A Solution for Handling Hybrid Traffic in Clustered Environments: The MultiMedia Router MMR

B. Caminero C. Carrión F.J. Quiles[1] J. Duato[2] S. Yalamanchili[3]

[1] Dept. of Computer Science - Univ. de Castilla-La Mancha (Spain)

[2] Dept. of Computer Engineering (DISCA) - Univ. Politecnica de Valencia (Spain)

[3] School of Electrical and Computer Engineering - Georgia Institute of Technology (U.S.A.)

Work partially supported by the Spanish CICYT under Grant TIC2000-1151-C07,

and by the Junta de Comunidades de Castilla-La Mancha (regional government) under Grant PBC-02-008.

Contents

Contents

Introduction

- Explosion of network-based applications ➔ Development of scalable media servers based on networks of workstations
  - Interconnected by high-speed SAN/LAN fabrics
- Applications pose a wide range of bandwidth and service requirements
  - Very long data streams
  - Quality of Service (QoS) requirements
- Examples:
  - VoD servers, GIS servers, . . .
  - Video-conference, multiuser games, . . .

Introduction (cont.)

- Most current interconnects are aimed at providing good performance to best-effort traffic
  - High throughput, low average latency
- Others are too complex to be efficiently used in a local/cluster environment
- The Multimedia Router (MMR) project arises as an attempt to provide QoS support within a compact interconnection element
  - Supports a large number of multimedia connections. . .
  - . . . as well as conventional best-effort traffic

Introduction (cont.)

- Multimedia Router objectives :
  - Satisfy the QoS requirements of the multimedia flows
  - Make an efficient use of the remaining bandwidth by best-effort traffic
  - Maximize link utilization
- Purpose of this work: to investigate the feasibility of the MMR architecture for supporting mixed workloads
  - Focus is placed on the impact of switch scheduling algorithms on the MMR performance with mixed workloads
    - Multimedia flows: CBR, VBR
    - Non multimedia traffic: best-effort and control

Contents

The Multimedia Router Architecture

- Switching technique ➔ *Hybrid scheme*
  - Connection oriented scheme for multimedia flows (PCS)
    - Resource reservation on the path
  - Virtual Cut-Through for best-effort messages
    - Lower latency, no resource reservation
- Buffer organization ➔ *Small input buffers*
  - One virtual-channel per connection
    - HOL-blocking is avoided
  - Implementation based on interleaved RAM modules

