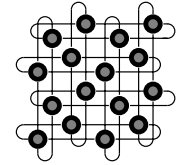


Switching Techniques, Adaptive Routing and Deadlock Handling in Interconnection Networks

José Duato

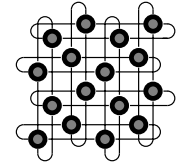
Dept. de Ingeniería de Sistemas, Computadores y Automática
Universidad Politécnica de Valencia, Spain



Adaptive Routing and Deadlock Handling in Interconnection Networks

José Duato

Dept. de Ingeniería de Sistemas, Computadores y Automática
Universidad Politécnica de Valencia, Spain



Outline

Introduction

Switching techniques

Optimized switching techniques

Deadlock handling

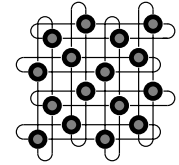
Theory of deadlock avoidance

Design methodologies

Application to deadlock recovery

Application to networks of workstations

Performance evaluation



Outline

Introduction

Deadlock handling

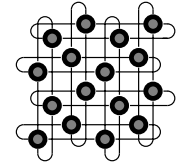
Theory of deadlock avoidance

Design methodologies

Application to deadlock recovery

Application to networks of workstations

Performance evaluation



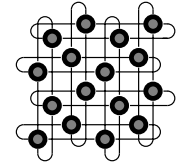
Introduction (From W. J. Dally)

The performance of most digital systems today is limited by their communication or interconnection, not by their logic or memory

Most of the power is used to drive wires and most of the clock cycle is spent on wire delay, not gate delay

As technology improves, pin density and wiring density are scaling at a slower rate than the components themselves. Also, the frequency of communication between components is lagging far beyond the clock rates of modern processors

These factors combine to make interconnection the key factor in the success of future digital systems

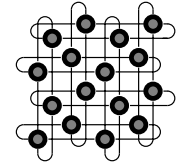


Introduction (From W. J. Dally)

As designers strive to make more efficient use of scarce interconnection bandwidth, interconnection networks are emerging as a nearly universal solution to the system-level communication problems for modern digital systems

Originally developed for the demanding communication requirements of multicomputers, interconnection networks are beginning to replace buses as the standard system-level interconnection

Interconnection networks are also replacing dedicated wiring in special-purpose systems as designers discover that routing packets is both faster and more economical than routing wires

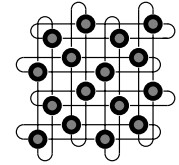


Introduction

Interconnection networks are currently being used for many different applications, ranging from internal buses in VLSI circuits to wide area computer networks. These applications include:

- System area networks
- Telephone switches
- Internal networks for ATM switches
- Processor/memory interconnects for vector supercomputers
- Interconnects for multicomputers
- Interconnects for distributed shared-memory multiprocessors
- Clusters of workstations
- Local area networks
- Metropolitan area networks
- Wide area networks

} Computer networks



Introduction

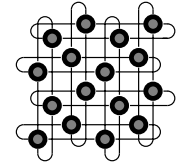
Parallel computers should be designed using commodity components to be cost-effective

Unfortunately, commodity communication subsystems have been designed to meet a different set of requirements, i.e., those arising in computer networks

Designing high performance interconnection networks becomes a critical issue to exploit the performance of parallel computers

Most manufacturers designed custom interconnection networks

Recently, several high performance switches have been developed to build inexpensive parallel computers by connecting cost-effective computers through those switches



Main design parameters

Topology: Defines how the nodes are interconnected by channels

- Direct networks, switch-based networks

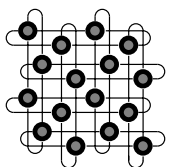
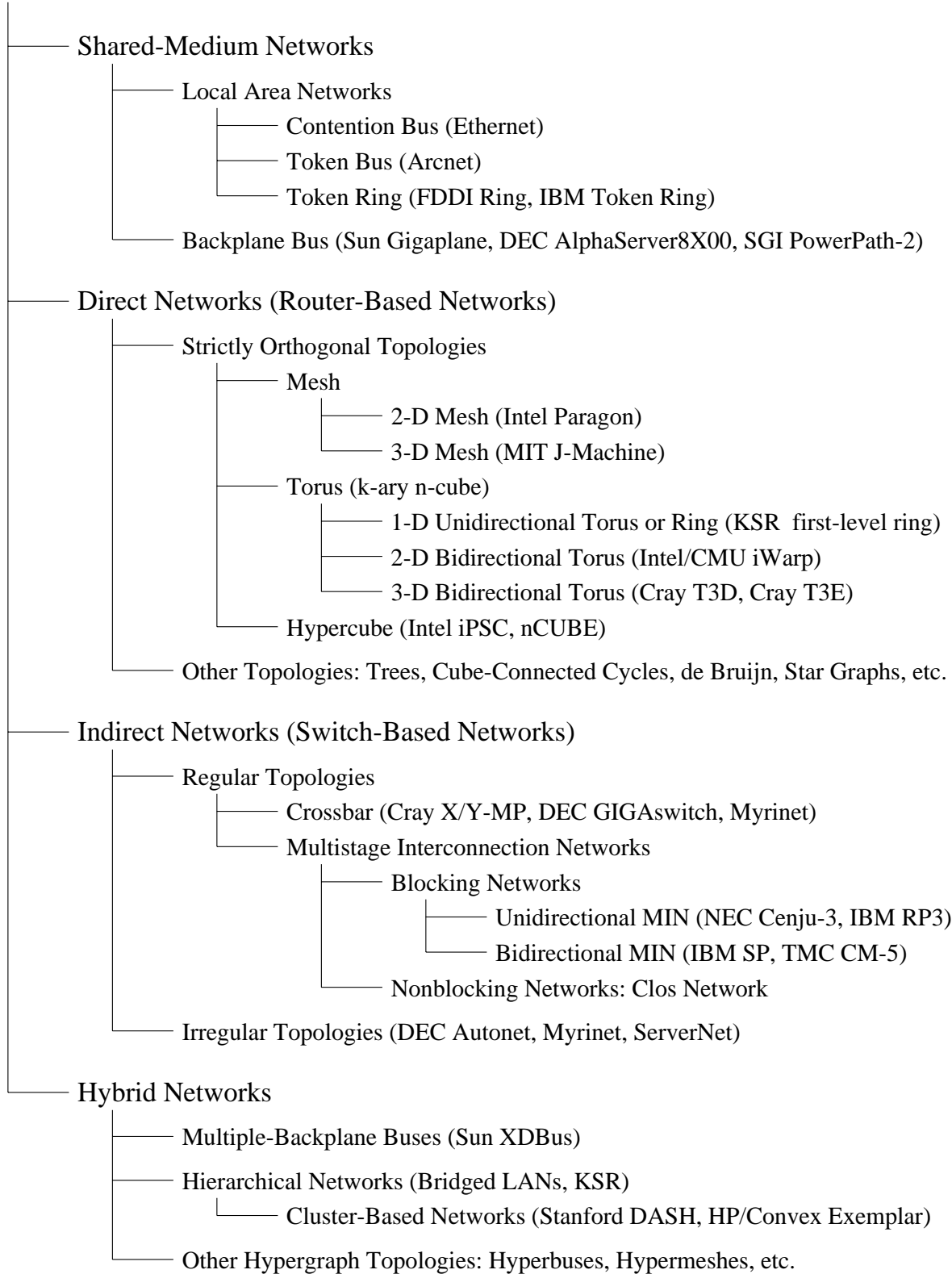
Routing algorithm: Determines the path selected by a message to reach its destination

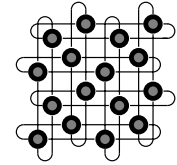
- Deterministic routing, adaptive routing

Switching technique: Determines how and when buffers are reserved and switches are configured

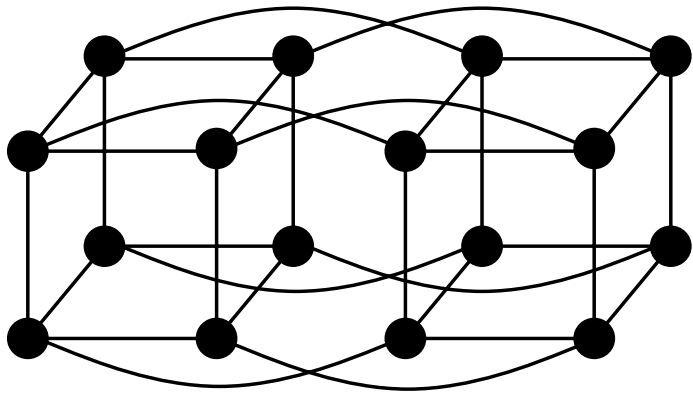
- Packet switching, circuit switching, wormhole, virtual cut-through

Interconnection Networks

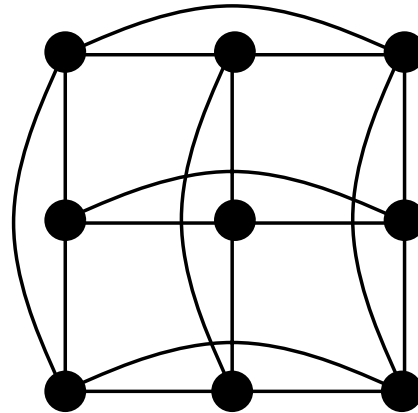




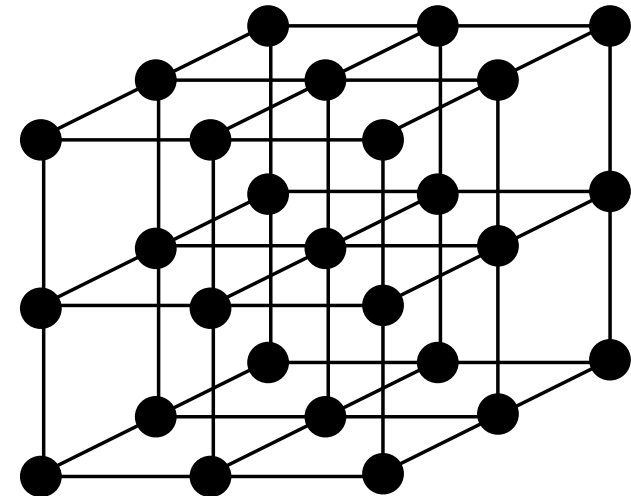
Direct networks



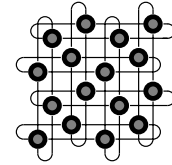
(a) 2-ary 4-cube (hypercube)



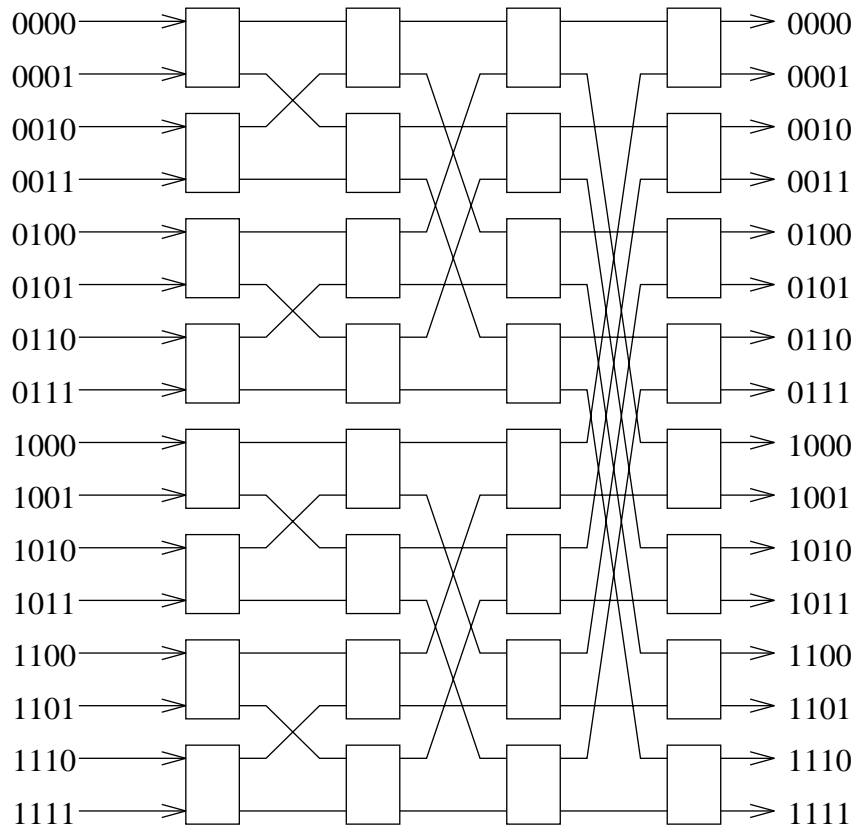
(b) 3-ary 2-cube



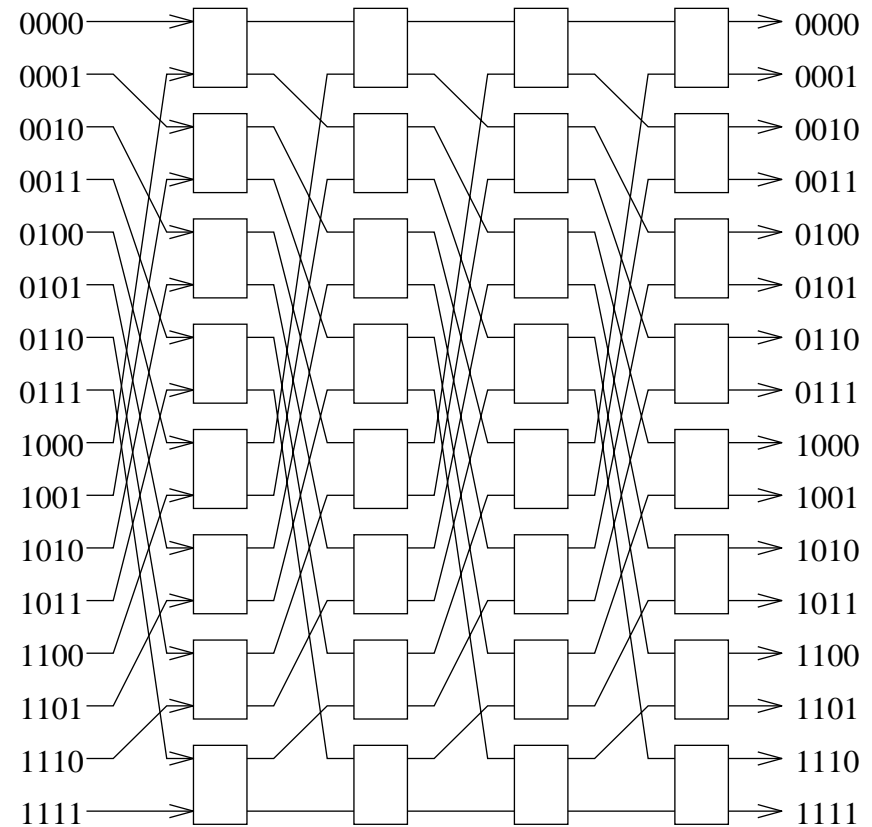
(c) 3-ary 3D-mesh



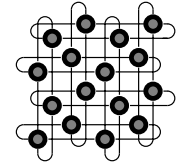
Multistage interconnection networks



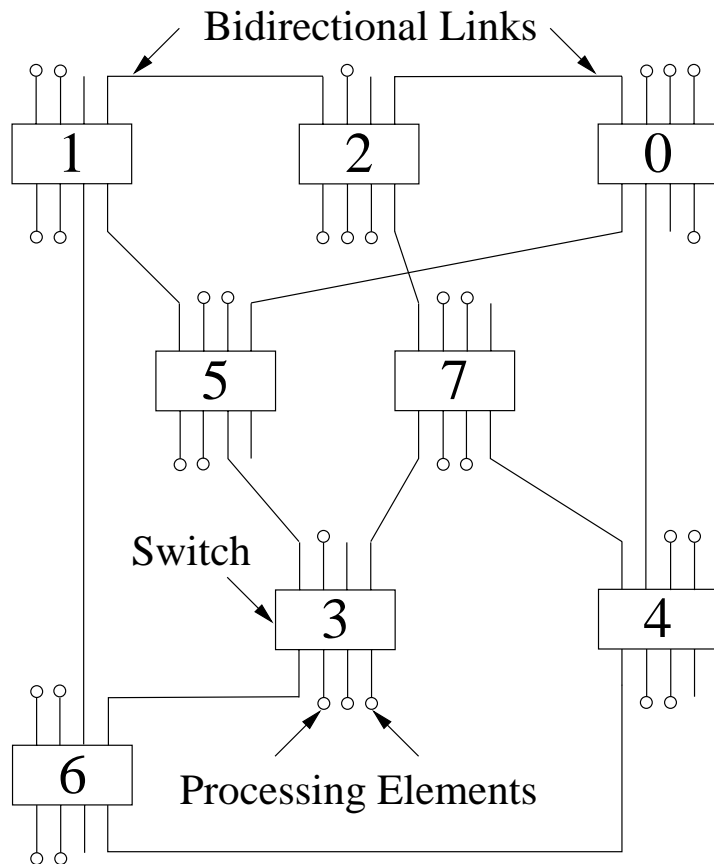
Multistage butterfly network



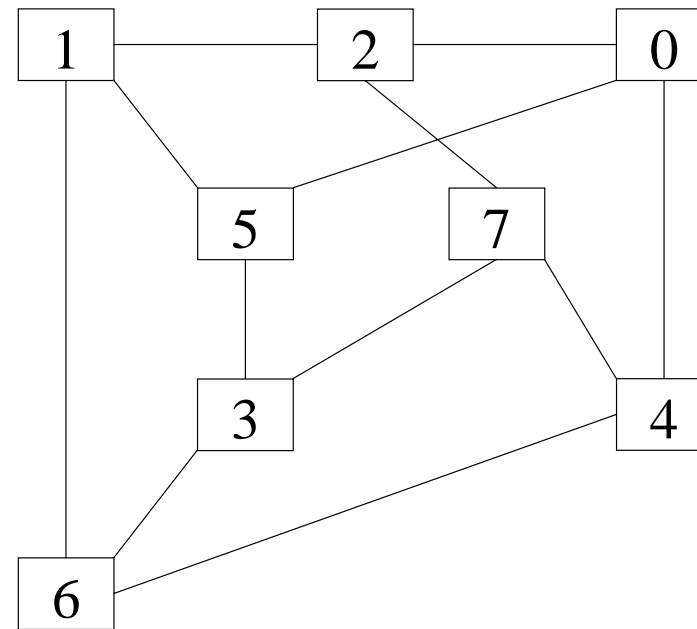
Omega network



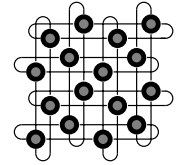
Switch-based irregular topologies



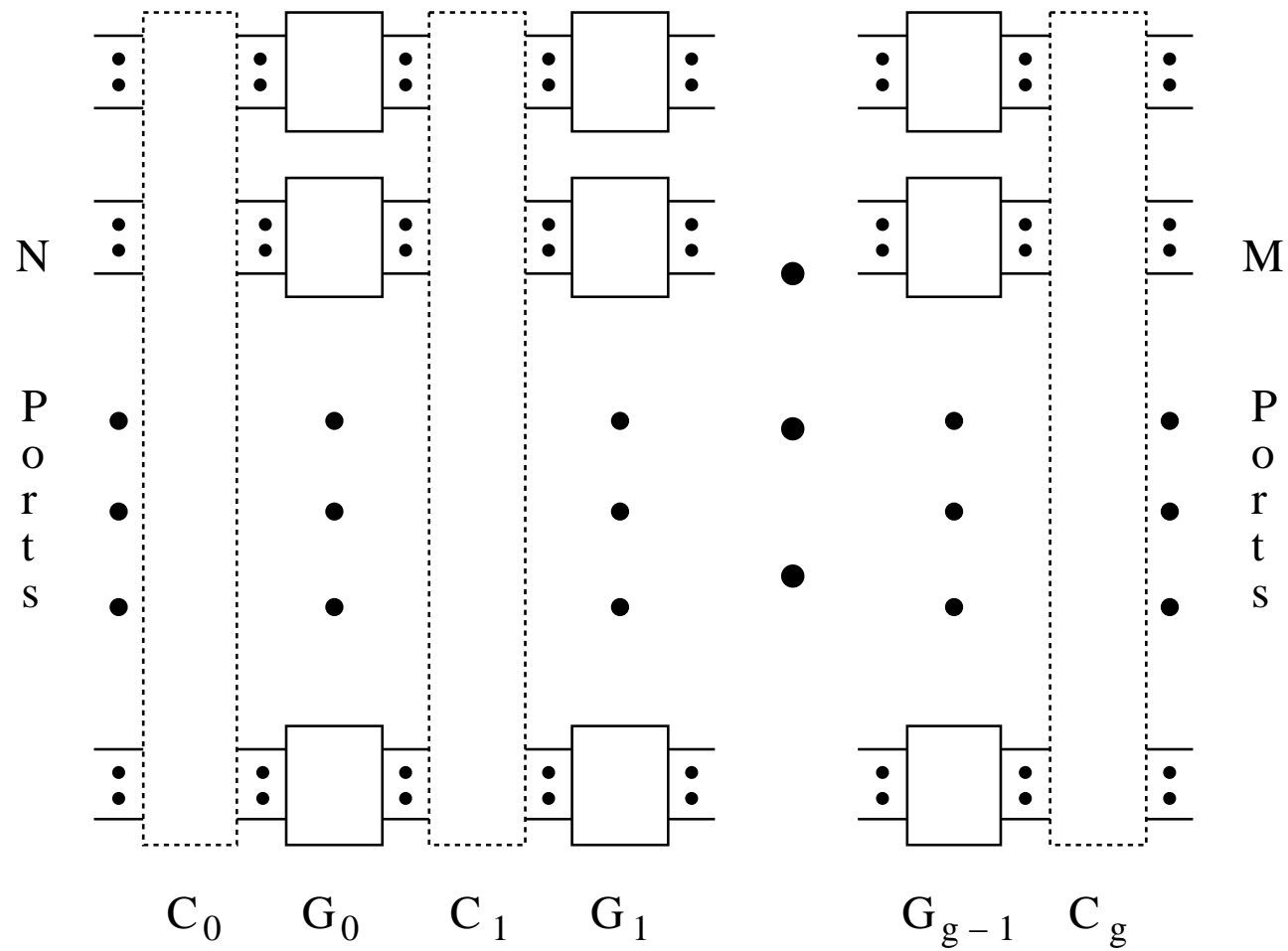
Switch-Based Network

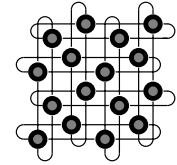


Graph Representation



Generalized MIN model



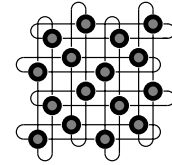


Unified View

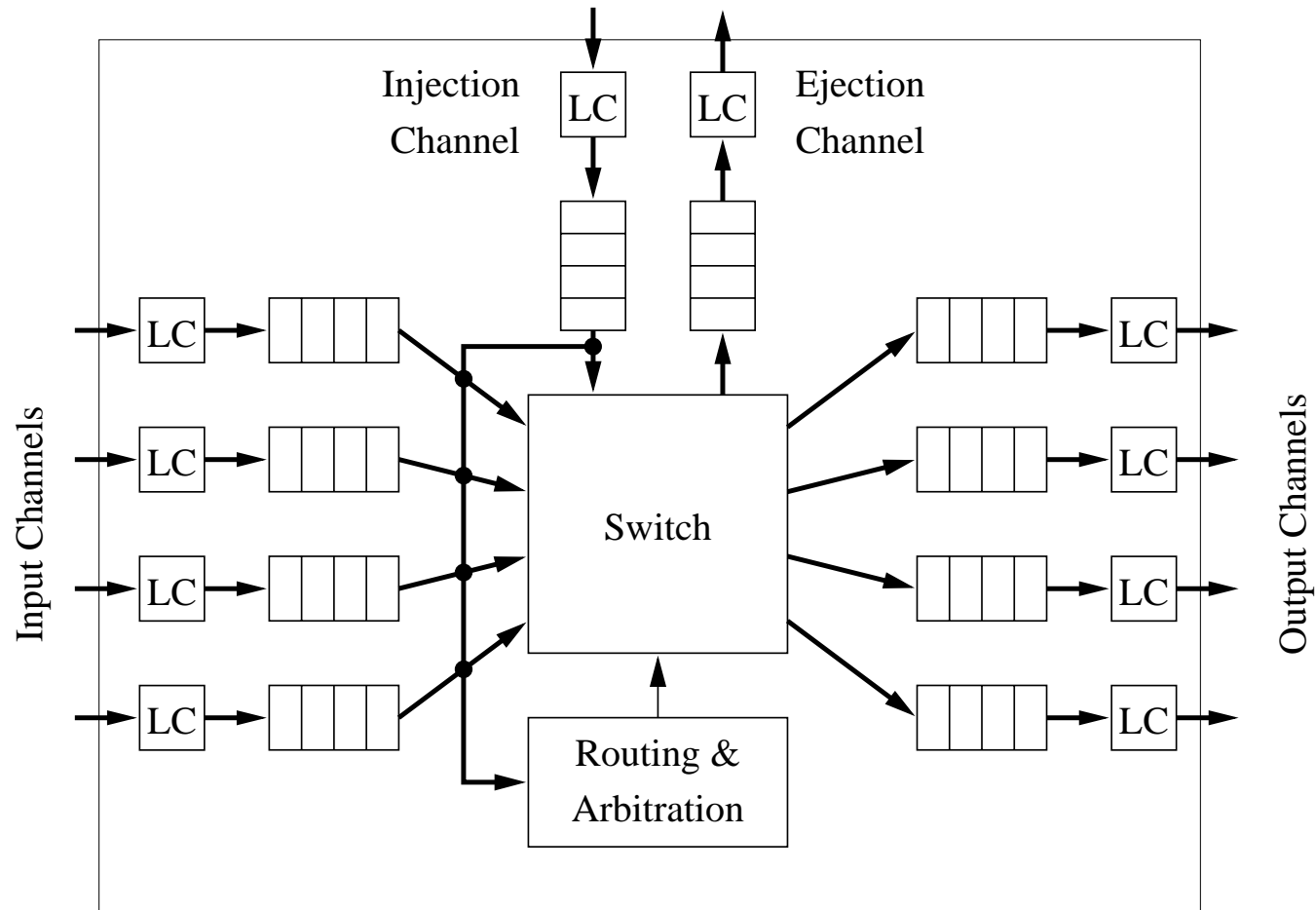
Some manufacturers developed switches that are suitable to implement either direct or indirect networks (Inmos C104, SGI SPIDER)

