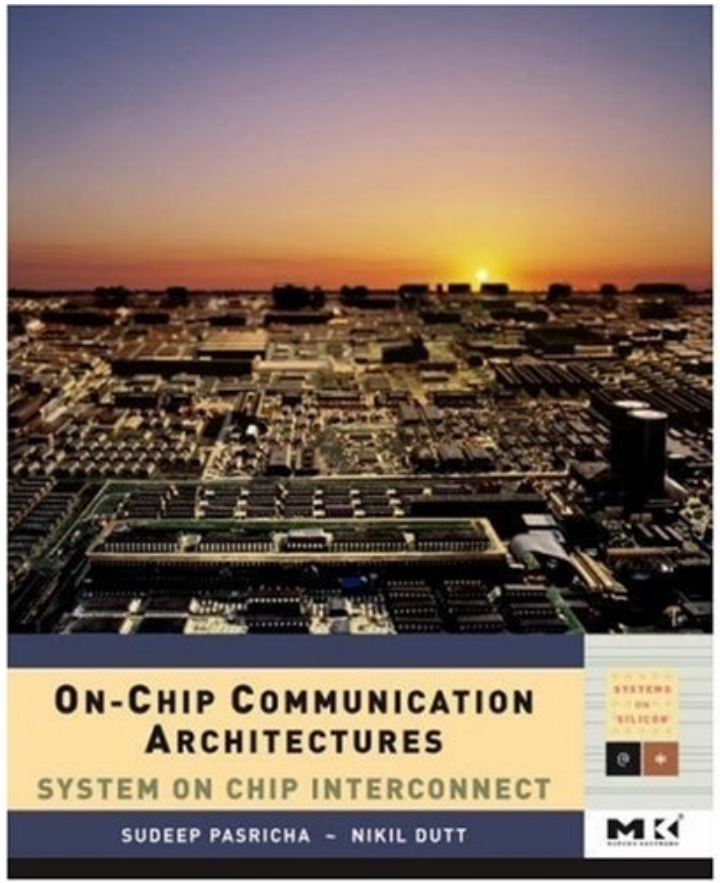
On-Chip Communication Architectures

Networks-on-Chip



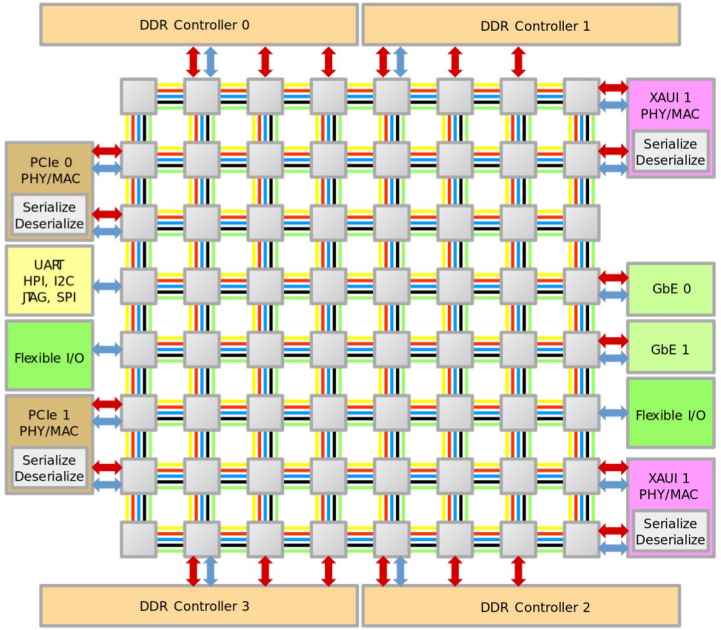
Sudeep Pasricha and Nikil Dutt

Slides based on book chapter 12

and from Moraes (16/ago/2026)

Outline

- Introduction
- NoC Topology
- Switching strategies
- Routing algorithms
- Estudo de caso: NoC HERMES



Introduction

- Evolution of on-chip communication architectures

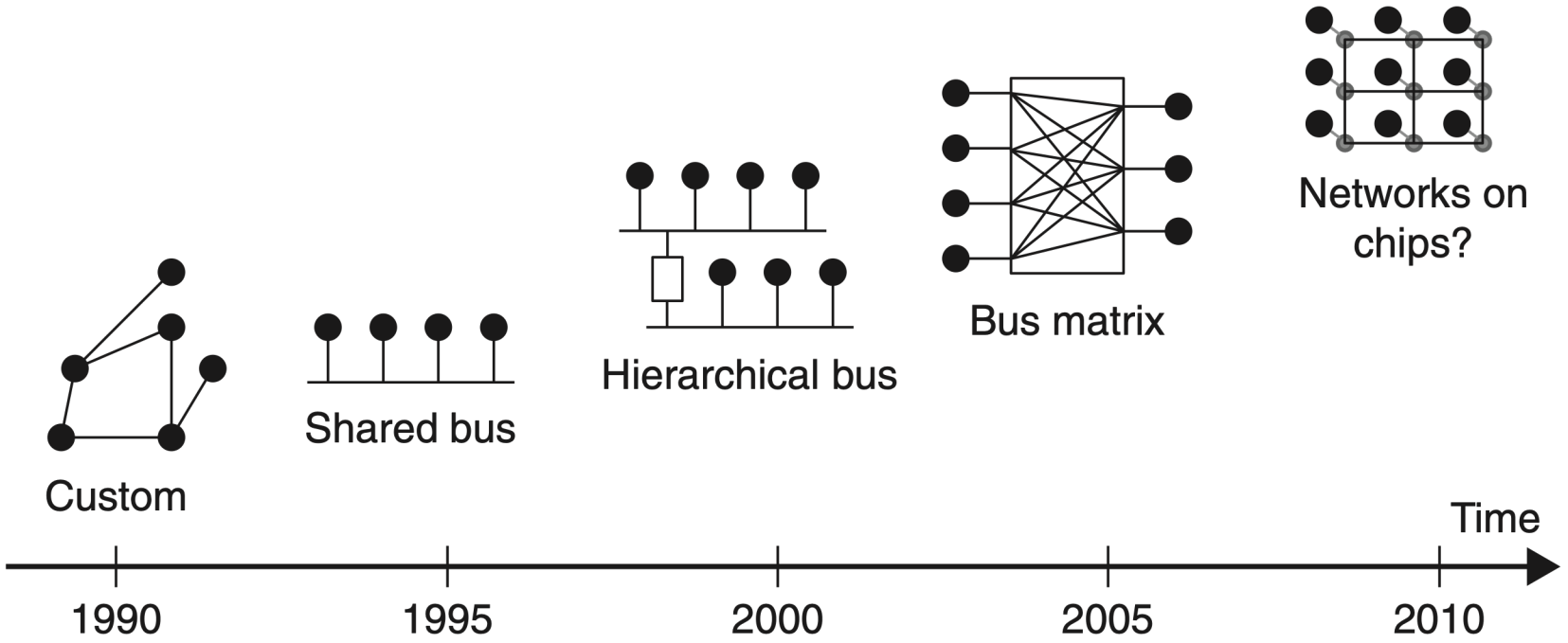
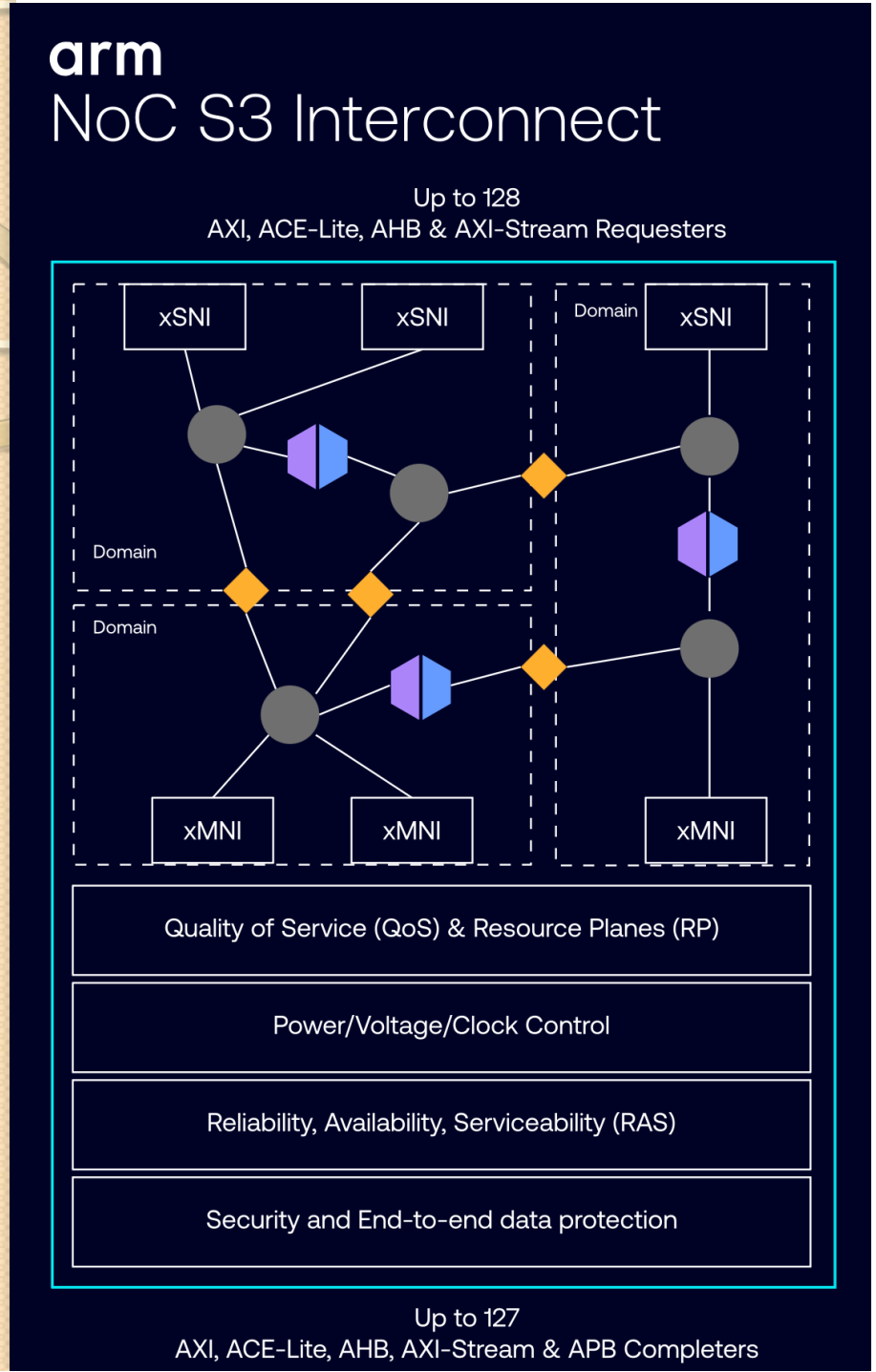


FIGURE 12.3

Evolution of on-chip communication architectures



INTEL:
Applying-the-benefits-of-network-on-a-chip-architecture-to-fpga-system-design-white-paper

https://docs.altera.com/v/u/docs/650306/applying-the-benefits-of-network-on-a-chip-architecture-to-fpga-system-design-white-paper

Meta's Second Generation AI Chip: Model-Chip Co-Design and Productionization Experiences

ISCA'25

https://dl.acm.org/doi/full/10.1145/3695053.3731409

Figure 1: MTIA 2i's overall architecture.

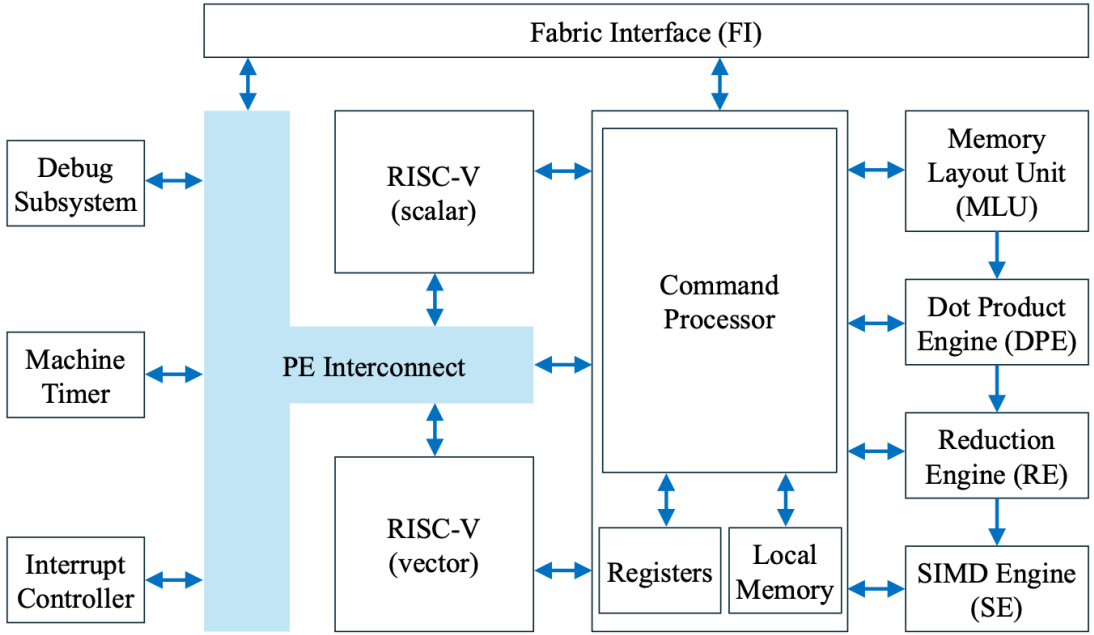
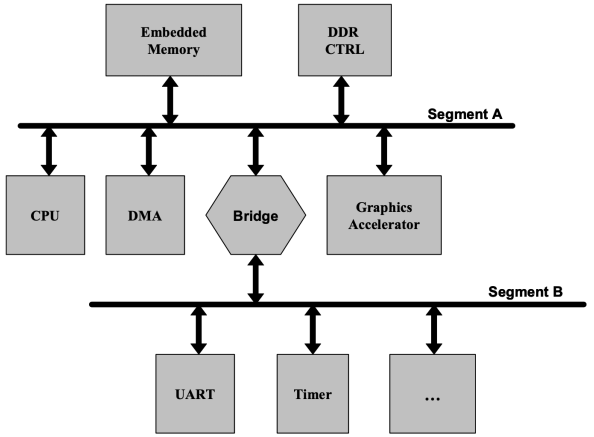


Figure 2: Internal architecture of Processing Element (PE).

Introduction



Bus-based design

- **Scaling**
  - Estimating delays becomes harder
    - wire geometry determined later in design flow
    - In advanced processes, 80% of the delay of critical path comes from interconnects
    - Electrical noise due to crosstalk, delay variations and synchronization failure results in bit upset

- **Conclusion**: transmission of digital values on wires are slow, power hungry and unreliable

NoC Definition

- NoC allows decoupling processing cores from communication fabric
  - The need for global synchronization is eliminated

- Benefits
  - Explicit parallelism
  - Modularity
  - Minimize the usage of global wires
  - Power minimization
  - Scalability
  - Better performance

Introduction

- Network-on-chip (NoC) is a packet switched on-chip communication network designed using a layered methodology

  "routes packets, not wires" → Dally & Towles 2001 - DAC

- NoCs use packets to route data from the source to the destination PE via a network fabric that consists of
  - routers
  - interconnection links (wires)

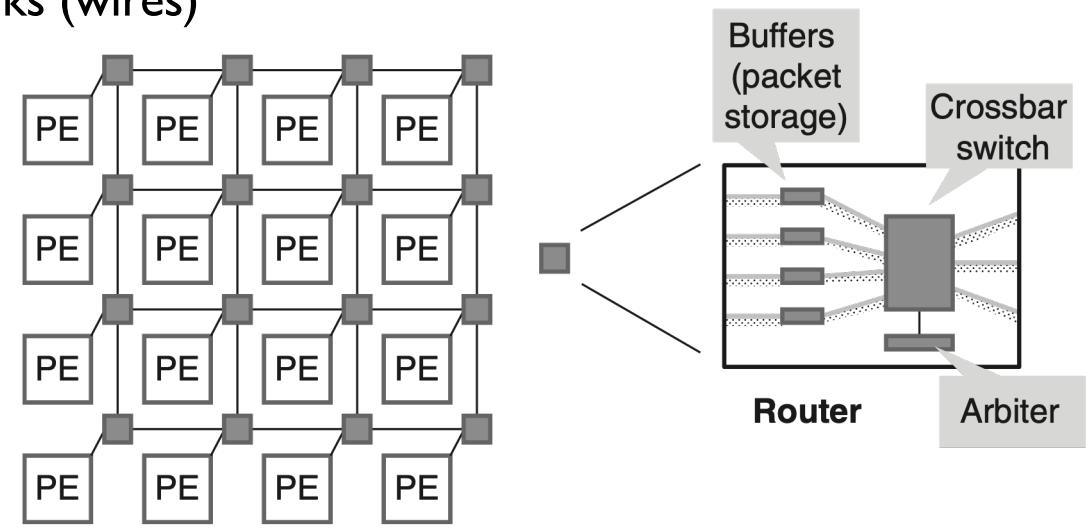


FIGURE 12.1

An example of a (mesh type) NoC interconnection fabric

Barramento × NoC

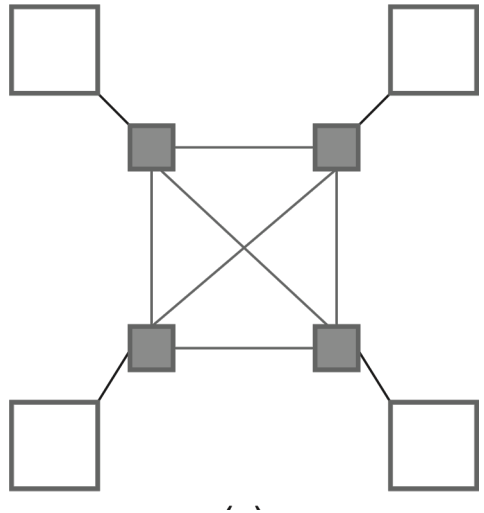
- **Barramento compartilhado**
  - banda dividida entre todos os mestres; apenas uma transferência por vez
  - arbitragem centralizada: latência cresce com o número de mestres
  - fios globais longos, com atraso e consumo dominados pela interconexão
- **NoC**
  - banda agregada cresce com o número de roteadores
  - enlaces curtos, regulares e ponto a ponto: temporização previsível
  - transferências simultâneas entre pares distintos de nós
  - **custo**: área dos roteadores e latência acumulada em múltiplos saltos
- A NoC compensa o custo do roteador a partir de algumas dezenas de núcleos, quando o barramento satura e os fios globais dominam o atraso

