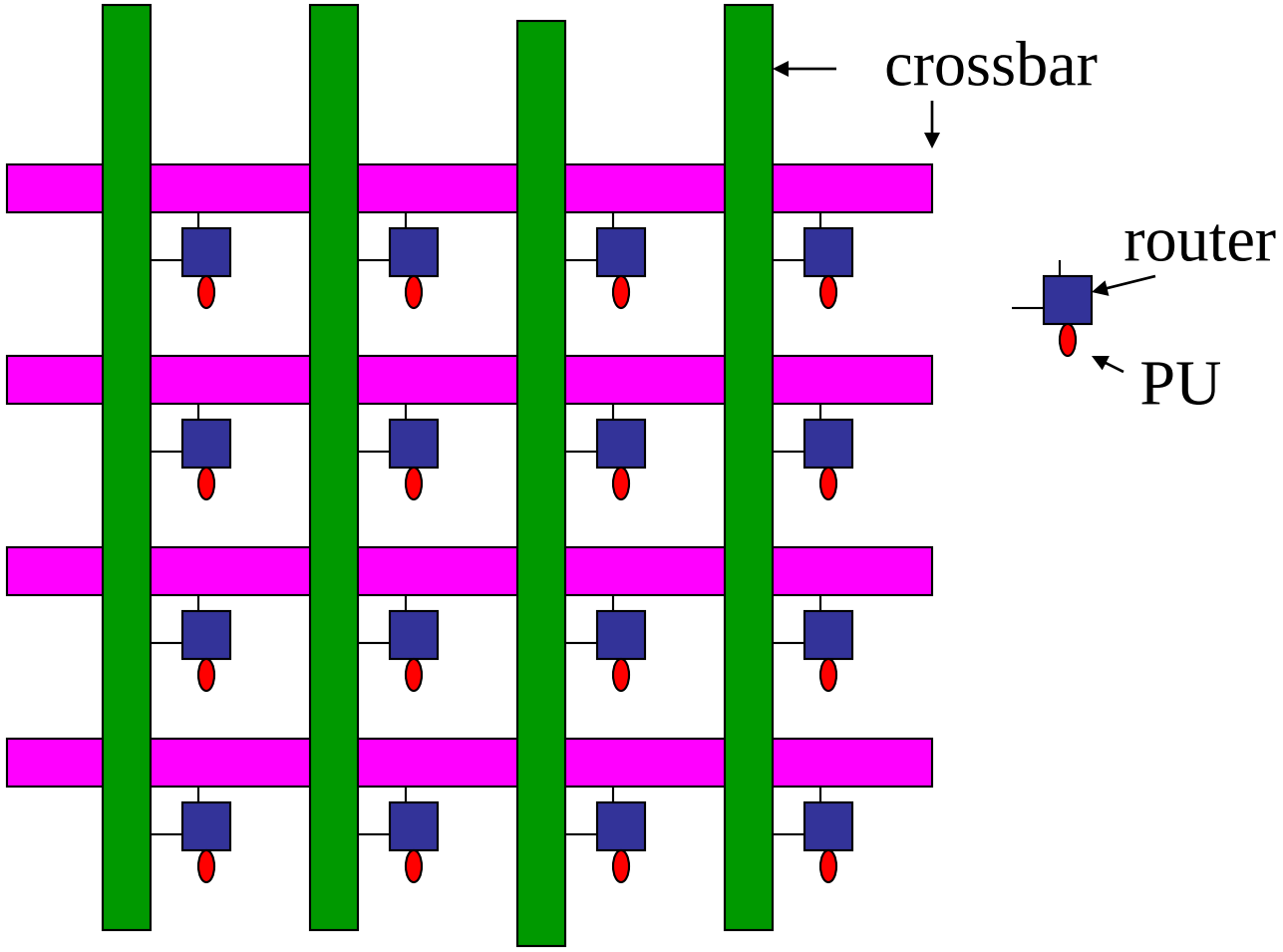
- 等距離間接網
  - Multistage Interconnection Network (MIN)
  - 最近はButterflyと呼ばれる
  - 局所性が生かせないので大規模なシステムに向かない
- 不等距離間接網
  - base-m n-cube
  - Fat Tree
  - Dragon FlyなどHigh Radixネットワーク

Omega網



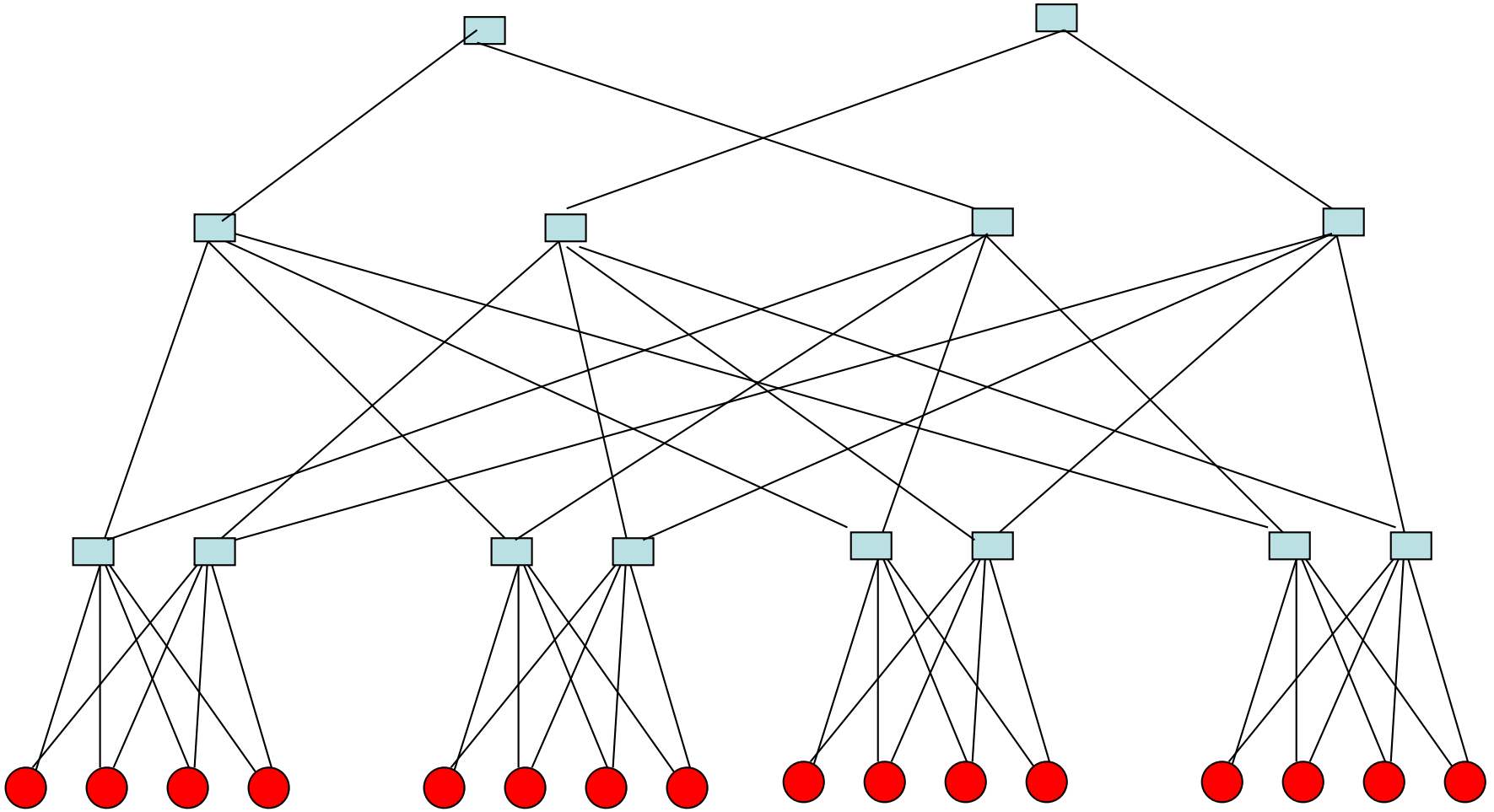
2x2のスイッチ素子を利用、Perfect Shuffleでステージ間を接続

base-m n-cube (Hyper crossbar)