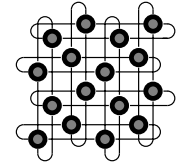
We can view networks using point-to-point links as a set of interconnected switches, each one connected to zero, one, or more nodes:

- Direct networks correspond to the case where every switch is connected to a single node
- Crossbar networks correspond to the case where there is a single switch connected to all the nodes
- Multistage interconnection networks correspond to the case where switches are arranged into several stages and the switches in intermediate stages are not connected to any processor



Router organization





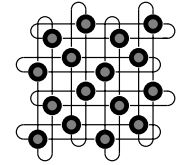
Switching

Switching: Determines how and when buffers are reserved and switches are configured

Flow control: Synchronization protocol for transmitting and receiving a unit of information

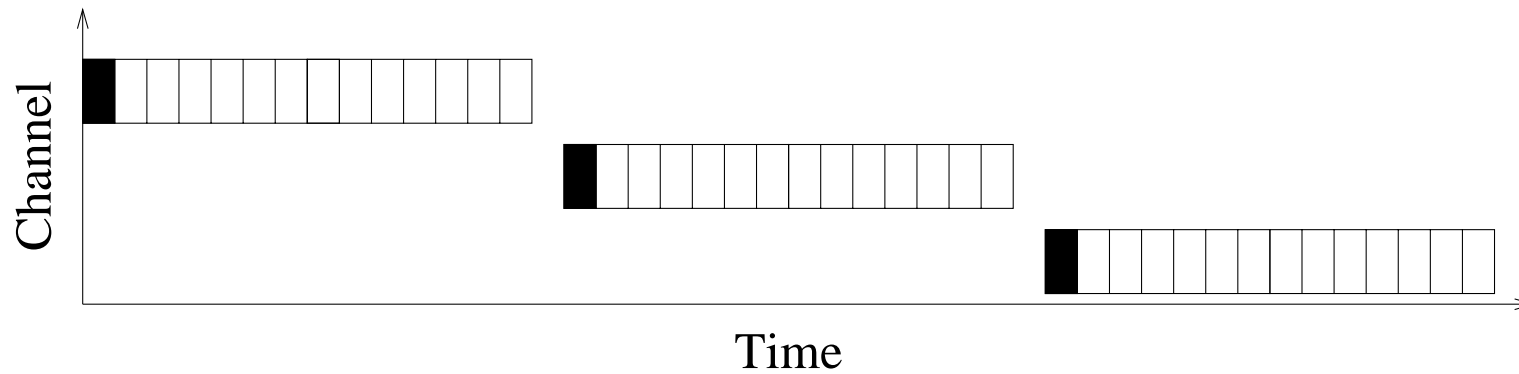
Unit of flow control: Portion of the message whose transfer must be synchronized

Flow control occurs at two levels: message flow control and physical channel flow control

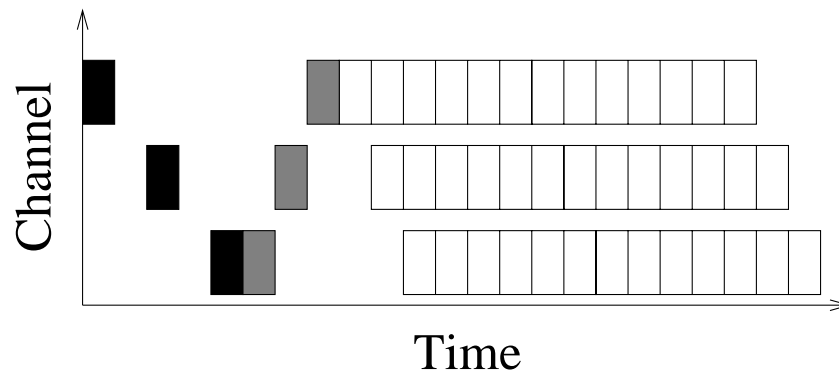


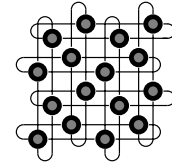
Packet switching and circuit switching

Time-space diagram (packet switching)

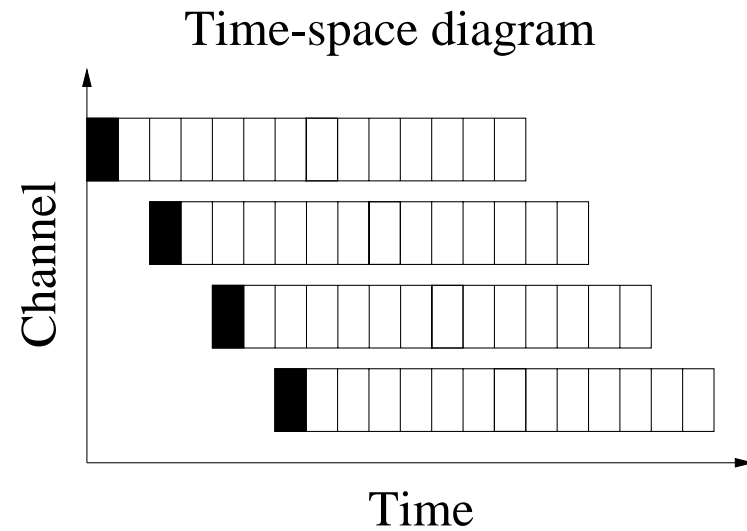
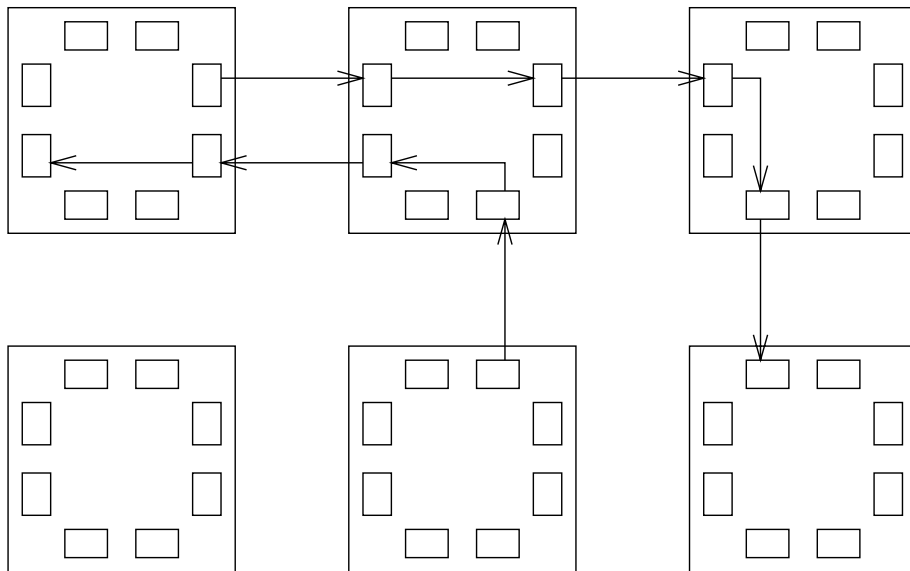
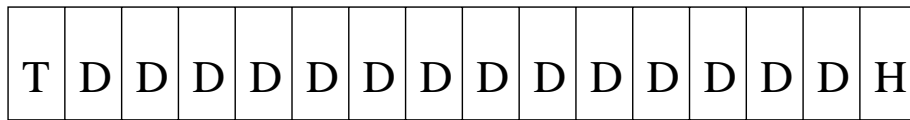


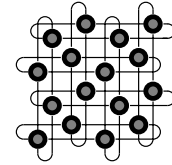
Time-space diagram (circuit switching)



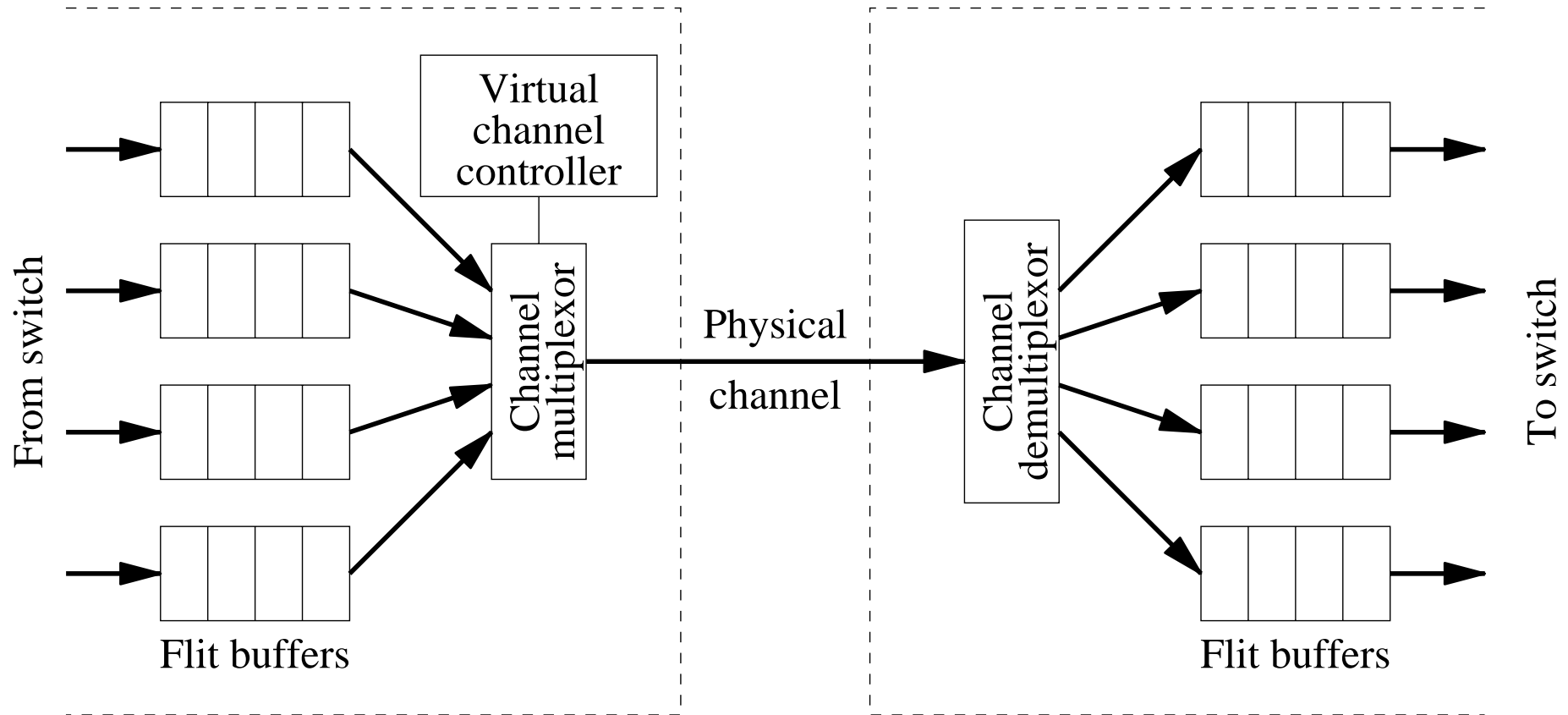


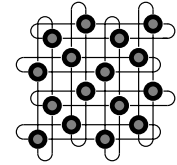
Virtual cut-through and wormhole switching





Virtual channels





Performance of switching techniques

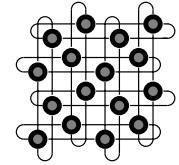
Packet switching is well suited for very short messages

Circuit switching is well suited for very long messages

Virtual cut-through switching is well suited for messages of any length but requires splitting messages into fixed-size packets

Wormhole switching is well suited for messages of any length but saturates at moderate loads. Virtual channels alleviate this situation

Wormhole switching has been preferred for electronic routers because buffers can be small and the resulting circuits are compact and fast



Optimized switching techniques

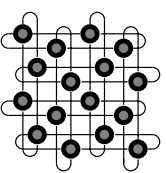
Traffic from real applications may be bimodal and may vary over time

Wormhole switching can be used for short messages

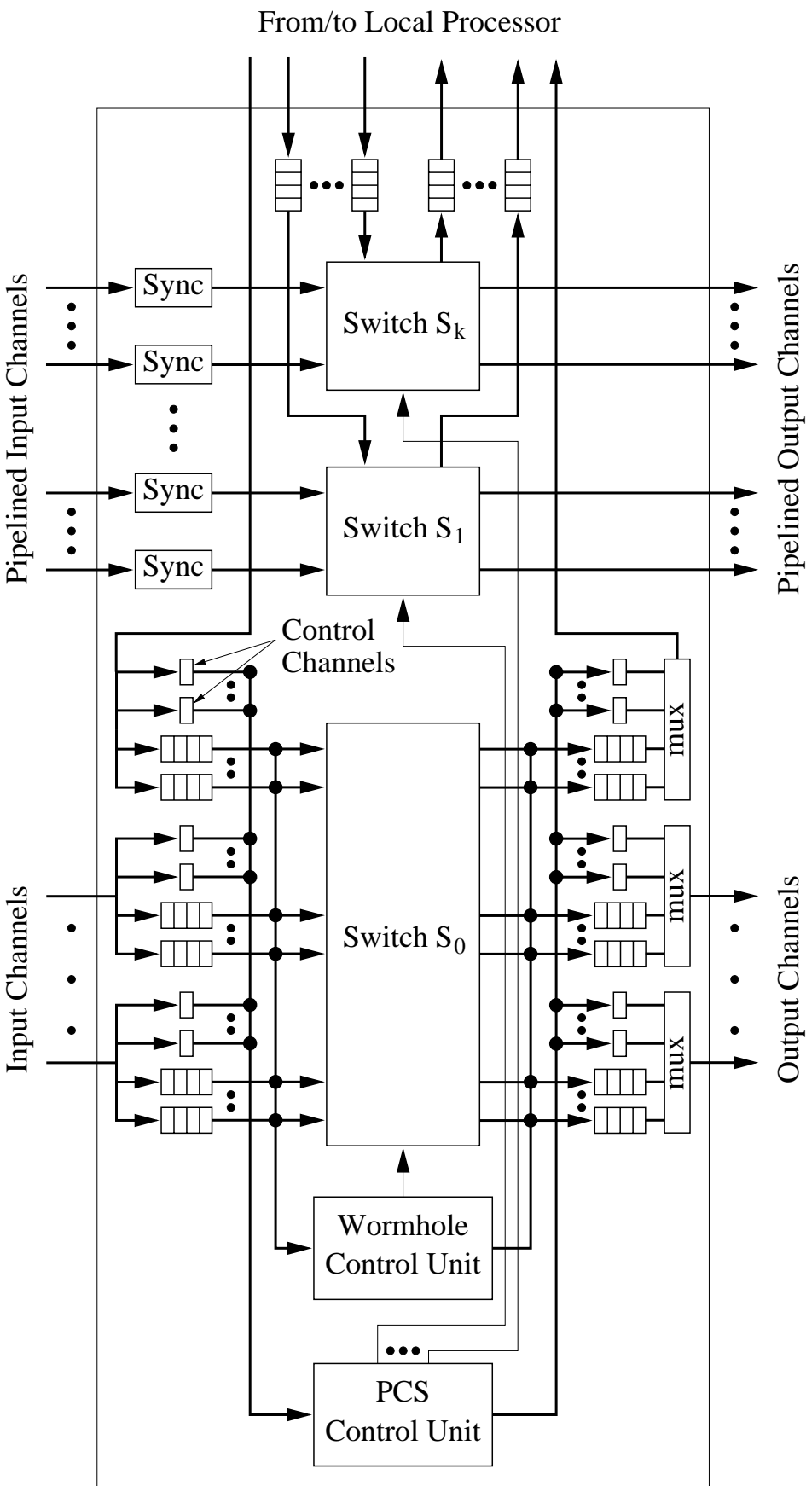
Circuit switching can be used for very long messages

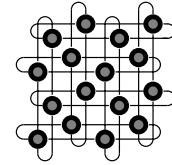
Path set-up can be overlapped with useful computation and/or circuits can be reused

Physical circuits do not need buffers at intermediate routers and can be made much faster than conventional links either by using wave pipelining or optical technology

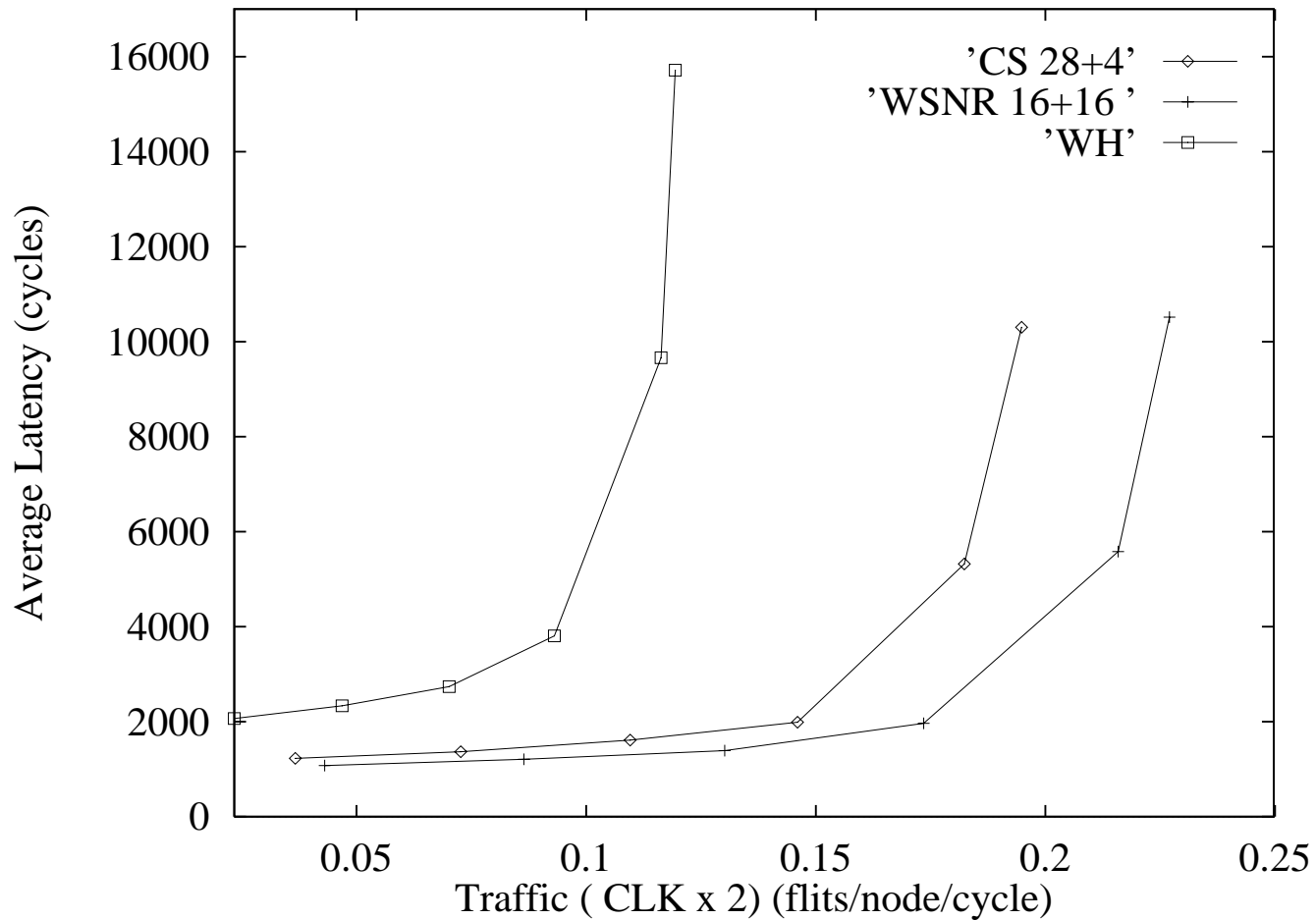


Optimized router organization



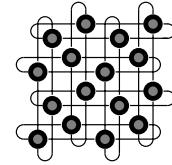


Performance for multimedia applications

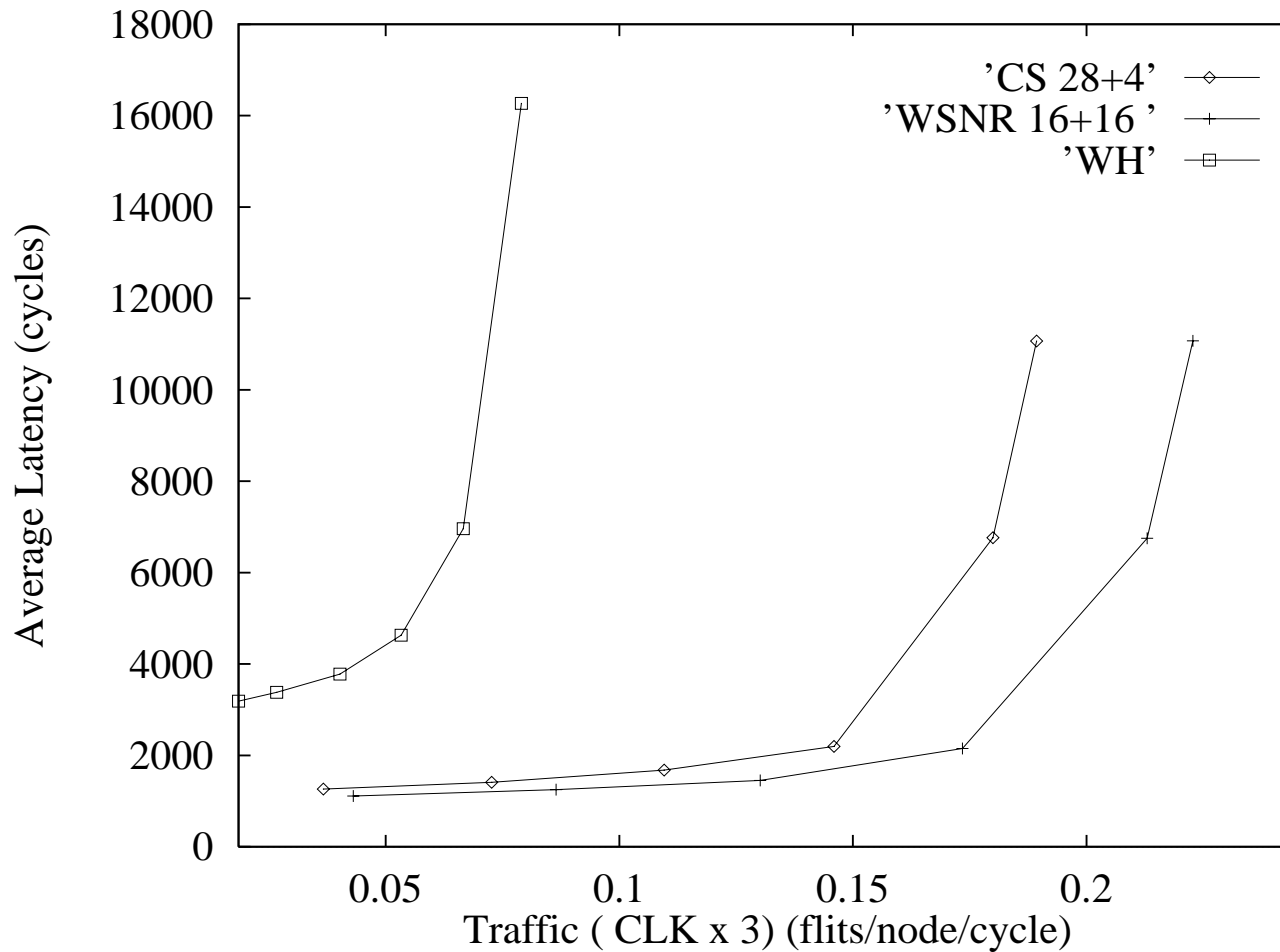


10% short
messages
(16 flits)

90% long
messages
(1024 flits)

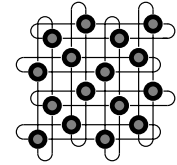


Performance for multimedia applications

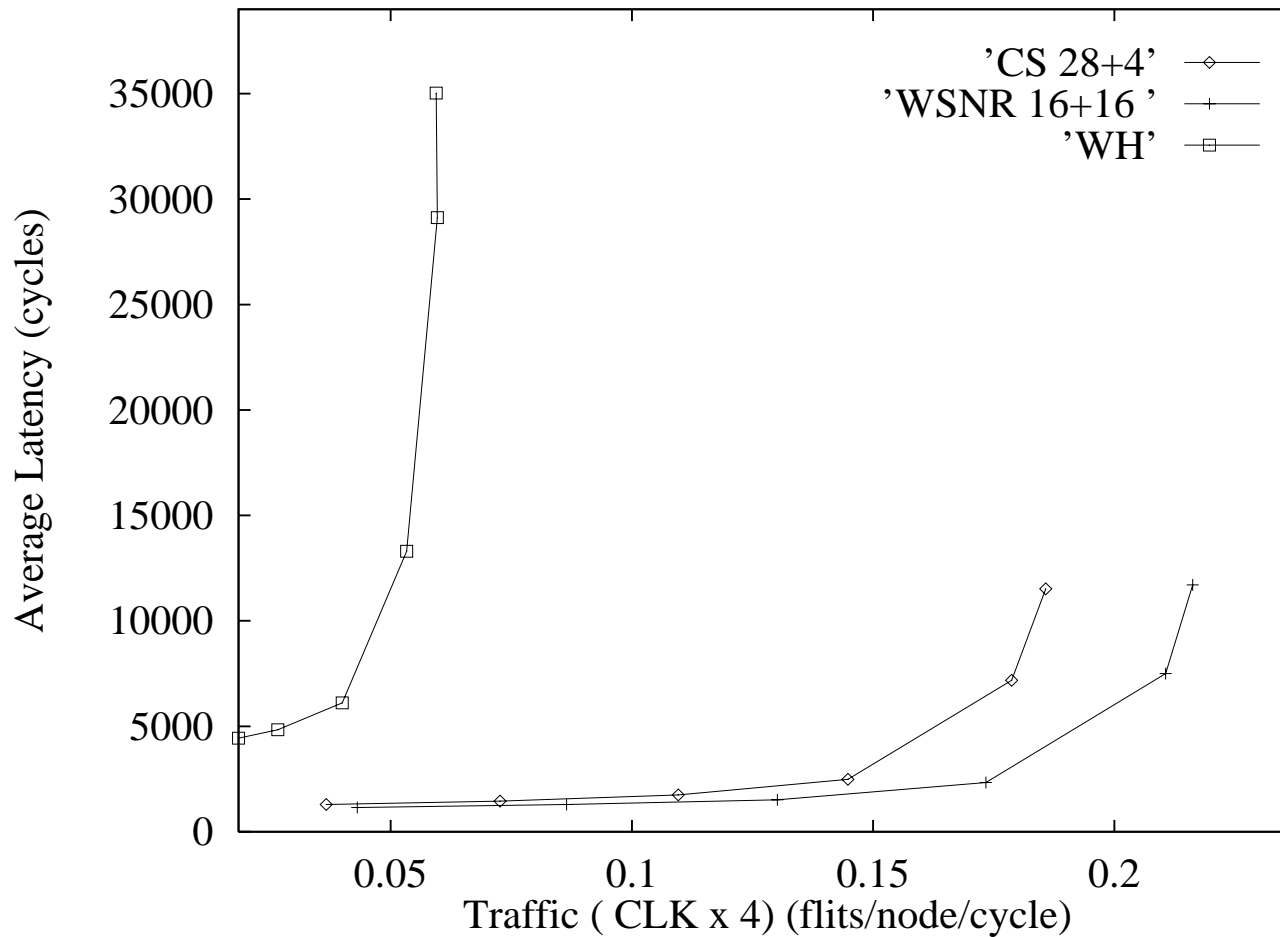


10% short
messages
(16 flits)