Outline

- Introduction
- NoC Topology
- Switching strategies
- Routing algorithms
- Estudo de caso: NoC HERMES

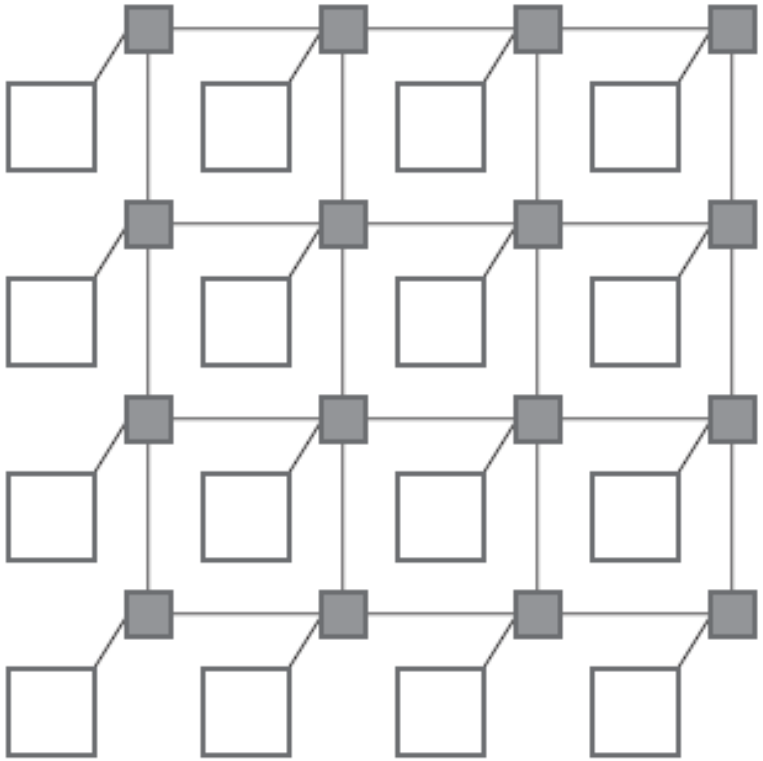
NoC Topology

- **Direct Topologies**
  - each node has direct link to router
  - nodes consist of computational blocks and/or memories, as well as a NI block that acts as a router
  - e.g. Nostrum, SOCBUS, Proteo, Octagon
  - **as the number of nodes in the system increases, the total available communication bandwidth also increases**
  - fundamental trade-off is between connectivity and cost



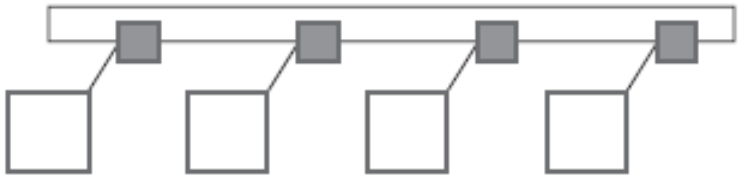
NoC Topology

- Most direct network topologies have an orthogonal implementation, where nodes can be arranged in an n-dimensional orthogonal space
  - routing for such networks is fairly simple
  - e.g. n-dimensional mesh, torus, folded torus, hypercube, and octagon
- **2D mesh is most popular topology**
  - all links have the same length
    - eases physical design
  - area grows linearly with the number of nodes
  - **must be designed in such a way as to avoid traffic accumulating in the center of the mesh**

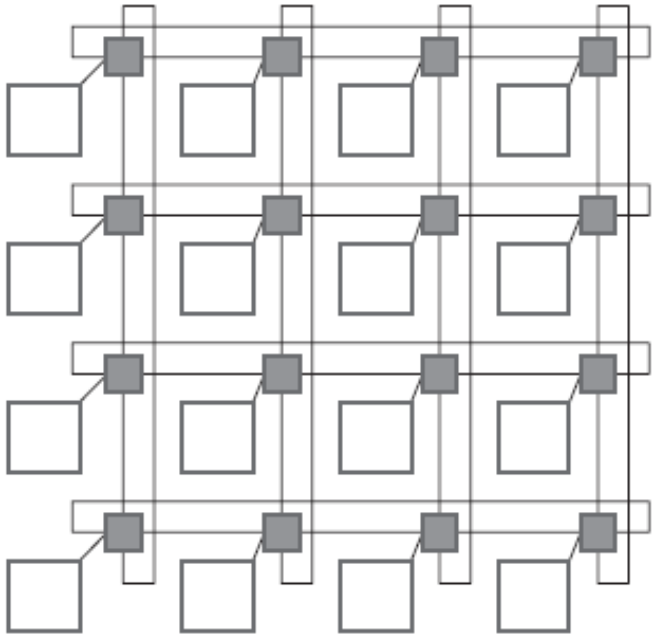


NoC Topology

- Torus topology, also called a k-ary n-cube, is an n-dimensional grid with k nodes in each dimension
  - k-ary 1-cube (1-D torus) is essentially a ring network with k nodes
    - limited scalability as performance decreases when more nodes

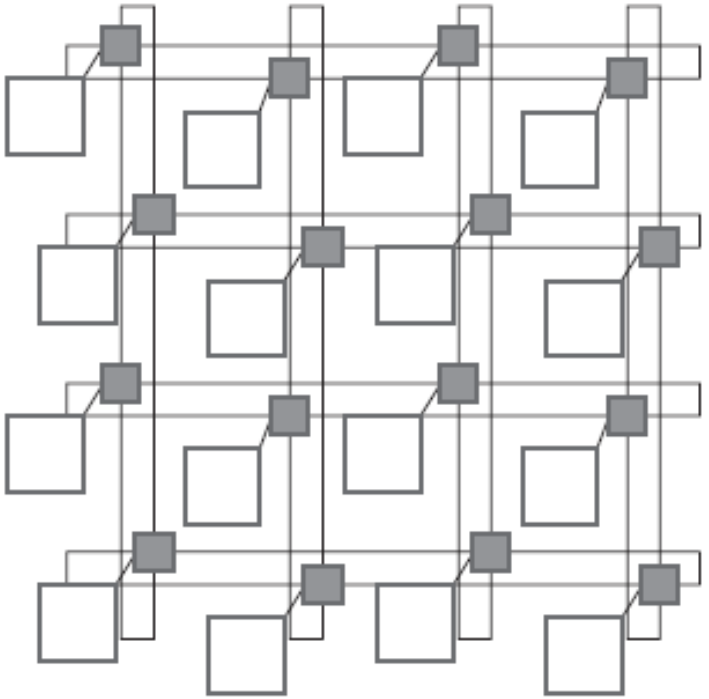


  - k-ary 2-cube (i.e., 2-D torus) topology is like a regular mesh
    - except that nodes at the edges are connected to switches at the opposite edge via wrap-around channels
    - **long end-around connections can, however, lead to excessive delays**



NoC Topology

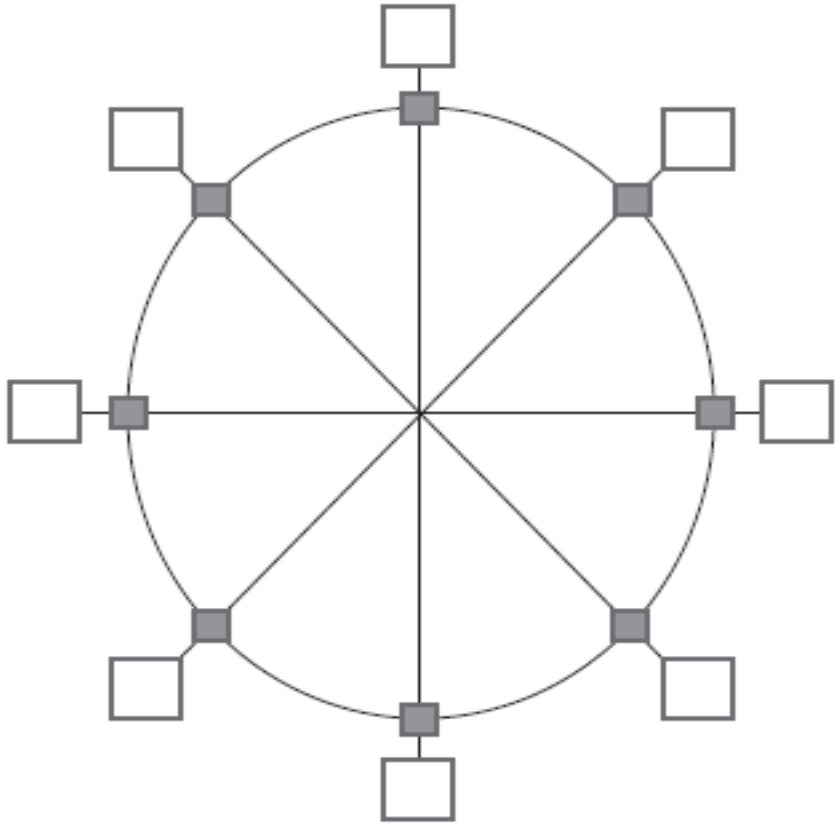
- Folding torus topology overcomes the long link limitation of a 2-D torus
  - links have the same size

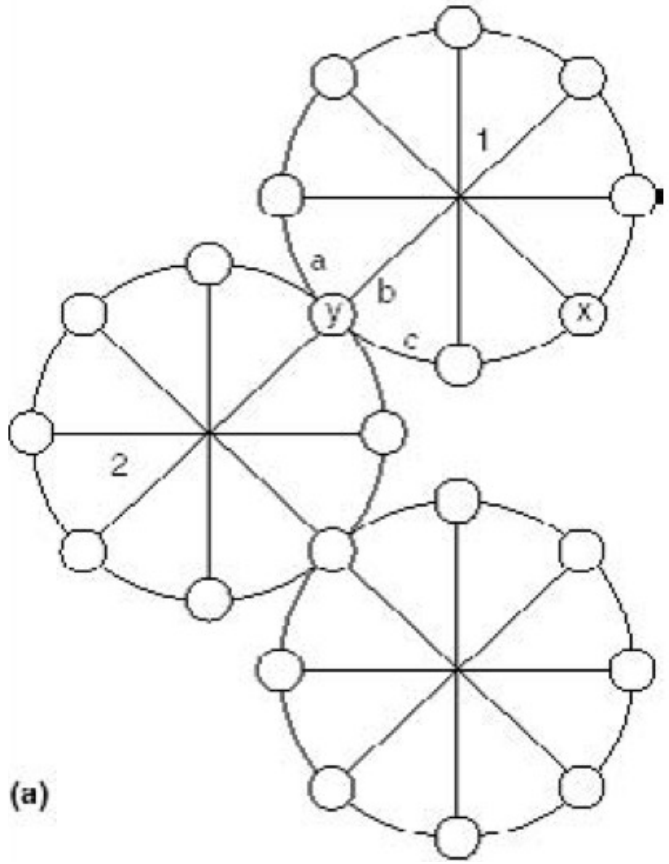


- Meshes and tori can be extended by adding bypass links to increase performance at the cost of higher area

NoC Topology

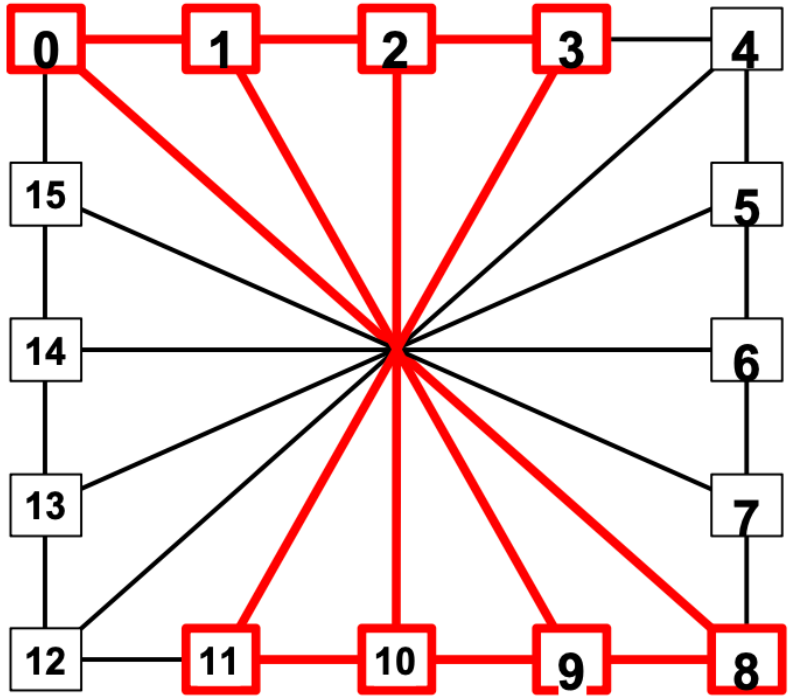
- Octagon topology is another example of a direct network
  - messages being sent between any 2 nodes require at most two hops
  - more octagons can be tiled together to accommodate larger designs
    - by using one of the nodes is used as a bridge node





NoC Topology

- Spidergon



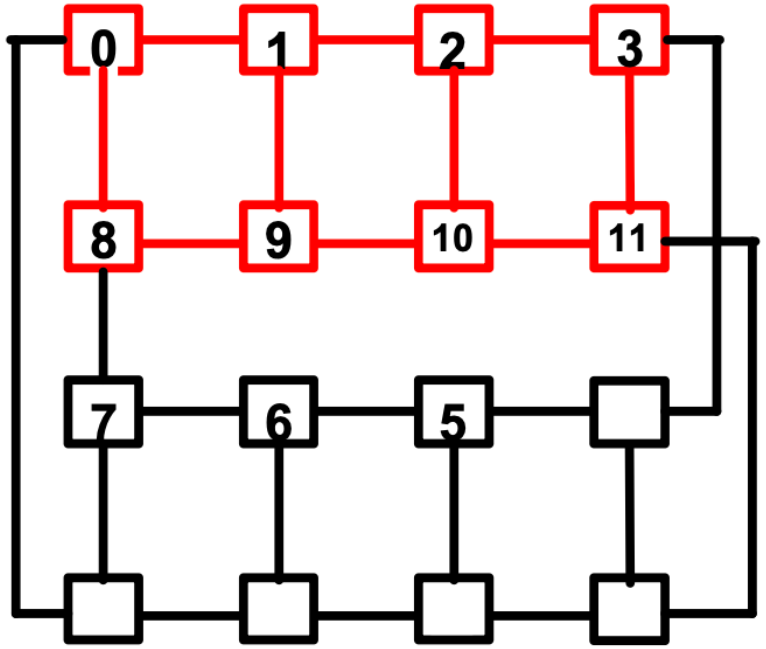
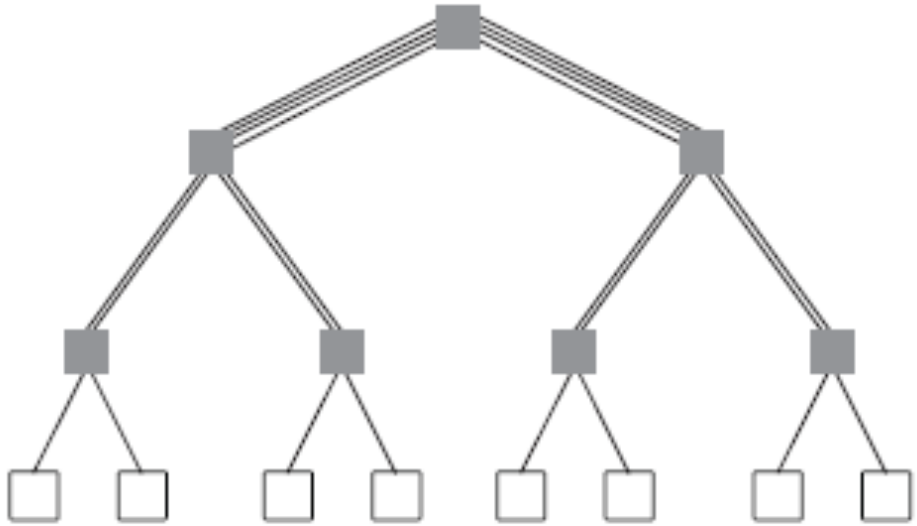