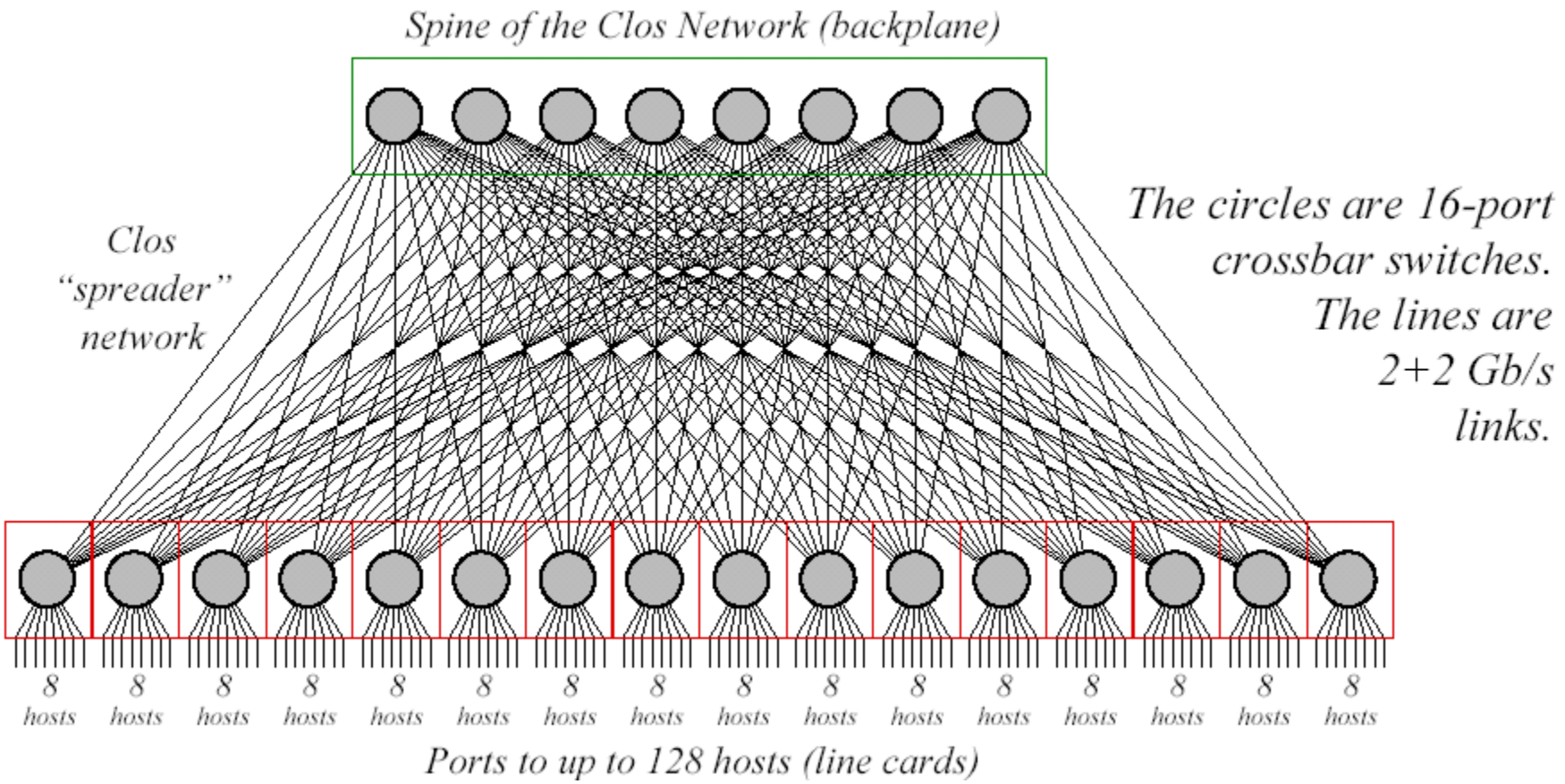
Used in Toshiba's Prodigy and Hitachi's SR8000

Fat Tree



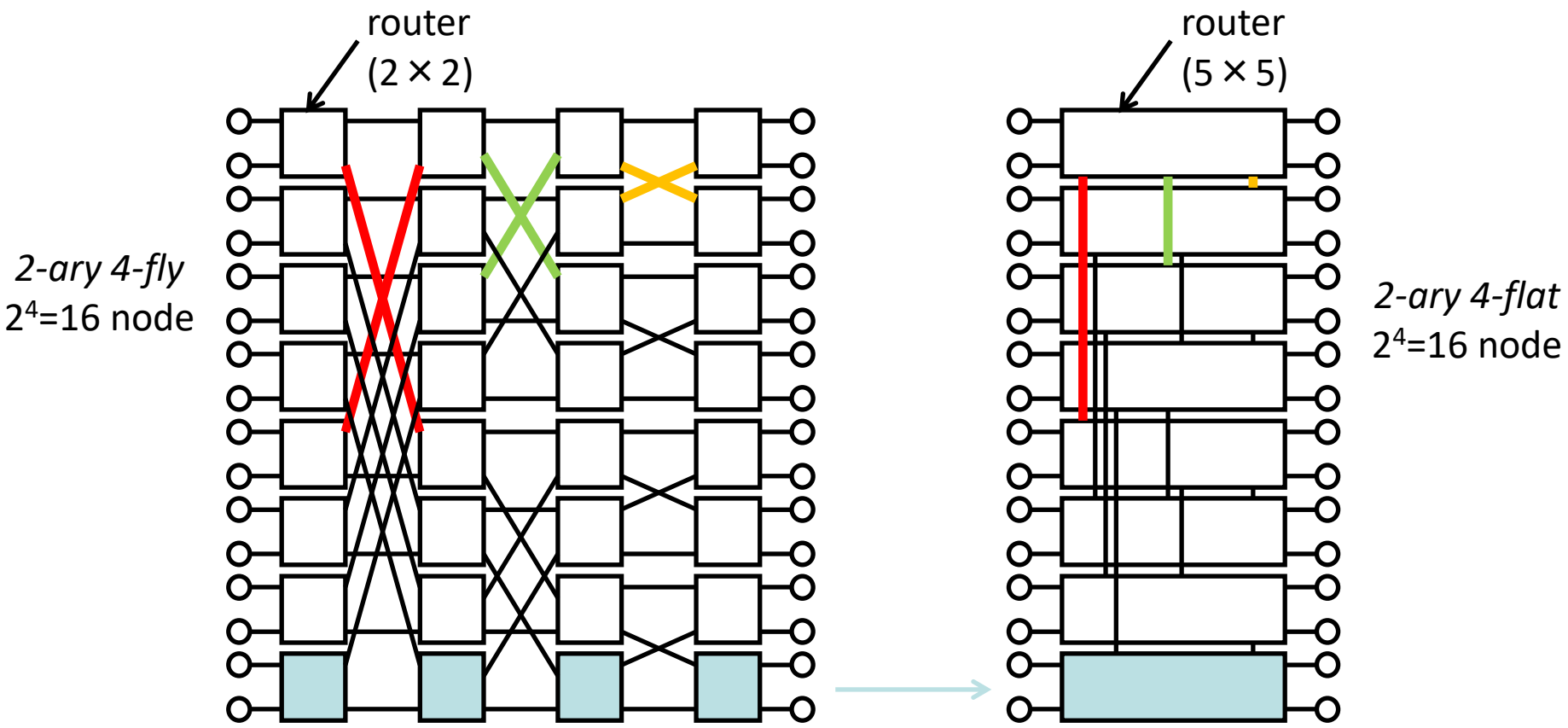
Myrinet-Clos はClosではなく Fat-tree

Myrinet-Clos（1/2)



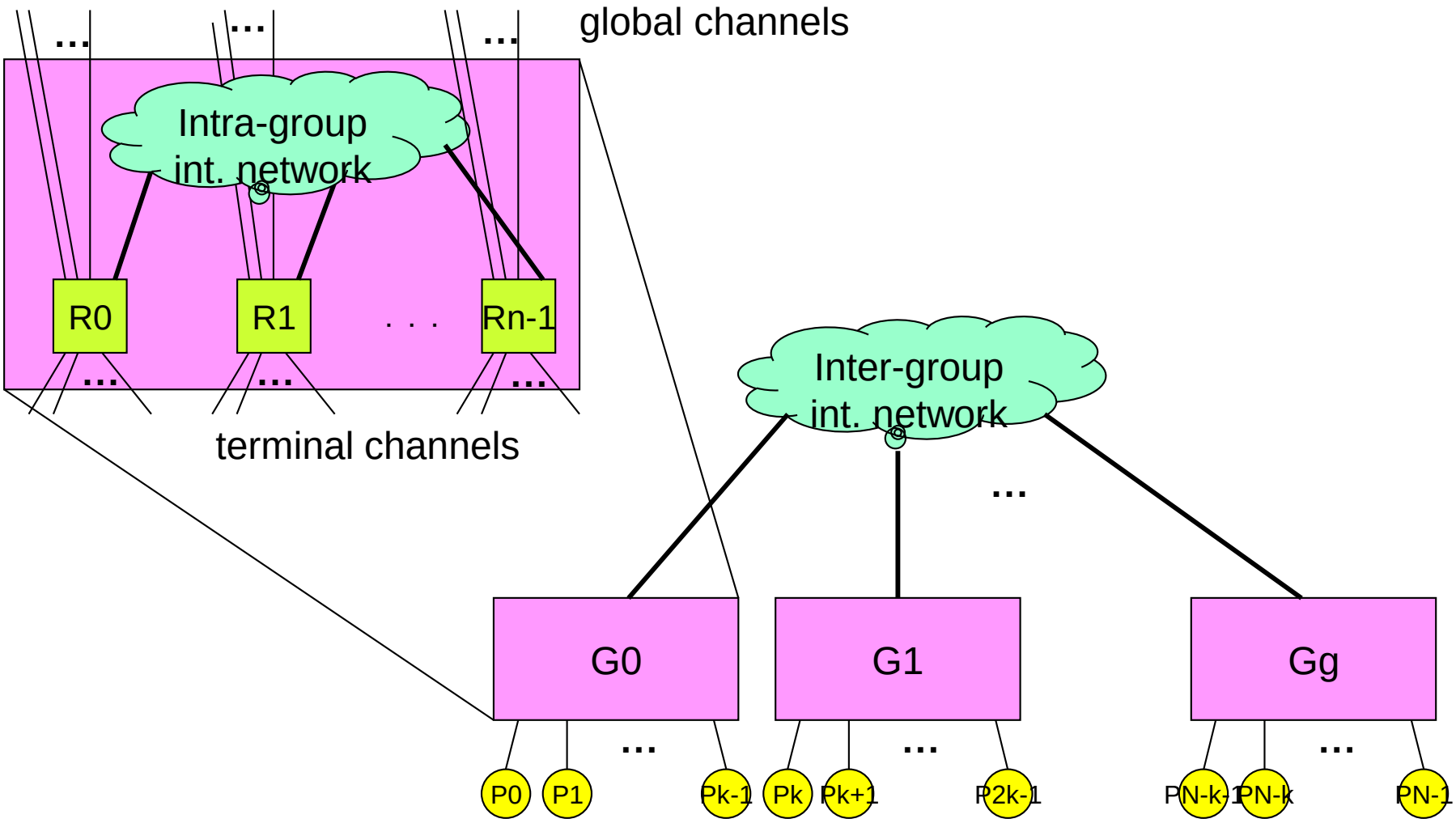
- 128nodes(Clos128)