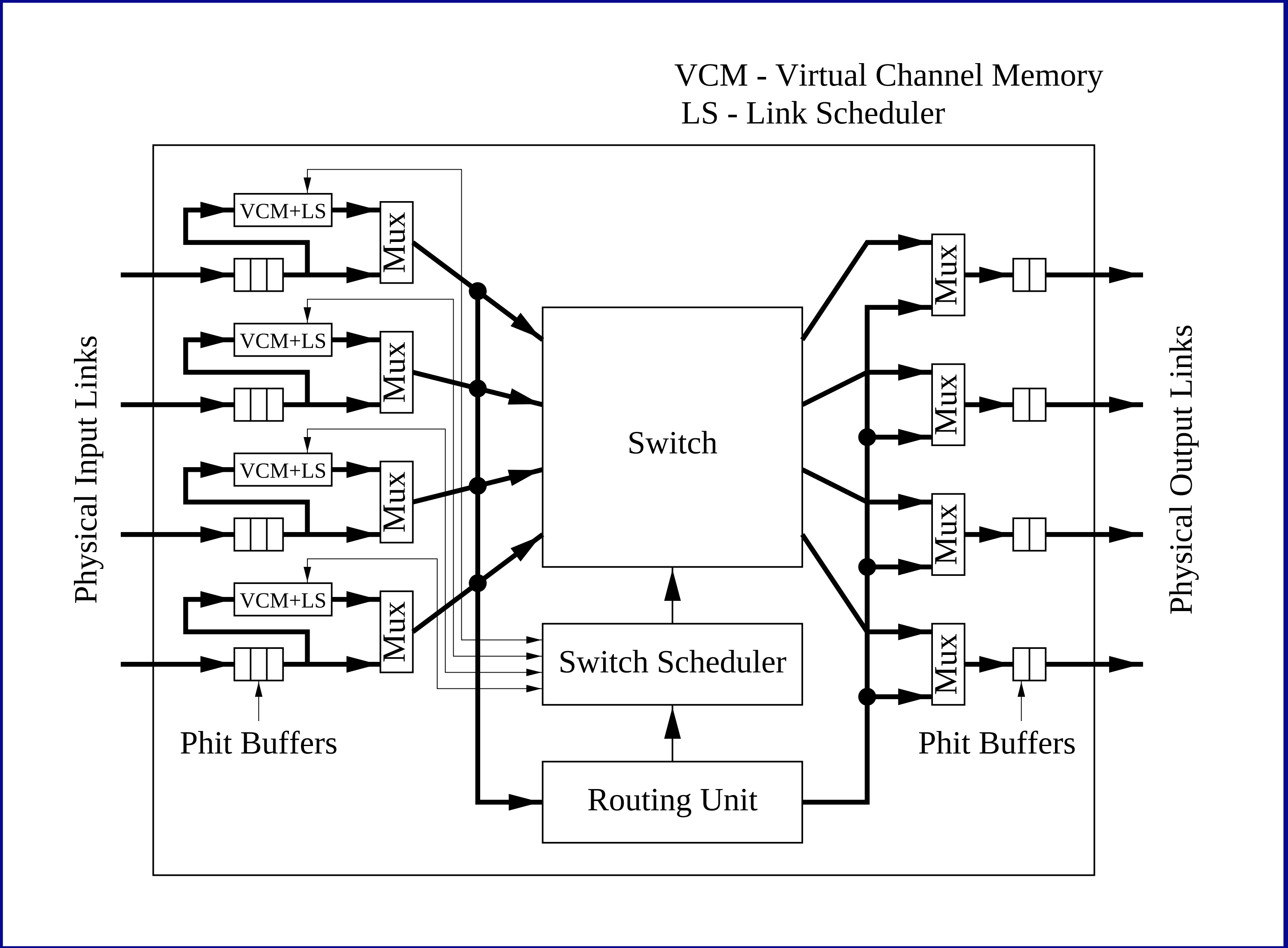
The Multimedia Router Architecture (cont.)

- Flow control ➔ *Credit-based scheme*
- Internal switch organization ➔ *Multiplexed crossbar*
  - Arbitration at input ports: link scheduling
  - Arbitration within the crossbar: switch scheduling
  - Arbitration is made in parallel with flit transmission
- *Large flits* are preferred
  - Allow to amortize flow-control overheads
  - Provide more time for the link/switch scheduler execution

The Multimedia Router Architecture (cont.)

- Establishment of Multimedia Connections
  - A probe is sent from source to destination, carrying bandwidth requirements
  - Bandwidth ➔ integer number of flit cycles
  - Flit cycles are grouped into rounds
    - A round is an integer multiple of the number of virtual channels per link
    - *A round is equivalent to the full link bandwidth*
  - Connections are accepted/rejected on the basis of their requirements, and the amount of bandwidth already allocated to other connections

The Multimedia Router Architecture (cont.)