FIGURE 3.27: Equivalent representation of Spidergon STNoC for $N = 16$

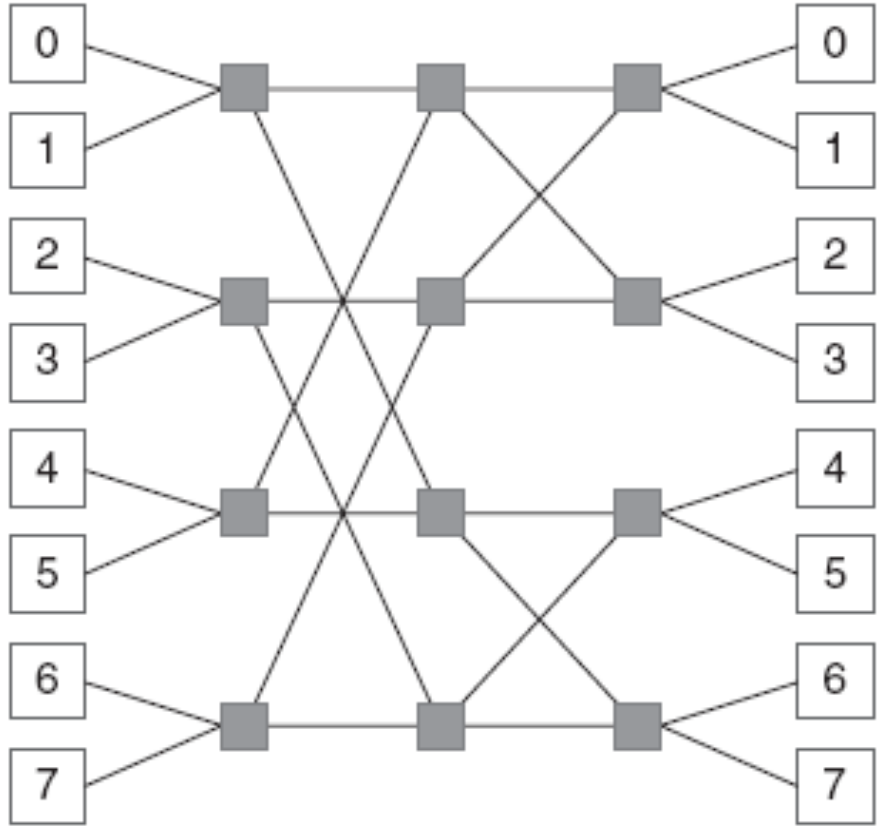
NoC Topology

- **Indirect Topologies**
  - each node is connected to an external router, and routers have point-to-point links to other switches
  - routers do not perform any information processing, and correspondingly nodes do not perform any packet switching
  - e.g. SPIN, **crossbar** topologies
- Fat tree topology
  - nodes are connected only to the leaves of the tree
  - more links near root, where bandwidth requirements are higher



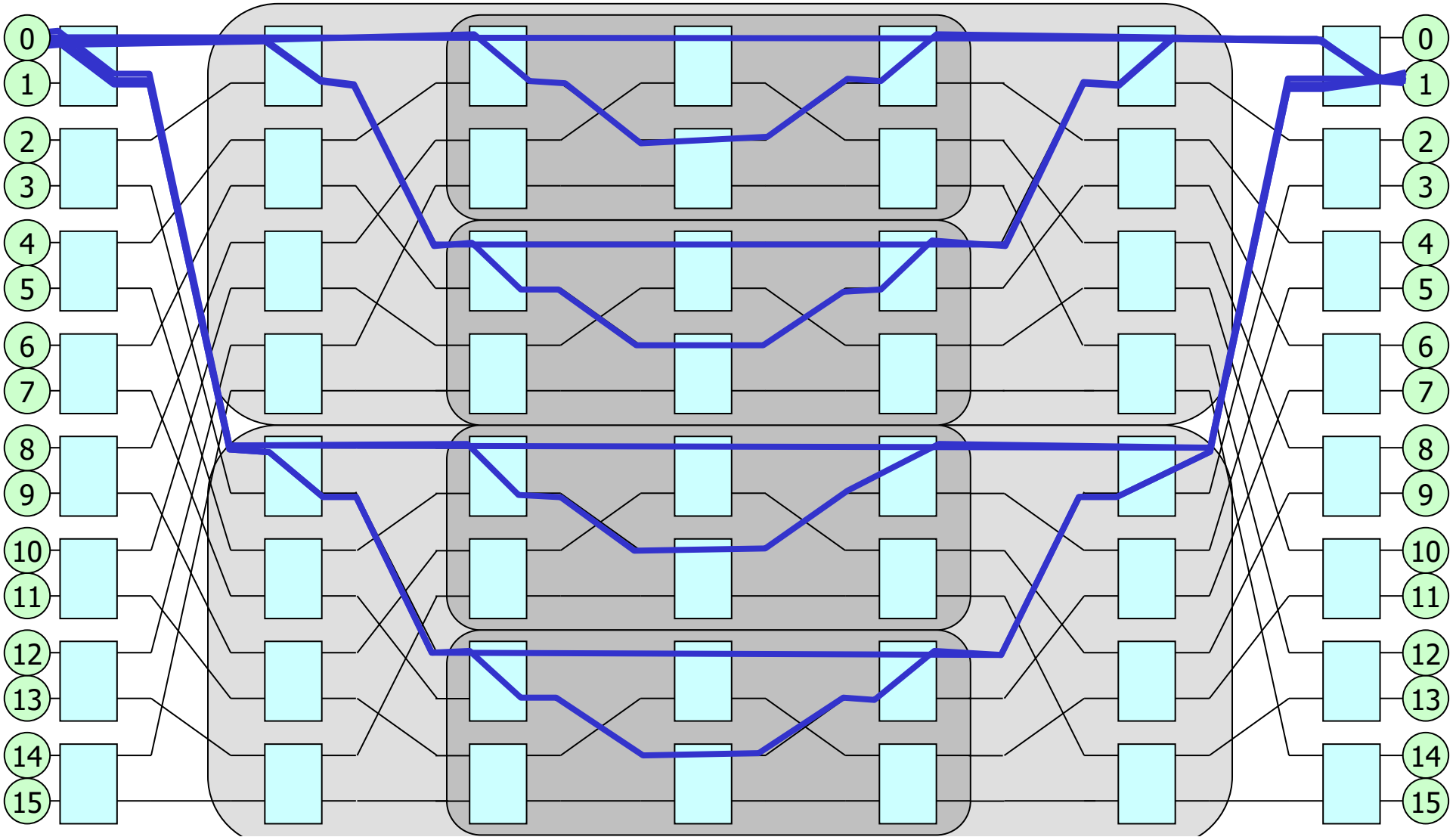
NoC Topology

- butterfly network
  - blocking multi-stage network – packets may be temporarily blocked or dropped in the network if contention occurs
  - e.g. 2-ary 3-fly butterfly network



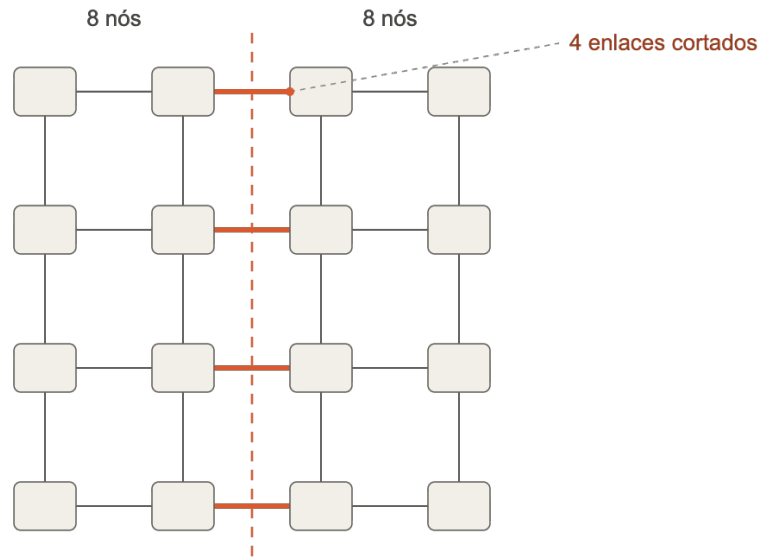
NoC Topology

Alternative paths from 0 to 1. 16 port



Topologia: métricas de avaliação

- **Grau do nó** — enlaces por roteador
  - define o número de portas, o tamanho do crossbar e a área
- **Diâmetro** — maior distância mínima entre dois nós
  - limita o pior caso de latência
- **Largura de bissecção** — enlaces cortados ao dividir a rede em duas metades
  - limite superior de vazão sob tráfego uniforme
- **Distância média** — número médio de saltos; determina a latência típica
- **Regularidade** — enlaces de mesmo comprimento simplificam o projeto físico
- **Compromisso**: mais conectividade reduz o diâmetro, mas eleva área e potência



Malha 4 × 4: qualquer corte em metades iguais atravessa pelo menos 4 enlaces

Topologia: comparação

- Rede com N nós; malha e toro $k \times k$, com $N = k^2$
- **Malha 2-D**: grau ≤ 4, diâmetro $2(k - 1)$, bissecção k
  - **layout regular, roteamento simples — escolha usual em silício**
- **Toro 2-D**: grau 4, diâmetro $2\lfloor k/2 \rfloor$, bissecção $2k$
  - **enlaces de fechamento longos; a versão dobrada equaliza os comprimentos**
- **Anel**: grau 2, diâmetro $\lfloor N/2 \rfloor$, bissecção 2 — não escala
- **Árvore gorda**: diâmetro proporcional a $2 \cdot \log N$, banda maior próxima à raiz
- **Crossbar**: diâmetro 1, custo de área proporcional a N^2

Outline

- Introduction
- NoC Topology
- Switching strategies
- Routing algorithms
- Estudo de caso: NoC HERMES

Switching strategies

- Determine how data flows through routers in the network
- Define granularity of data transfer and applied switching technique
  - phit is a unit of data that is transferred on a link in a single cycle
  - typically, **phit size = flit size**

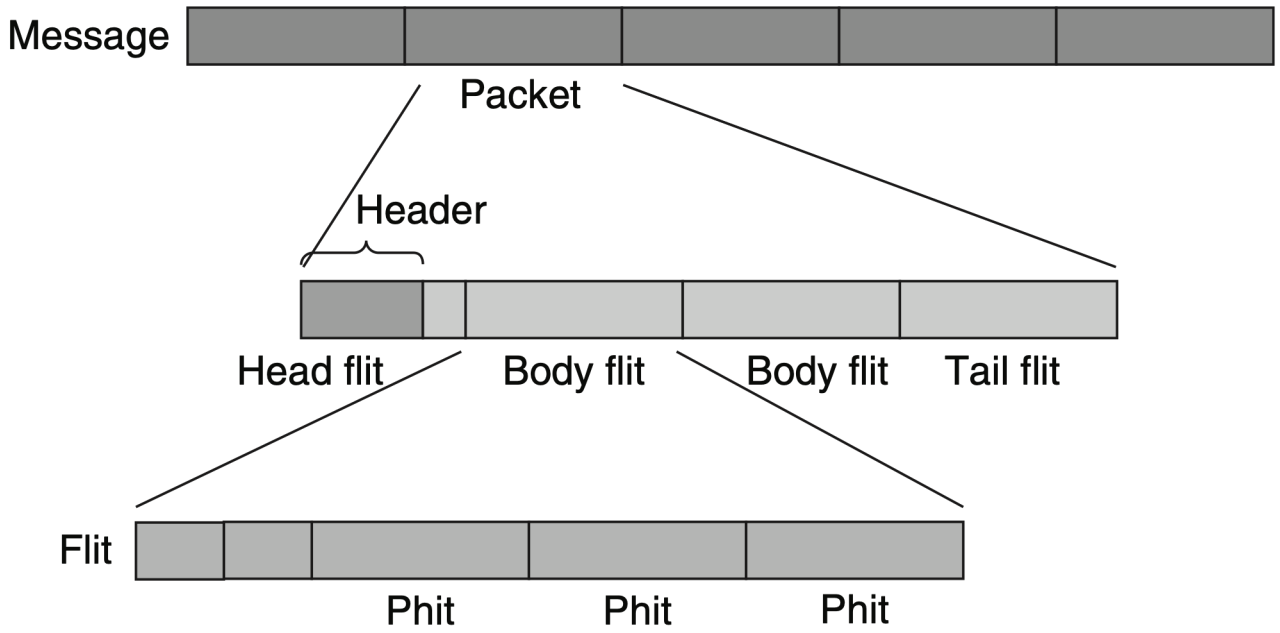


FIGURE 12.7

Structure of messages, packets, flits, and phits

Switching strategies

- Two main modes of transporting flits in a NoC are **circuit switching** and **packet switching**

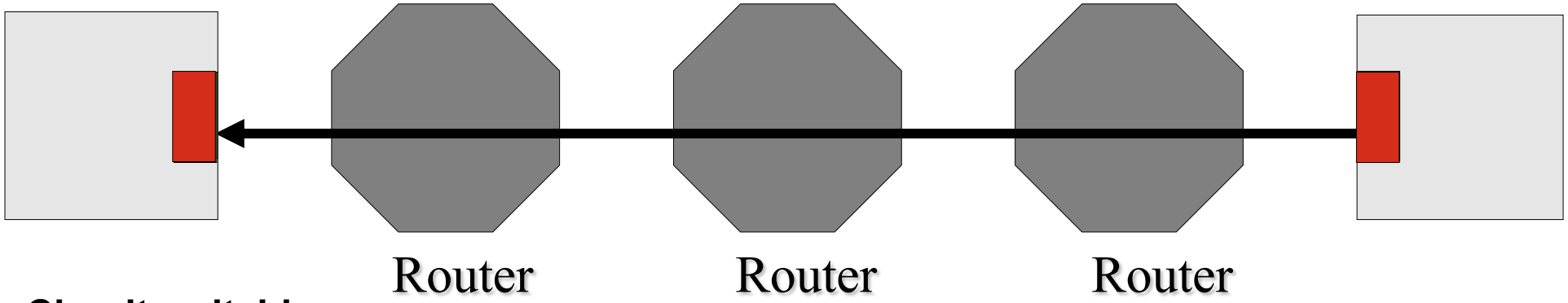
- **Circuit switching**
  - physical path between the source and the destination is reserved prior to the transmission of data
  - message header flit traverses the network from the source to the destination, reserving links along the way
  - **Advantage**: low latency transfers, once path is reserved
  - **Disadvantage**: pure circuit switching does not scale well with NoC size
    - several links are occupied for the duration of the transmitted data, even when no data is being transmitted
      - for instance in the setup and tear down phases

Switching strategies

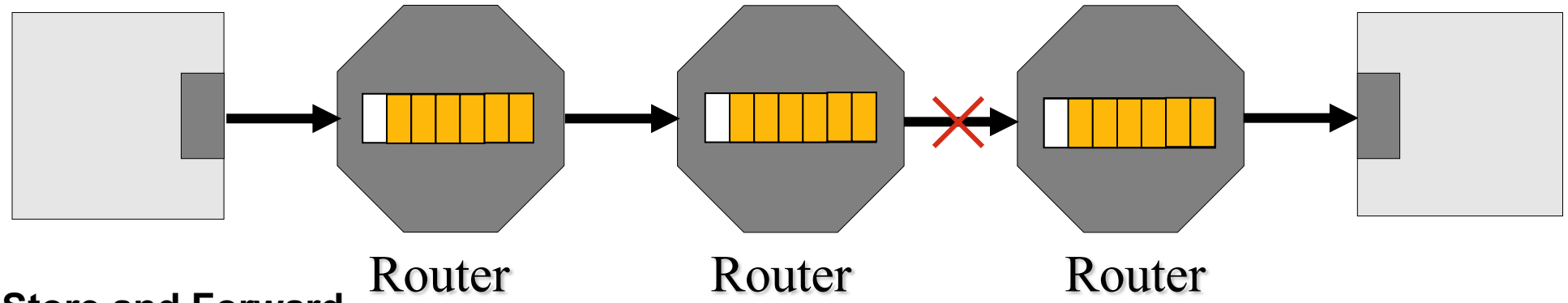
- **Packet Switching**
  - packets are transmitted from source and make their way independently to receiver
    - possibly along different routes and with different delays
  - zero start up time, followed by a variable delay due to contention in routers along packet path
  - QoS guarantees are harder to make in packet switching than in circuit switching
  - three main packet switching scheme variants

Packet switching

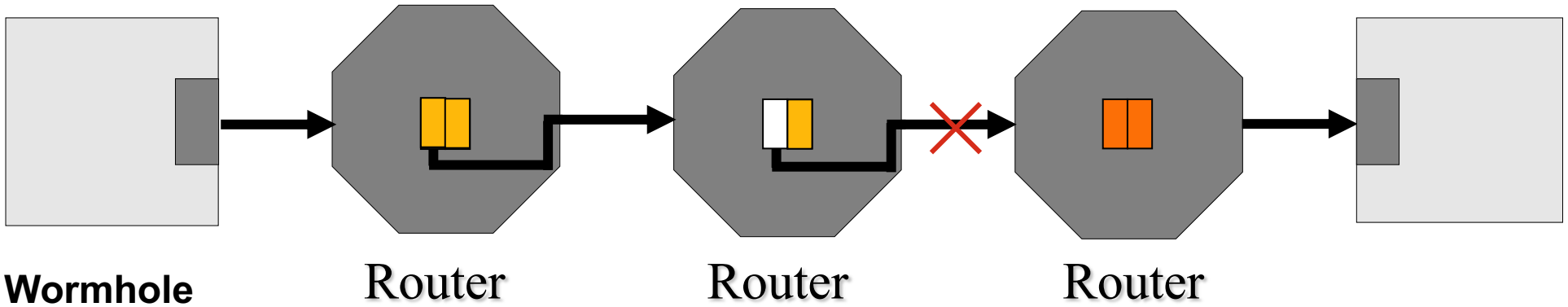
- **Store and forward (STF)**
  - packet is sent from one router to the next only if the receiving router has buffer space for entire packet
  - buffer size in the router is at least equal to the size of a packet
  - **Disadvantage**: excessive buffer requirements
- **Virtual-cut-through**
  - **reduces router latency** over SAF switching by forwarding first flit of a packet as soon as space for the entire packet is available in the next router
  - if no space is available in receiving buffer, no flits are sent, and the entire packet is buffered
  - **same buffering** requirements as SAF switching
- **Wormhole**
  - flit from a packet is forwarded to receiving router if space exists for that flit
  - parts of the packet can be distributed among two or more routers
  - buffer requirements are reduced to one flit, instead of an entire packet
  - more susceptible to deadlocks due to usage dependencies between links