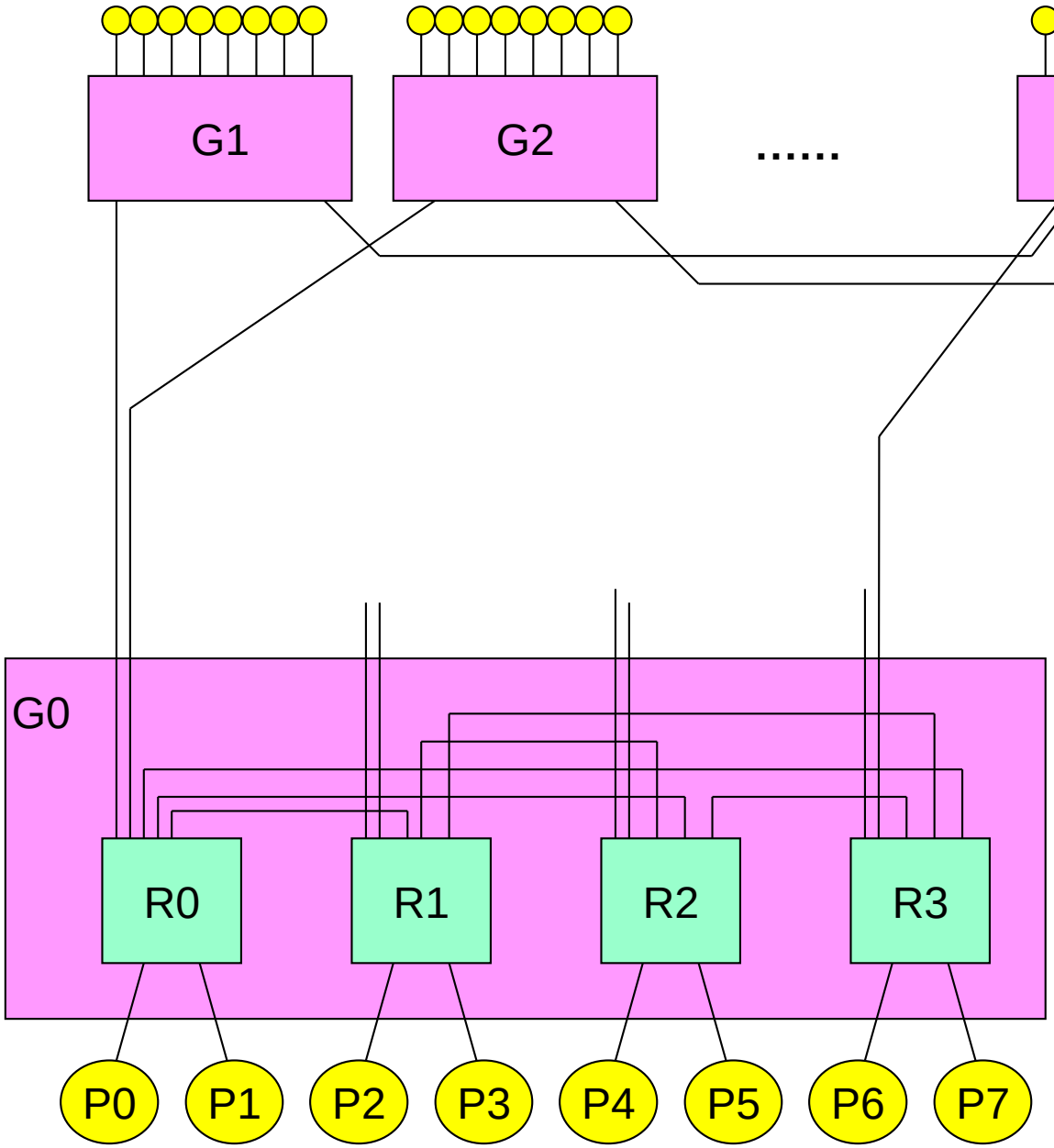
Flattened butterfly



A row of MIN is fused.
High radix → High bandwidth
Multiple paths can be formed

Dragonfly

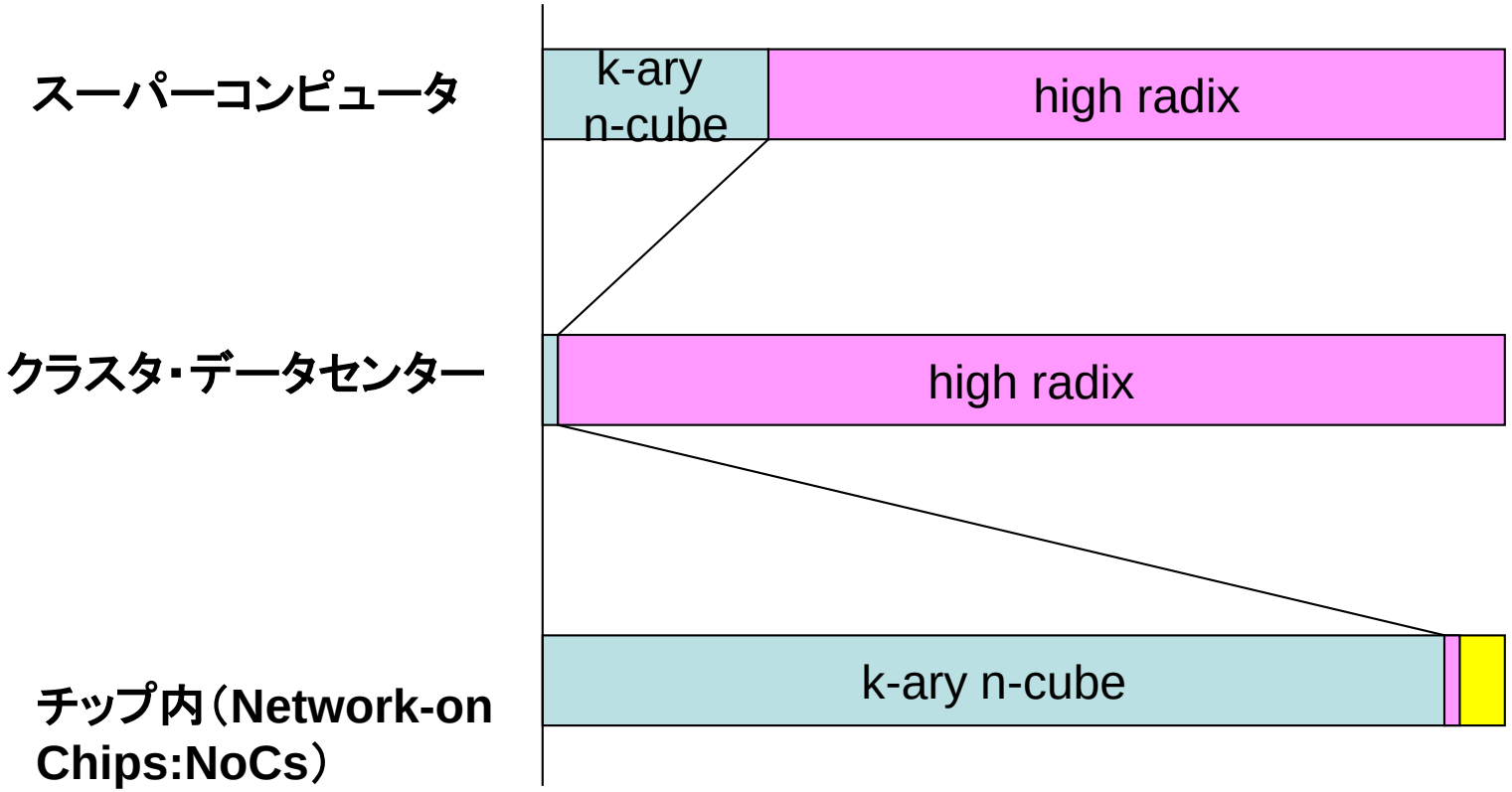

global channels
Intra-group
int. network
R0
R1
Rn-1
terminal channels
Inter-group
int. network
G0
G1
Gg
P0
P1
Pk-1
Pk
Pk+1
P2k-1
PN-k-1
PN-k
PN-1