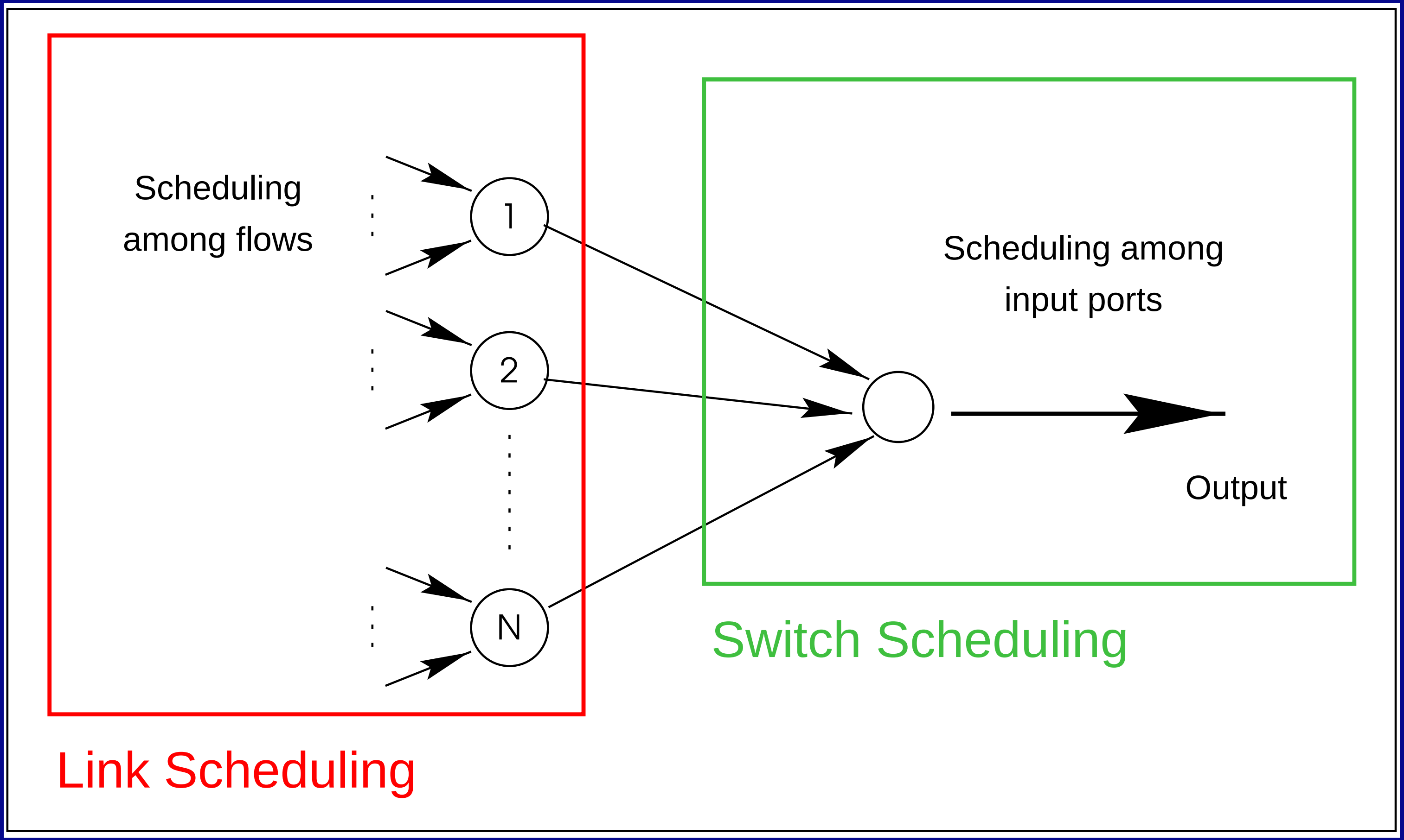


Contents

- Introduction
- The Multimedia Router Architecture
- Scheduling Algorithms
- Performance Analysis
- Conclusions

Scheduling Algorithms

Scheduling in Input Buffered Organizations



Scheduling Algorithms (cont.)

- Traffic scheduling considers the QoS requirements of the multimedia connections
- Three basic decisions:
  - *Candidate selection* ➔ Link scheduling
  - *Port ordering*
  - *Arbitration* ➔ Switch scheduling
- Algorithms suitable for pipelining and parallelization

Scheduling Algorithms (cont.)

LINK SCHEDULING

- Objective
  - Selection of the C virtual channels whose head flits have the maximum priorities ➔ Candidate vector
  - Candidates are sorted into levels
- Priority function ➔ *SIABP*
  - Priority is related to the QoS requirements of the connection, and is biased according to the QoS it is receiving
- One link scheduler per physical link

Scheduling Algorithms (cont.)

Switch Scheduling

- Objective
  - Compute a conflict-free matching among input and output ports
  - Crossbar throughput is maximized while . . .
  - . . . QoS requirements of the connections are considered
- Two decissions must be taken:
  - *Port ordering* ➔ Output port selection
  - *Arbitration* ➔ Input port selection

Scheduling Algorithms (cont.)

SWITCH SCHEDULING (CONT.)