Circuit switching



Store and Forward



Wormhole

Controle de fluxo

- Evita a perda de flits quando o buffer do roteador vizinho está cheio
- **Handshake (req/ack)**
  - protocolo simples; uma transferência a cada dois ou mais ciclos
- **Baseado em créditos**
  - o emissor mantém um contador das posições livres no receptor
  - sustenta um flit por ciclo; custo de fios adicionais de crédito
- **On/off (stall/go)**
  - sinal único de parada; exige margem de buffer para o atraso de propagação

- O chaveamento wormhole exige controle de fluxo por flit, e não por pacote

Canais virtuais

- Vários buffers lógicos compartilham o mesmo enlace físico
- **Motivações**
  - quebrar dependências cíclicas: roteamento adaptativo livre de deadlock
  - reduzir o bloqueio de cabeça de fila (head-of-line blocking)
  - separar classes de tráfego e sustentar prioridades e QoS
- **Custo**
  - multiplexação do enlace, árbitro por canal, mais buffers
  - aumento de área, potência e caminho crítico do roteador

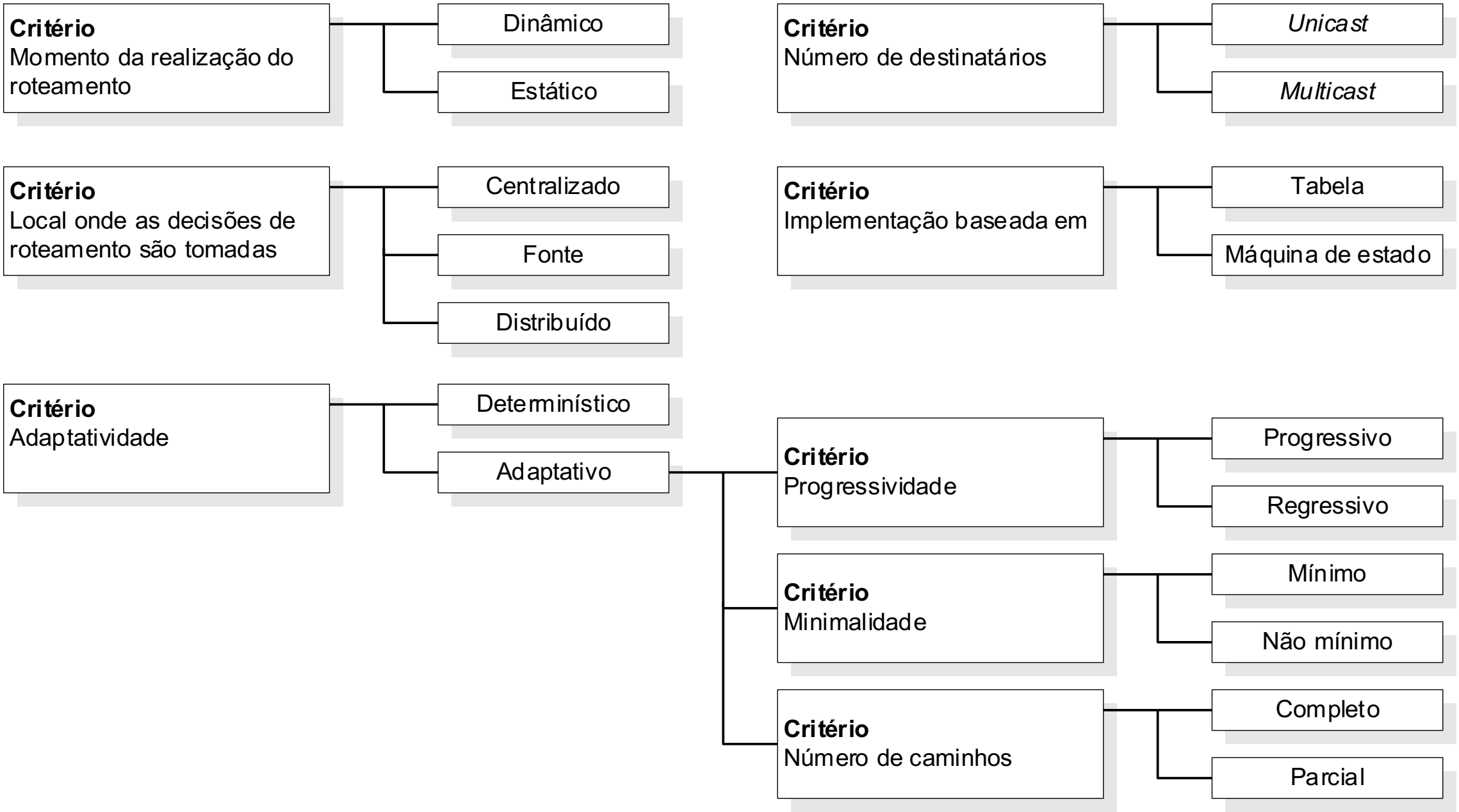
Outline

- Introduction
- NoC Topology
- Switching strategies
- Routing algorithms
- Estudo de caso: NoC HERMES

Routing algorithms

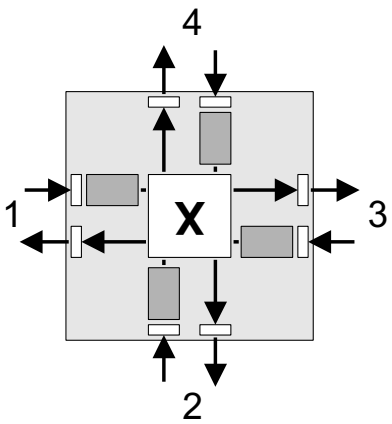
- Responsible for routing packets or circuits from the source to the destination
- Choice of a routing algorithm depends on trade-offs between several potentially conflicting metrics
  - minimizing power required for routing
  - minimizing logic and routing tables to achieve a lower area footprint
  - increasing performance by reducing delay and maximizing traffic utilization of the network
  - improving robustness to better adapt to changing traffic needs
- Routing schemes can be classified into several categories
  - static or dynamic routing
  - distributed or source routing
  - minimal or non-minimal routing

Roteamento: Classificações

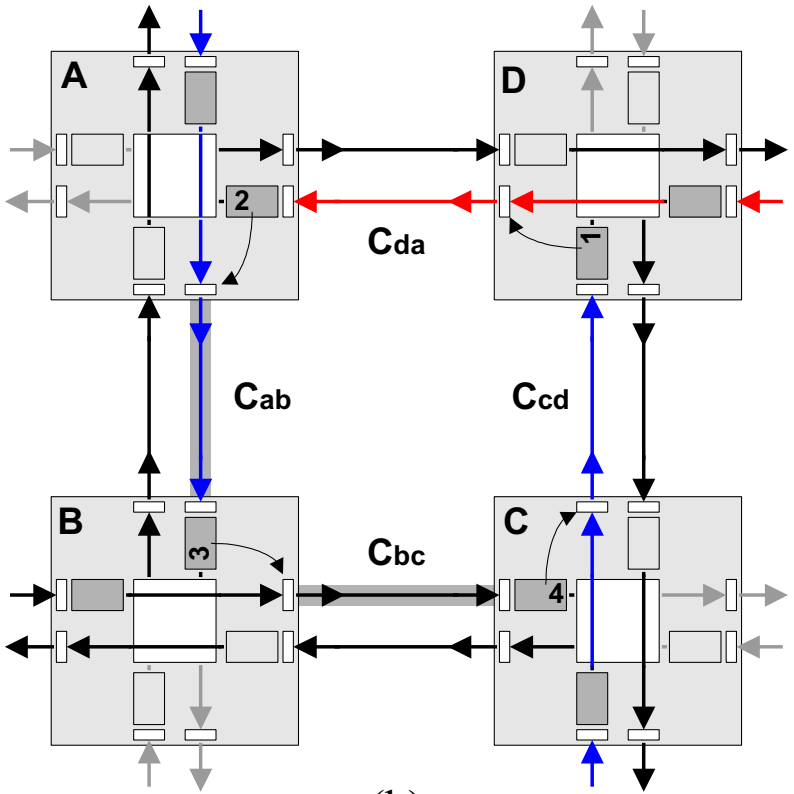


Problemas: Livelock e Deadlock

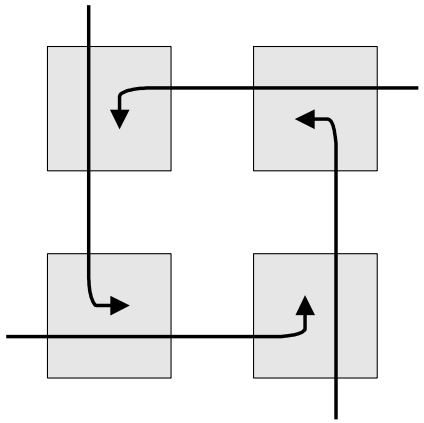
- Livelock
  - Quando uma mensagem trafega permanentemente pela rede sem chegar ao seu destino
- Deadlock
  - Ocorre quando existe uma dependência cíclica de recursos na rede e as mensagens são paralisadas



(a)

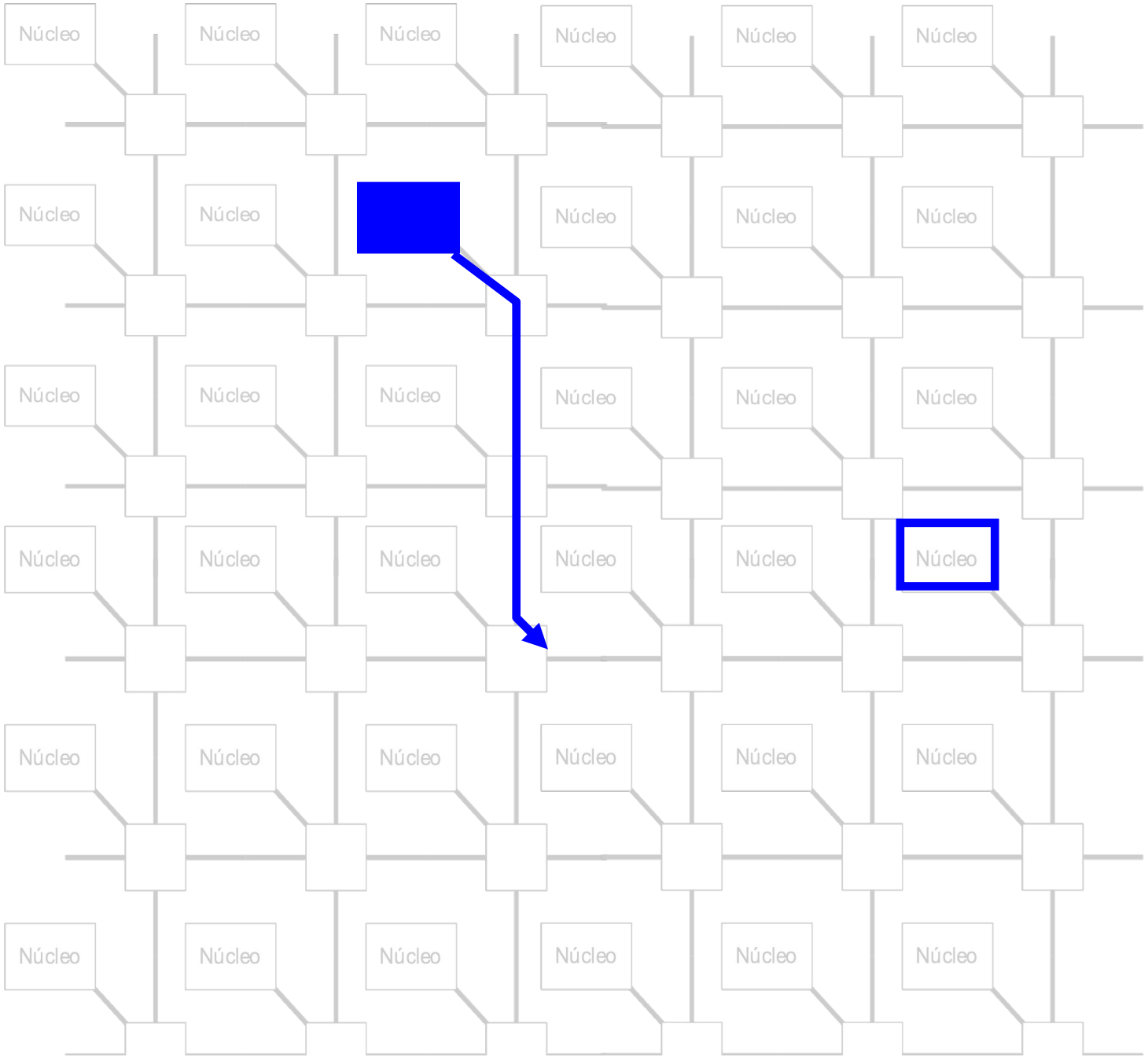


(b)

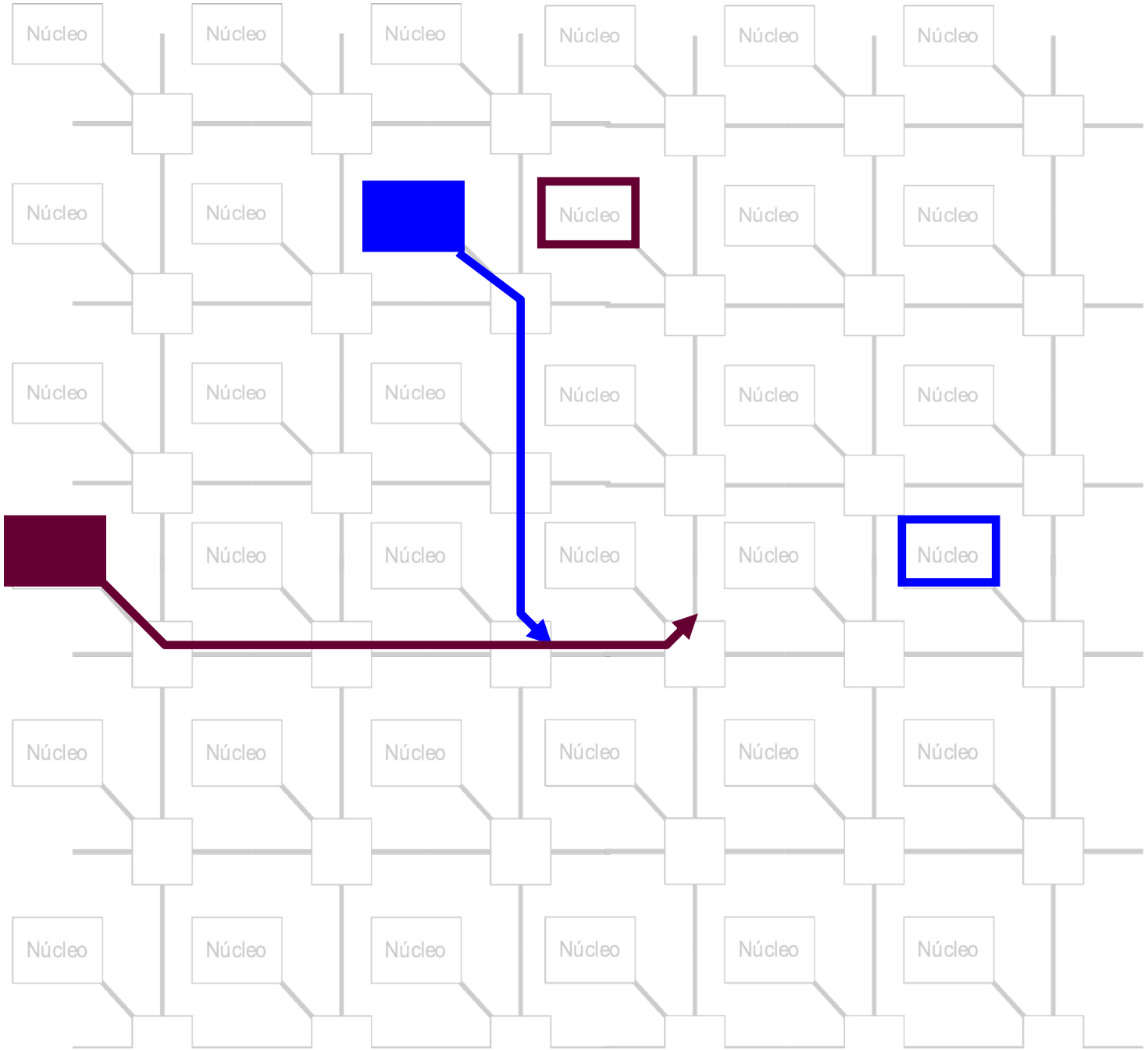


(c)