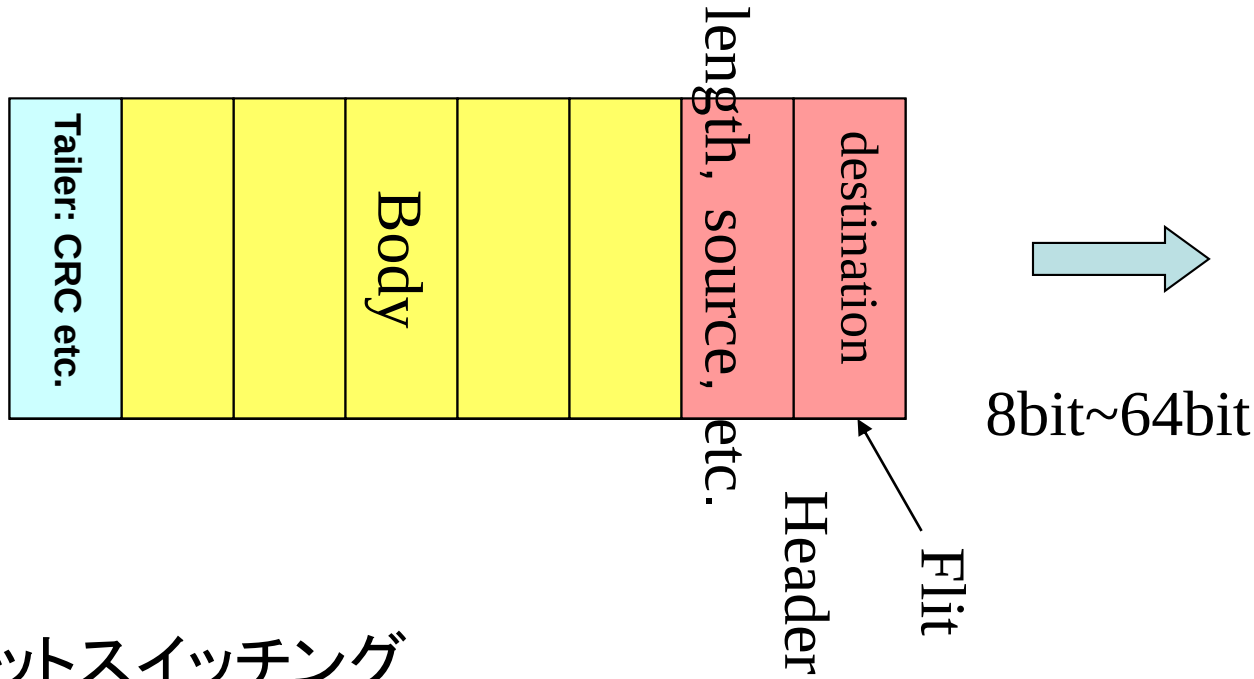
An example of Dragonfly
(N=72)

The interconnection of
this part can be Flatten Butterfly

k-ary n-cube 対 high radix



3. パケットの流し方



パケットスイッチング

⇕

サーキットスイッチング

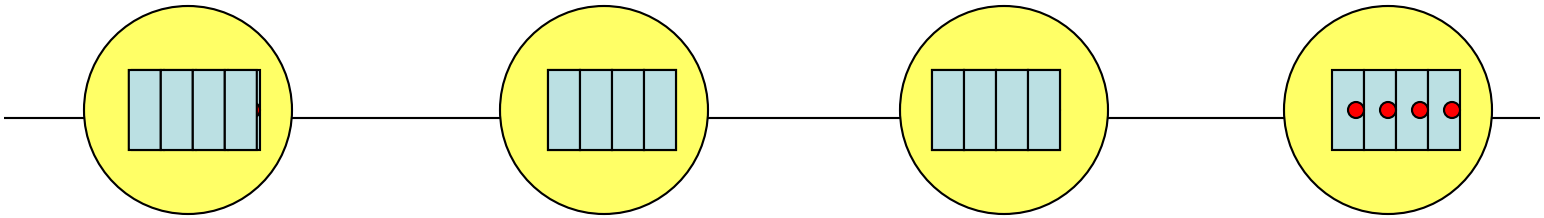
Flit：転送の基本単位
配線のbit幅でない場合もある

クラスタではパケットスイッチングを使う

パケット転送手法

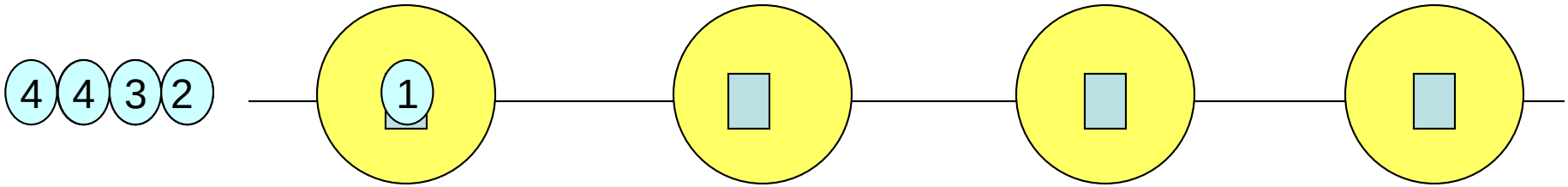
- ストア・アンド・フォワード（Store-&-Forward)
  - パケット全体をノードのバッファに蓄えてから次のノードに送る
  - TCP/IP は、これを使っている
- ウォームホール（Wormhole）
  - フリット単位で先に進んでいける
  - 先頭が進めなくなると全体が停止する
- バーチャル・カットスルー（Virtual Cut Through)
  - 先頭のflitが進めなくなるとパケットの残りをノード中のバッファに格納する

Store and Forward



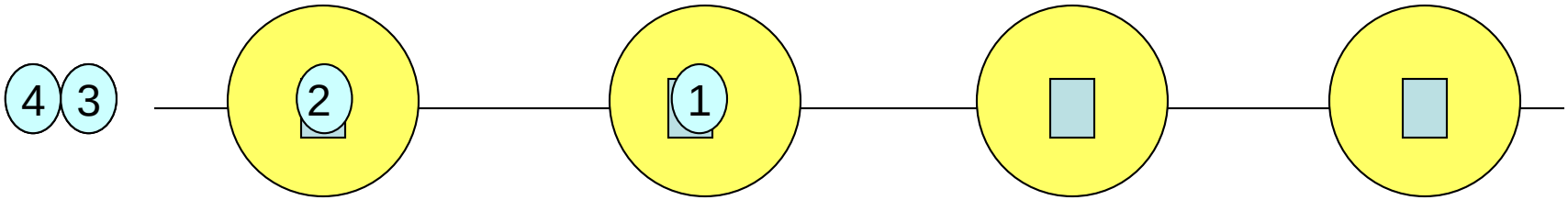
- パケット全体をバッファに格納してから進む
- ノード単位で再送が可能
- しかし転送レイテンシィが大きい：D(h+b)
- バッファサイズも大きいものが必要
- パケット転送、エラー時の再送制御などはソフトウェアで可能→TCP/IPで使われる

Wormhole

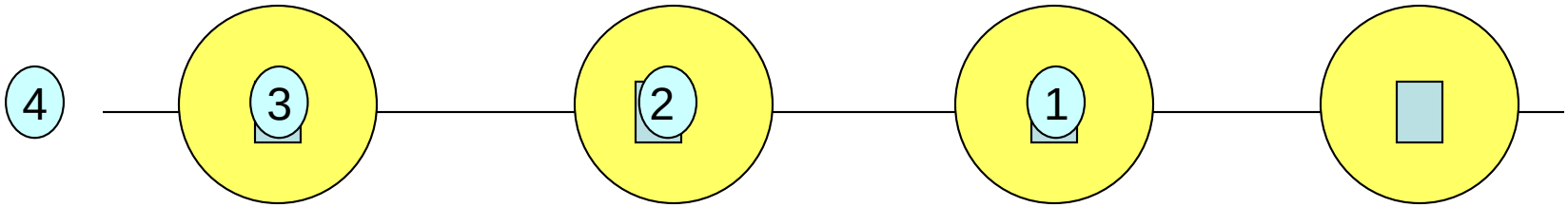


- パケットのフリットはどんどん先に進める
- 転送遅延が小さい　hD+b
  h: header, D:Diameter, b: body
- バッファ要求量も小さい（ヘッダが入ればいい）
- しかし、先頭が進めないと複数のノードにまたがってパケットがストップしてしまう→混雑の原因に！
  →仮想チャネルが有効
- ハードウェアの専用ルータ（スイッチ）が必要