90% long
messages
(1024 flits)

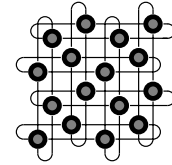


Performance for multimedia applications

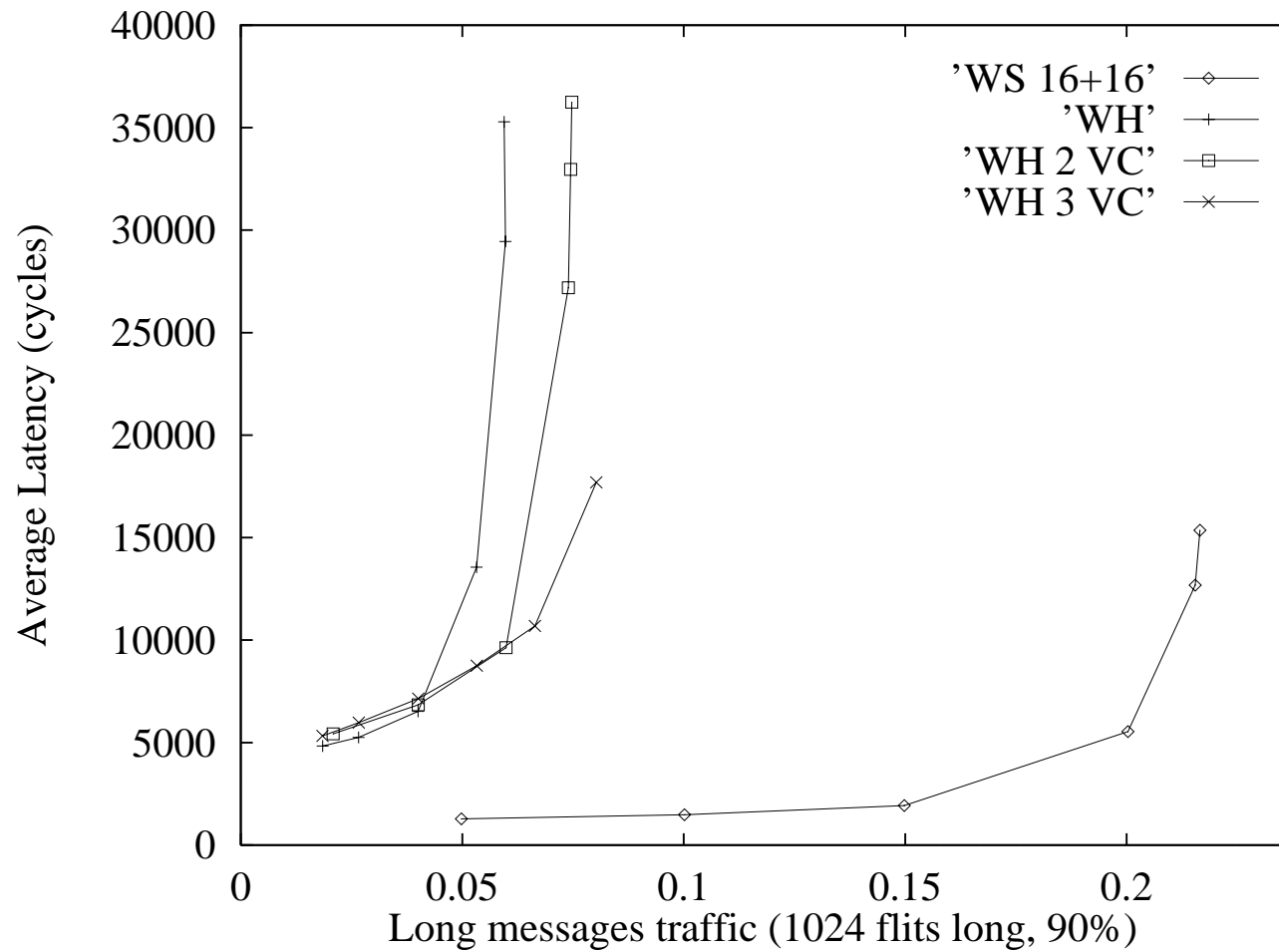


10% short
messages
(16 flits)

90% long
messages
(1024 flits)



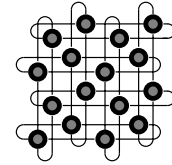
Performance for multimedia applications



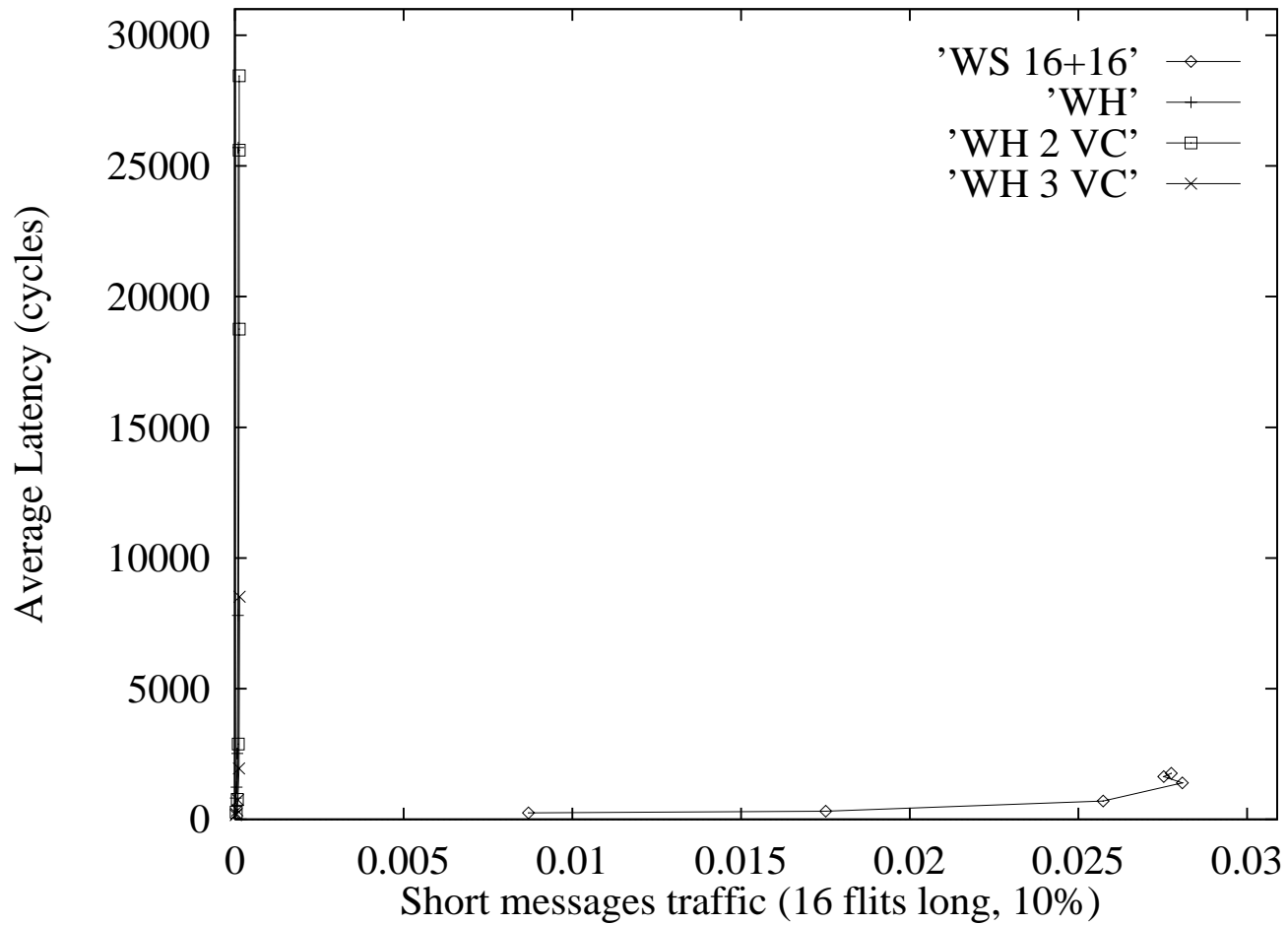
10% short
messages
(16 flits)

90% long
messages
(1024 flits)

Only long
messages
are shown



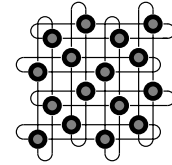
Performance for multimedia applications



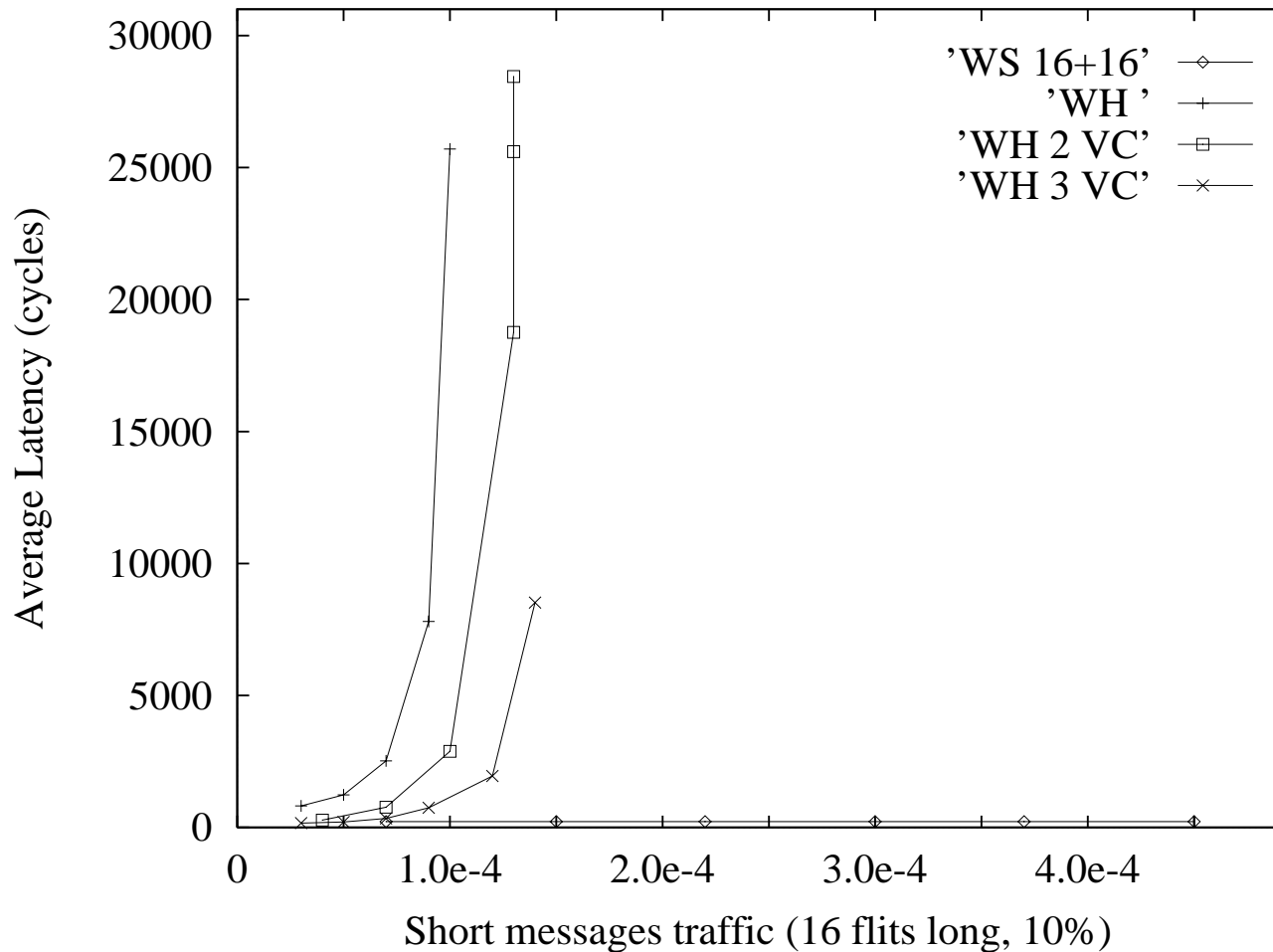
10% short
messages
(16 flits)

90% long
messages
(1024 flits)

Only short
messages
are shown



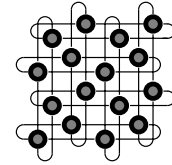
Performance for multimedia applications



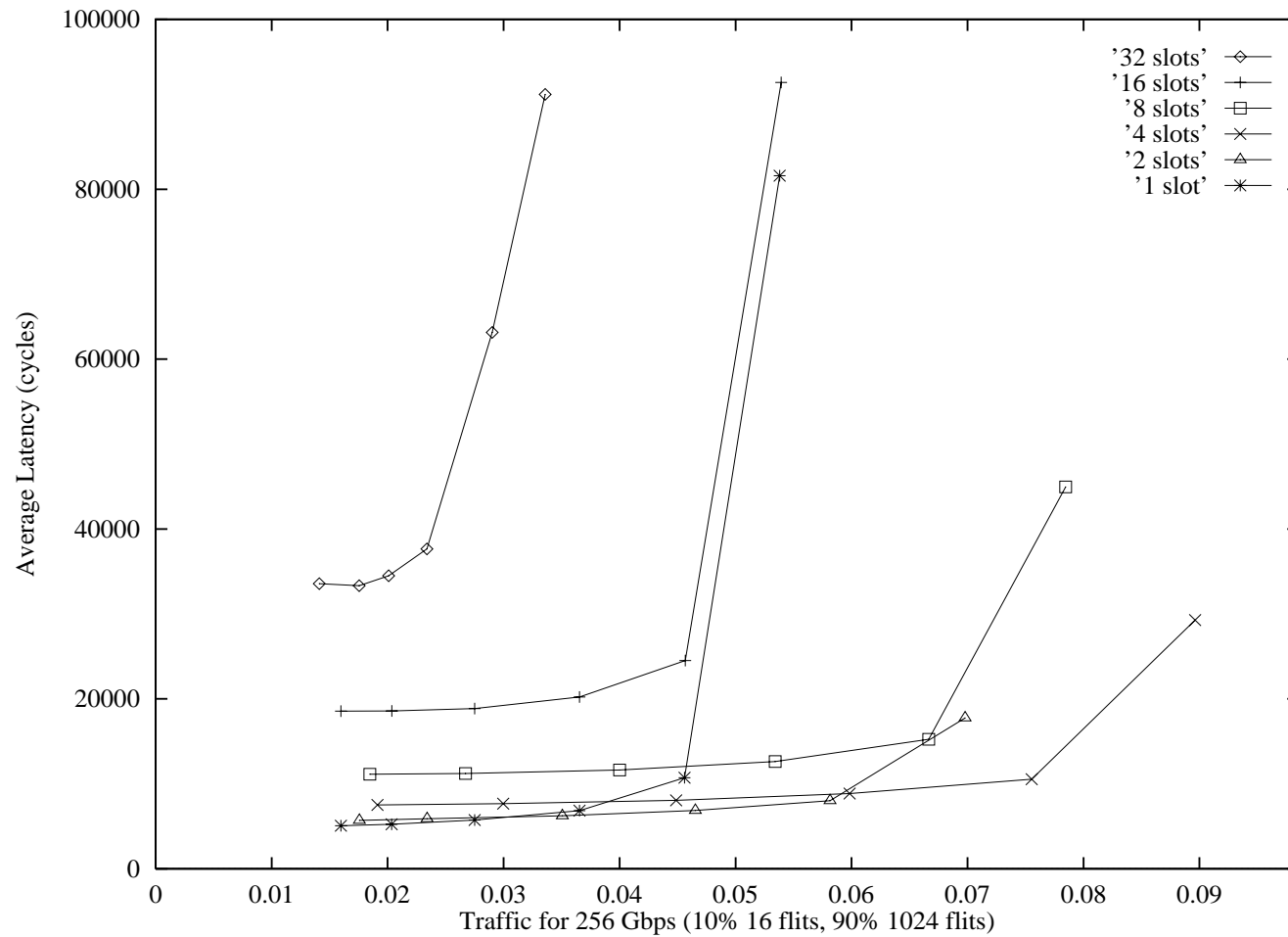
10% short
messages
(16 flits)

90% long
messages
(1024 flits)

Only short
messages
are shown



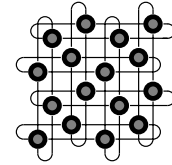
Performance for multimedia applications



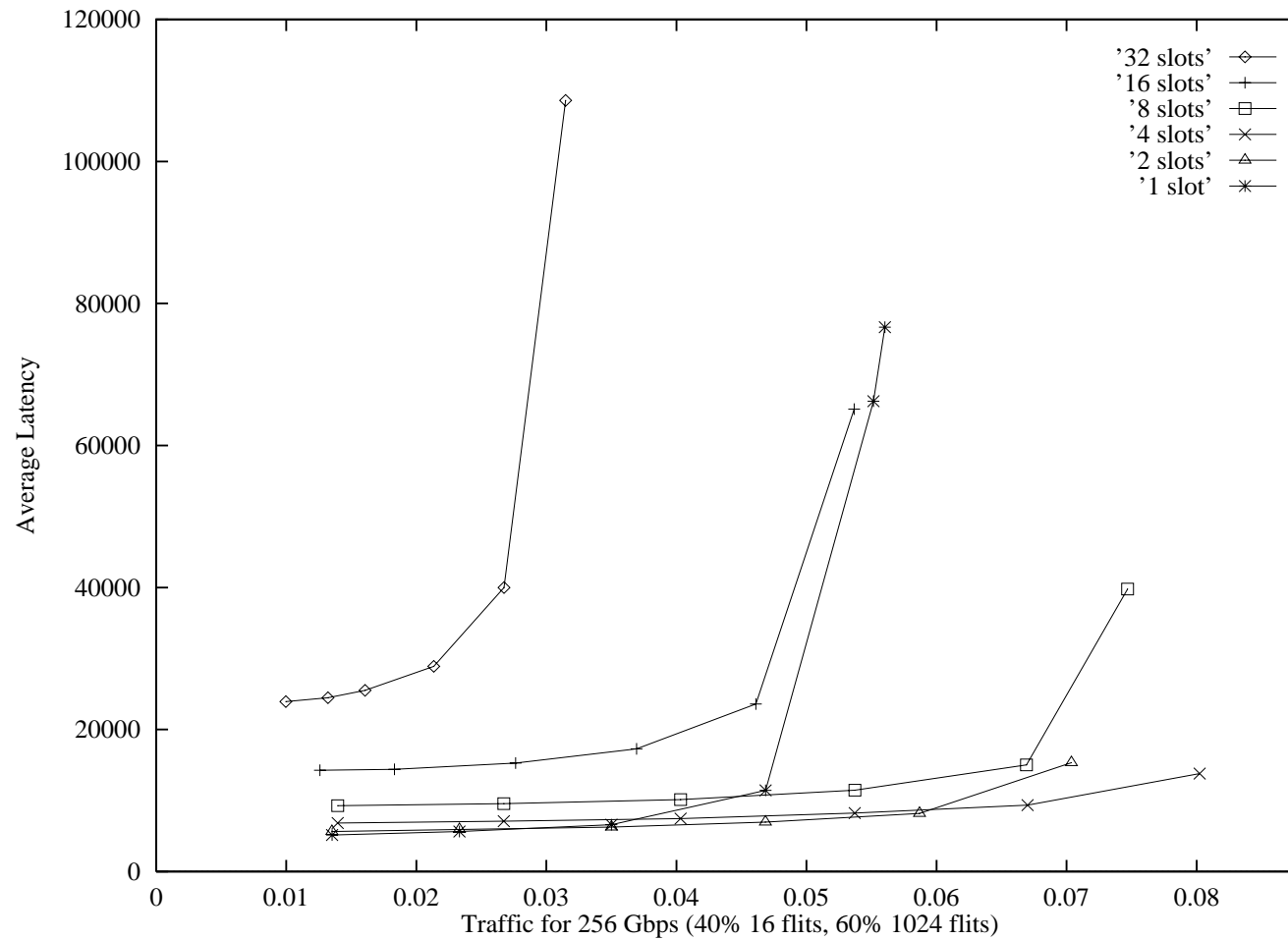
10% short
messages
(16 flits)

90% long
messages
(1024 flits)

256 Gbps
link band-
width



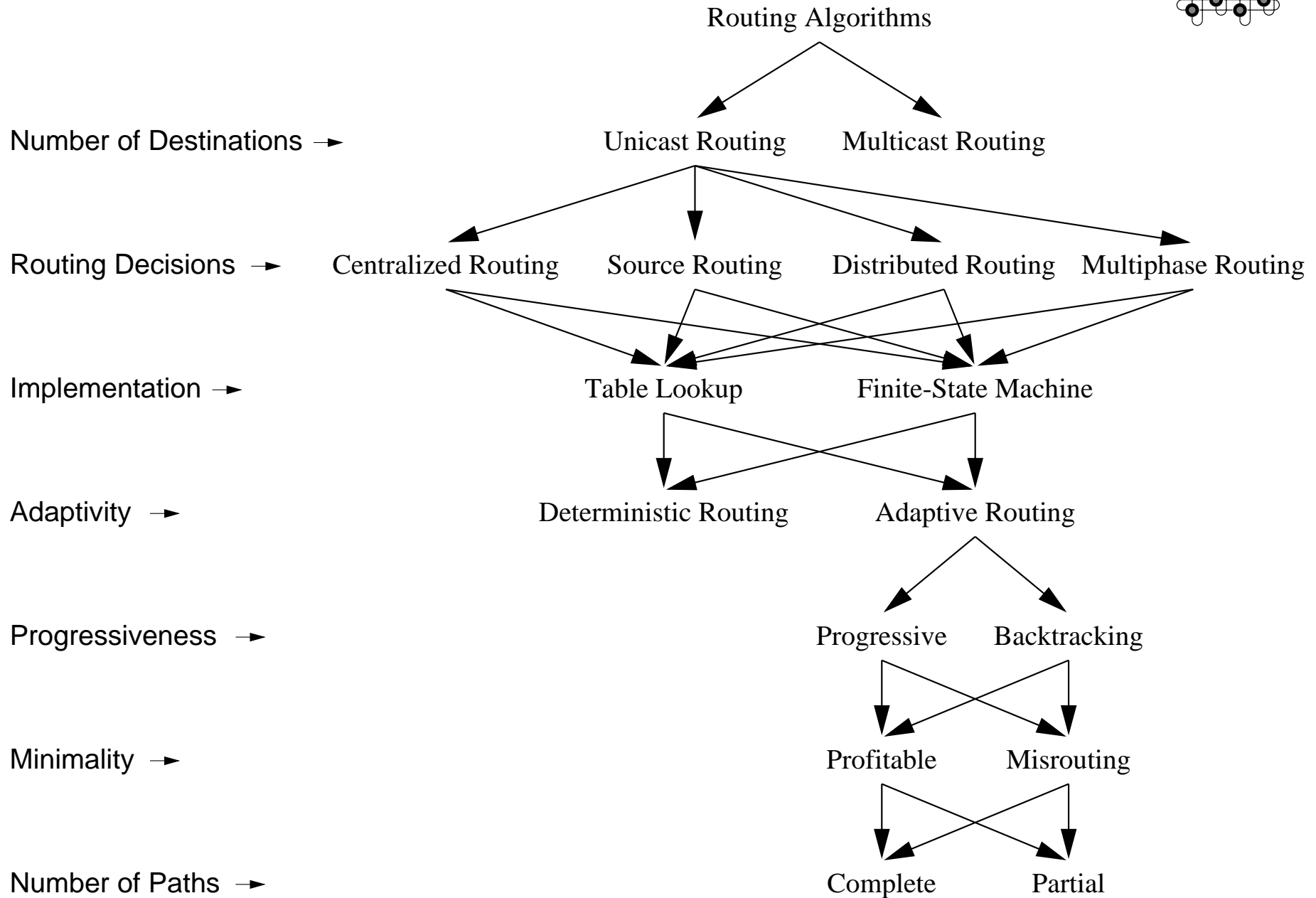
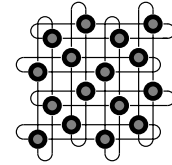
Performance for multimedia applications

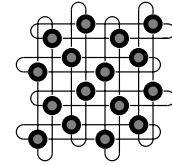


40% short
messages
(16 flits)

60% long
messages
(1024 flits)