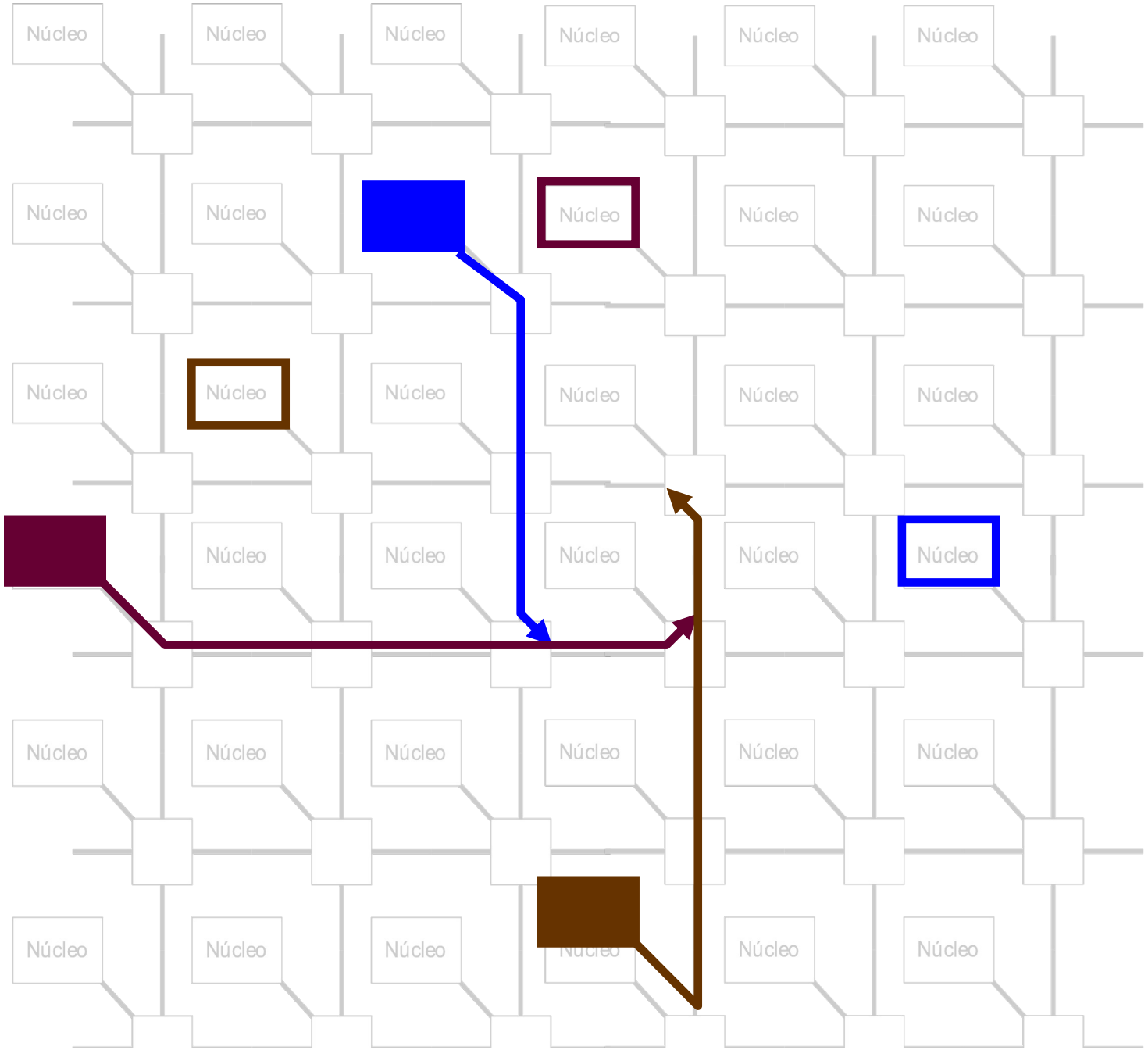
Exemplo de deadlock



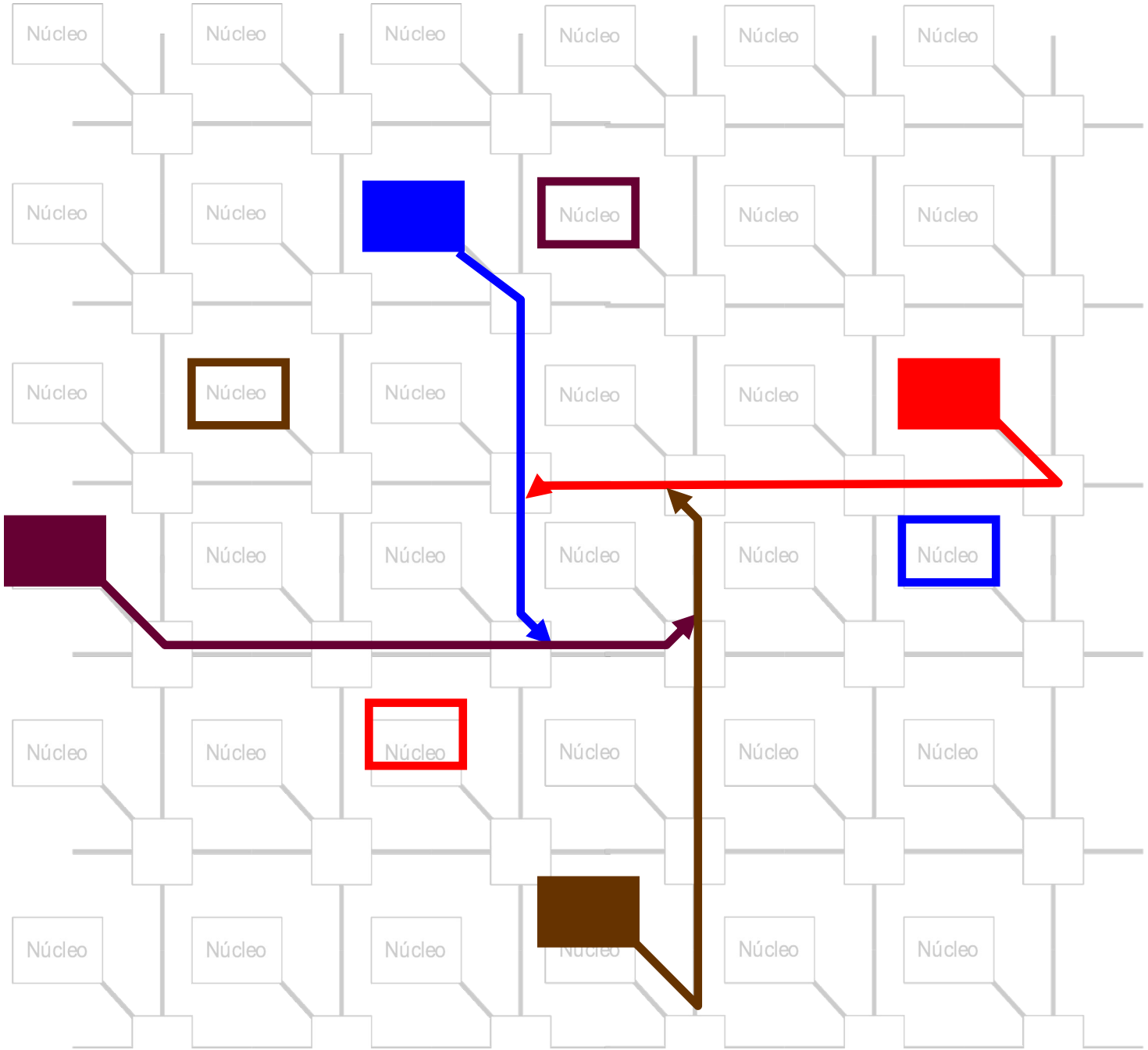
Exemplo de deadlock



Exemplo de deadlock

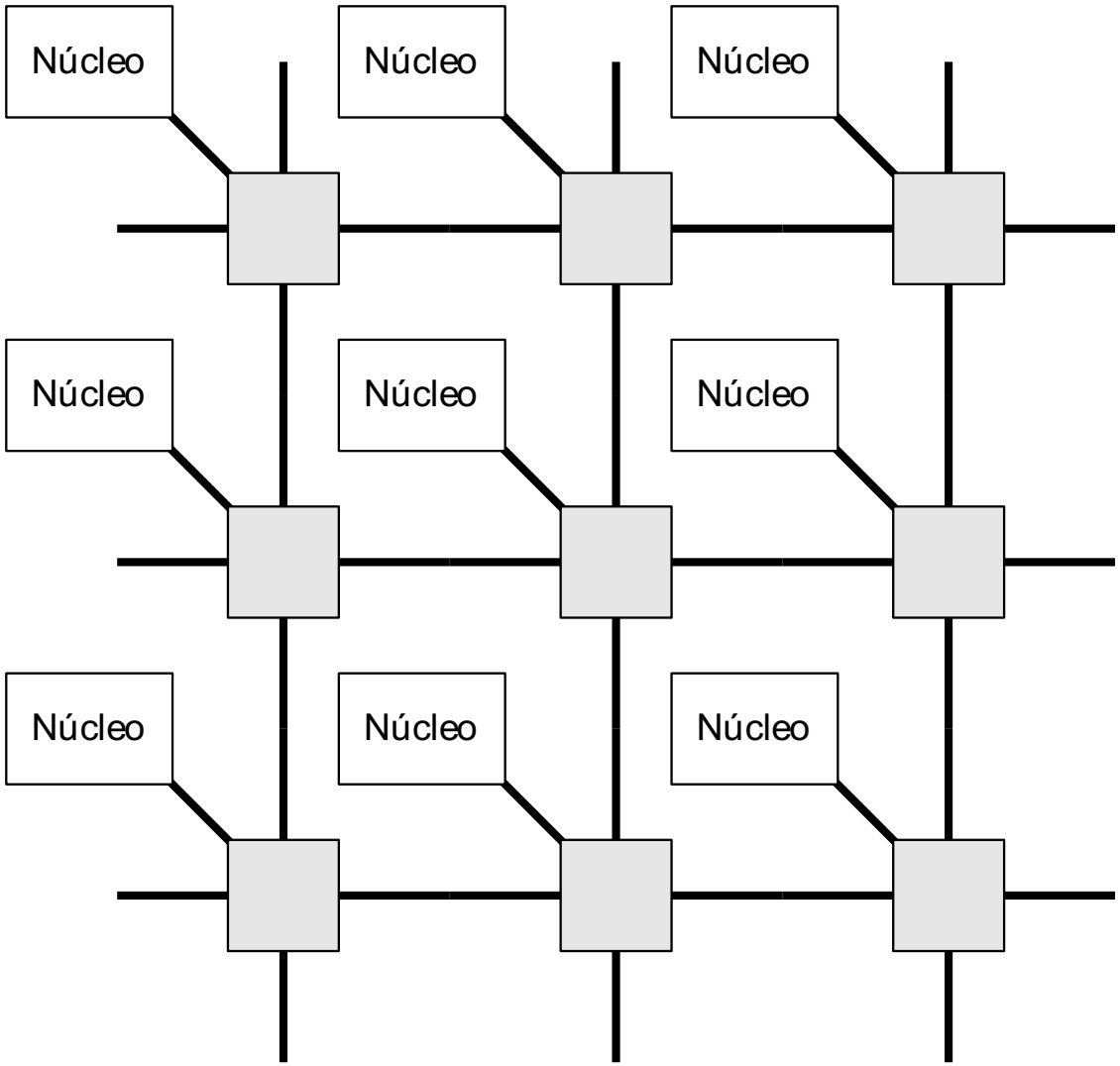


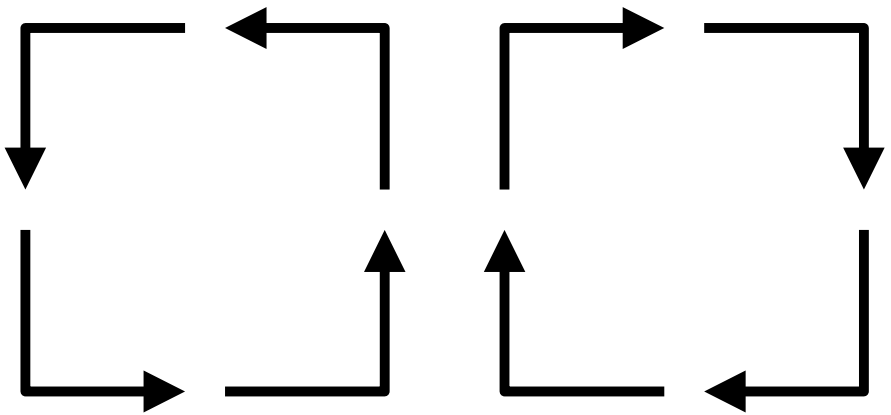
Exemplo de deadlock



Roteamento: Evitando o deadlock

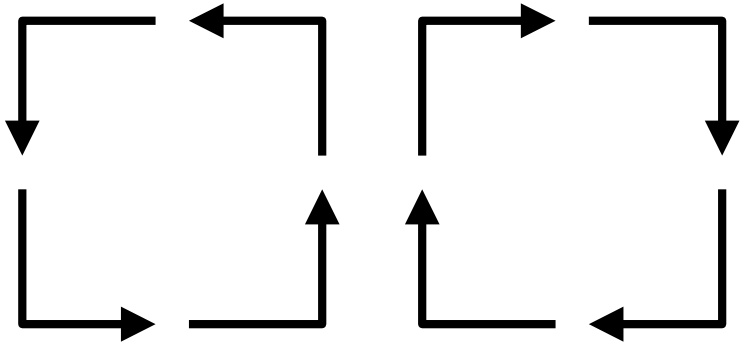
Deve-se evitar o surgimento de **ciclos** na rede





Ciclos na malha 2-D

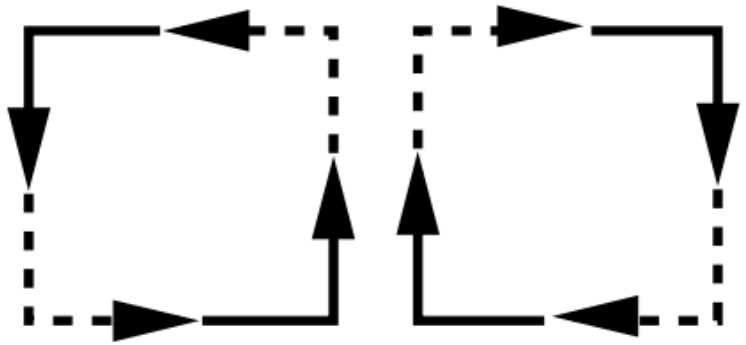
Roteamento: *turn model*

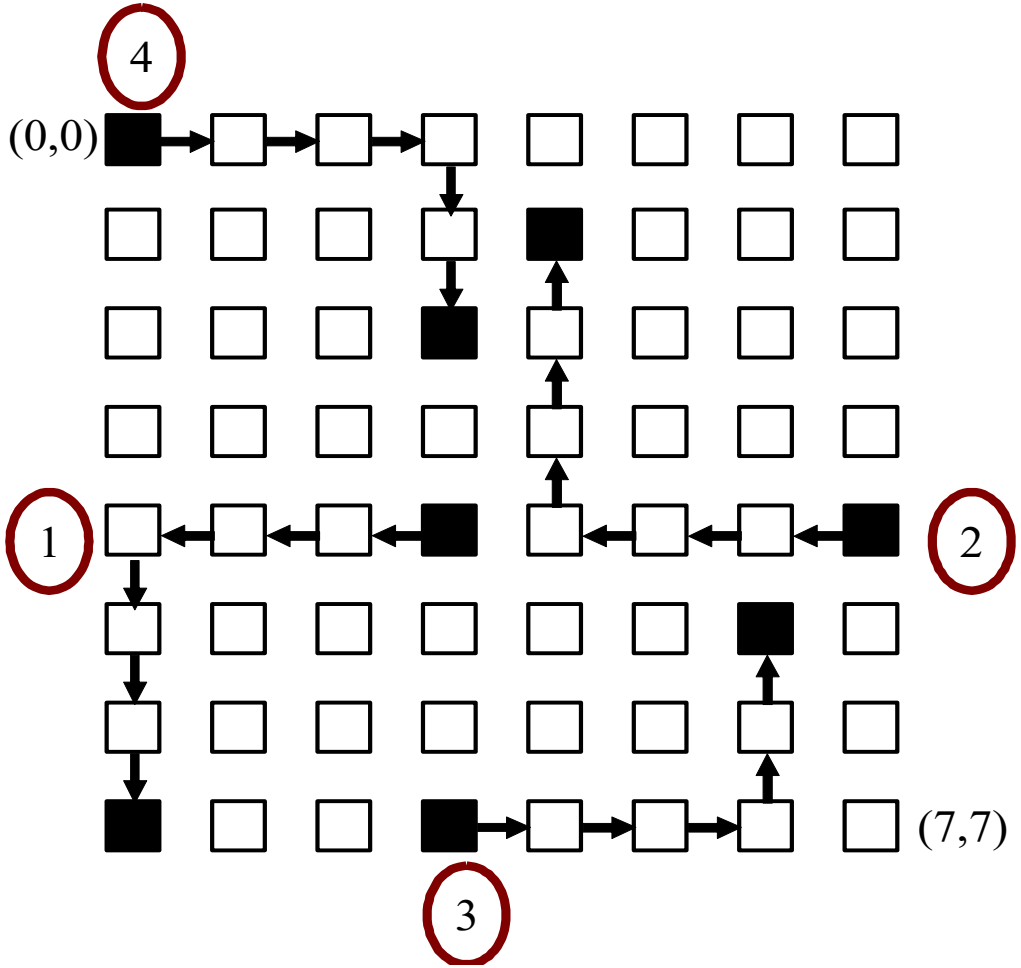


- Turn-based model to avoid deadlock
- Possible turns = {NW, NE, SW, SE, WN, WS, EN, ES}
- **Disallow >= 2 turns**
- XY routing only allows turns from X to Y {EN, ES, WN, WS}
- West-first routing prohibits turns to west {NW, SW}
  - Offers full adaptiveness to paths that route east
  - Not fair to all paths