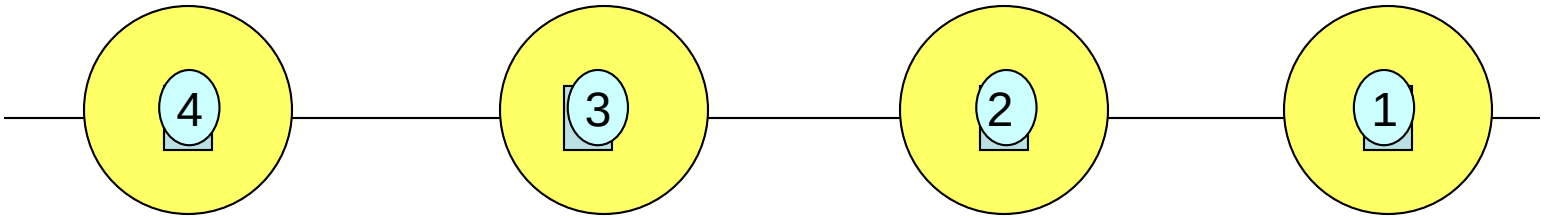
Wormhole



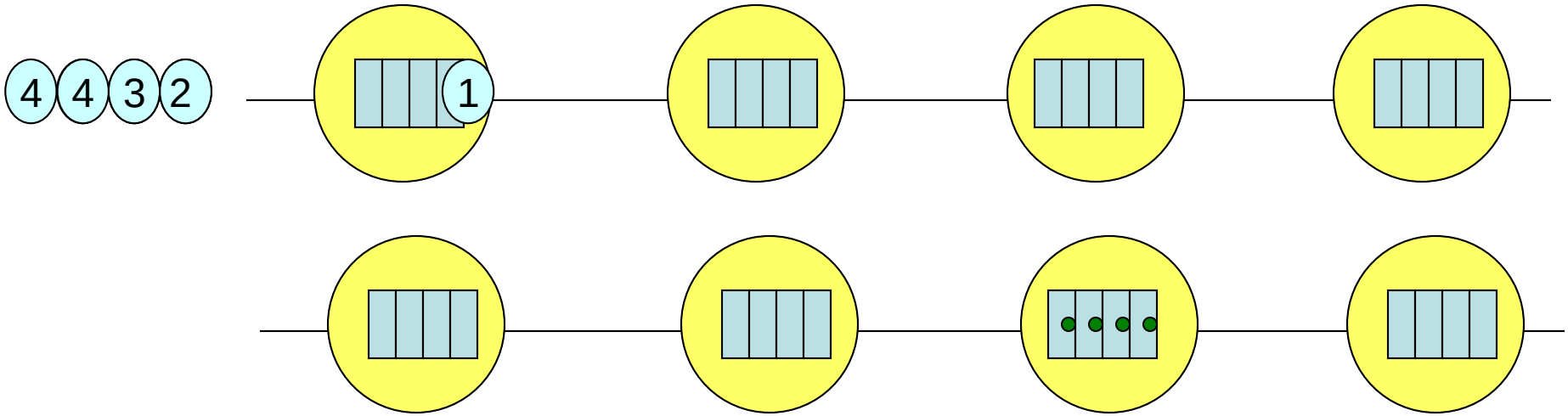
Wormhole



Wormhole

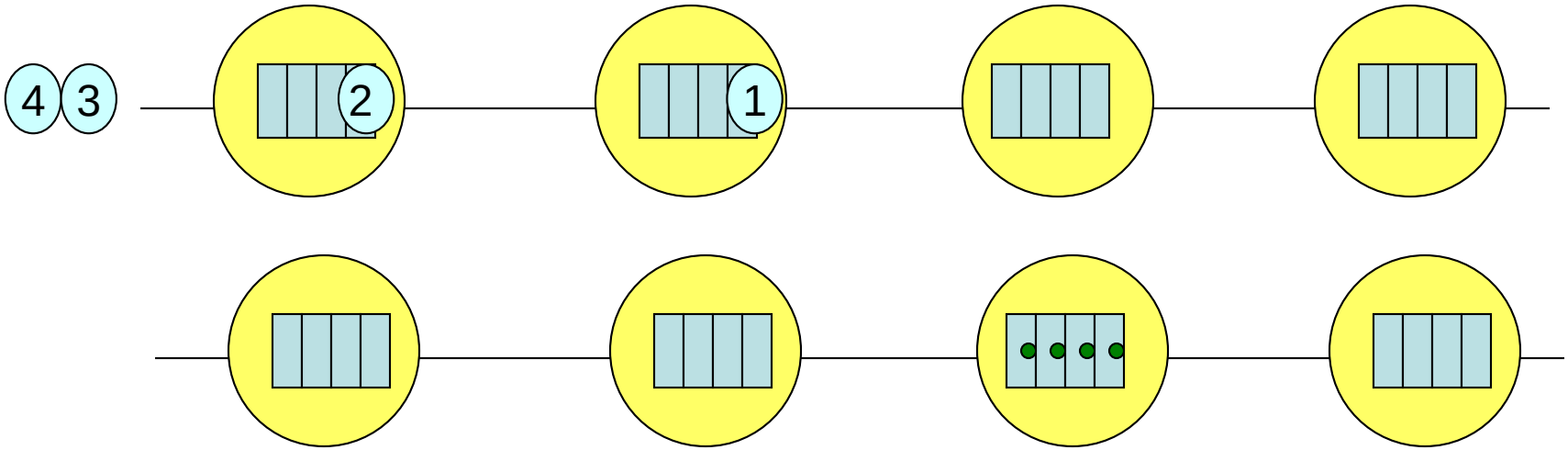


Virtual Cut Through

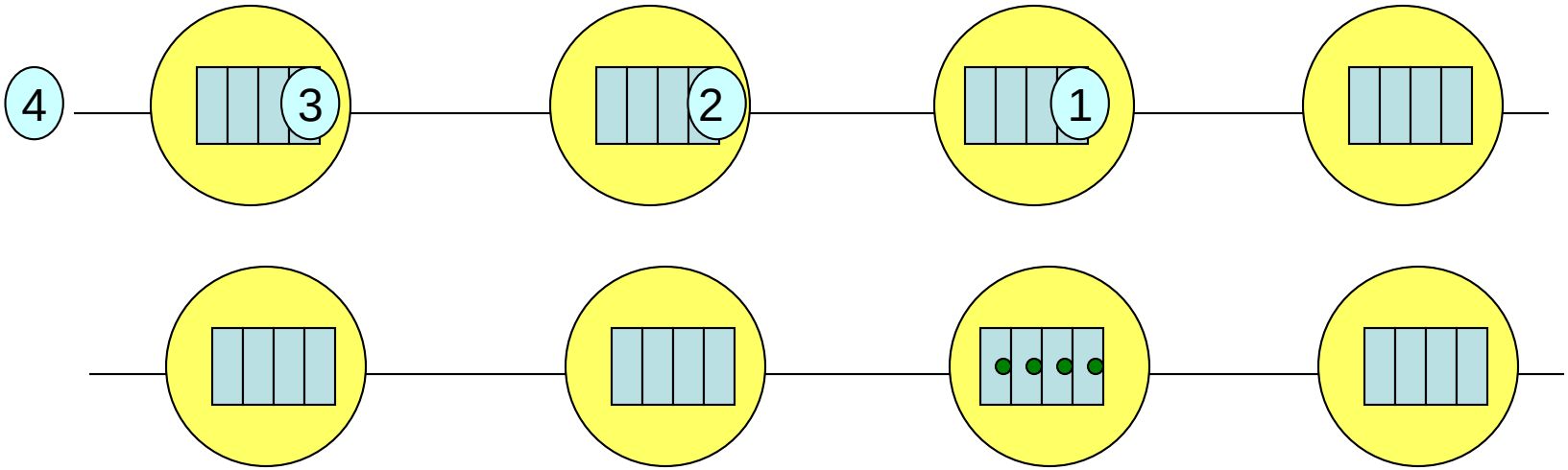


- Wormholeと同じくどんどん先に進める
- 先頭が進めなくなると、残りがバッファに入る
  →ノードにまたがってバッファを占領しない
  →Wormholeほど混雑をおこさない。
- Wormholeと同じ転送遅延時間
- Store and Forwardと同じバッファが要求される
- ハードウェアルータが必要