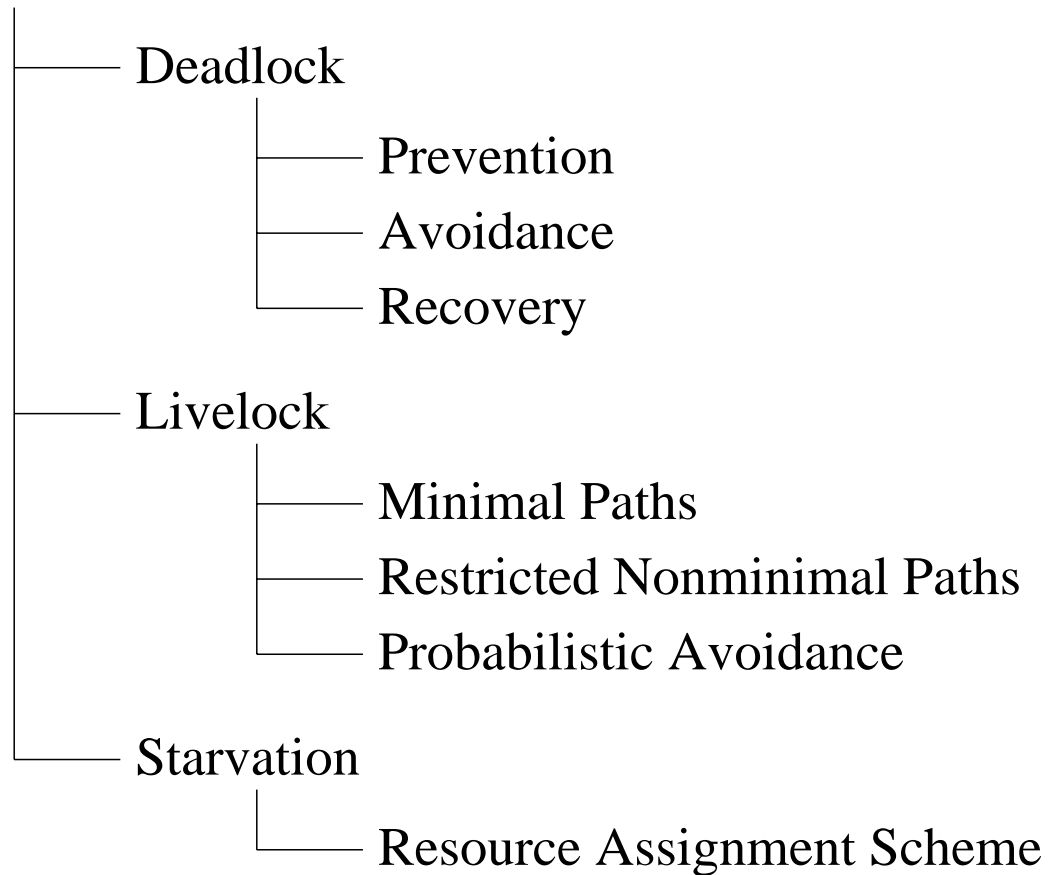
256 Gbps
link band-
width

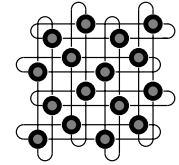




Situations that may prevent packet delivery

Undeliverable Packets





Deadlock handling

Deadlock prevention: Backtracking

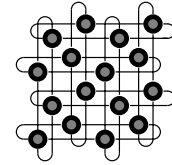
Deadlock avoidance: Acyclic graph, acyclic subgraph

Regressive deadlock recovery: Message removal, message abortion

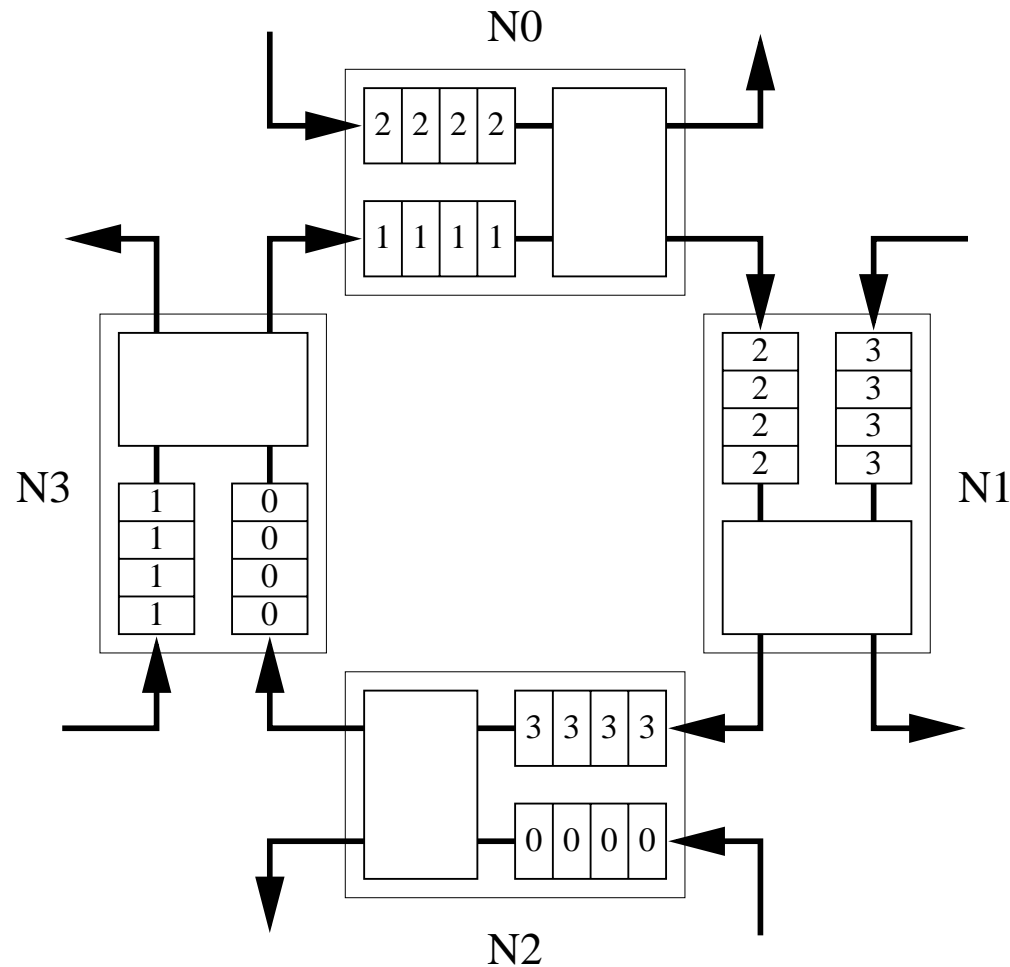
Progressive deadlock recovery: Disha

Main goal

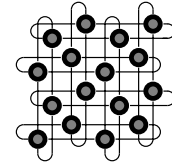
Design of efficient deadlock-free fully adaptive routing algorithms



Deadlocked configuration

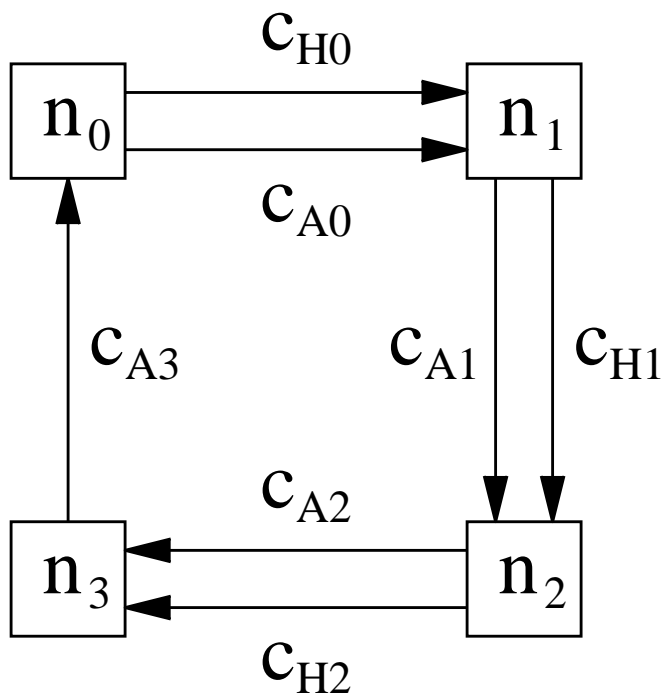


Messages wait for resources held by other messages in a cyclic way
⇒ Removing cyclic dependencies will avoid deadlock



Allowing cyclic dependencies

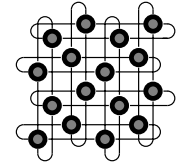
Example for the unidirectional ring: c_{Ai} channels can be used to forward messages to all the destinations. c_{Hi} channels can only be used if the destination is higher than the current node.



There exist cyclic dependencies between c_{Ai} channels

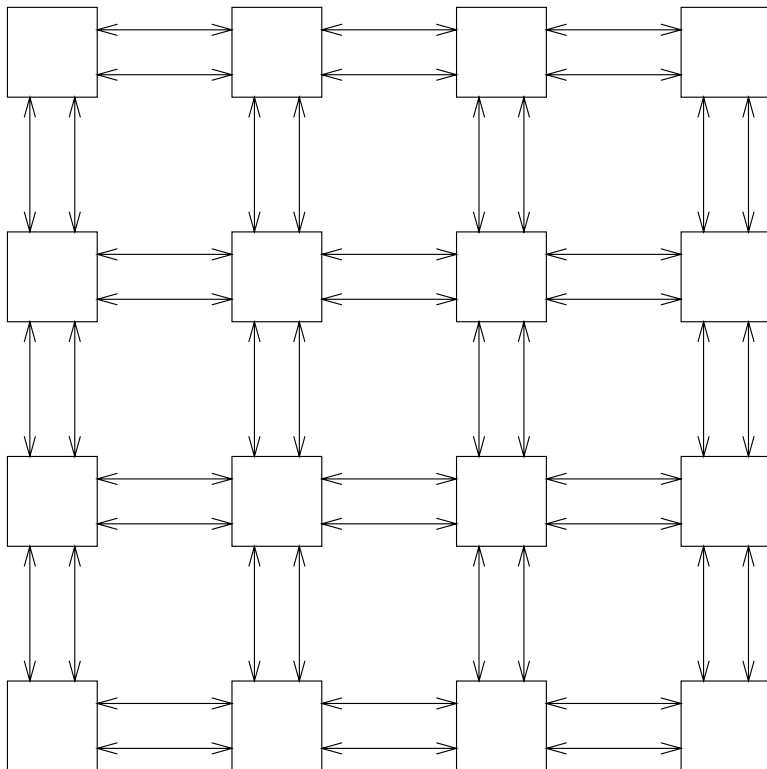
However, c_{Hi} channels have no cyclic dependencies

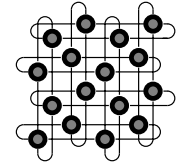
There is no deadlock because messages waiting for resources can always escape by using c_{Hi} channels



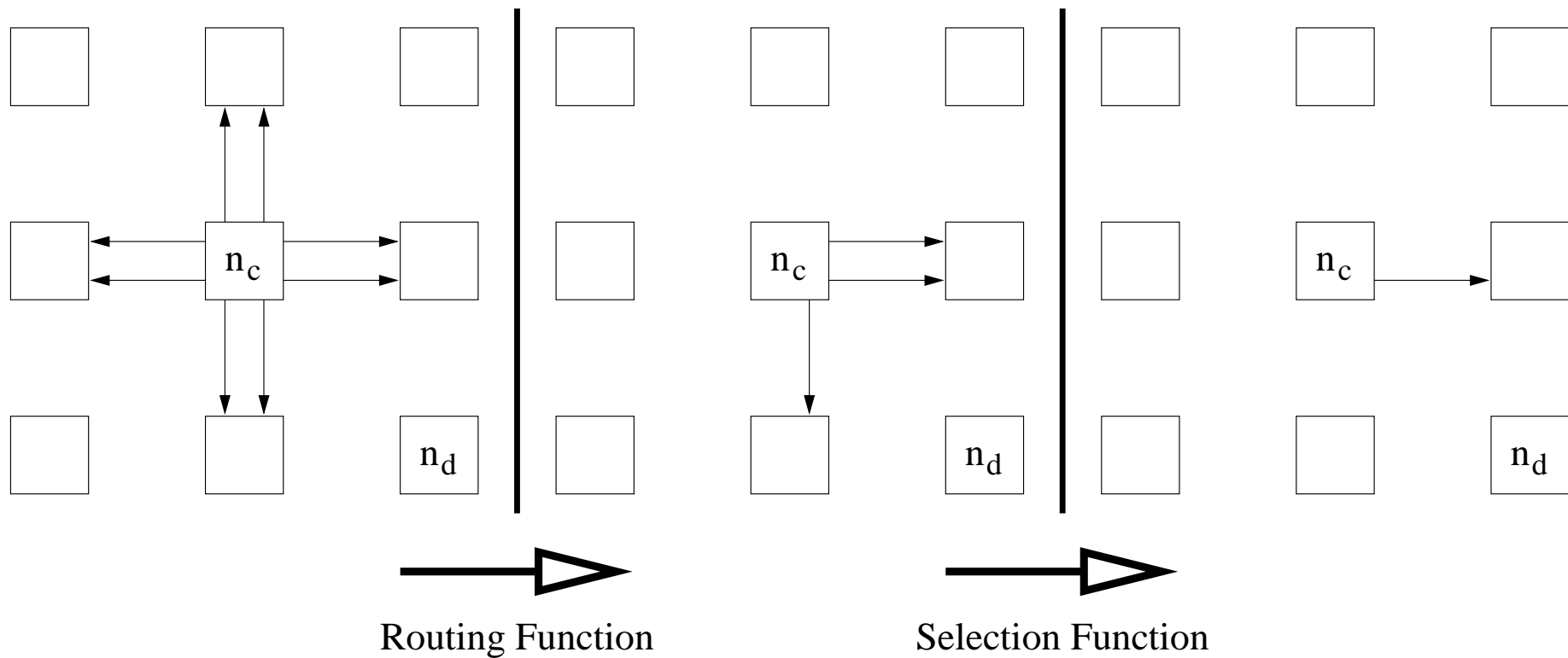
Theory of deadlock avoidance (informal)

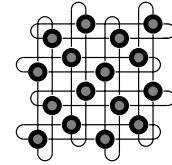
Interconnection network





Adaptive routing function and selection function

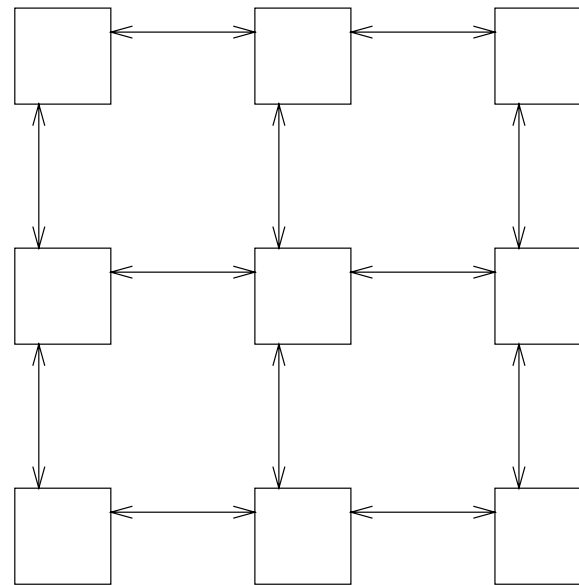
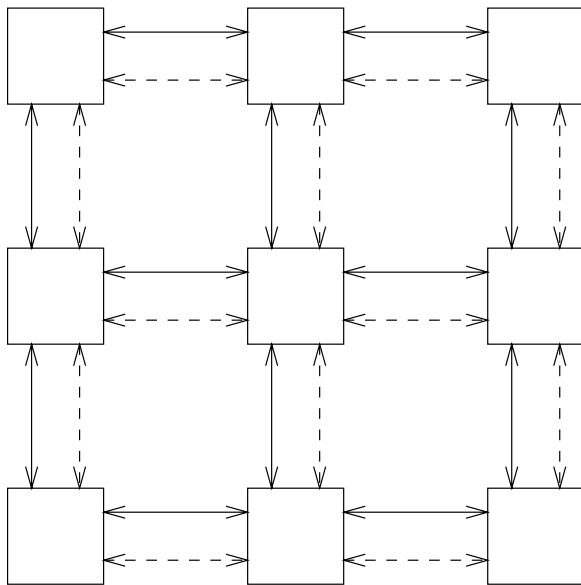


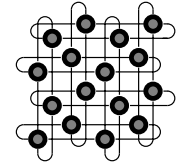


Routing subfunction

Network channels can be split into two subsets: *adaptive* and *escape* channels

The routing function will be referred to as routing subfunction when restricted to escape channels

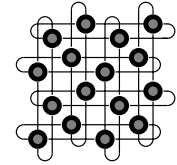




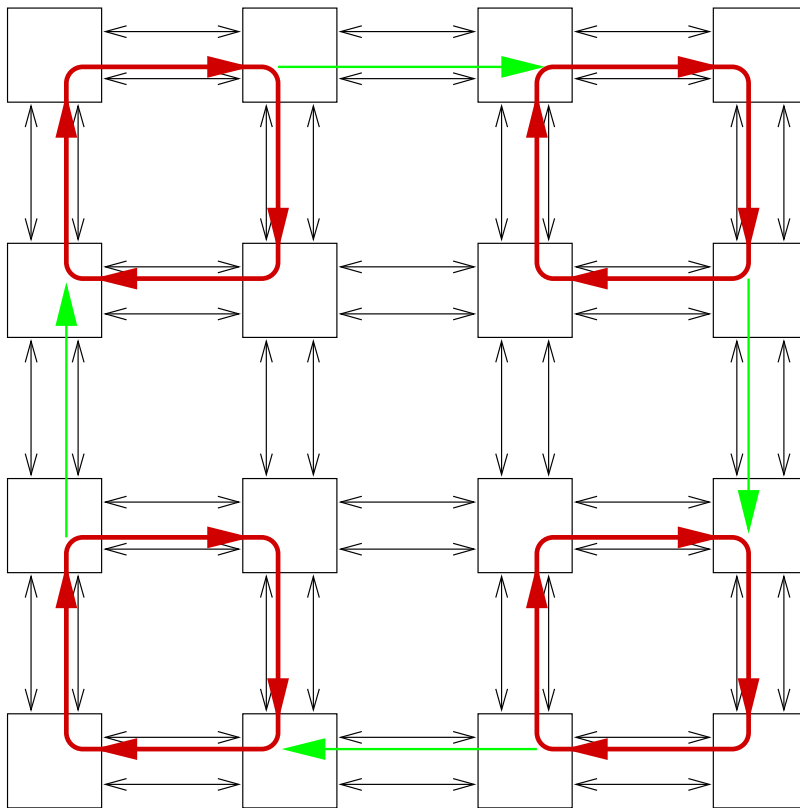
Approach to avoid deadlock

An adaptive routing function may allow cyclic dependencies between channels as long as:

- There exist a subset of channels (escape channels) that have no cyclic dependencies between them
- It is possible to establish a path from the current node to the destination node using only escape channels
- For wormhole switching, when a message reserves an escape channel and then an adaptive channel, it must be able to select an escape channel at the current node, i.e., escape channels should have no cyclic dependencies *indirectly* through adaptive channels



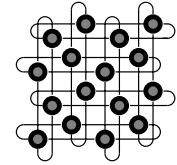
Deadlock produced by indirect dependencies



A set of messages are cyclically waiting for channels occupied by other messages in the set

Some messages are able to use escape channels but reach another cycle. Messages using escape channels are cyclically waiting indirectly through adaptive channels

⇒ There is a deadlock



Design methodology

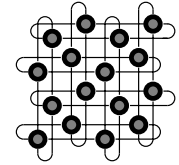
Based on the extension of other routing functions

Allows the use of all the alternative minimal paths

Does not increase the number of physical channels

Provides a way to:

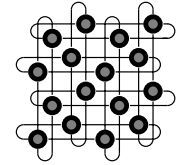
- Extend the network topology and the routing function
- Guarantee the absence of deadlocks



Design methodology

Steps:

- Given an interconnection network I_1 , define a minimal path connected deadlock-free routing function R_1
- Split each physical channel into a set of additional virtual channels. The new routing function can use any of the new channels belonging to a minimal path or, alternatively, the channels supplied by R_1
- Verify that the extended channel dependency graph for R_1 is acyclic. If it is, the routing algorithm is valid. Otherwise, it must be discarded. This step is not required for store-and-forward and virtual cut-through



Design example

Routing algorithm for n -dimensional meshes

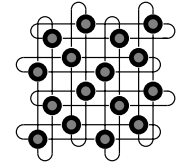
Basic algorithm: Dimension order routing

Step2: Split each physical channel c_i into k virtual channels

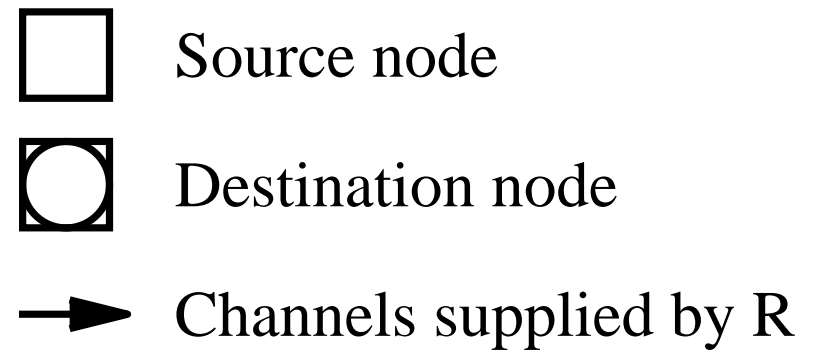
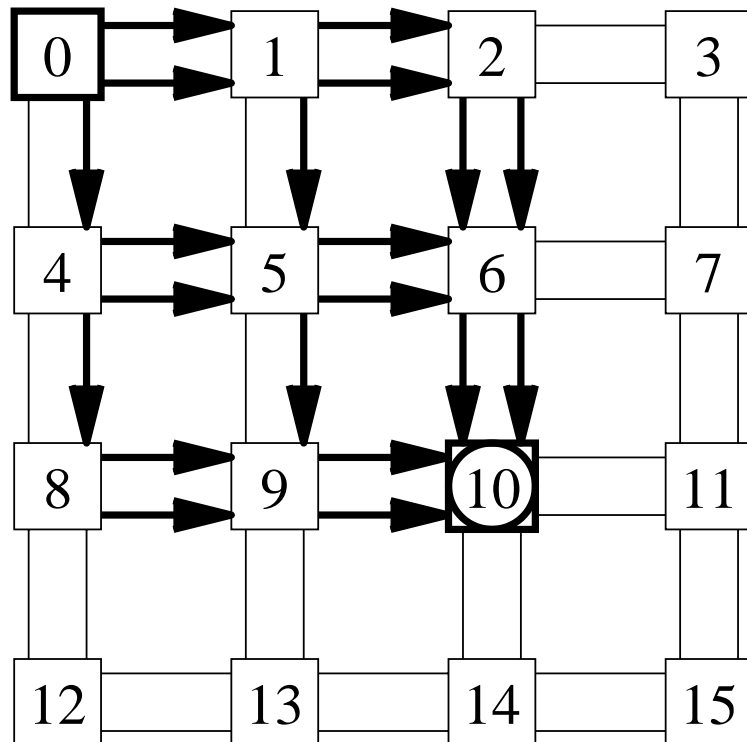
$a_{i,1}, a_{i,2}, \dots, a_{i,k-1}, b_i$

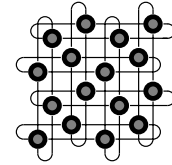
New algorithm: Route over any minimal path using any of the a channels. Alternatively, route over the lowest useful dimension using the corresponding b channel

The MIT Reliable Router uses two virtual channels for fully adaptive minimal routing and two virtual channels for dimension-order routing in the absence of faults (on a 2-D mesh)