- Two different algorithms . . .
  - Candidate Order Arbiter (COA)
  - Candidate Conflict Arbiter (CCA)
- . . . with different *port ordering* functions
- Common reasoning ➔ Ports with less conflicts are matched first, since they have less chances to get matched
  - COA : Order first by level, then in increasing order of conflicts within a level
  - CCA : Consider candidates from all the levels at the same time, in increasing order of conflicts

Scheduling Algorithms (cont.)

Switch Scheduling (cont.)

- Info from candidate vectors is organized into a selection matrix and a conflict vector :

COA

Output ports		Input port 1	Input port 2	Input port 3	Input port 4	Conflict Vector
Level 1	1		Request		Request	2
Level 1	2			Request		1
Level 1	3	Request				1
Level 1	4					0
Level 2	1	Request				1
Level 2	2		Request			1
Level 2	3					0
Level 2	4				Request	1

CCA

Output ports	Input port 1	Input port 2	Input port 3	Input port 4	Conflict Vecto
1	Req 2	Req 1		Req 1	3
2		Req 2	Req 1		2
3	Req 1				1
4				Req 2	1

Scheduling Algorithms (cont.)

SWITCH SCHEDULING (CONT.)

- Arbitration function ➔ If there is more than one conflict for the selected output port . . .
  - Pick the candidate with the highest priority
- Same arbitration function for COA and CCA
- The matched input and output ports drop their requests

Scheduling Algorithms (cont.)

Switch Scheduling (cont.)

- The process ends when there are no more pending requests
- At the end of the scheduling process, the head flits from the selected ports are forwarded synchronously through the crossbar
- At the same time, a new execution of the link and switch schedulers is started

Contents

Performance Analysis

- Discrete event simulator of a 4×4 MMR router, with its attached NICs
- Credit-based flow control between NICs and MMR
- Simulation configuration
  - 1.24 Gbps, 16-bit wide links
  - Flit size = 1024 bits
  - MMR buffer size = 1 flit
  - SIABP link schedulers with four levels of candidates
- WFA (Wave Front Arbiter) is also tested as an alternative switch scheduler

Performance Analysis (cont.)

TRAFFIC MODELS

- CBR traffic : Mix of connections with low, medium and high bandwidth requirements
- VBR traffic : Traces from real MPEG-2 video sequences
- Best-effort traffic : Realistic web workload generator, with self-similar properties (SURGE)
- Control traffic : Information used to control the network itself, needs *minimum delay*

Performance Analysis (cont.)