Roteamento: *XY*

- Deterministic
- All messages from Src to Dest will traverse the **same** path
- AKA: **Dimension Order Routing (DOR)**
  - Message traverses network dimension by dimension

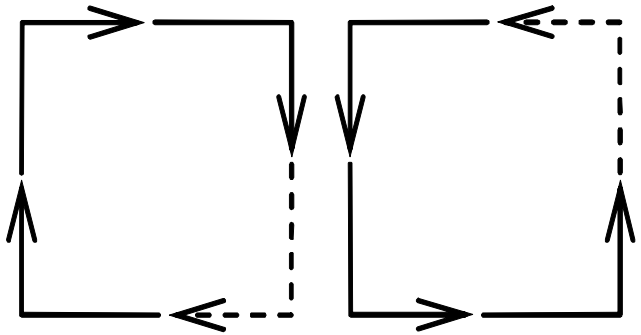


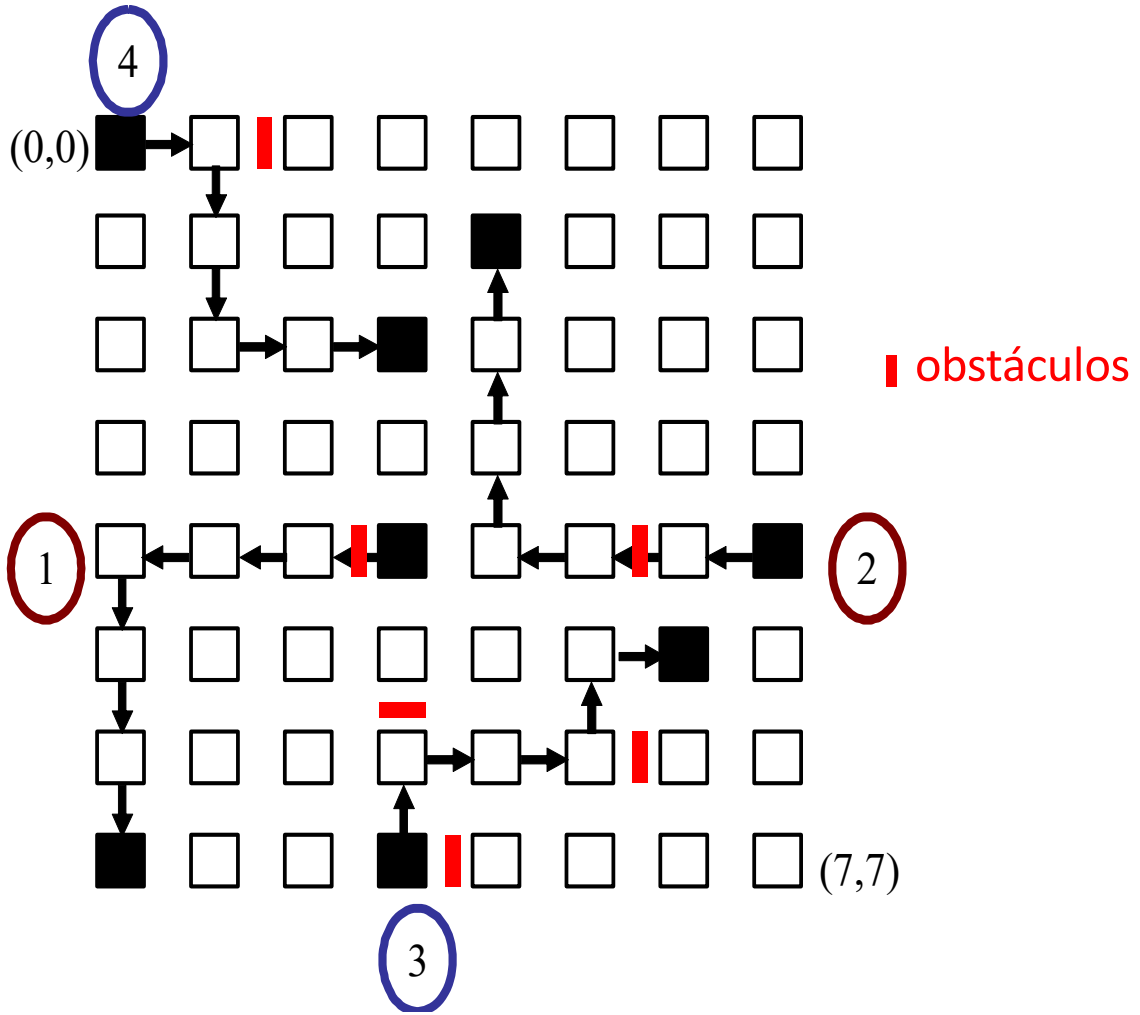


- Cons:
  - Eliminates any path diversity provided by topology
  - Poor load balancing
- Pros:
  - Simple and inexpensive to implement
  - Deadlock free

Roteamento: *west-first*

- The prohibited turns are the two to the West
  - if XT ≤ XS, packets are routed deterministically, as in the XY algorithm, (paths 1 and 2)
  - if XT > XS packets can be routed adaptively in East, North or South directions (paths 3 and 4)





(1) e (2) – caminhos determinísticos

(3) e (4) – caminhos adaptativos

Roteamento: *west-first não mínimo*

- Route a packet first west if necessary, and then adaptively south , east, and north.
- Both minimal and non-minimal paths are shown.
- [Dally and Seitz] proof show that a routing algorithm is deadlock free if the channels in the interconnection network can be numbered so that the algorithm routes every packet along channels with strictly decreasing numbers.

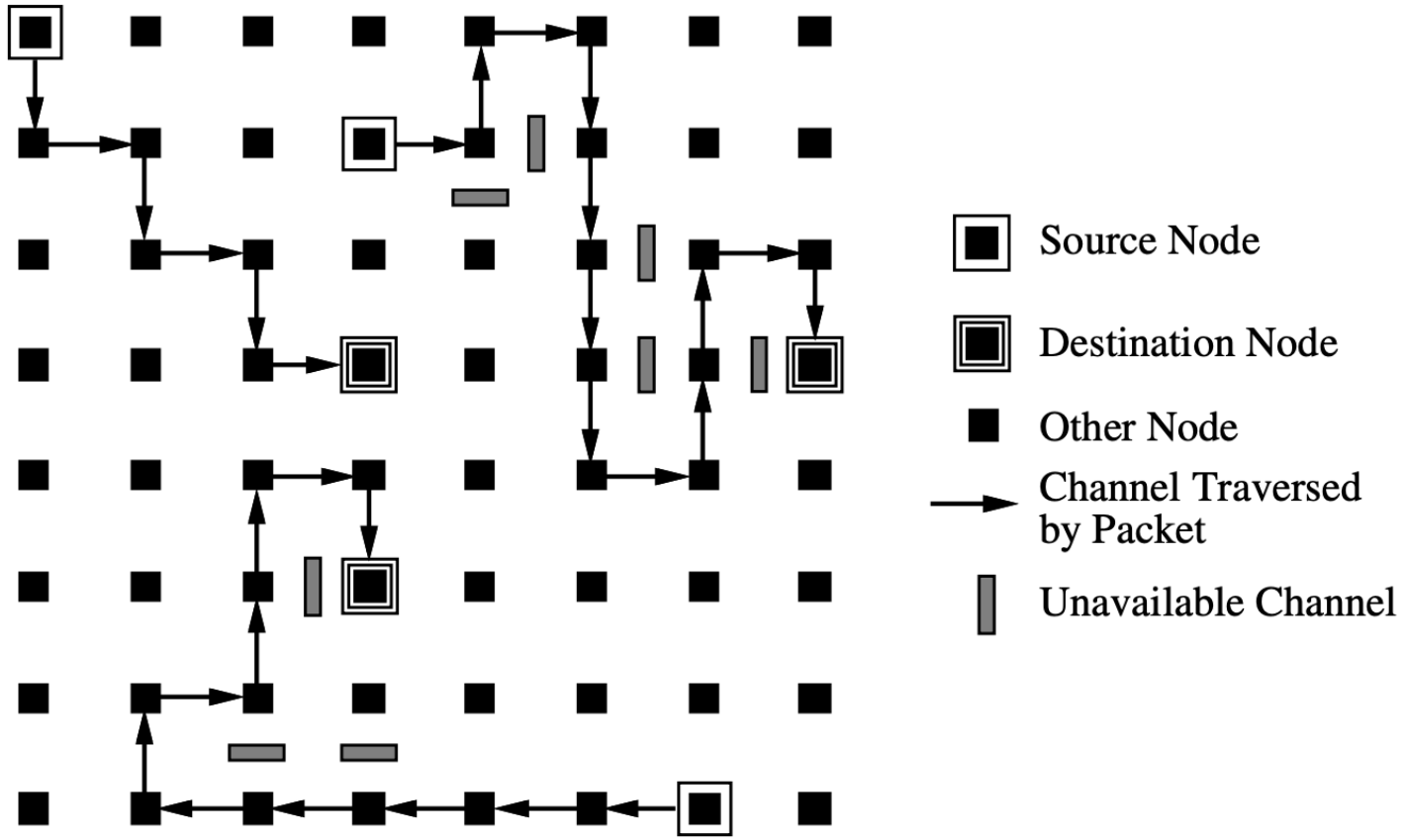


Figure 4.11 Examples of west-first routing in an 8 × 8 2-D mesh.

Métricas de desempenho

- **Latência** — ciclos entre a injeção do primeiro flit e a recepção do último
  - componentes: roteamento e arbitragem, travessia do crossbar e do enlace, contenção
  - sem contenção: **L ≈ H ·t_roteador + tamanho do pacote** (wormhole)
- **Vazão** — flits entregues por ciclo e por roteador
- **Carga de saturação** — taxa de injeção a partir da qual a latência cresce sem limite
- **Custo** — área, potência e frequência de operação
  - separar a contribuição de roteadores e de enlaces
- Comparações só têm sentido com topologia, tamanho de pacote e tráfego declarados

Padrões de tráfego para avaliação

- **Sintéticos**
  - uniforme: destino sorteado com igual probabilidade — caso otimista
  - transpose, bit-reversal e shuffle: exercitam a bissecção da rede
  - complemento: pior caso para a malha 2-D com roteamento XY
  - hotspot: fração do tráfego dirigida a um nó (memória ou acelerador)
- **De aplicação**
  - traços extraídos de benchmarks ou de grafos de tarefas mapeados na rede
- **Parâmetros do experimento**
  - taxa de injeção, tamanho de pacote e distribuição temporal (rajadas)

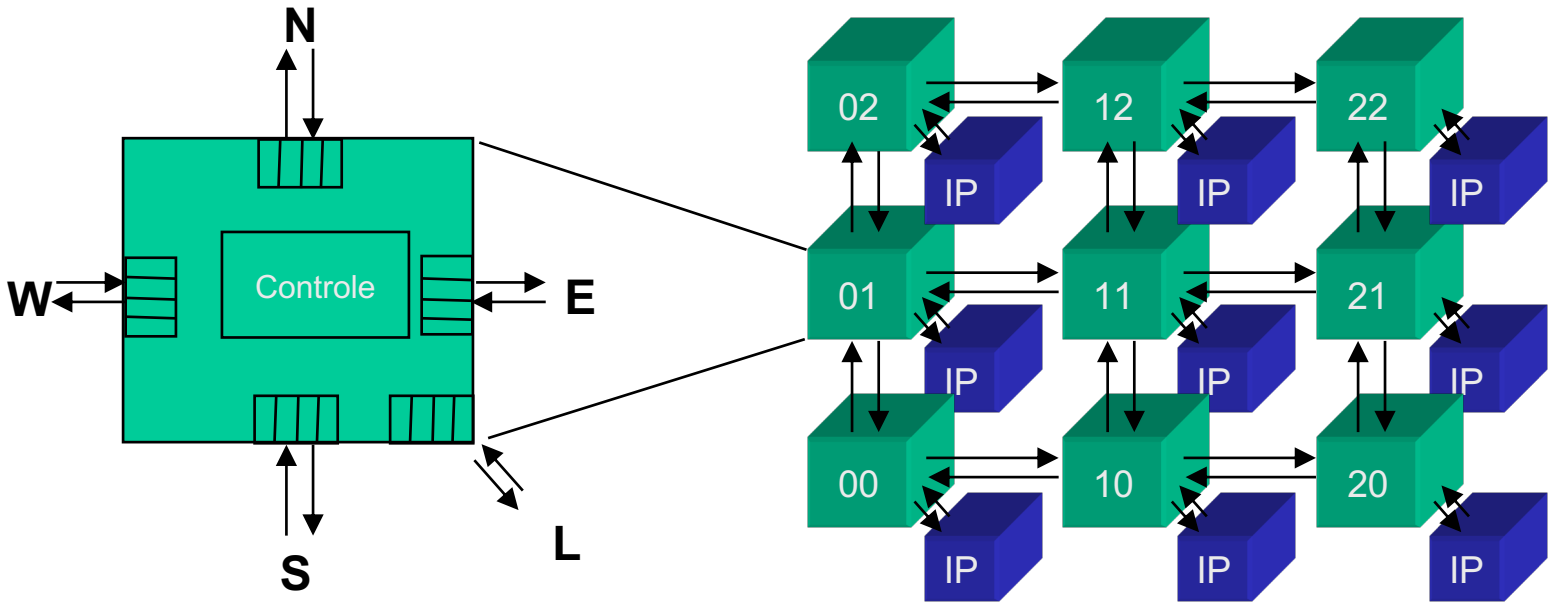
HERMES

TÍTULO	CITADO POR	ANO
HERMES: an infrastructure for low area overhead packet-switching networks on chip F Moraes, N Calazans, A Mello, L Möller, L Ost Integration 38 (1), 69-93	831	2004

NoC Hermes

Características

- Todo IP é conectado a um roteador (**rede direta**)
- Endereçamento **XY**
- Até 5 portas unidirecionais por roteador (N / S / W / E + Local)
- Buffers de entrada



Árbitro e roteamento

Características

- Apenas o primeiro *flit é* tratado (endereço de destino)
- Arbitragem **rotativa** das prioridades de acesso às portas de saída (***round robin***)

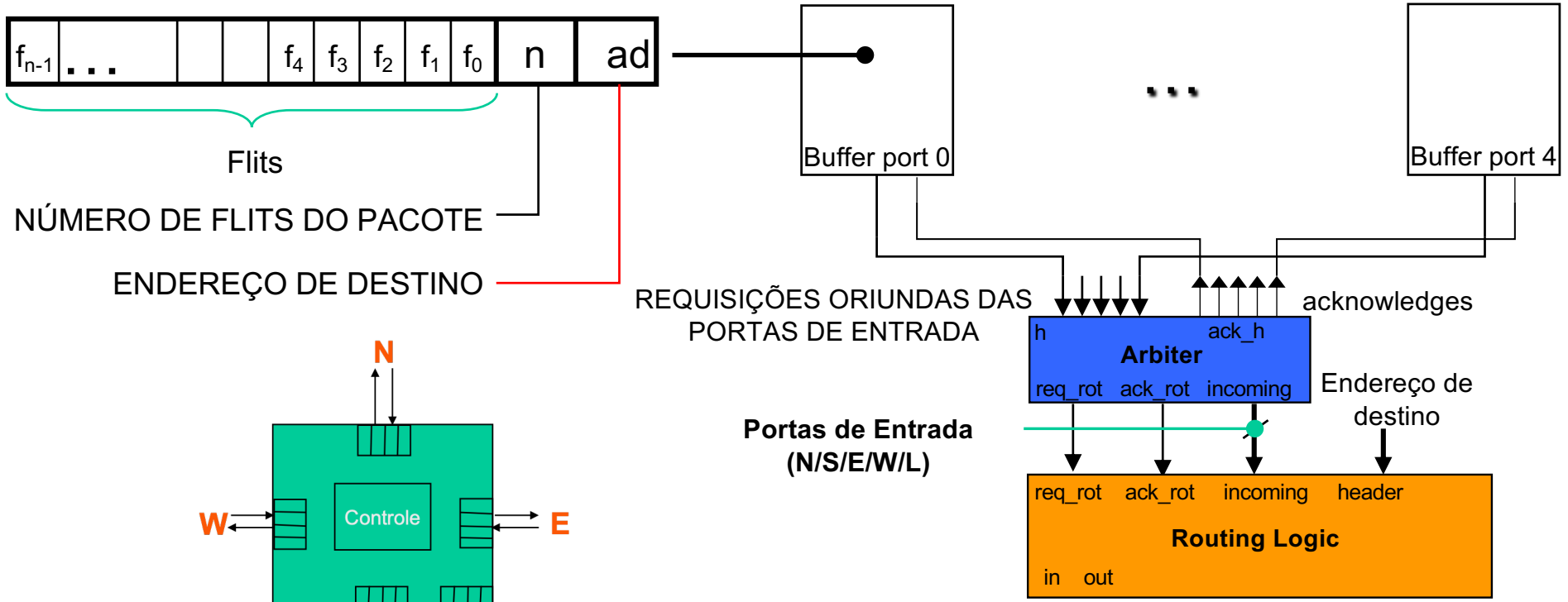
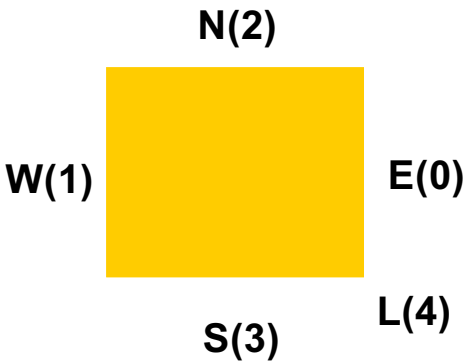