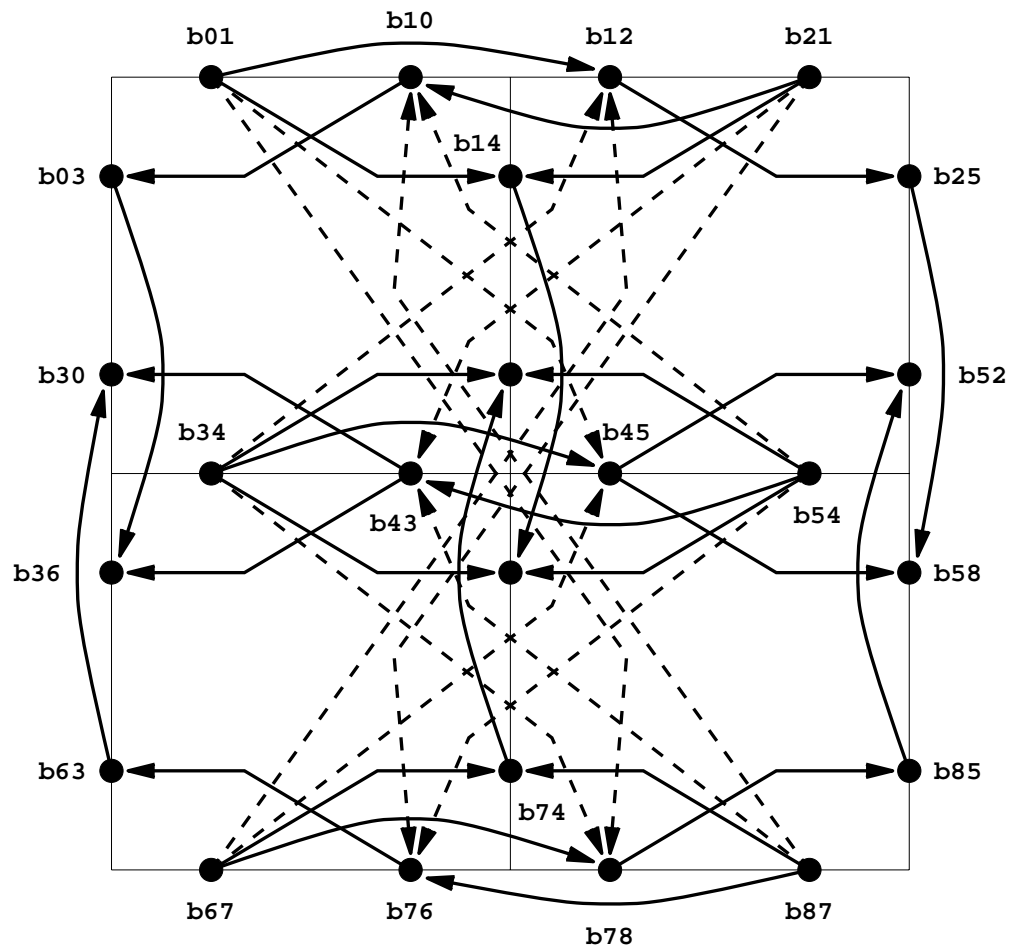


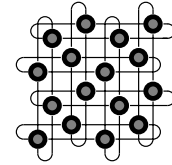
Example routing paths for 2-D meshes



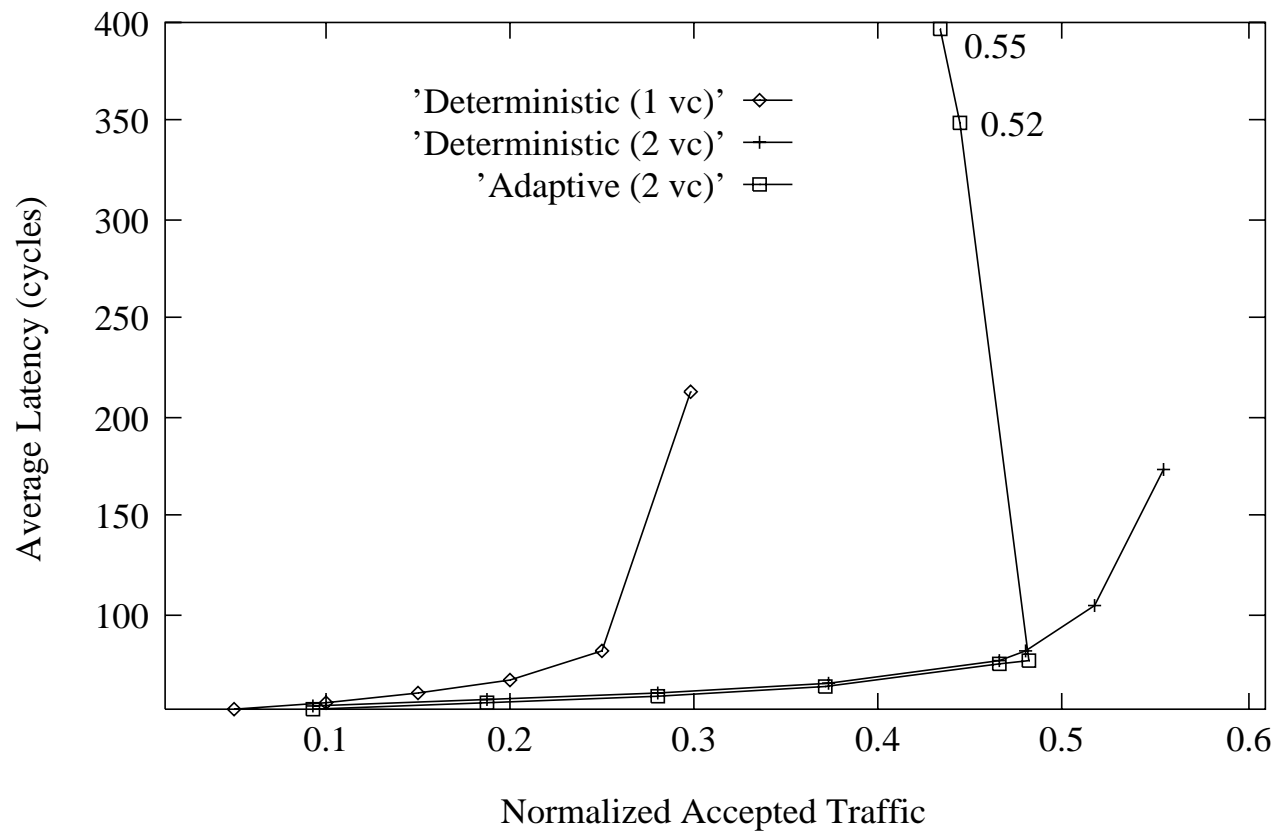


Extended channel dependency graph for R_1

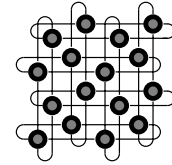




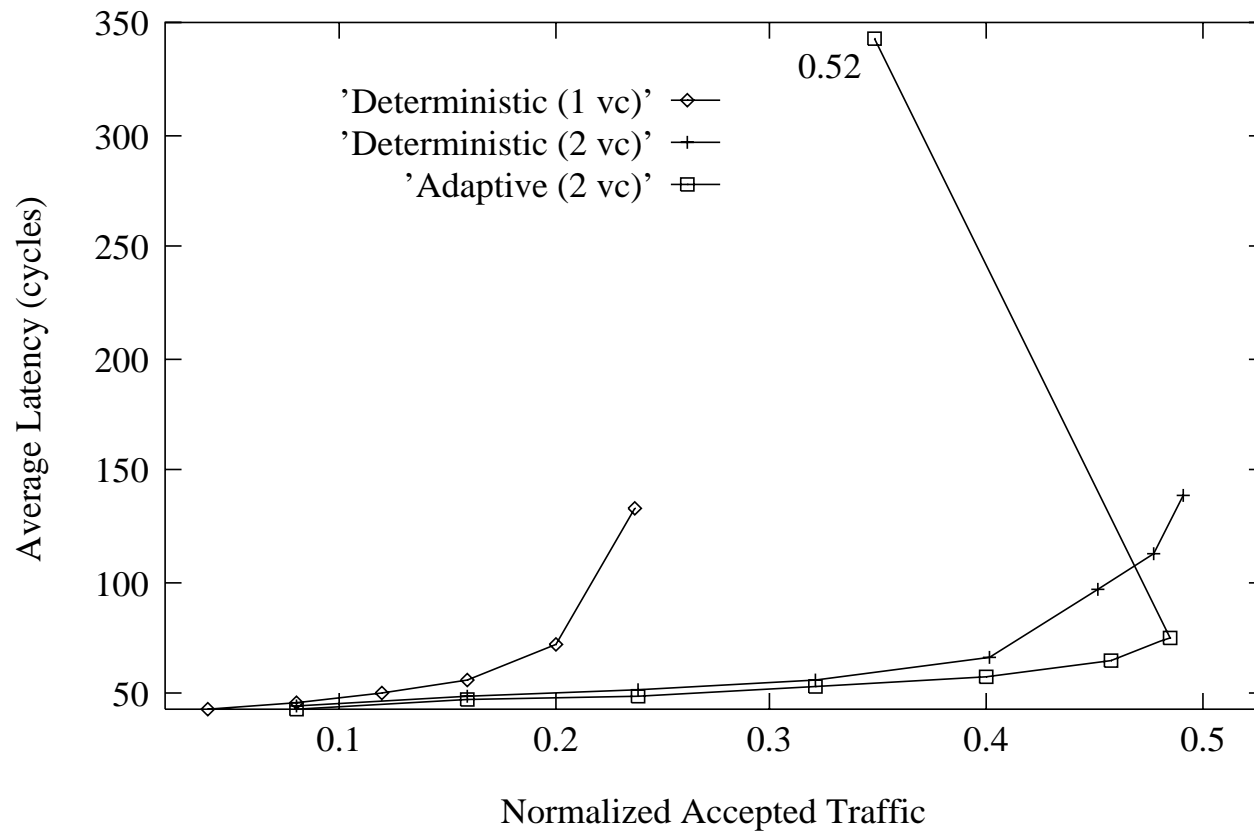
Performance evaluation for the 2-D mesh



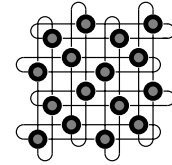
Network size:
256 processors.
Message length:
16 flits.
Random traffic



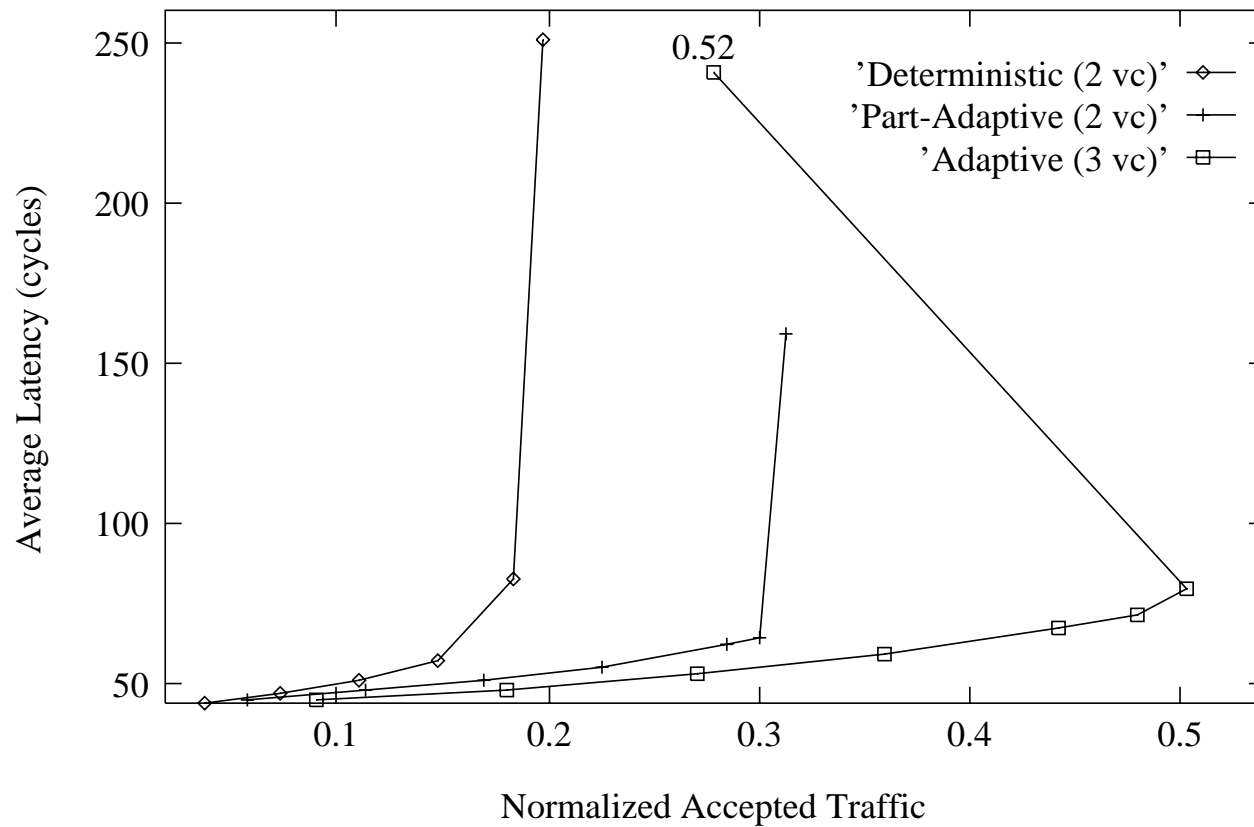
Performance evaluation for the 3-D mesh



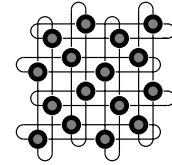
Network size:
512 processors.
Message length:
16 flits.
Random traffic



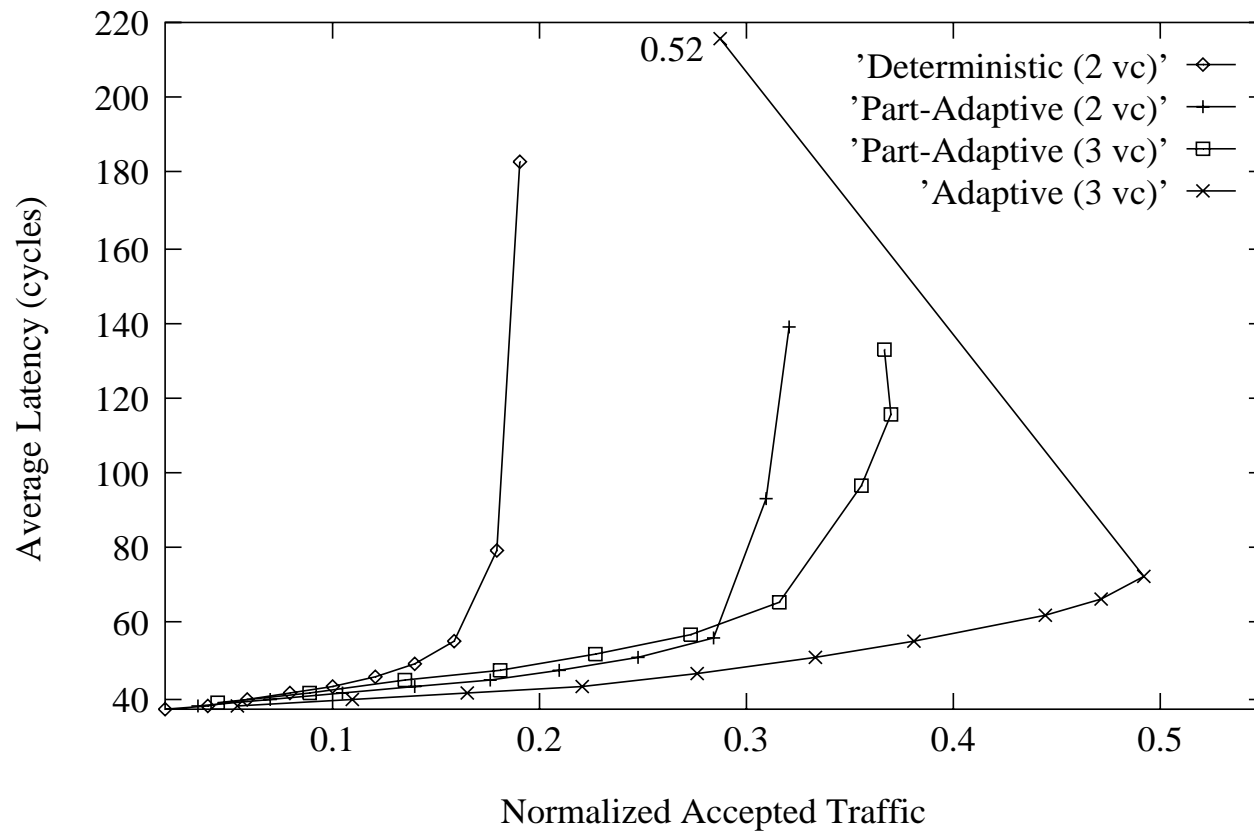
Performance evaluation for the 2-D torus



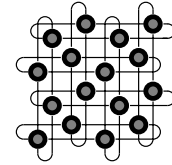
Network size:
256 processors.
Message length:
16 flits.
Random traffic



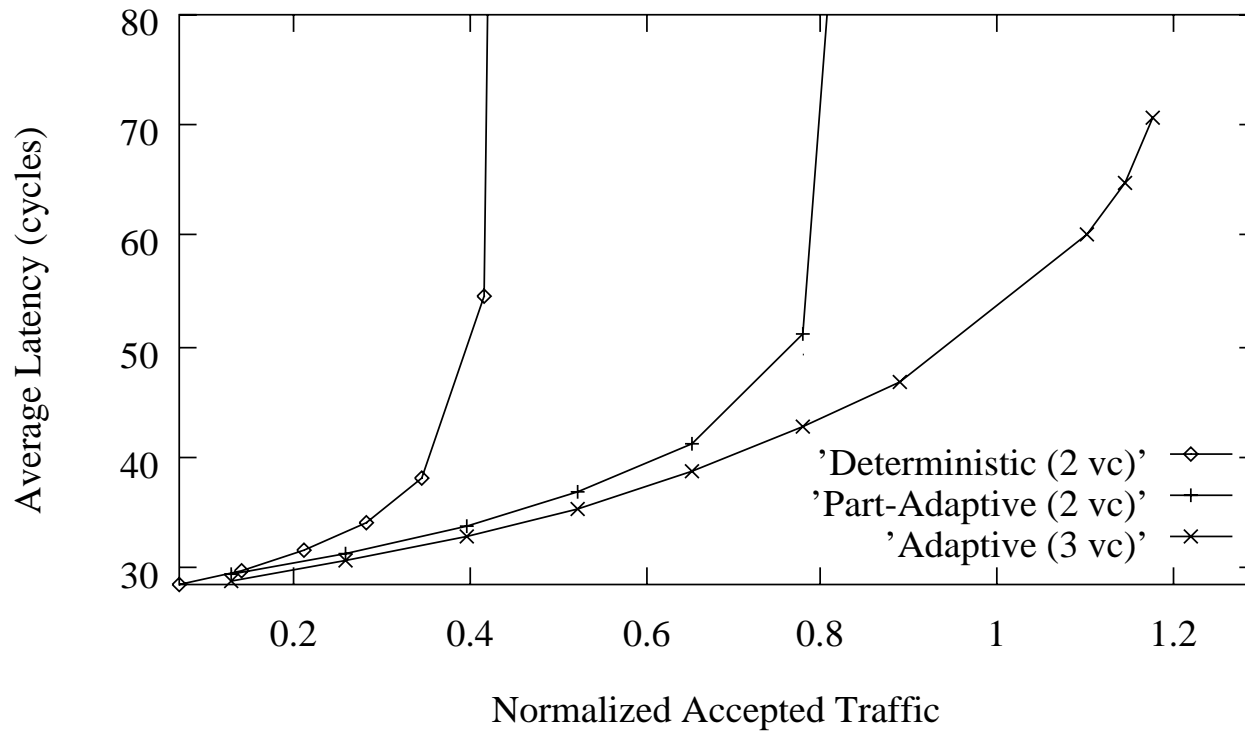
Performance evaluation for the 3-D torus



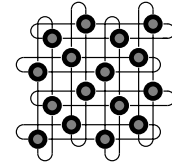
Network size:
512 processors.
Message length:
16 flits.
Random traffic



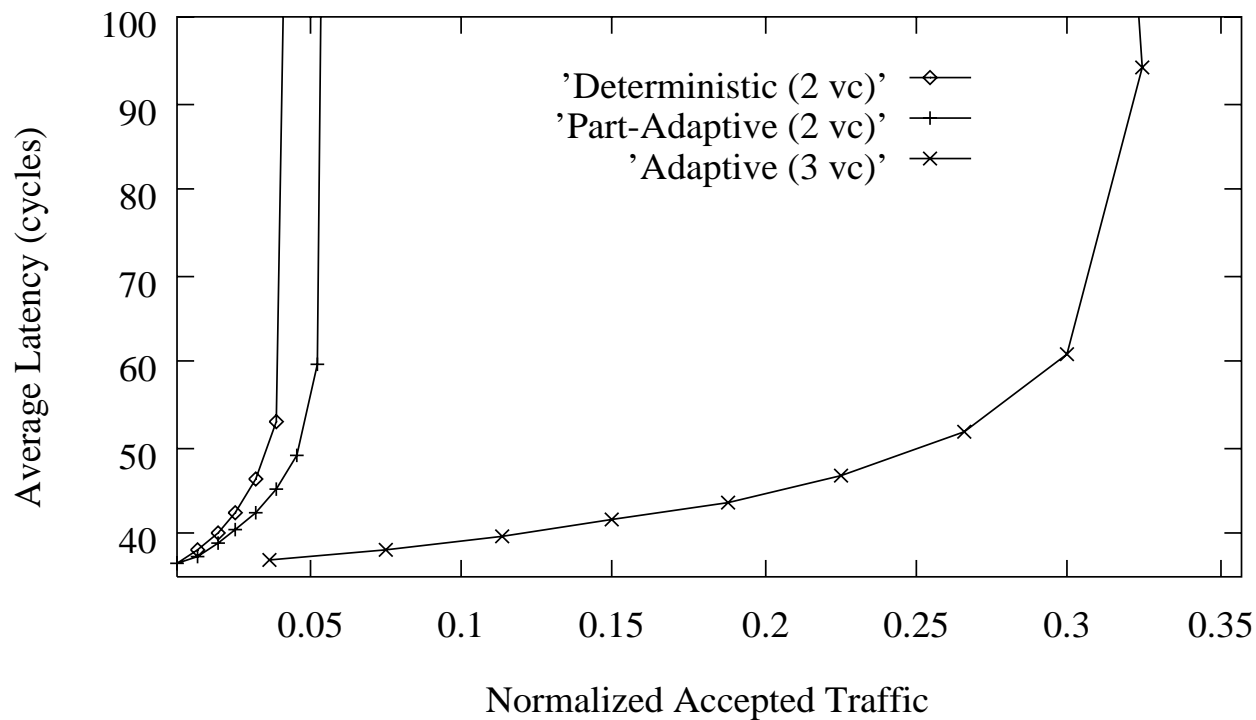
Performance evaluation for the 3-D torus (II)



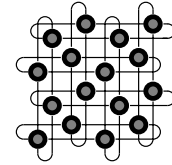
Network size:
512 processors.
Message length:
16 flits.
Local traffic



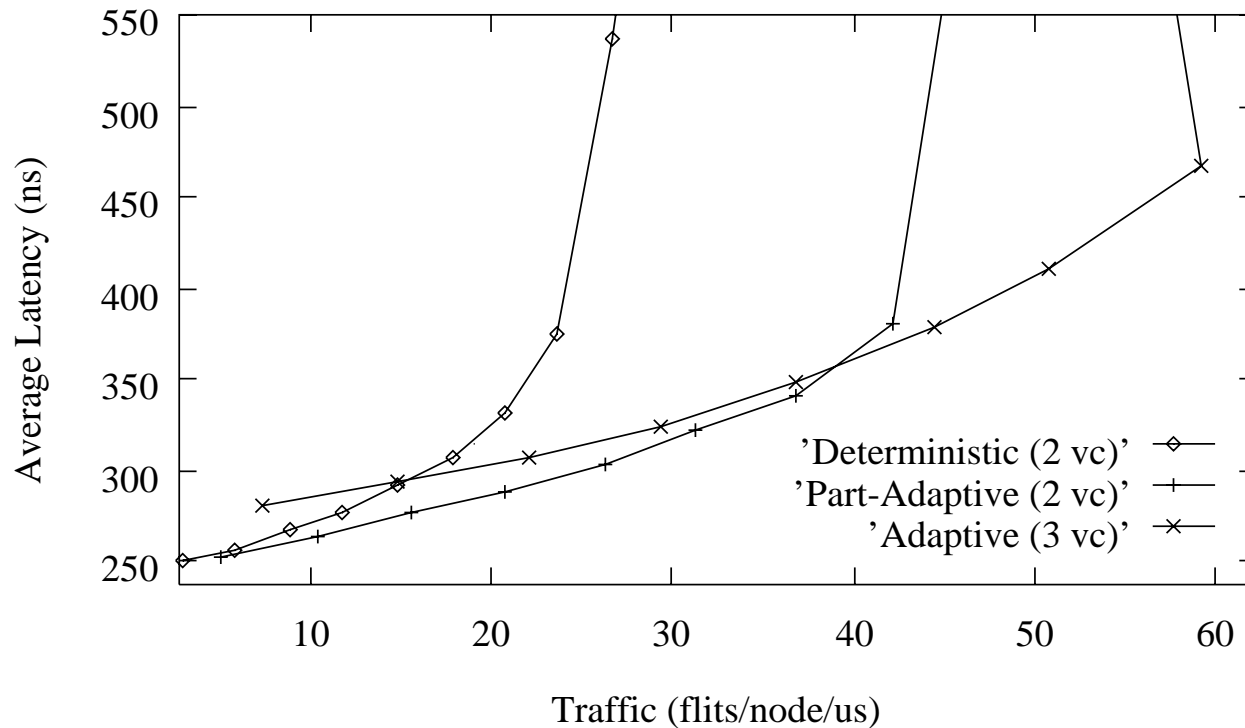
Performance evaluation for the 3-D torus (III)



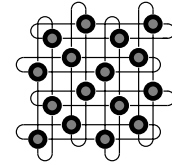
Network size:
512 processors.
Message length:
16 flits.
Bit-reversal
traffic pattern



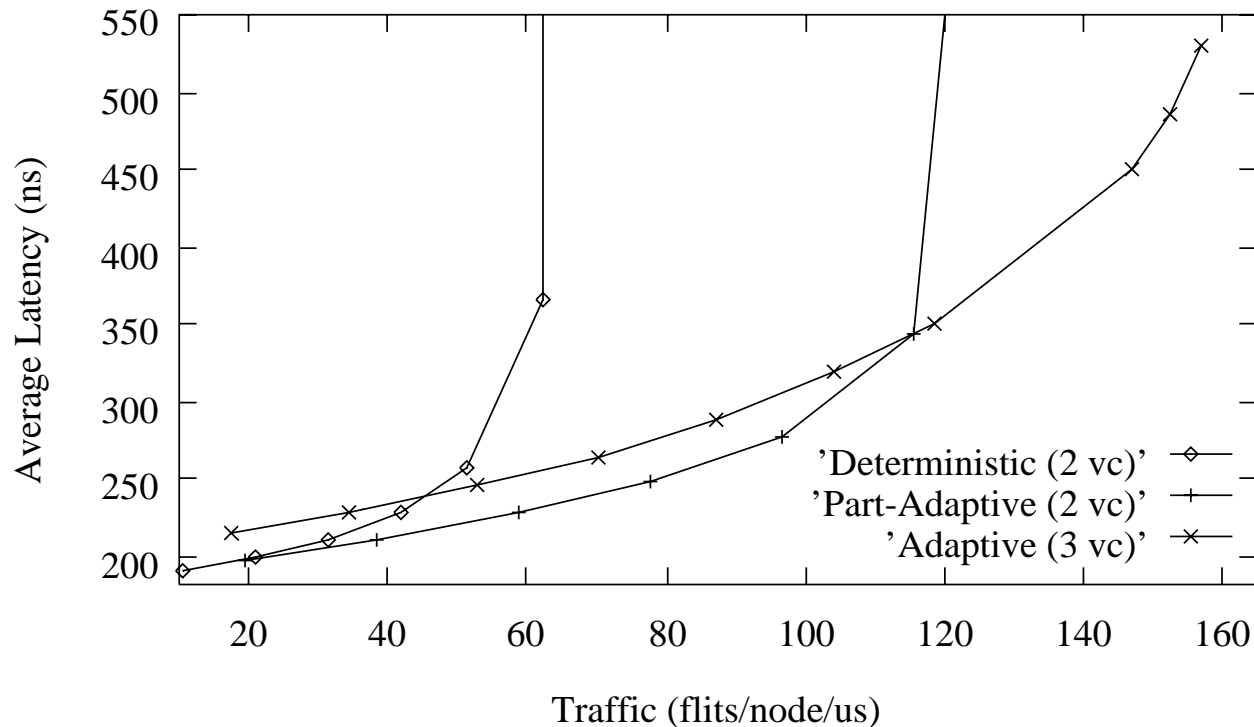
Accurate performance evaluation for the 3-D torus