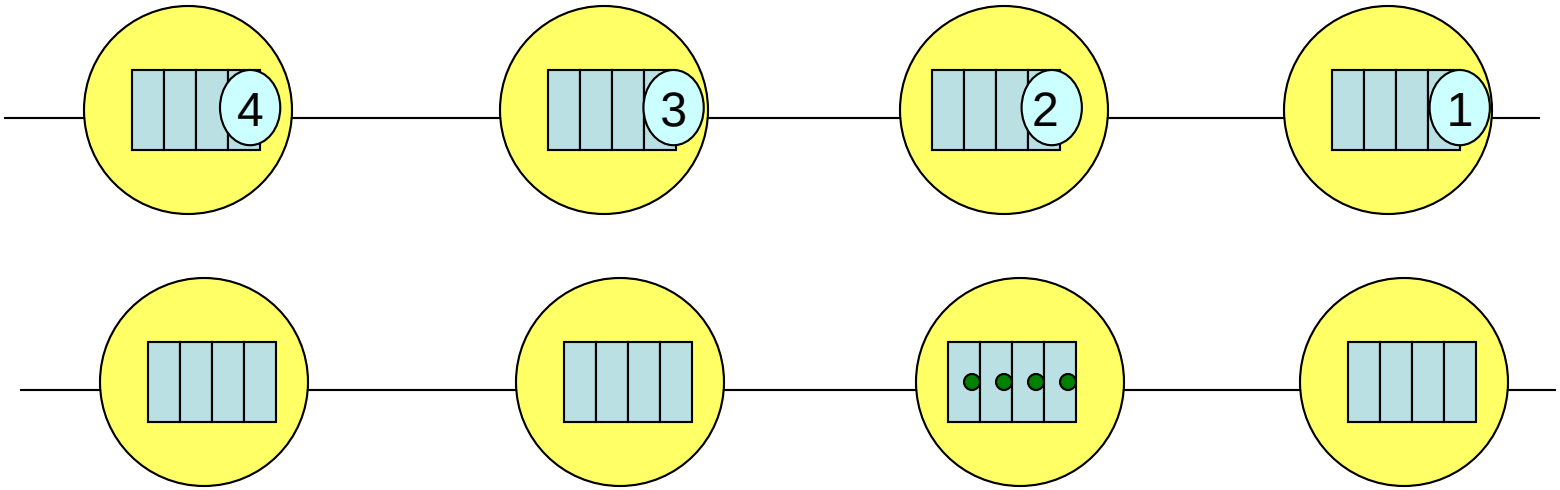
Virtual Cut Through



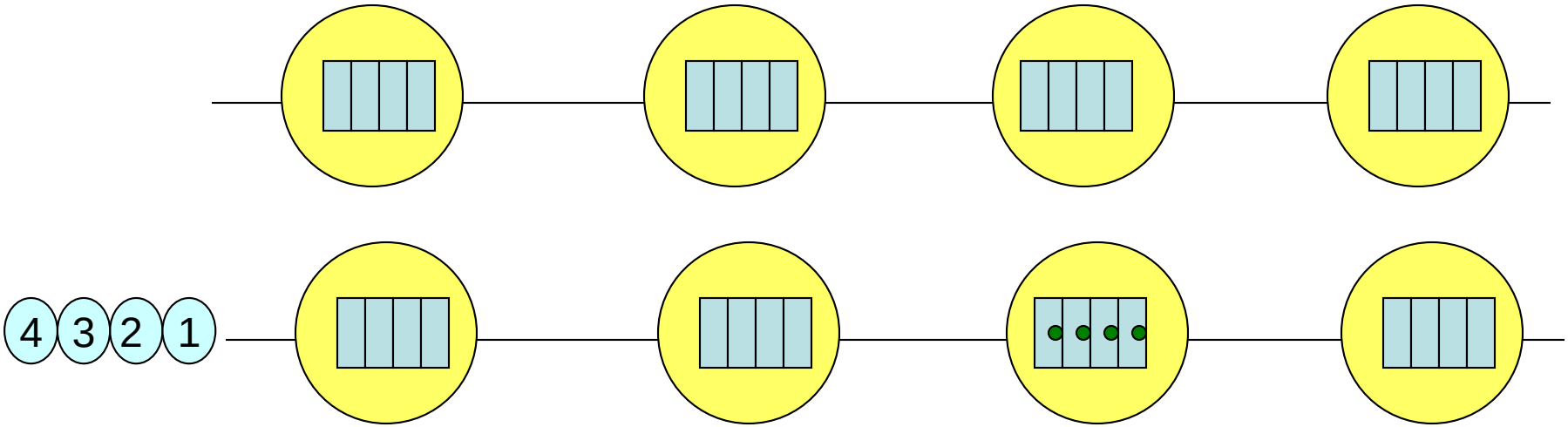
Virtual Cut Through



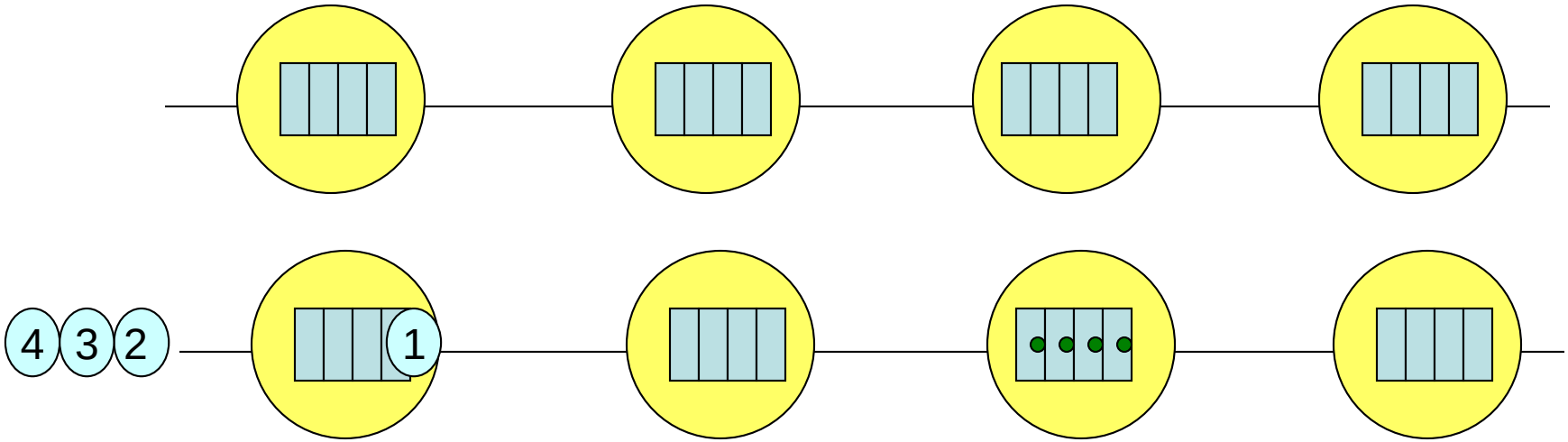
Virtual Cut Through



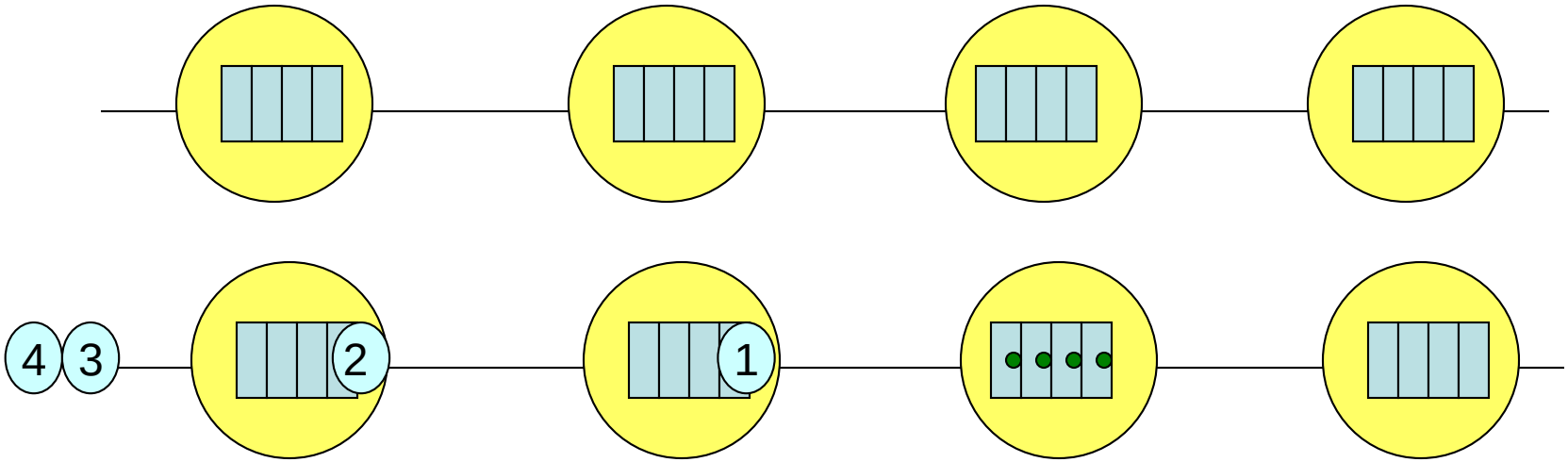
Virtual Cut Through



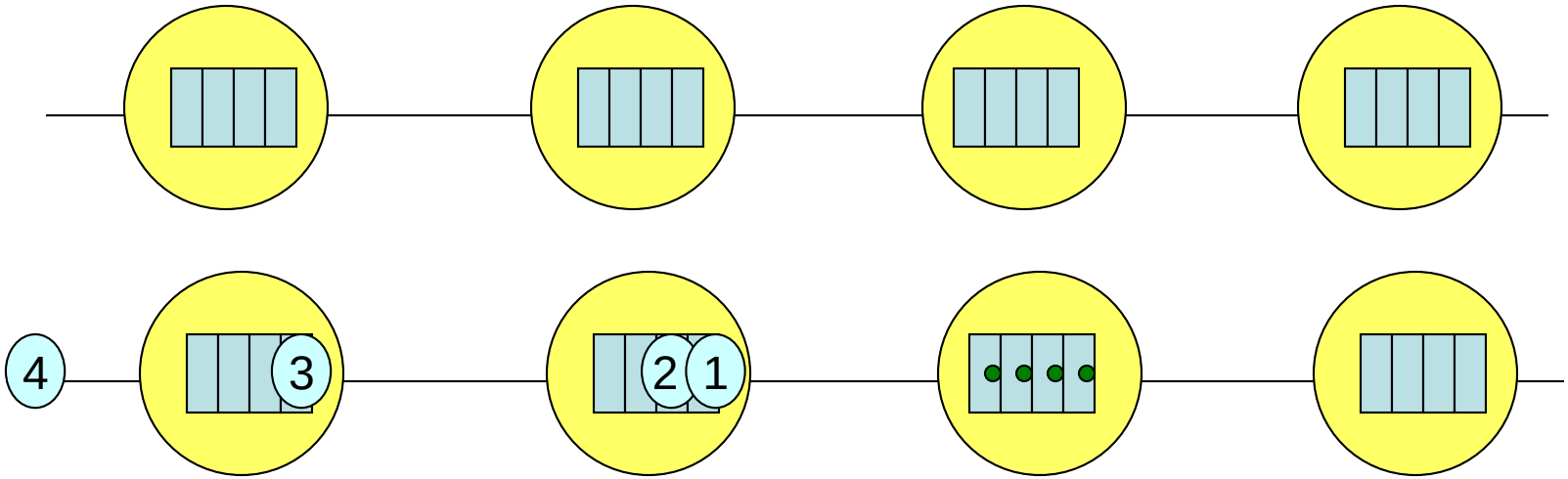
Virtual Cut Through