Tabela de roteamento

- Tabela de conexão
  - Quando uma conexão é estabelecida o caminho entre a porta de entrada e saída é armazenado
  - Três vetores são utilizados: IN, OUT, FREE

	0 – E	1 – W	2 – N	3 – S	4 – L
Free	1	1	1	1	1
In					
Out					



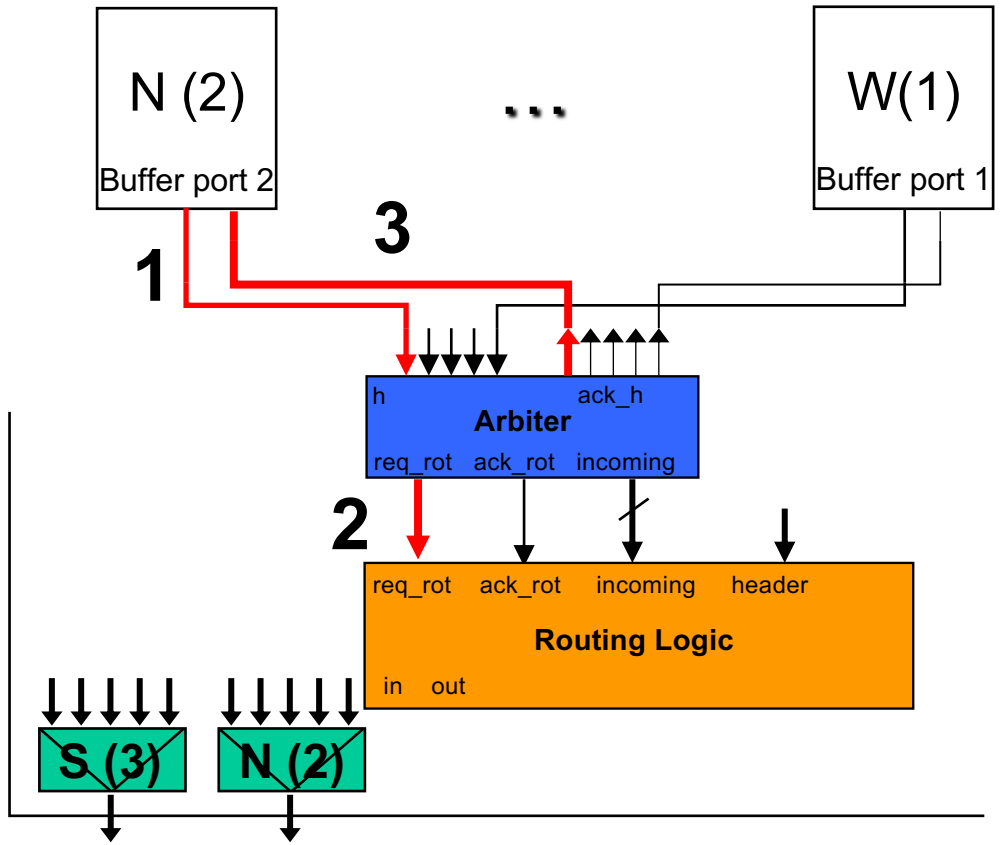
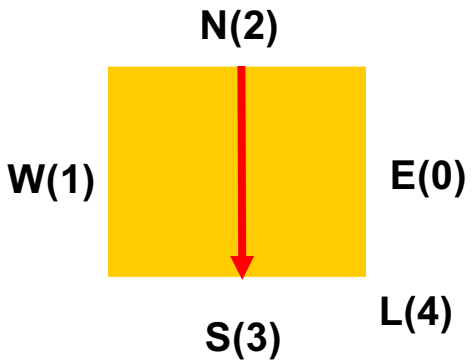


Tabela de roteamento

- Tabela de conexão
  - Quando uma conexão é estabelecida o caminho entre a porta de entrada e saída é armazenado
  - Tres vetores são utilizados: IN, OUT, FREE

	0 – E	1 – O	2 – N	3 – S	4 – L
Free	1	1	1	0	1
In			3		
Out				2	



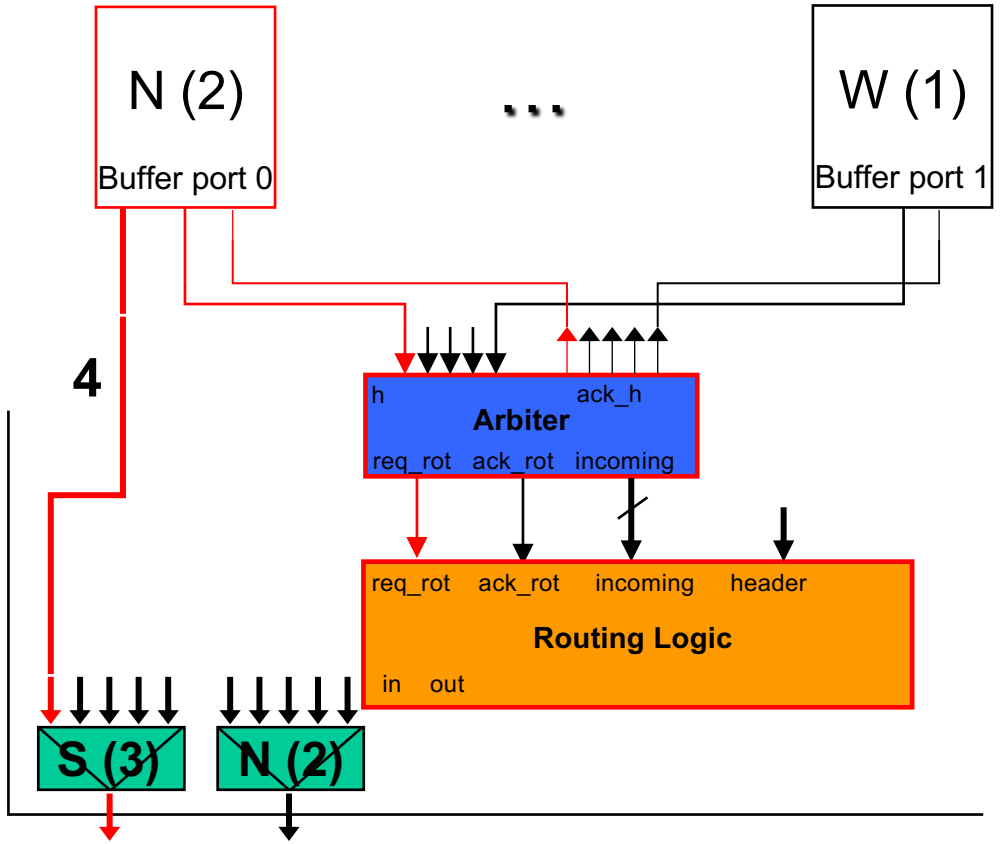
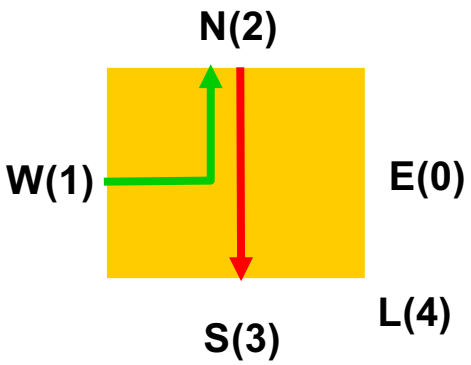
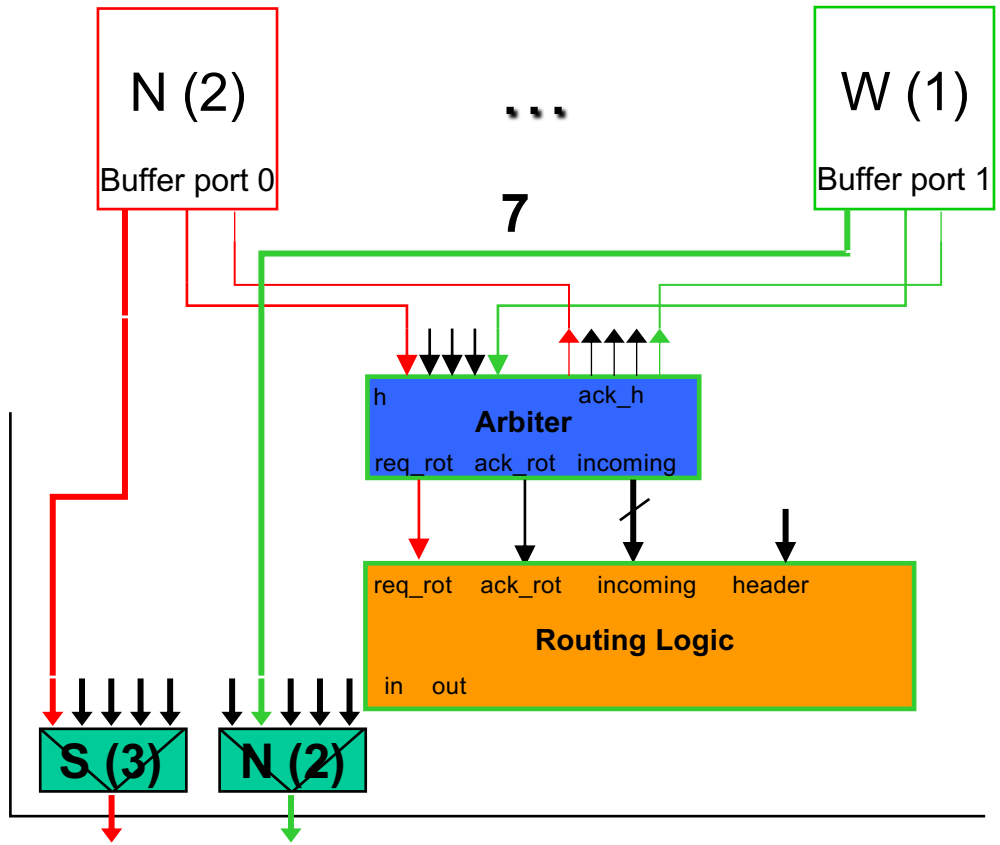


Tabela de roteamento

- Tabela de conexão
  - Quando uma conexão é estabelecida o caminho entre a porta de entrada e saída é armazenado
  - Tres vetores são utilizados: IN, OUT, FREE

	0 – E	1 – W	2 – N	3 – S	4 – L
Free	1	1	0	0	1
In		2	3		
Out			1	2	



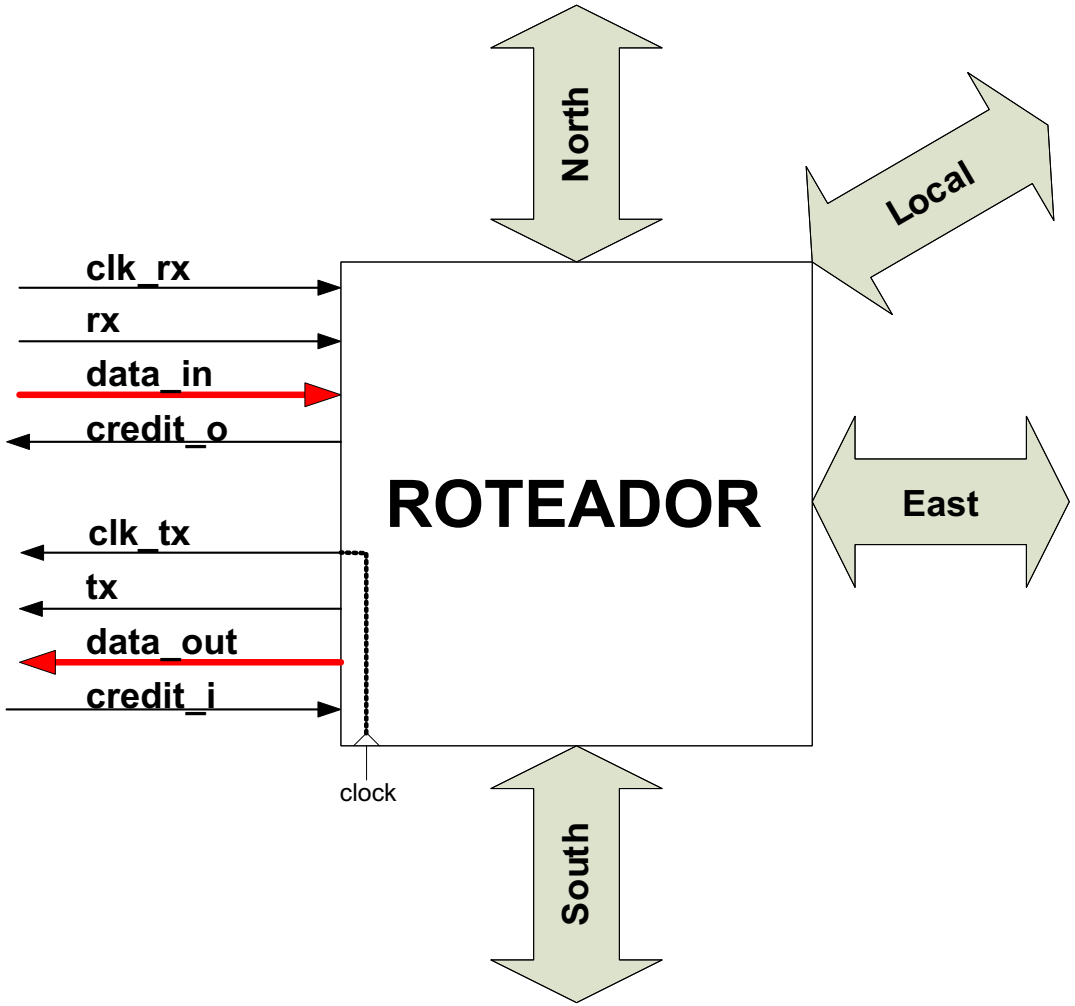


Roteador – routerCC

```
module HermesRouter
import HermesPkg::*;
#(
   parameter logic [15:0] ADDRESS = 0,
   parameter BUFFER_SIZE = 8, /* Power of 2 */
   parameter FLIT_SIZE = 32 /* Minimum: 20 */
)
(
input logic clk_i,
input logic rst_ni,


input logic rx_i [(HERMES_NPORT - 1):0],
input logic eop_i [(HERMES_NPORT - 1):0],
output logic credit_o [(HERMES_NPORT - 1):0],
input logic [(FLIT_SIZE - 1):0] data_i [(HERMES_NPORT - 1):0],


output logic tx_o [(HERMES_NPORT - 1):0],
output logic eop_o [(HERMES_NPORT - 1):0],
input logic credit_i [(HERMES_NPORT - 1):0],
output logic [(FLIT_SIZE - 1):0] data_o [(HERMES_NPORT - 1):0]
);
```

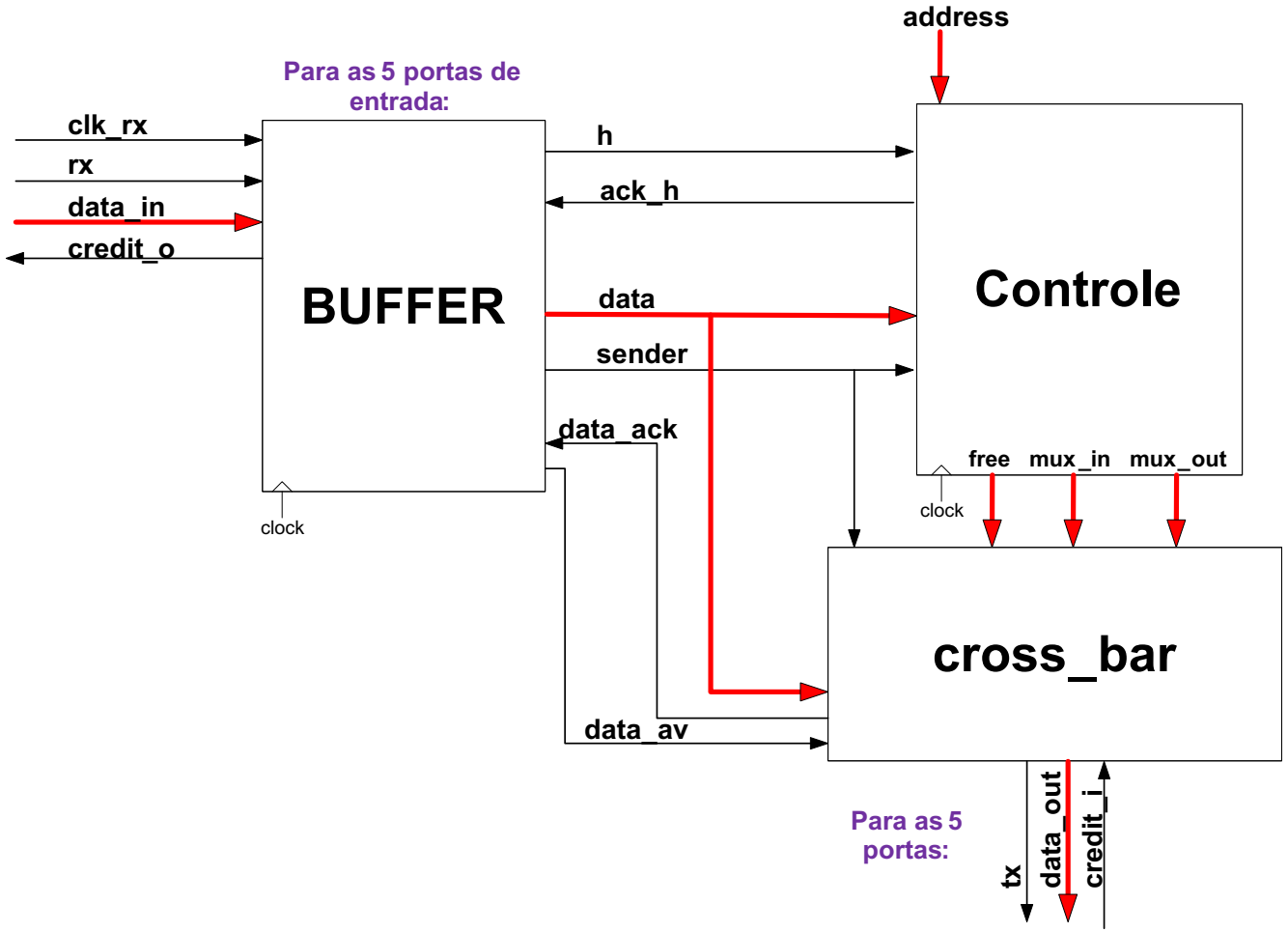


Roteador – routerCC

```
genvar port;
generate
for (port = 0; port < HERMES_NPORT; port++) begin
buffer
(…)
);
end
endgenerate;

HermesCrossbar #(
.FLIT_SIZE(FLIT_SIZE)
)
crossbar
(…
);


HermesSwitch #(
.ADDRESS(ADDRESS),
.FLIT_SIZE(FLIT_SIZE)
)
switch
( …);
```