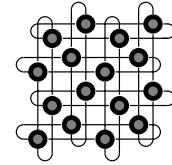
Network size:
512 processors.
Message length:
16 flits.
Random traffic



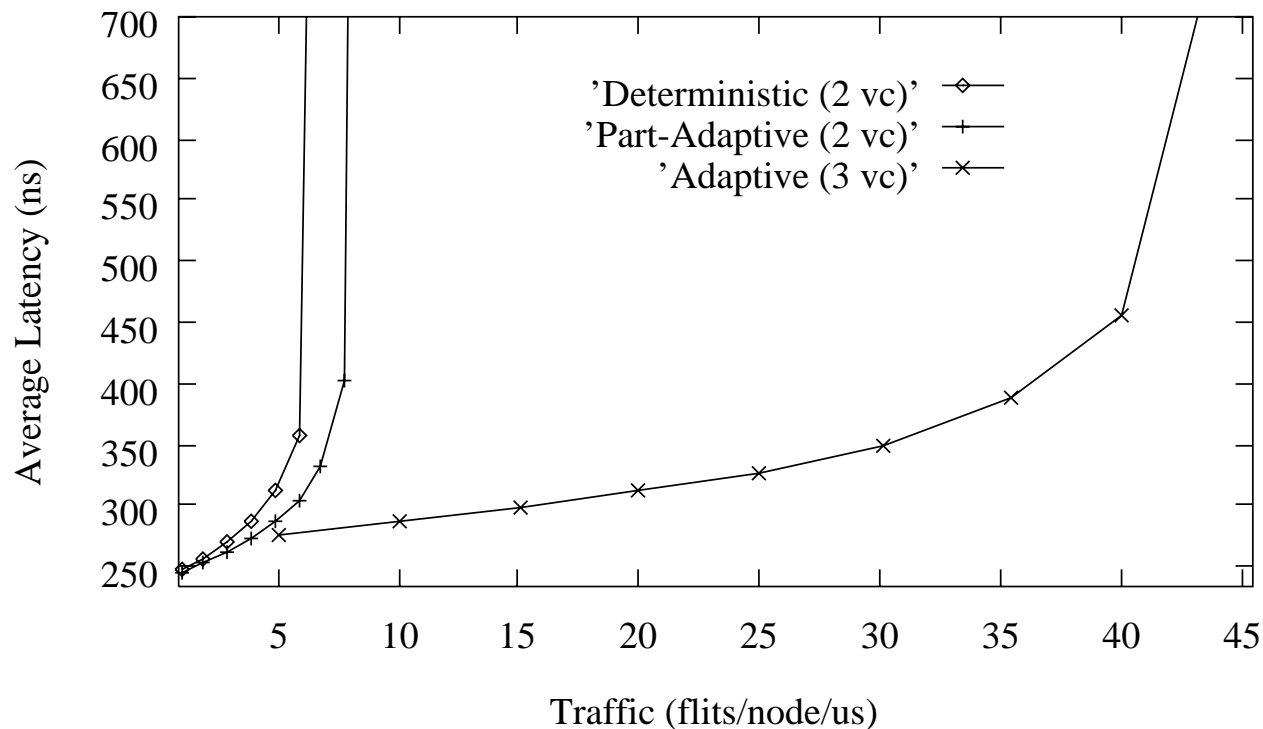
Accurate performance evaluation for the 3-D torus (II)



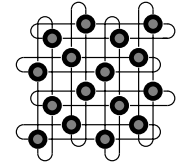
Network size:
512 processors.
Message length:
16 flits.
Local traffic



Accurate performance evaluation for the 3-D torus (III)



Network size:
512 processors.
Message length:
16 flits.
Bit-reversal
traffic pattern



Application to deadlock recovery

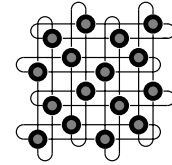
Routing resources (channels or buffers) are split into two classes: adaptive and escape

Adaptive resources can be freely used by all the packets

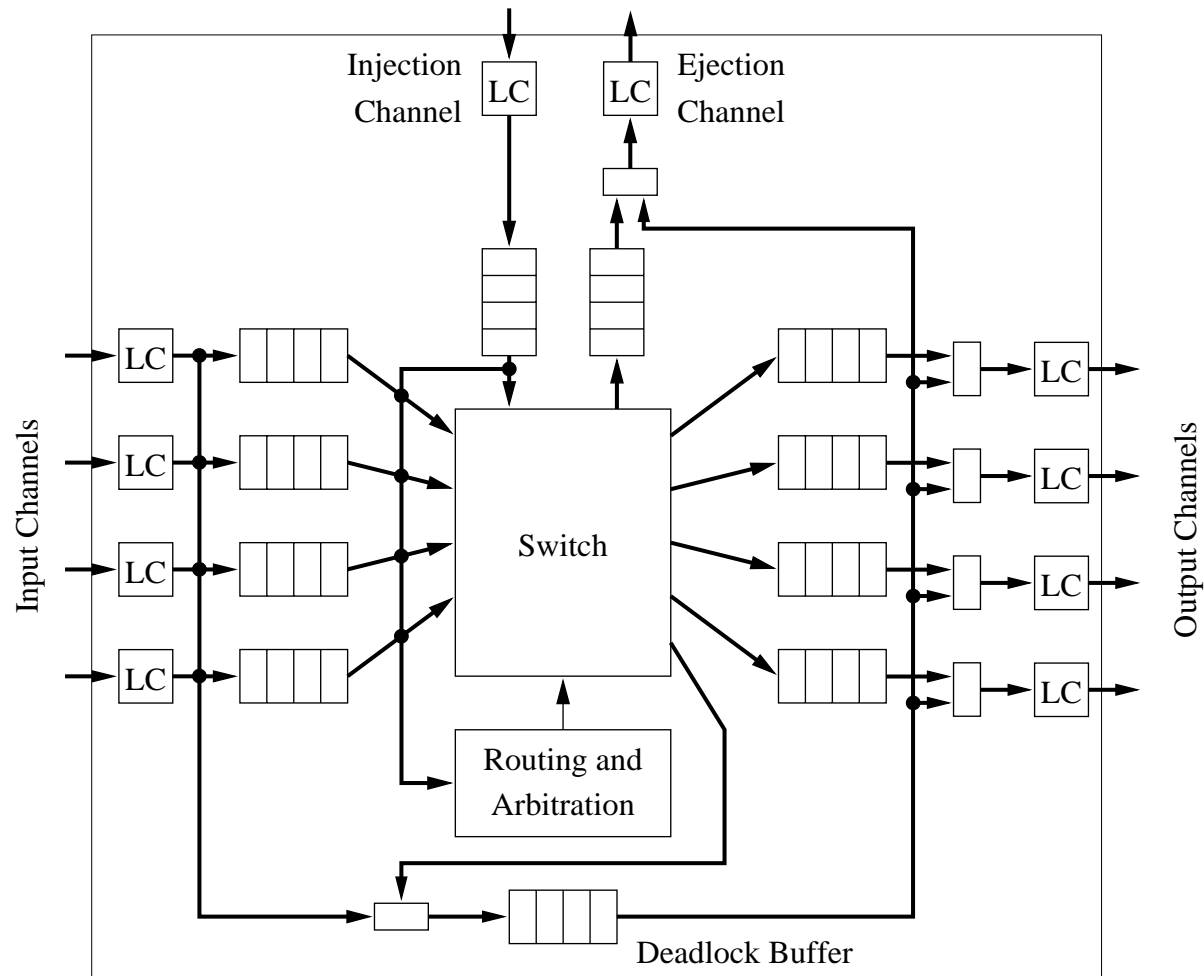
When a packet is waiting for longer than a timeout, it moves to an escape resource

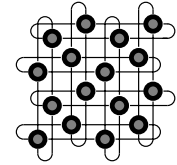
Once a packet uses an escape resource, it cannot use an adaptive resource again

This routing scheme eliminates all the indirect dependencies between adaptive and escape resources

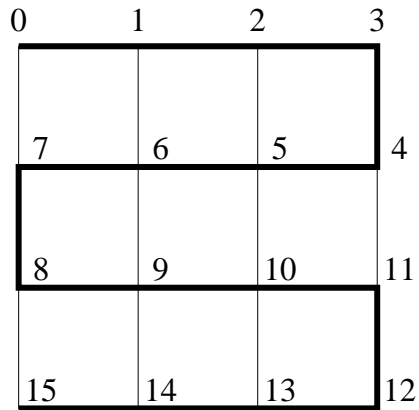


Router organization for Disha



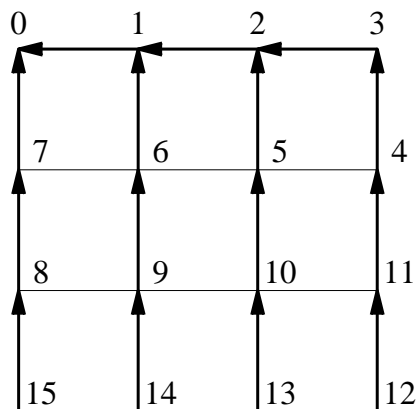


Routing on edge and deadlock buffers



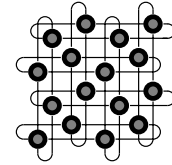
Deadlock buffers can only be used in increasing label order

When a deadlock is detected, the packet header can be routed to the deadlock buffer

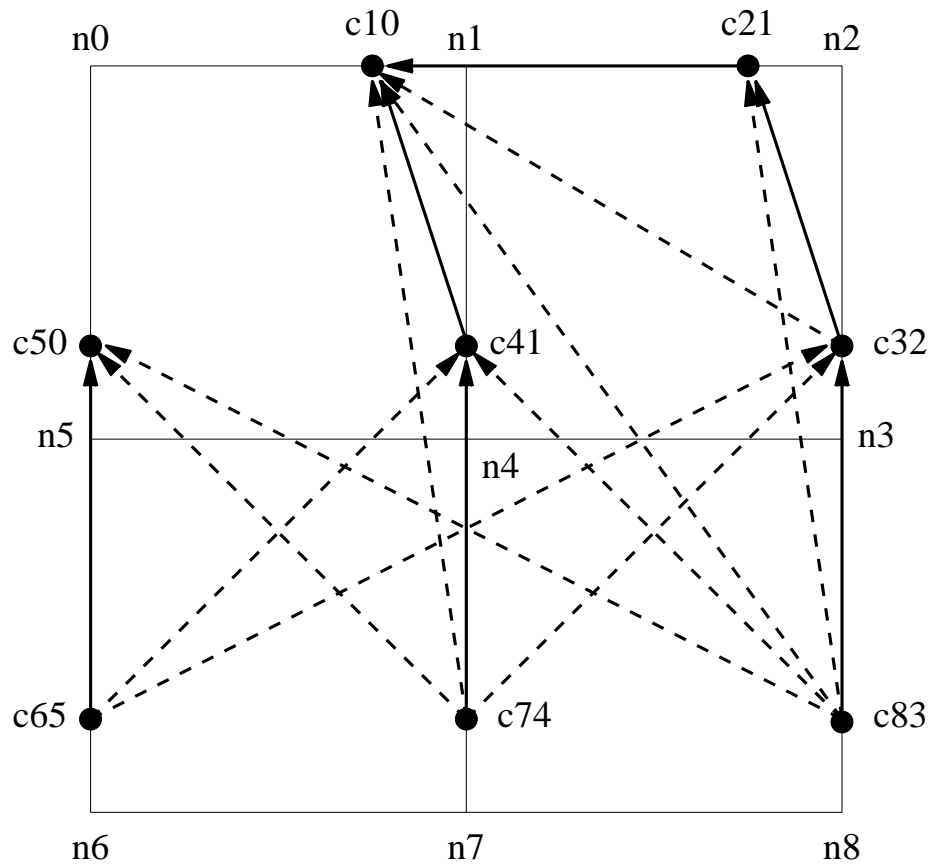


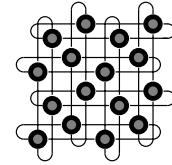
Edge buffers allow fully adaptive minimal routing

Escape channels are defined so that the routing subfunction is able to deliver messages for any destination (including deadlock buffers)

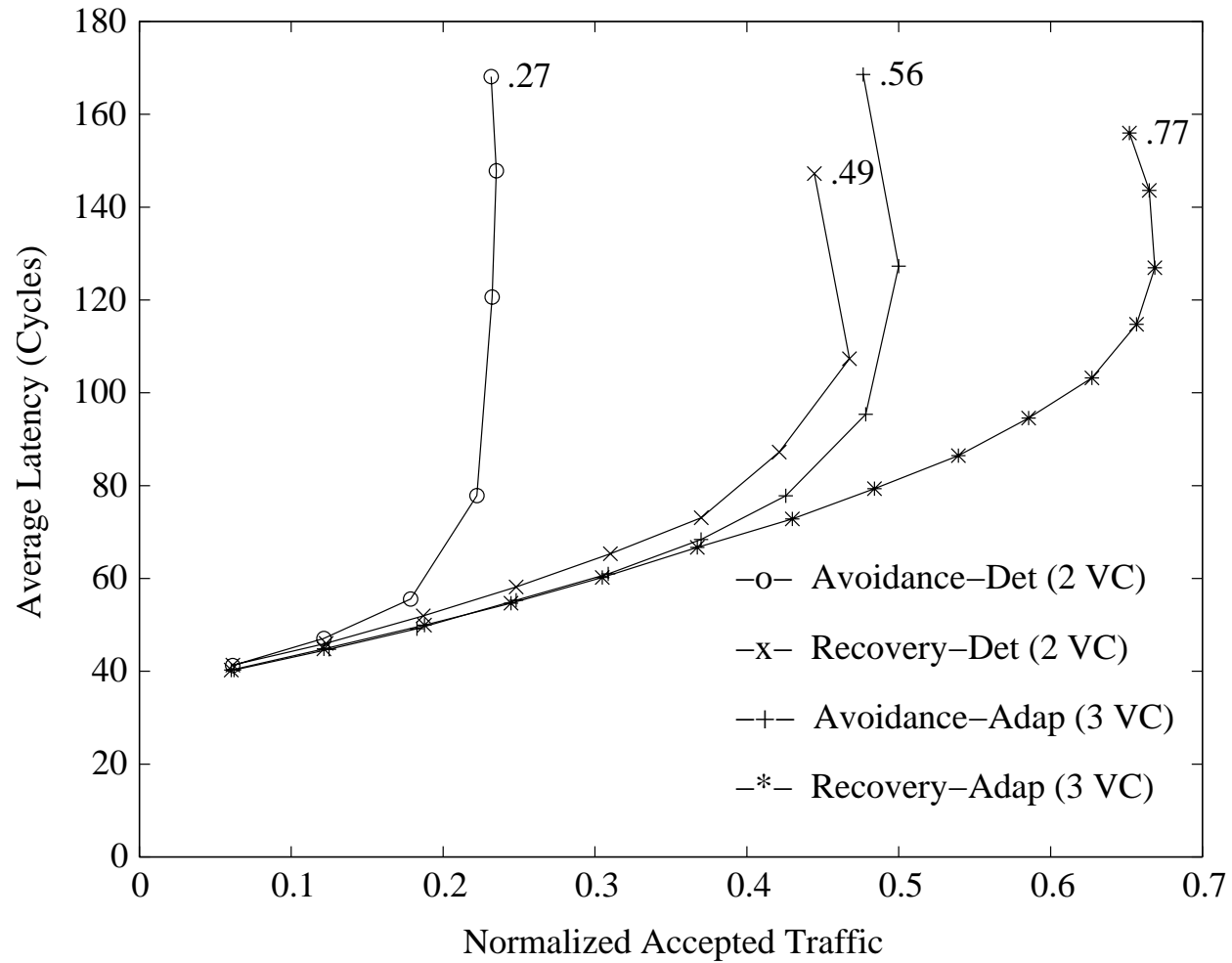


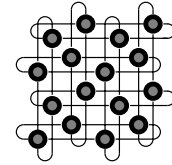
Extended channel dependency graph for edge buffers





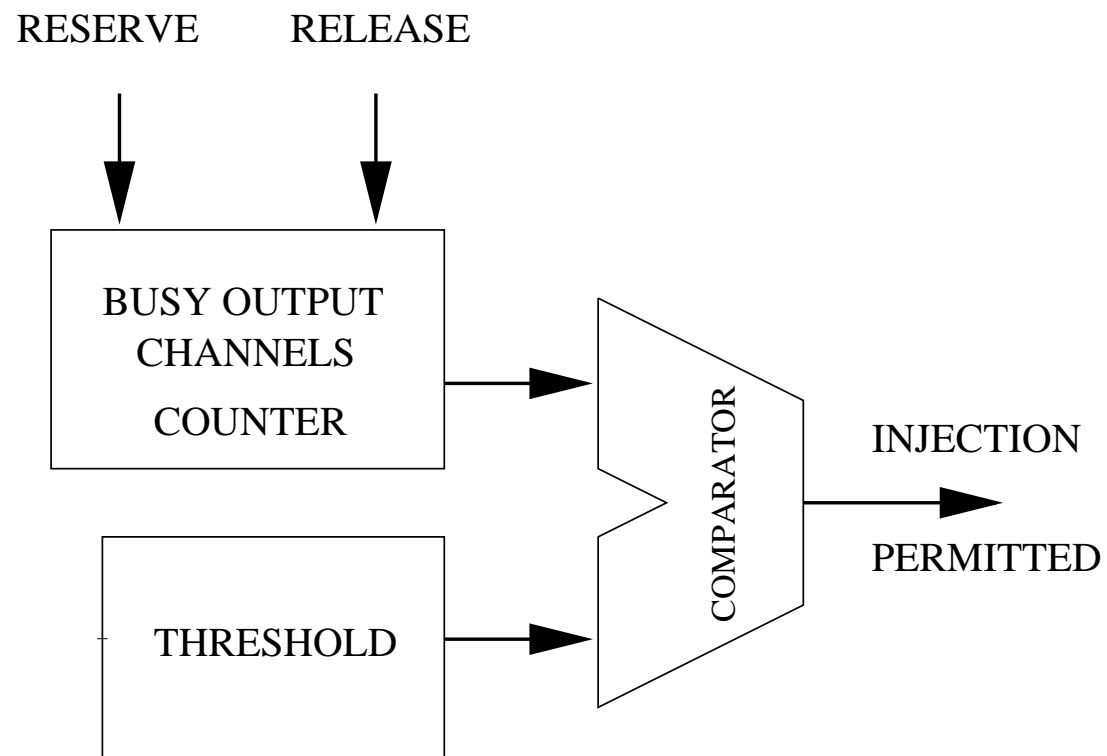
Performance evaluation

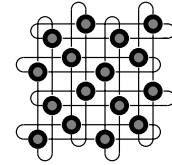




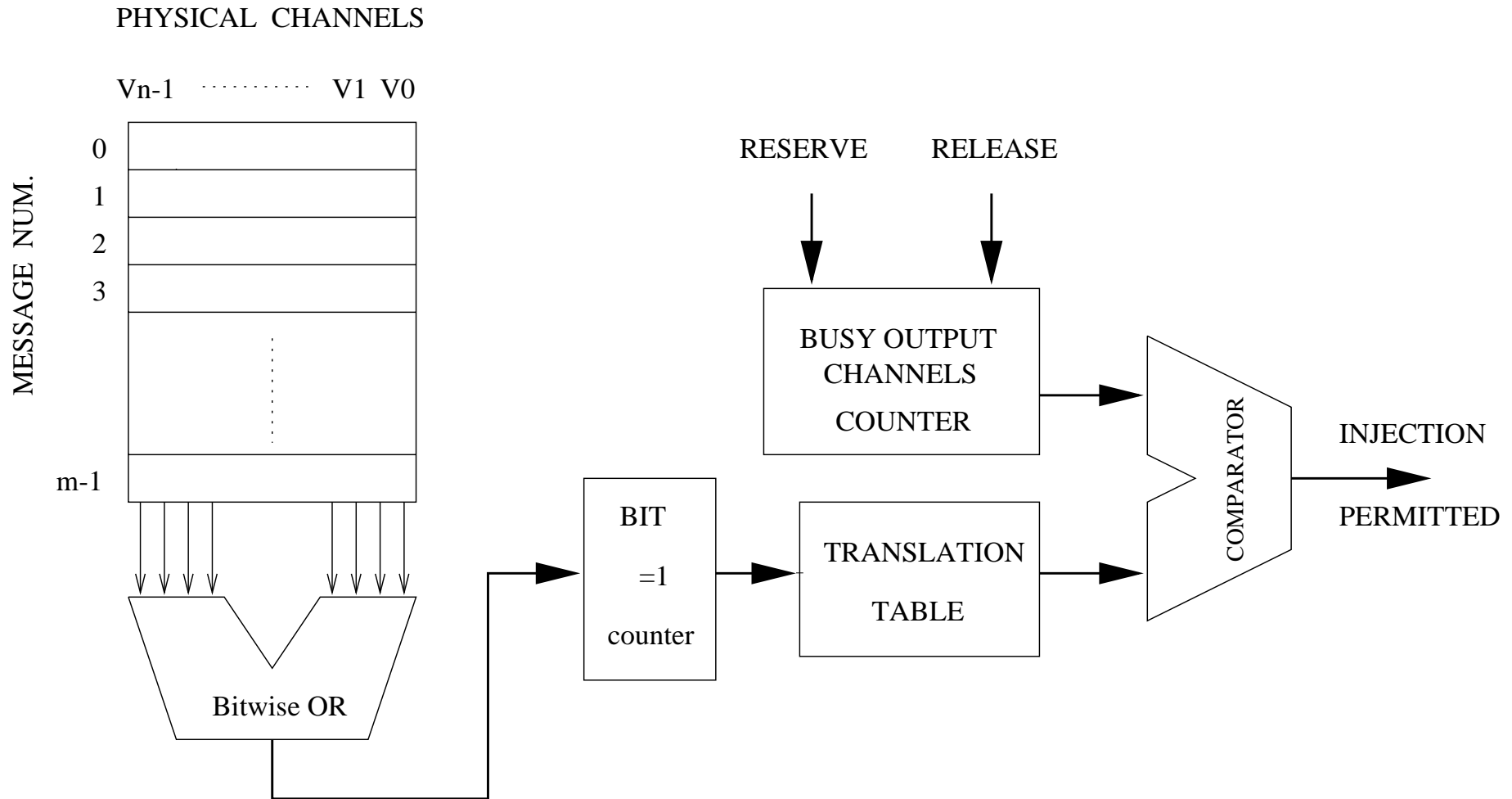
Injection limitation

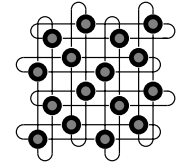
- Prevents performance degradation at saturation
- Reduces the frequency of deadlock occurrence to negligible values



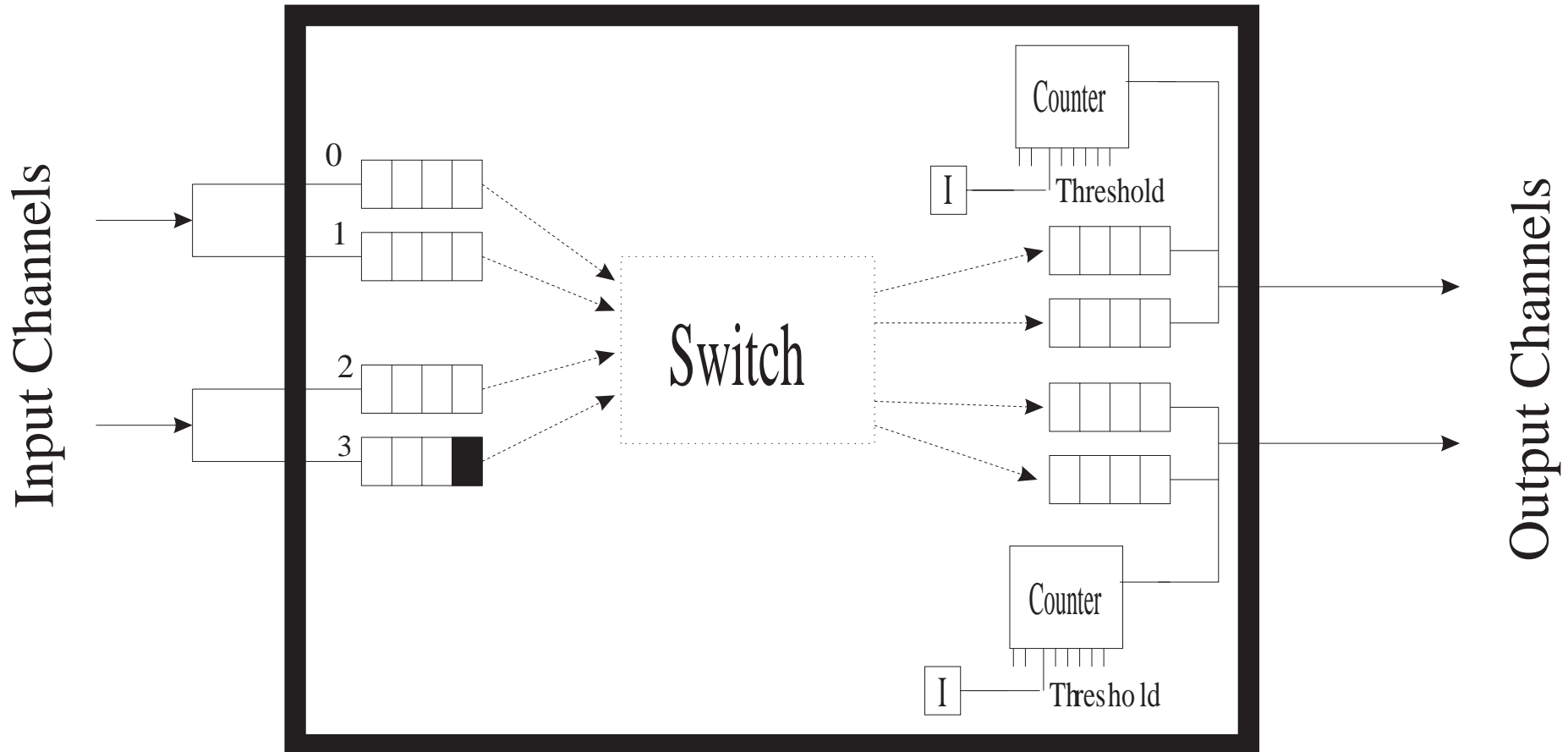


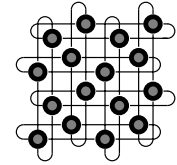
Improved injection limitation mechanism





Improved deadlock detection mechanism





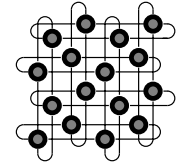
Application to networks of workstations

Networks of workstations are emerging as a cost-effective alternative to parallel computers.