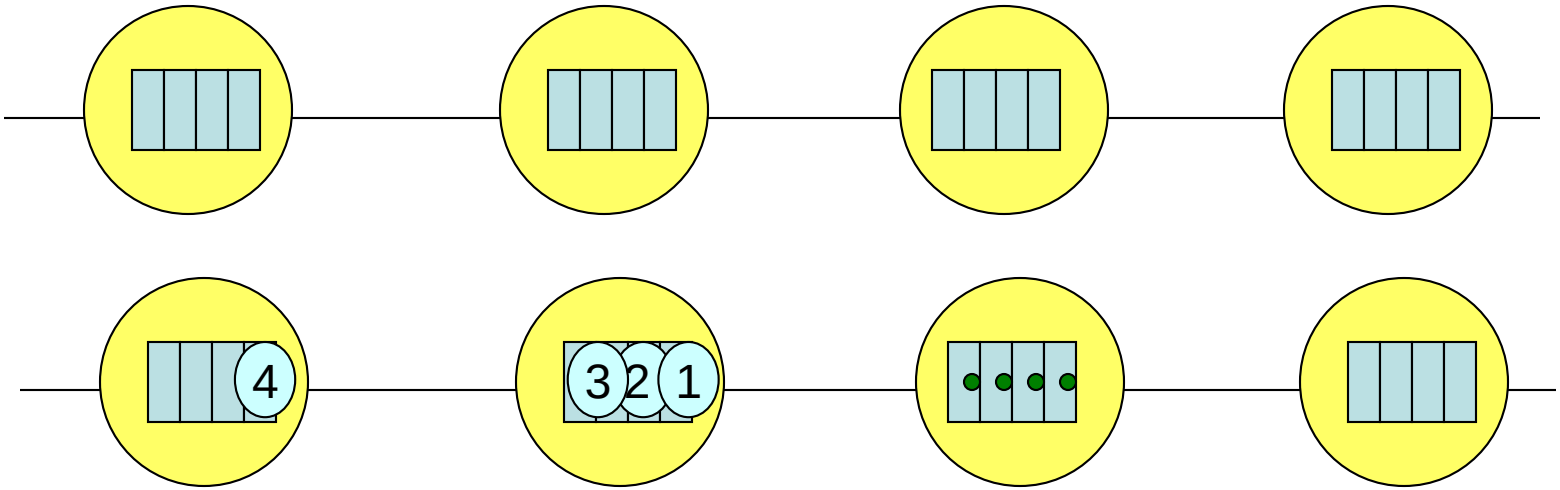
Virtual Cut Through



Virtual Cut Through



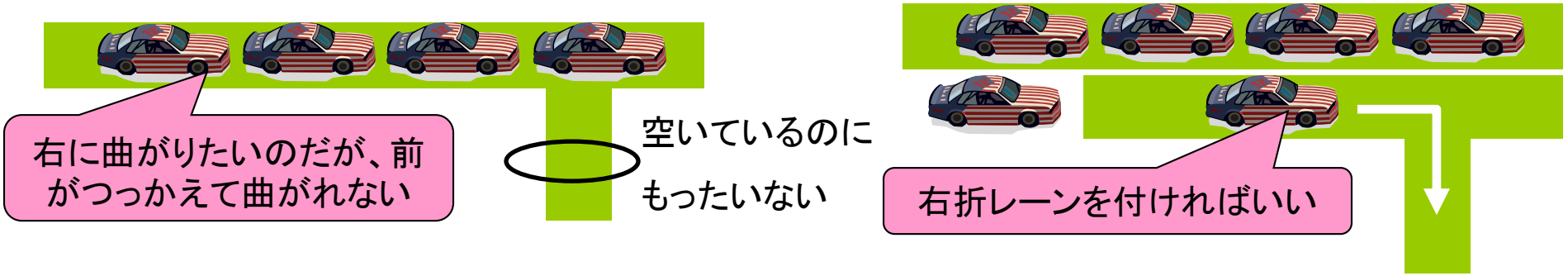
Virtual Cut Through



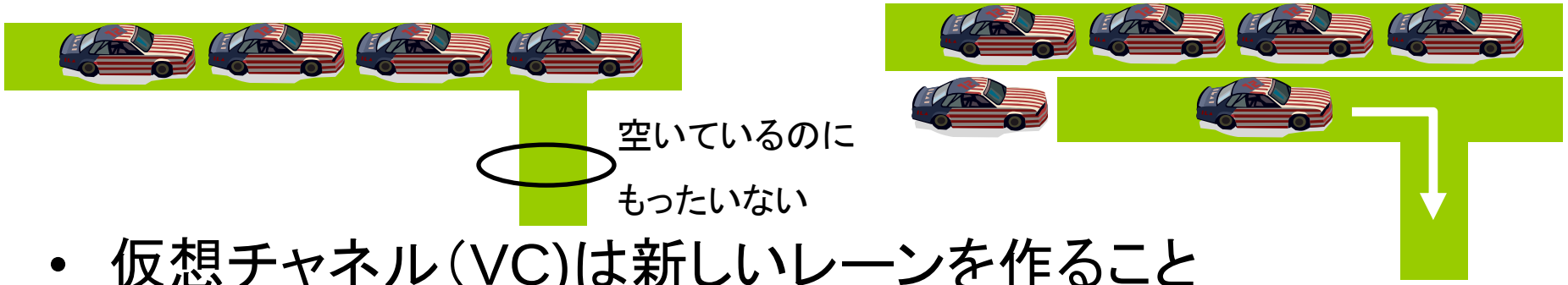
仮想チャネル（Virtual Channel)

- 特にWormholeでは、パケットが複数のノードのバッファを占有
- 行先のバッファが空いていても、途中が占有されて先に進めない
- 独立したバッファとハンドシェーク線を用意することで、物理的な配線を増やさずに空いたバッファを有効活用
- デッドロック回避（後程説明）にも有効