Control

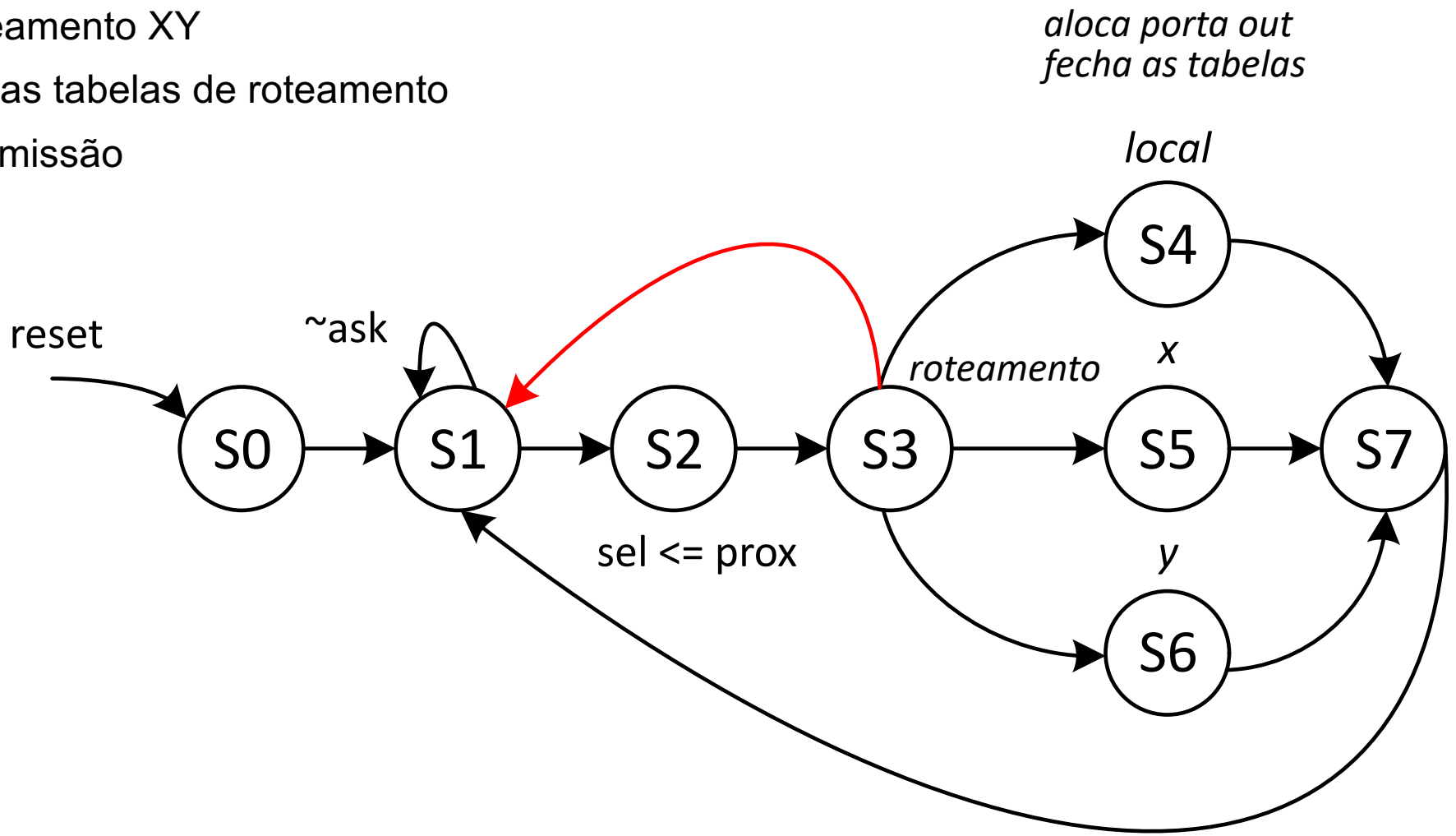
→ S1 arbitra

→ S2 armazena resultado do arbitragem

→ S3 realiza o roteamento XY

→ S4/S5/S6 aloca as tabelas de roteamento

→ S7 inicia a transmissão



Crossbar

Multiplexadores

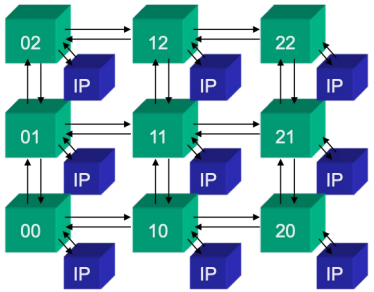
tabelas definidas na FSM do control

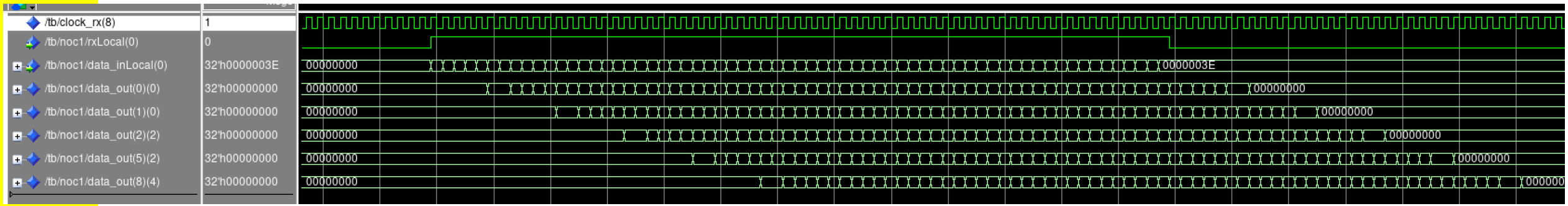
controle de fluxo: baseado no sinal "*credit*"

```
always_comb begin
     for (int i = 0; i < HERMES_NPORT; i++) begin
          ack_o[i] = data_av_i[i] ? credit_i[outport_i[i]] : 1'b0;
          if (!free_i[i]) begin
                tx_o[i] = data_av_i[inport_i[i]];
                eop_o[i] = eop_i [inport_i[i]];
                data_o[i] = data_i [inport_i[i]];
          end
          else begin
                tx_o[i] = 1'b0;
                eop_o[i] = 1'b0;
                data_o[i] = '0;
          end
     end
end
```

Transmissão em uma 3x3 e latência

00→ 32: 5 arbitragens & roteamento: 25 ciclos de clock

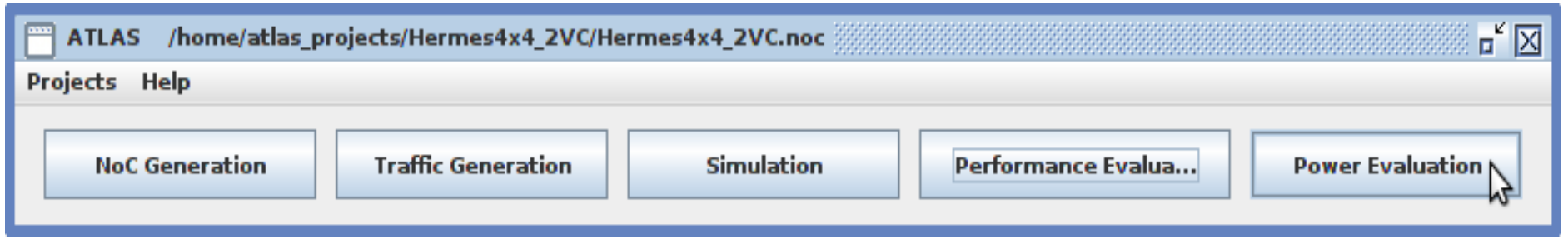


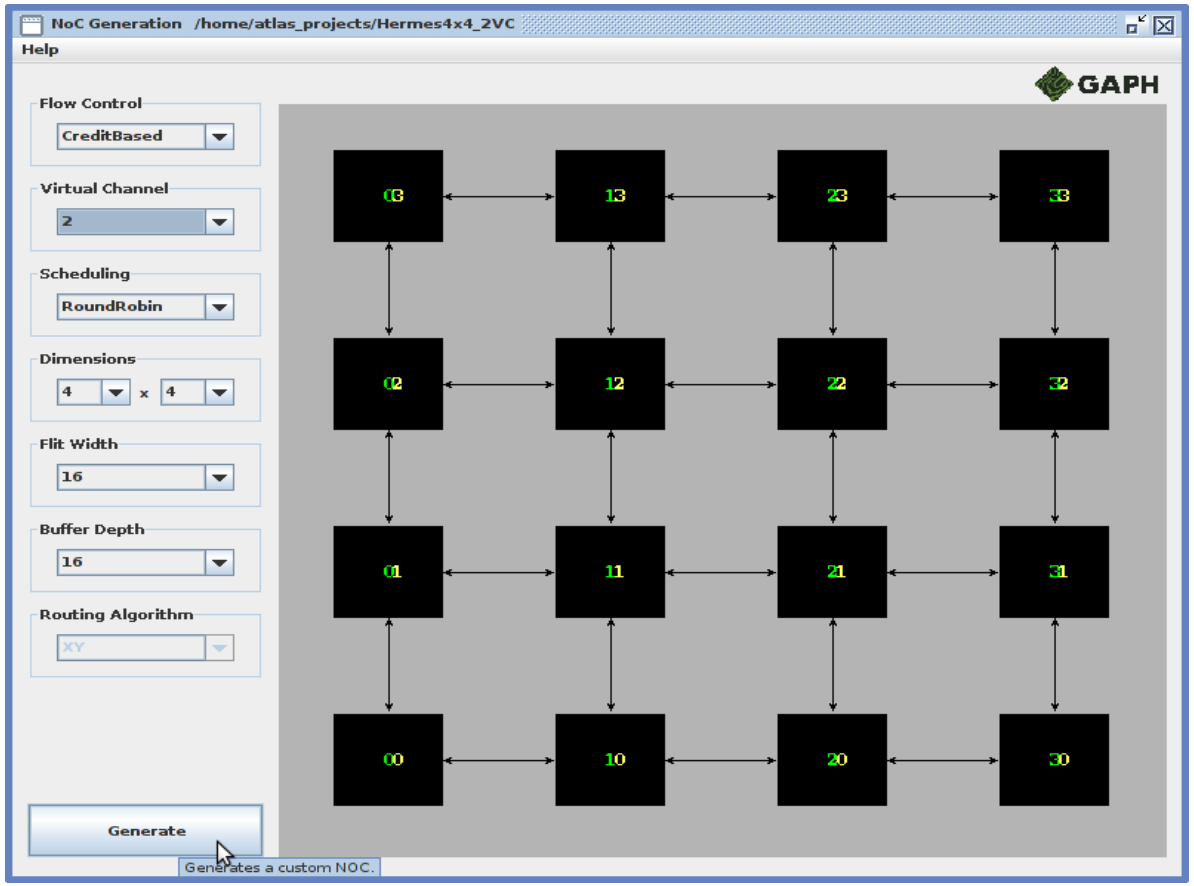


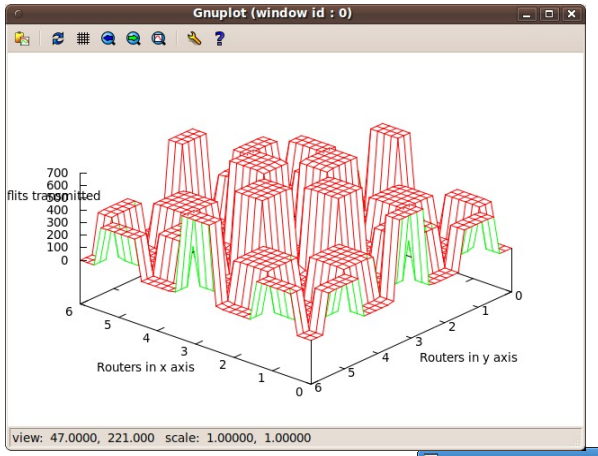
HERMES versions

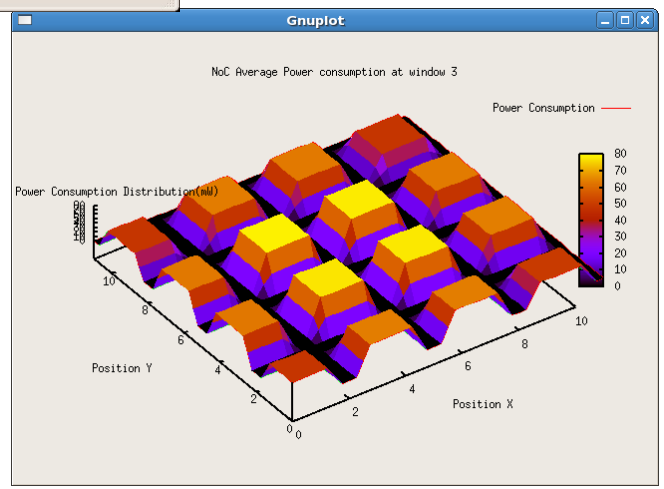
Parameters	Hermes [2][3]	Hermes TB	Hermes TU	Hermes SR	Hermes CRC
Topology	2D Mesh	2D Torus	1D Torus	2D Mesh	2D Mesh
Virtual Channels	1, 2 or 4	1	2	1 or 4	2
Flit Width	8, 16, 32 or 64			16	16
Buffer Depth	4, 8, 16 or 32				
Routing Algorithm	XY or West-first	West-first non miminal	Unidirec-tional XY	XY	XY
Scheduling Algorithm	Round Robin or Priority	Round Robin	Round Robin	Age Based	Round Robin
CRC	No	No	No	No	Yes
QoS	Yes	No	No	Yes	No

ATLAS


ATLAS /home/atlas_projects/Hermes4x4_2VC/Hermes4x4_2VC.noc
Projects Help
NoC Generation
Traffic Generation
Simulation
Performance Evalua...
Power Evaluation


NoC Generation /home/atlas_projects/Hermes4x4_2VC
Help
GAPH
Flow Control
CreditBased
Virtual Channel
2
Scheduling
RoundRobin
Dimensions
4 x 4
Flit Width
16
Buffer Depth
16
Routing Algorithm
XY
Generate
Generates a custom NOC.
03 13 23 33
02 12 22 32
01 11 21 31
00 10 20 30


Gnuplot (window id : 0)
flits transmitted
Routers in x axis
Routers in y axis
view: 47.0000, 221.000 scale: 1.00000, 1.00000


Gnuplot
NoC Average Power consumption at window 3
Power Consumption
Power Consumption Distribution(mW)
Position Y
Position X

Qualidade de serviço

- **Best-effort (BE)**
  - sem garantias de latência ou banda; uso eficiente dos recursos
  - adequado a tráfego elástico, como acessos a cache
- **Serviço garantido (GS)**
  - reserva de banda e limite de latência por conexão
  - circuitos virtuais com divisão de tempo (TDM), como na Æthereal
- **Híbrido**: tráfego BE ocupa a banda não reservada
- **Mecanismos**: prioridades por canal virtual, controle de taxa na interface de rede, escalonamento no árbitro

Ferramentas de avaliação

- **Simuladores de rede**
  - BookSim, Noxim e Garnet/gem5: exploração de topologia, roteamento e canais virtuais
  - resultados rápidos, mas dependentes do modelo de roteador adotado
- **Descrição RTL**
  - Hermes e Memphis em VHDL/SystemVerilog; simulação com Questa ou Verilator
  - medidas obtidas com o mesmo hardware que será sintetizado
- **Síntese e prototipação**
  - área, frequência e potência em ASIC ou FPGA

Tendências

- **Aceleradores de IA**: malhas de grande porte, tráfego de difusão e redução, NoC estendida ao encapsulamento
- **Integração 3D e chiplets**: NoCs hierárquicas, TSVs e enlaces entre pastilhas (interposer, UCIe)
- **Enlaces fotônicos e sem fio**: alta banda e latência quase independente da distância; custo de integração ainda restritivo
- **Segurança**: isolamento de tráfego, zonas seguras e canais laterais decorrentes do compartilhamento de roteadores
- **Tolerância a falhas**: roteamento reconfigurável e redundância de enlaces

Resumo

- A NoC troca fios globais por uma rede de pacotes: paralelismo e escalabilidade
- A topologia fixa os limites de latência e de vazão; a malha 2-D domina em silício
- Wormhole reduz os buffers; o controle por créditos sustenta a vazão
- XY é simples e livre de deadlock; adaptatividade exige turn model ou canais virtuais
- Canais virtuais e QoS separam classes de tráfego, ao custo de área e potência
- A Hermes concretiza esses conceitos em hardware sintetizável