Switch-based interconnects like Autonet, Myrinet and ServerNet have been proposed to build networks of workstations with irregular topology.

The irregularity provides:

- Wiring flexibility.
- Scalability.
- Incremental expansion capability.



Drawback: The irregularity makes deadlock avoidance and routing quite complicated.

Simplest solution: Avoid deadlock by eliminating all the cyclic dependencies between channels

⇒ Many messages are routed following non-minimal paths.

→ Higher message latency

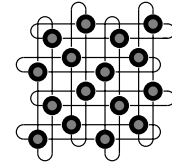
→ Waste of resources

→ Lower throughput

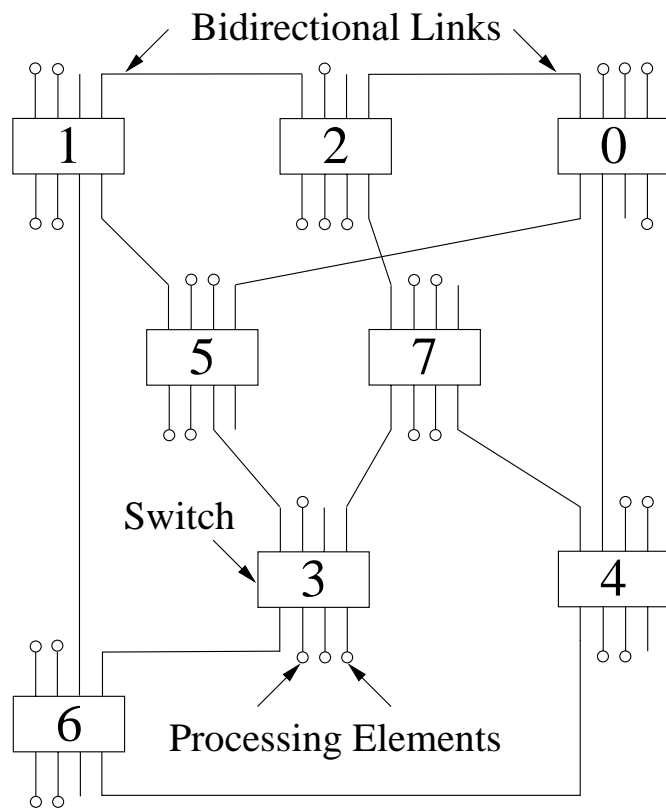
Alternative solution: Allow cyclic dependencies between channels

→ Reduces contention by increasing routing adaptivity

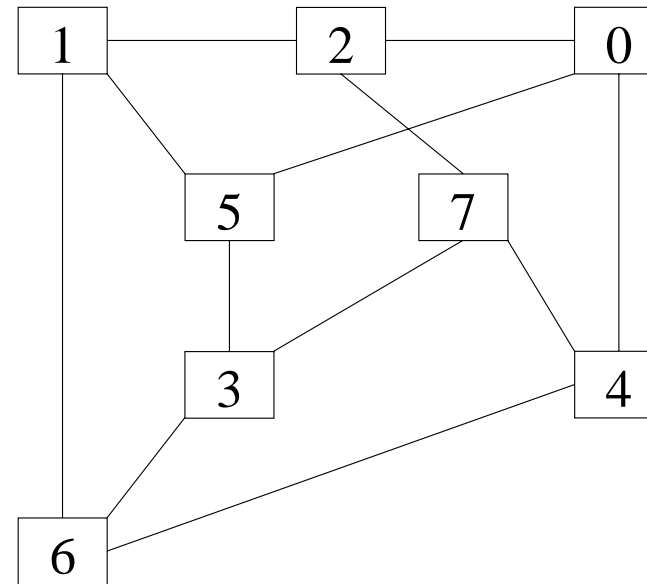
→ Allows more messages to follow minimal paths



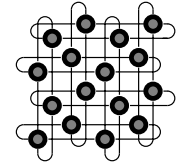
Switch-based networks with irregular topologies



Switch-Based Network



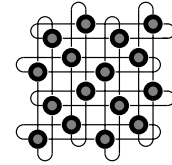
Graph Representation



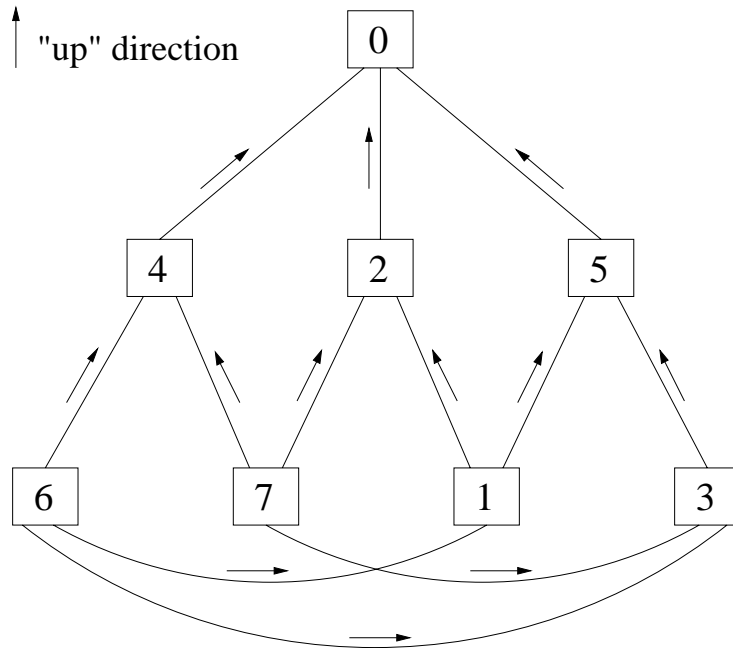
The Autonet routing algorithm

General characteristics:

- Deadlock-free routing scheme (*up/down* routing).
- Provides partially adaptive communication between nodes.
- Distributed.
- Implemented using table-lookup.



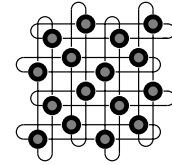
The up/down routing algorithm



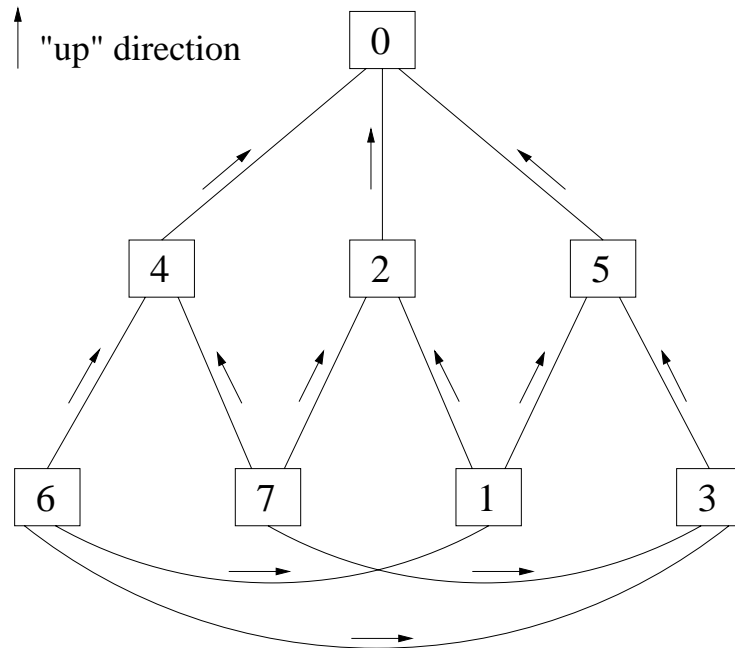
Routing is based on an assignment of direction to the operational links.

Routing rule: a legal route must traverse zero or more links in the “up” direction followed by zero or more links in the “down” direction.

- Each cycle has at least one link in the “up” direction and one link in the “down” direction.
- Cyclic dependencies are avoided: messages cannot cross a link in the “up” direction after one in the “down” direction.



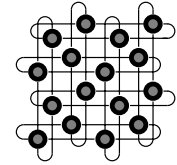
Routing efficiency



- From 7 to 0: OK
- From 2 to 5: lack of adaptivity
- From 4 to 1: non-minimal routing

The basic routing rule prevents from using minimal routing and adaptivity in most cases because of “down” to “up” conflicts.

Probability of non-minimal routing increases with network size.

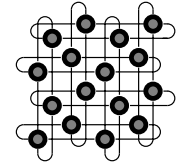


A design methodology for adaptive routing algorithms

interconnection
network
+
deadlock-free
routing function

new
methodology
⇒

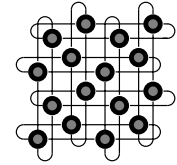
physical channels
duplicated or split into
two virtual channels
(*original* and *new*)
+
extended routing
function



Extended routing function

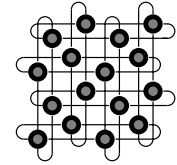
- Newly injected messages can use the new channels without any restriction. For performance reasons, only minimal paths are allowed
- Original channels are used *exactly* in the same way as in the original routing function
- Once a message reserves one of the original channels, it cannot use any of the new channels again
- When the routing table provides both kinds of channels, give preference to new channels

The extended routing function is deadlock-free



Improving the efficiency of the methodology

- *Idea:* Focus on minimal routing, even if adaptivity is reduced
- Restrict the transition from new channels to original channels
- Improved adaptive routing function:
 - Newly injected messages can **only** use new channels
 - At intermediate switches, a higher priority is assigned to the new channels belonging to minimal paths
 - If all the new channels are busy, then an original channel belonging to a minimal path (if any) is selected
 - If none exists, then the one that provides the shortest path is used (this ensures deadlock-freedom)
- Once a message reserves an original channel, it can no longer reserve a new one

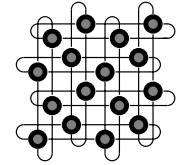


Performance evaluation

Evaluation of four routing schemes:

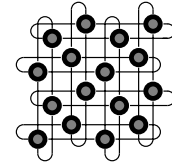
- Basic up/down routing scheme (UD).
- Up/down routing scheme using two virtual channels per physical channel (UD-2VC).
- Adaptive routing scheme using two virtual channels per physical channel (A-2VC).
- Improved adaptive routing scheme using two virtual channels per physical channel (MA-2VC).

Performance evaluation carried out by simulation.

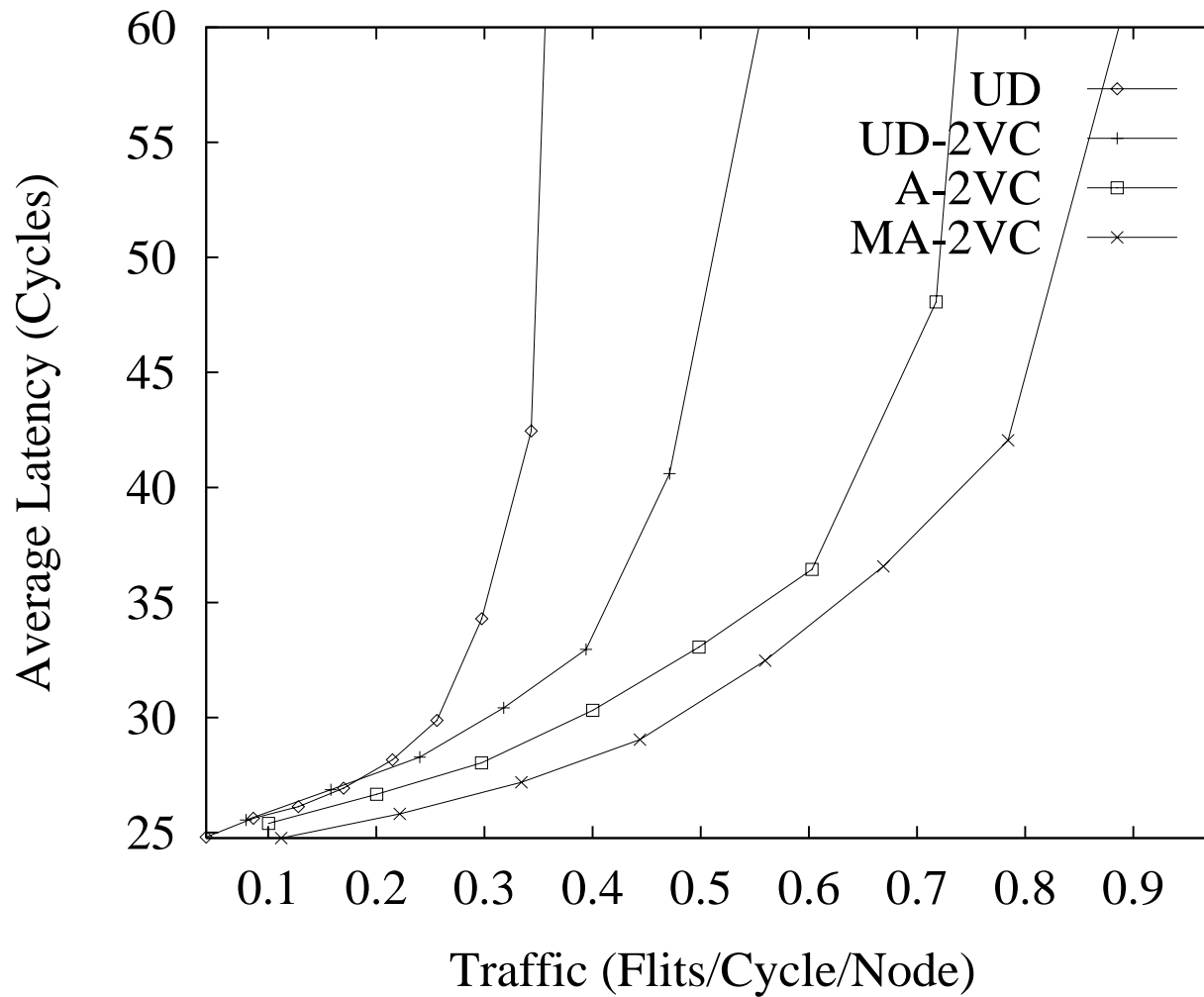


Network model:

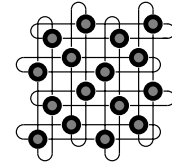
- Topology generated randomly (8-port switches)
- 4 nodes (processors) connected to each switch
- Two adjacent switches are connected by a single link
- One routing control unit per switch (assigned in a round-robin fashion)
- Message destination is randomly chosen among nodes
- It takes one clock cycle to compute the routing algorithm, to transfer one flit from an input buffer to an output buffer, or to transfer one flit across a physical channel



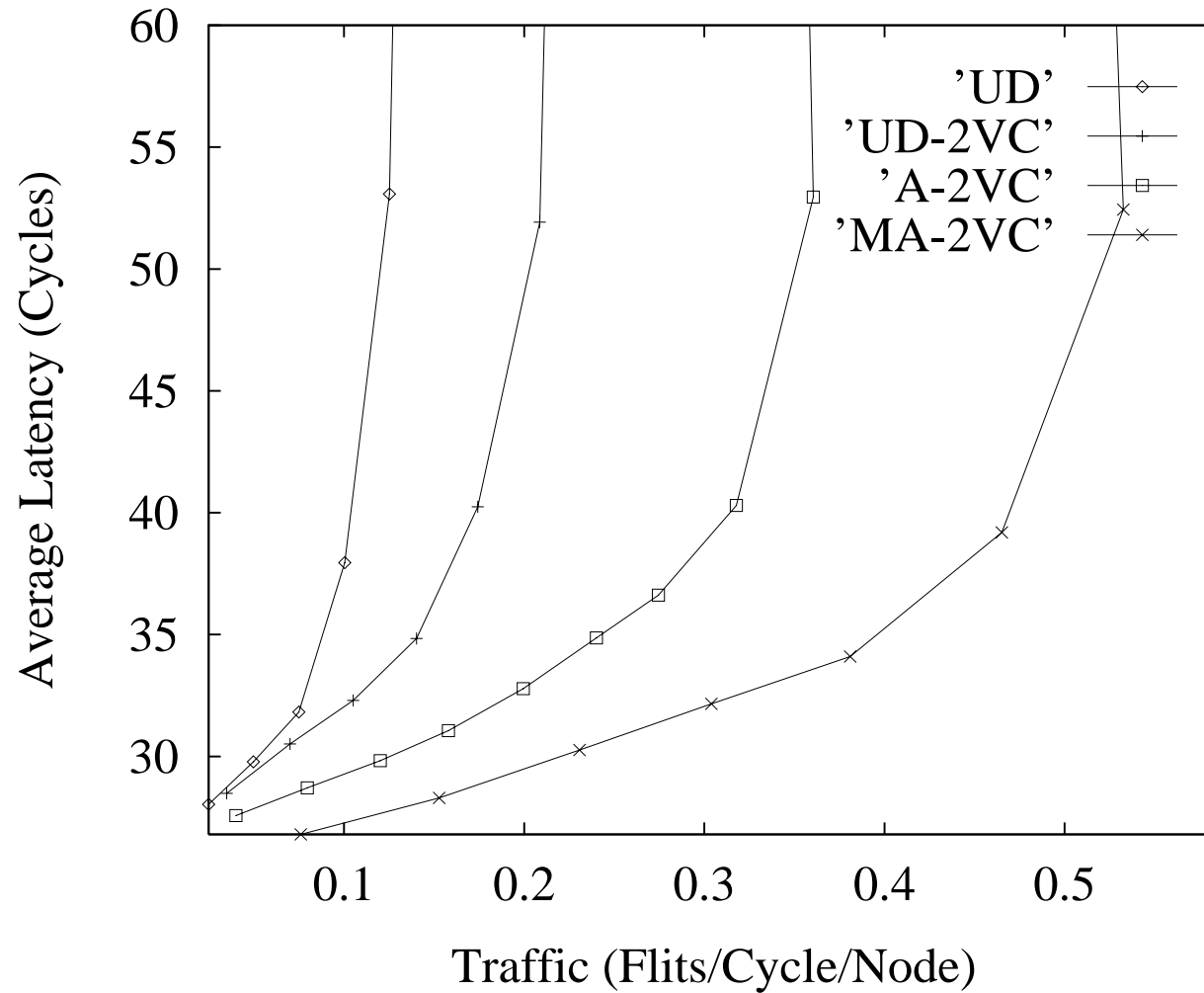
Simulation results (I)



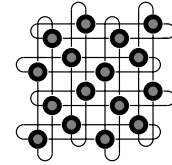
Network size: 16
switches.
Message length:
16 flits.



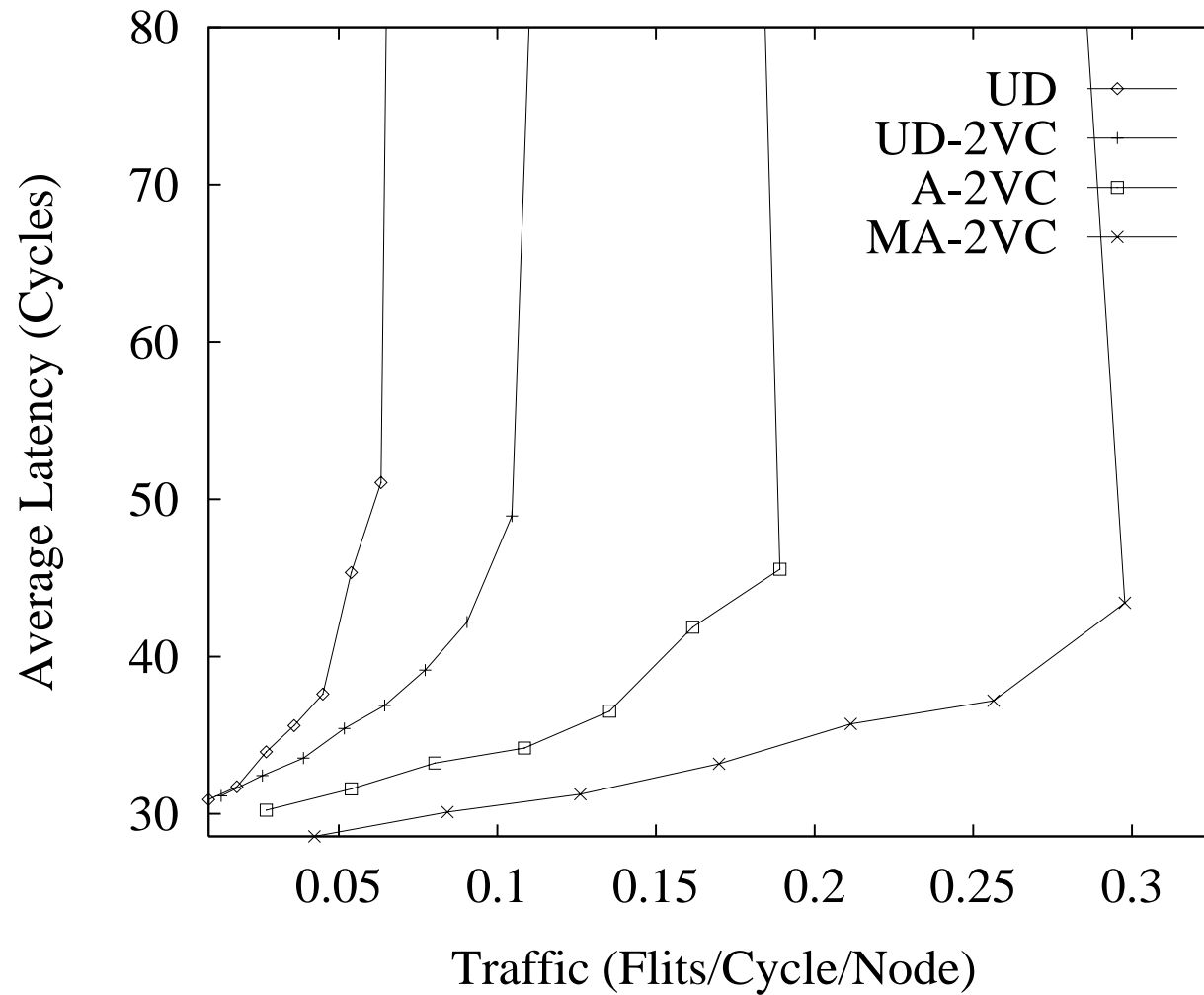
Simulation results (II)



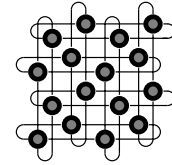
Network size: 32
switches.
Message length:
16 flits.



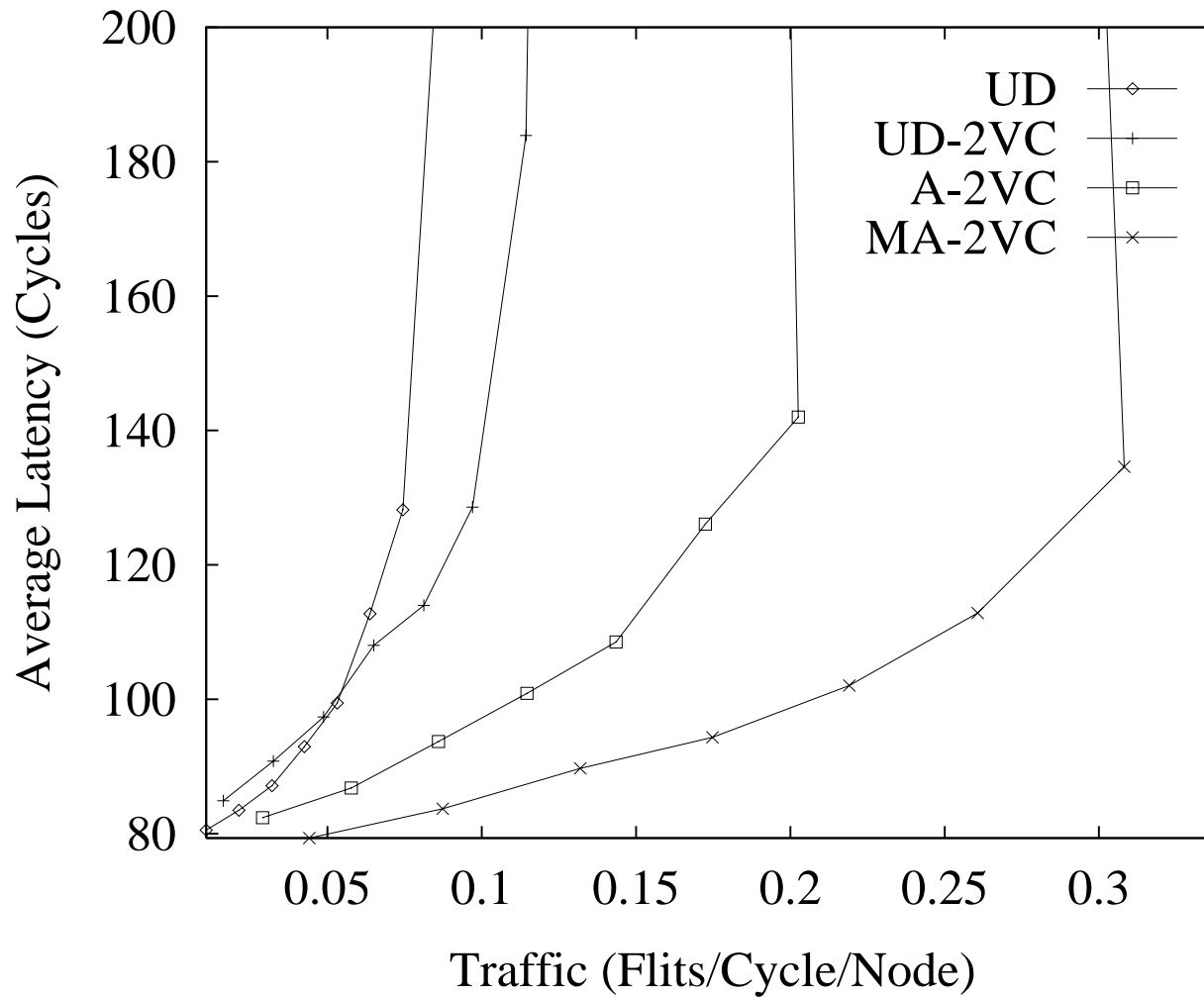
Simulation results (III)



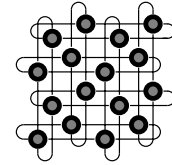
Network size: 64
switches.
Message length:
16 flits.