仮想チャネル：混雑の回避

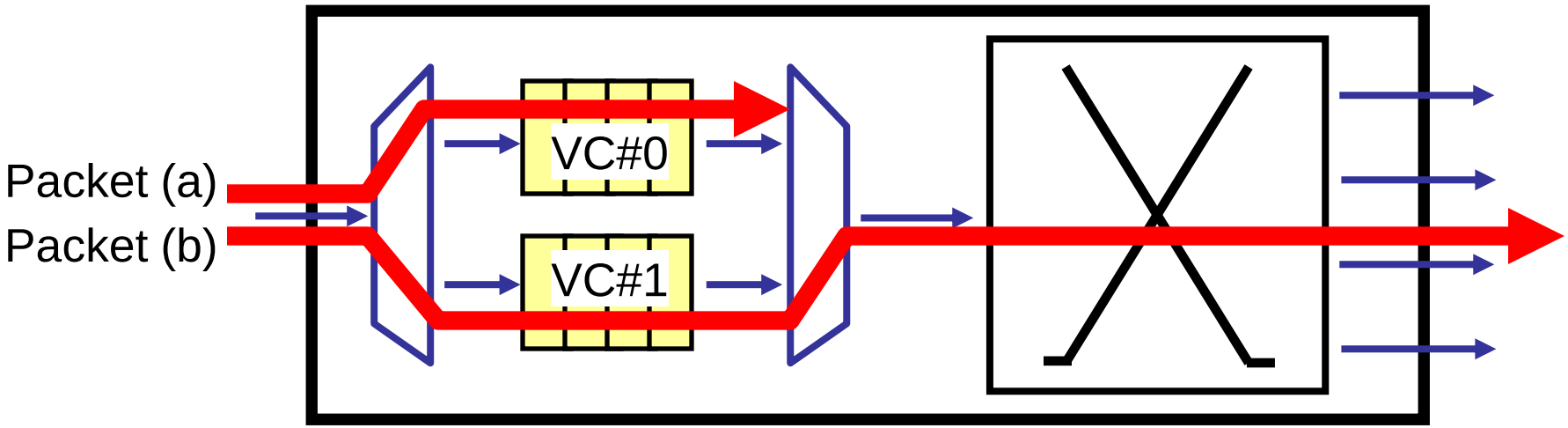


仮想チャネルの実装

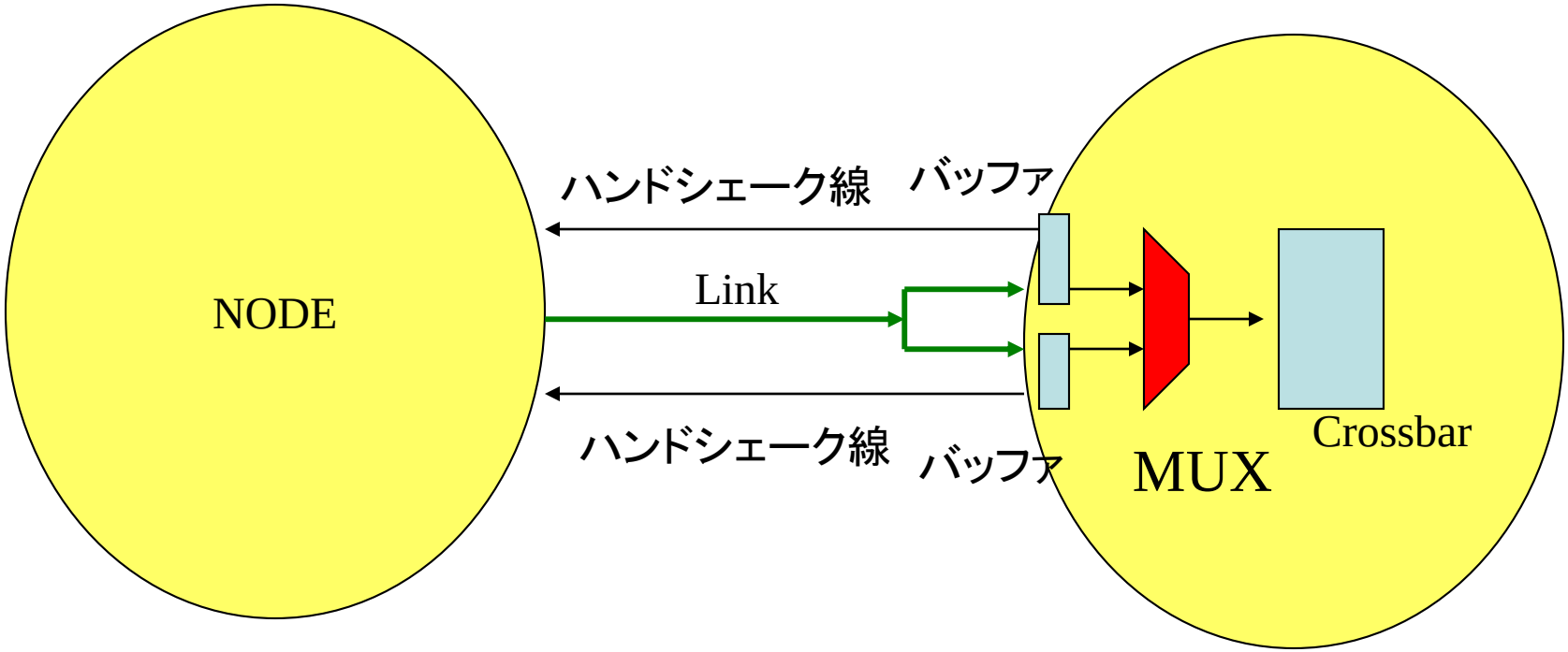


- 仮想チャネル(VC)は新しいレーンを作ること
  - 物理的なワイヤを増やすのではないことに注意！

[Dally,TPDS'92]

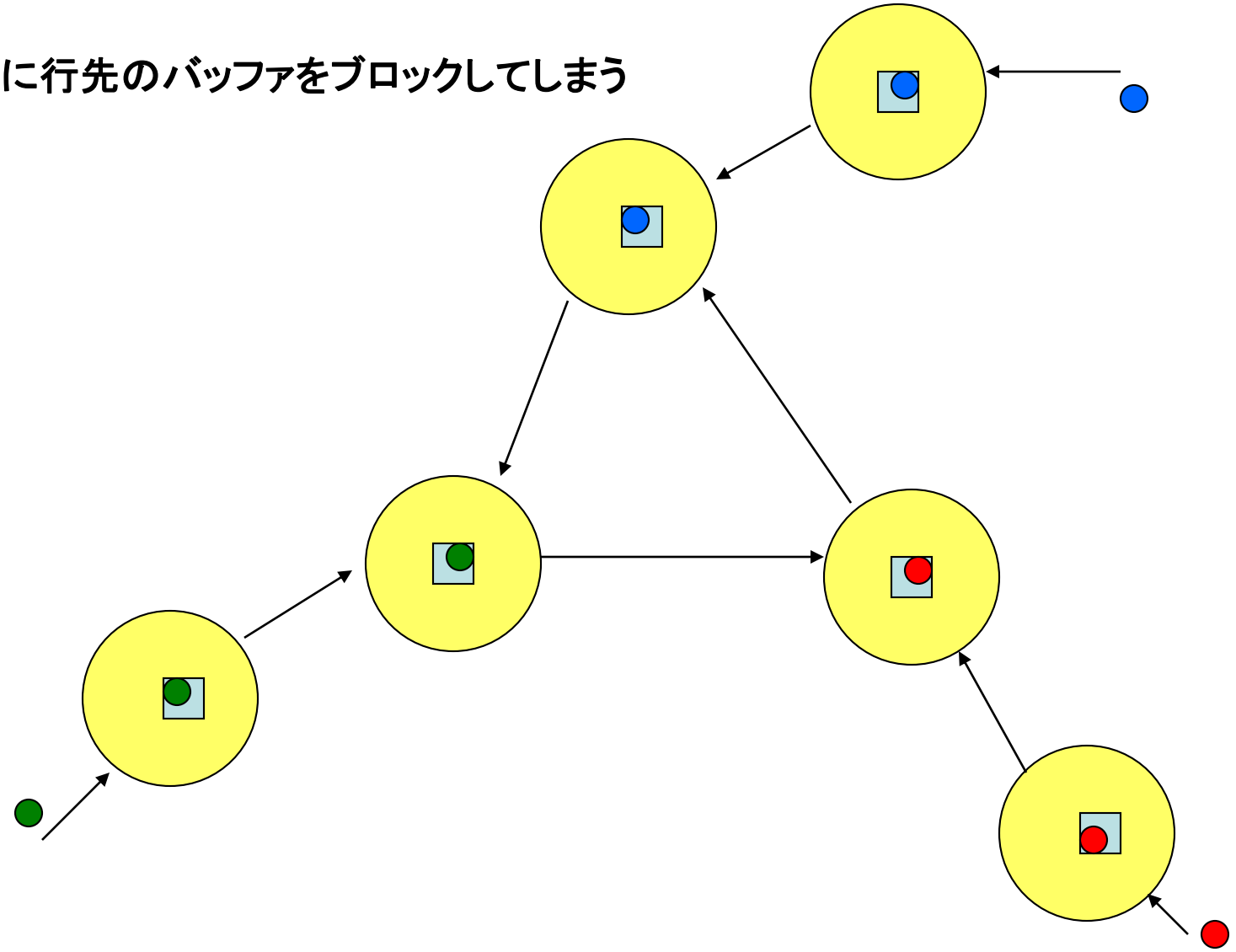


バッファの空きを知らせる線が必要



デッドロック

互いに行先のバッファをブロックしてしまう



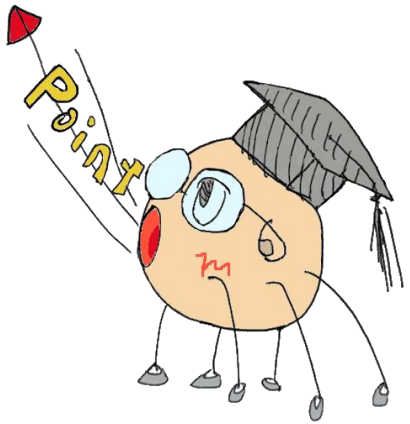
デッドロックの回避

- バッファ間の依存関係が循環構造を形成しないようにする。
  - 循環しない位多量にバッファを持たせる
    - 構造化バッファ法
  - 曲がる方向を制限する
    - XY Routing (Dimension Order Routing)
    - Turn model
  - 仮想チャネルを使う
- ネットワークに応じて様々な構造が提案されている

まとめ

- 共有メモリを持たないクラスタは、最も簡単に大規模な並列コンピュータを構成できる
- 独立したジョブを扱うデータセンターなどに向いている
- ノード間のネットワークが重要
  - 直接網
  - 間接網
  - ルーチング手法
- プログラムは、メッセージパッシングライブラリを用いてノード間のデータ交換を明示的に指定する

→次回演習



演習

- 4-ary 8-cubeで、ヘッダサイズ1、ボディ16フリットのパケットをWormhole方式で、転送した場合、最大遅延は何クロックになるか？
  - 1クロックに1フリット転送可能と考える
  - パケットの衝突は考えない