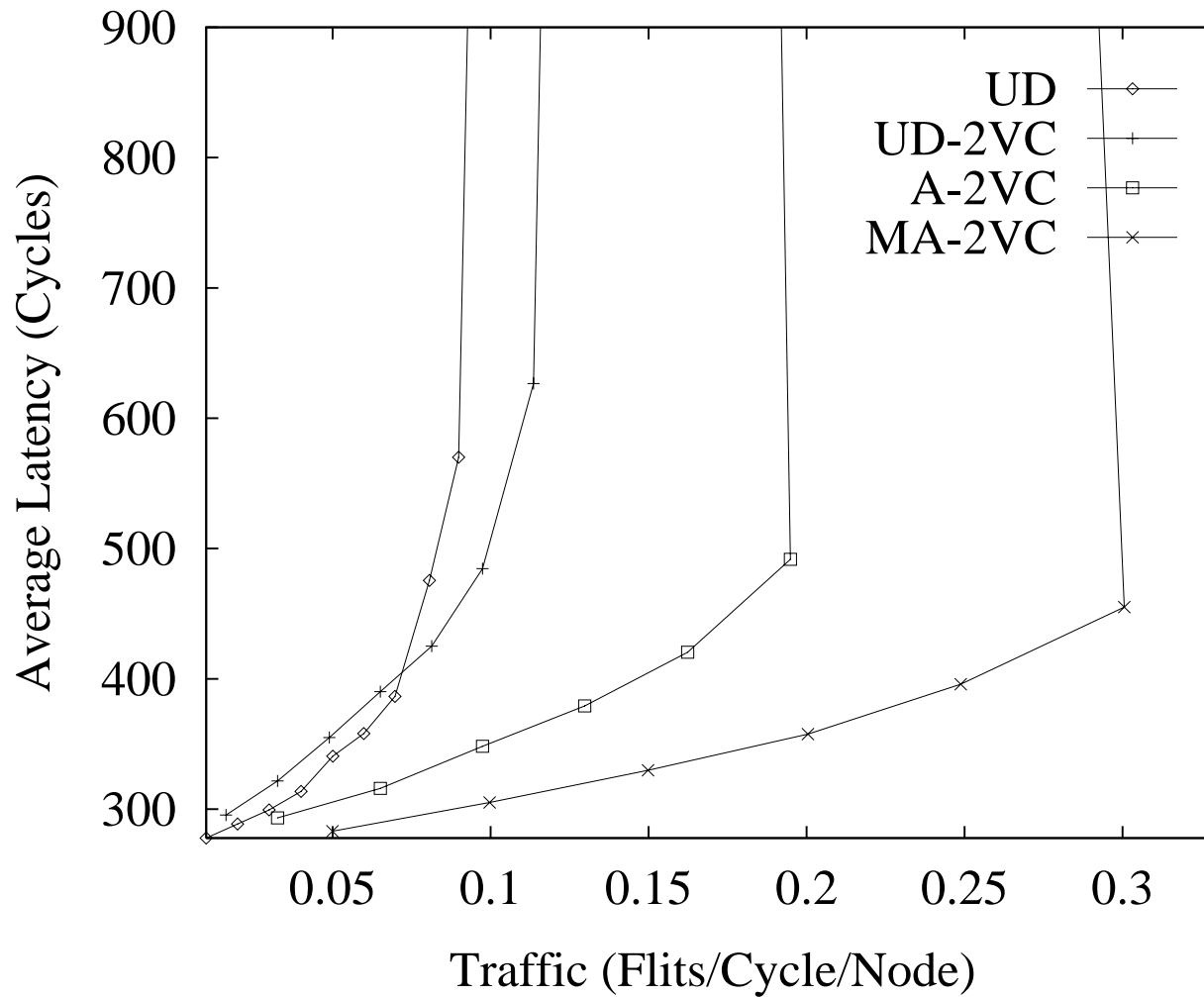
Simulation results (IV)



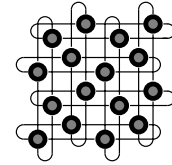
Network size: 64
switches.
Message length:
64 flits.



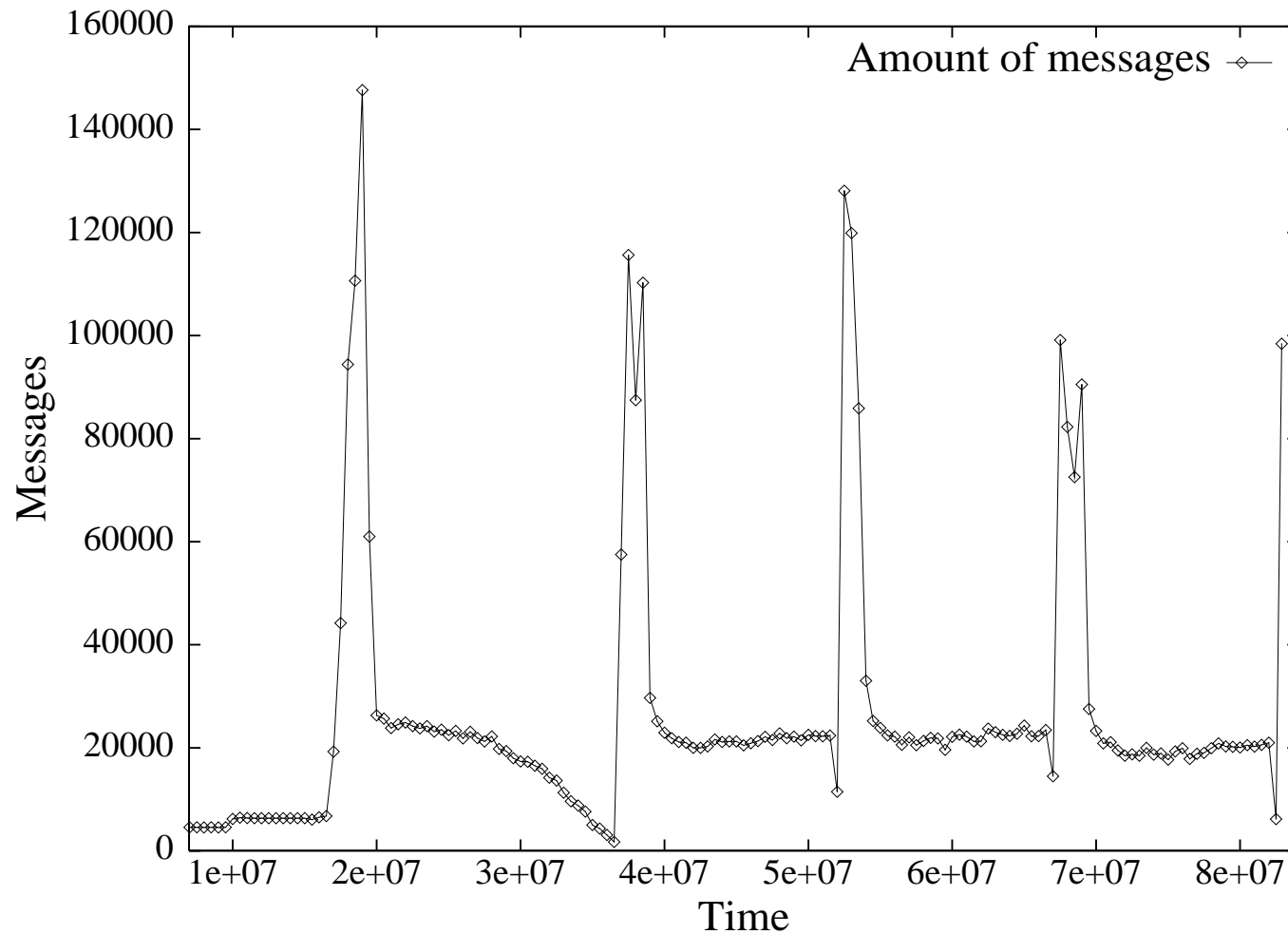
Simulation results (V)



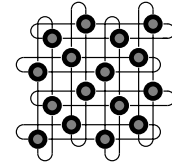
Network size: 64
switches.
Message length:
256 flits.



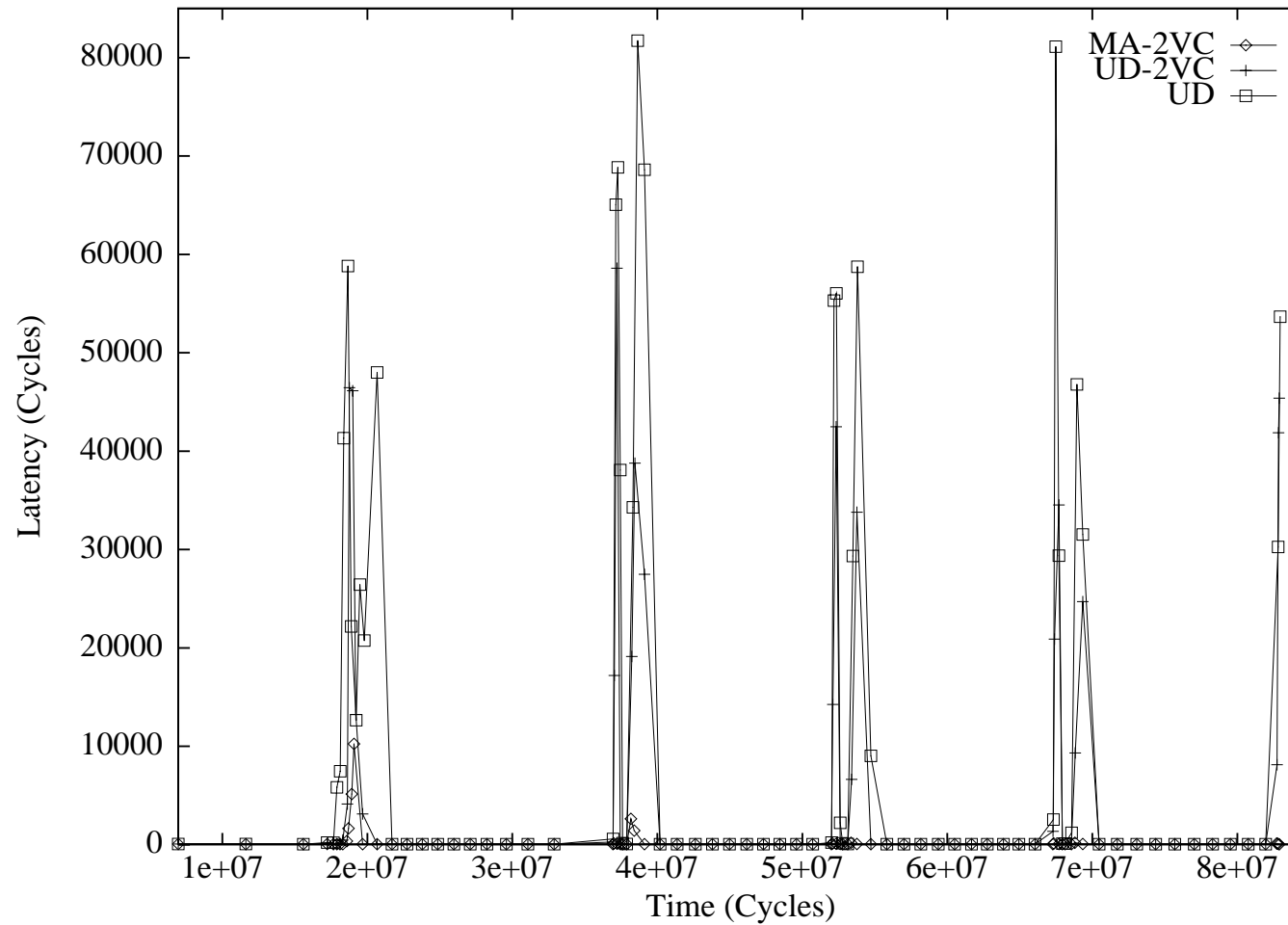
Simulation results for application traces

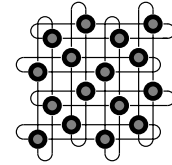


Traces from
Barnes-Hut
executed on
64 processors

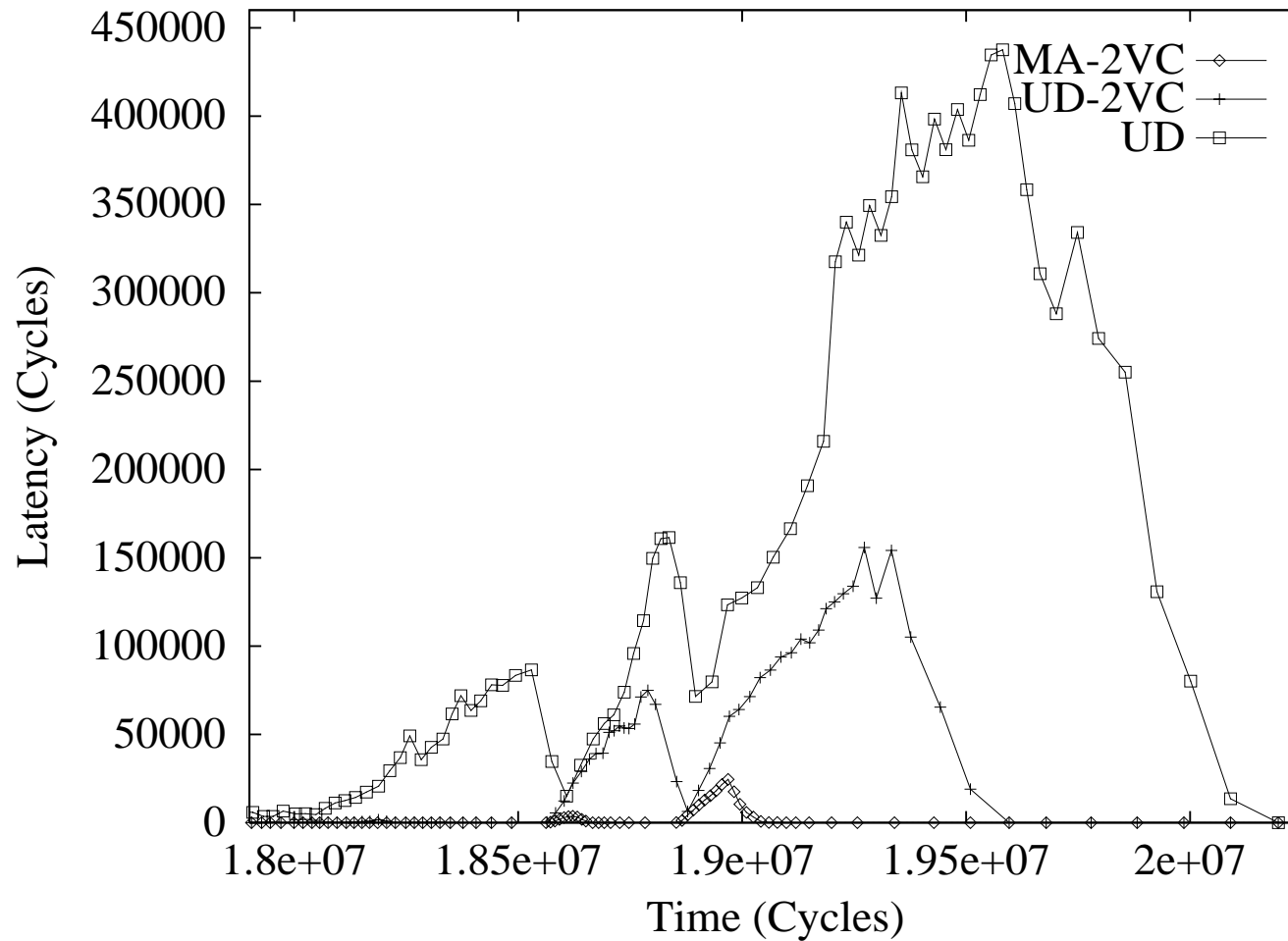


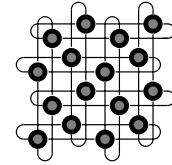
Simulation results for application traces



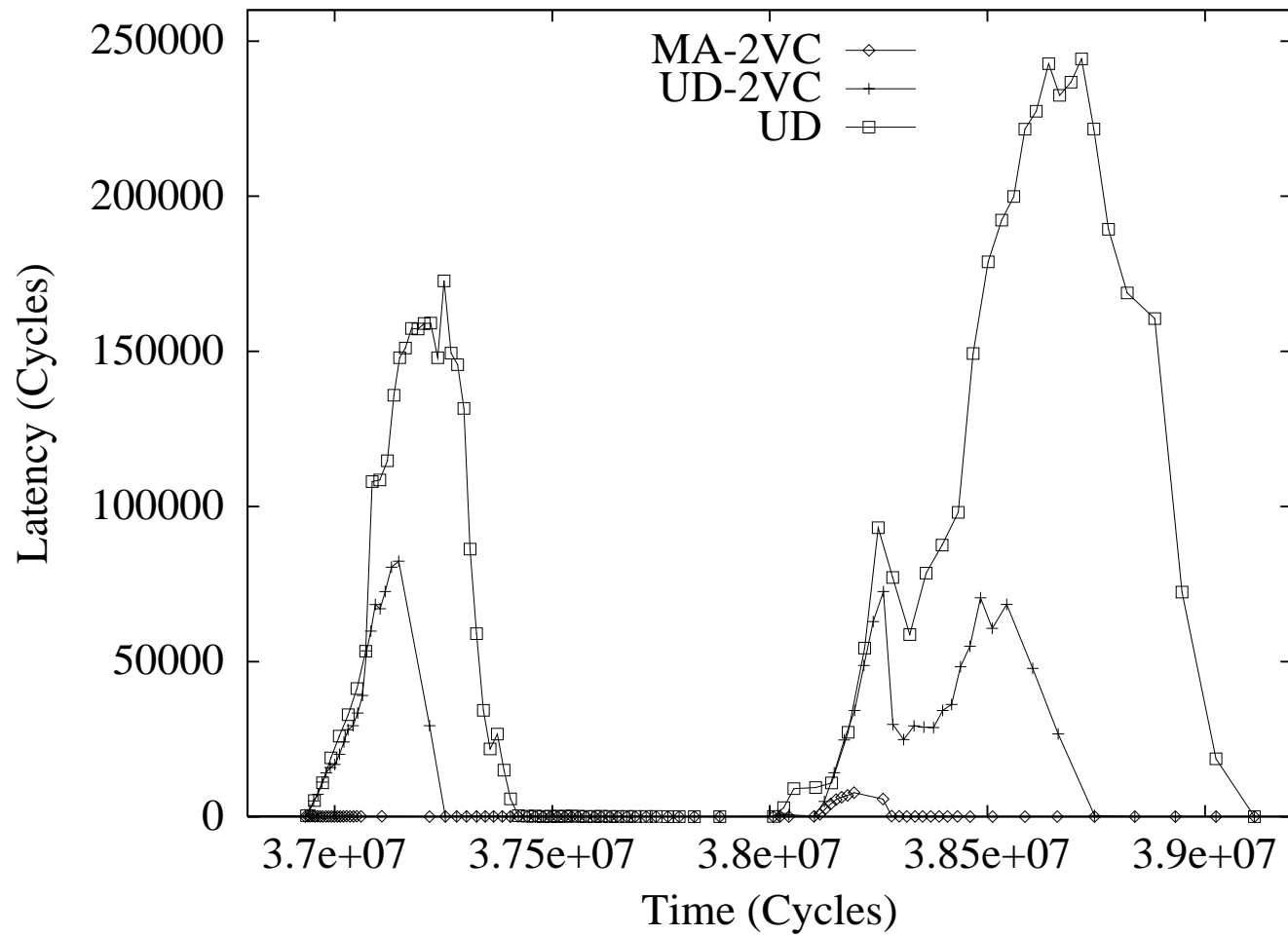


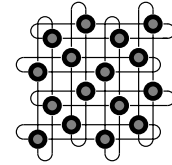
Zoom of the first peak



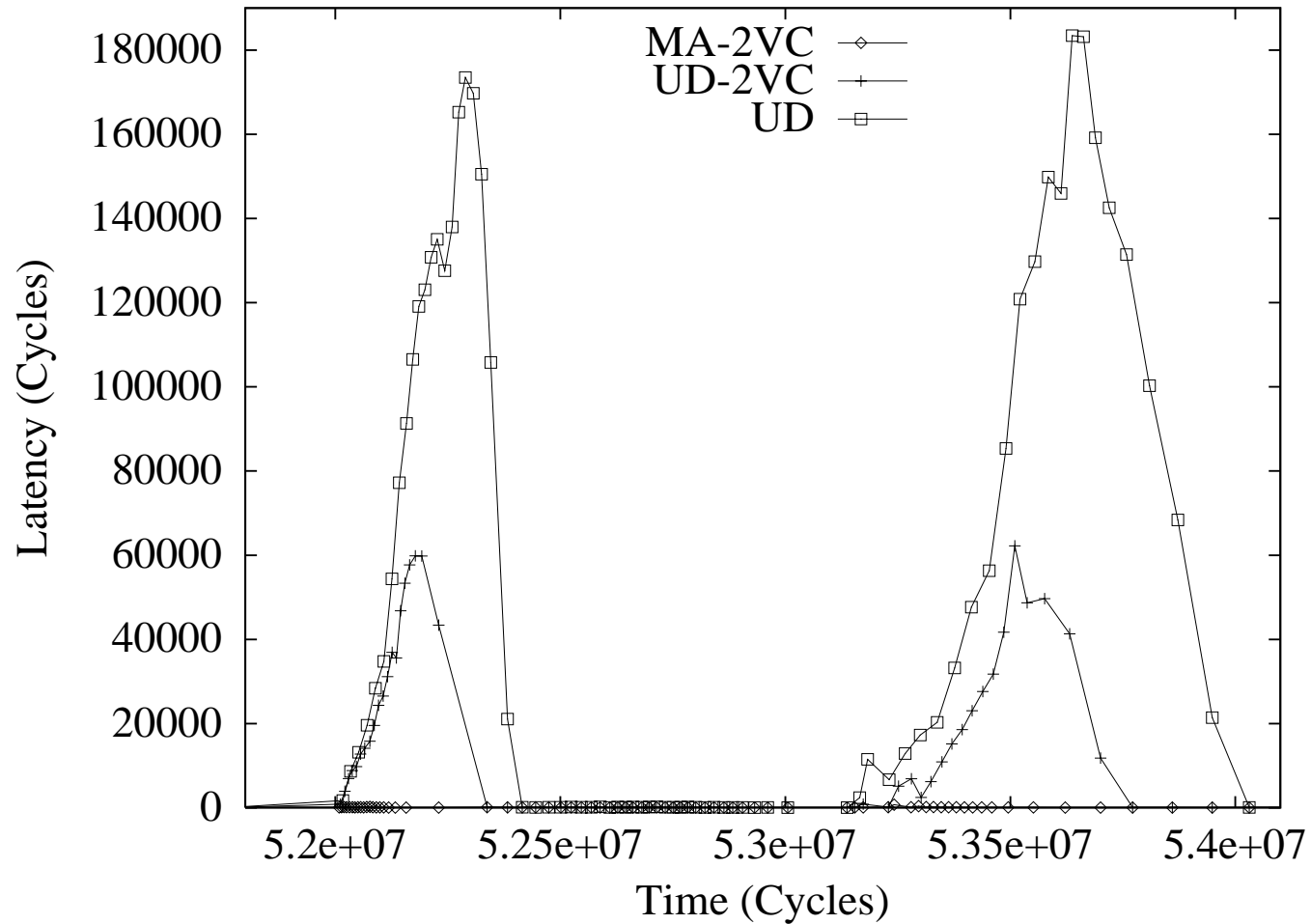


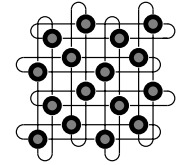
Zoom of the second peak





Zoom of the third peak





Final Remarks

Hybrid switching techniques may considerably increase performance by using the appropriate switching technique for each message class

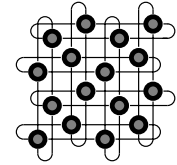
Circuit switching can take advantage of wave pipelining and optical technology to increase link bandwidth

Flexible deadlock avoidance and recovery schemes allow the design of more efficient routing algorithms

These routing algorithms have been implemented in the MIT Reliable Router and the Cray T3E

Adaptive routing and virtual channels are especially interesting when applications produce bursty traffic that saturates the network during some time intervals (usually prior to synchronization points)

Adaptive routing and virtual channels must be implemented efficiently to minimize the increment in clock cycle time



Final Remarks

Flexible deadlock avoidance and recovery schemes allow the design of more efficient routing algorithms

These routing algorithms have been implemented in the MIT Reliable Router and the Cray T3E

Adaptive routing and virtual channels are especially interesting when applications produce bursty traffic that saturates the network during some time intervals (usually prior to synchronization points)

Adaptive routing and virtual channels must be implemented efficiently to minimize the increment in clock cycle time