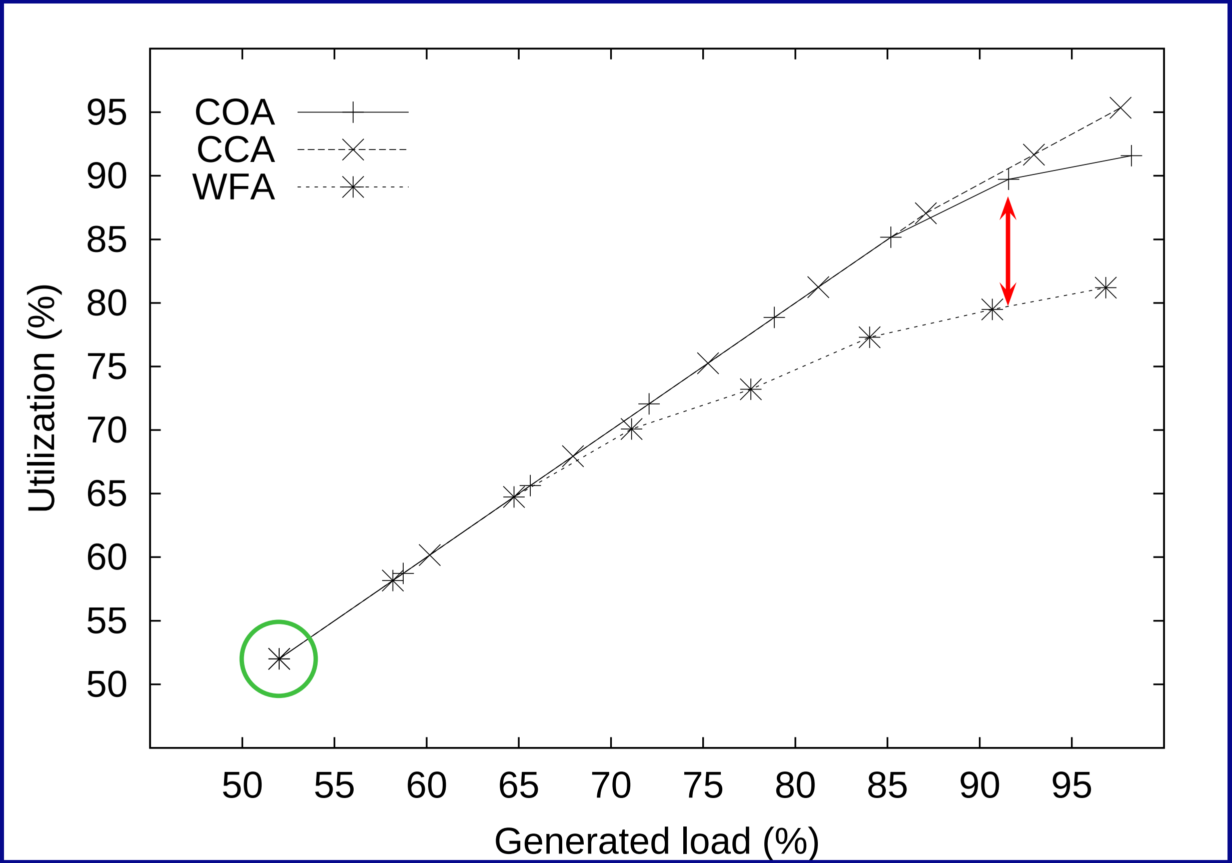
Workload Configuration

- Workload is generated as a percentage of link bandwidth
- A fixed amount of the workload is composed of multimedia connections (CBR and VBR)
- Increasing percentages of best-effort traffic have been generated for each simulation point
- A fixed and negligible amount of control traffic has also been introduced

Performance Analysis (cont.)

Simulation Results

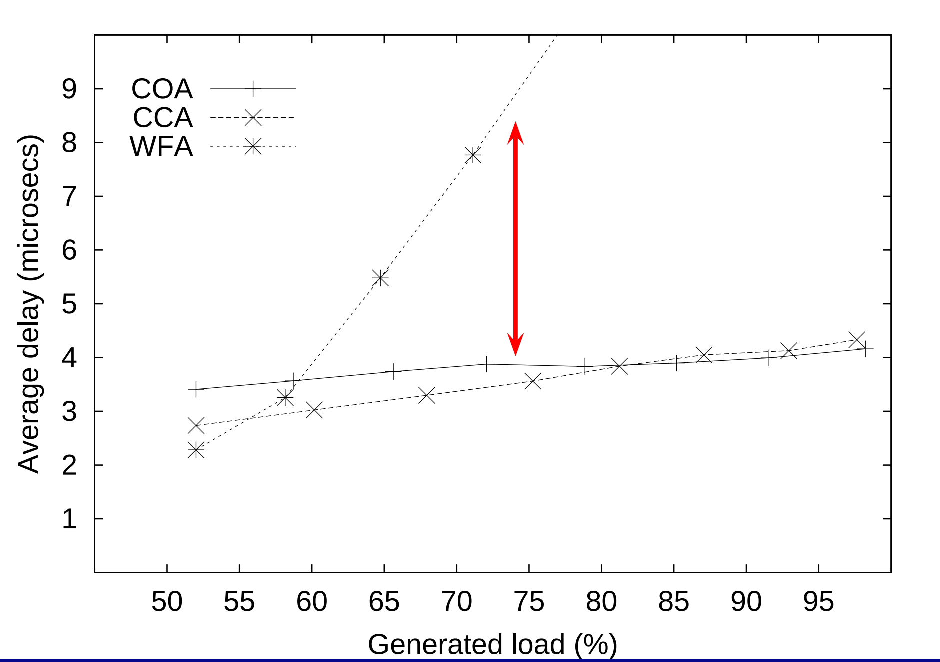
Average crossbar utilization



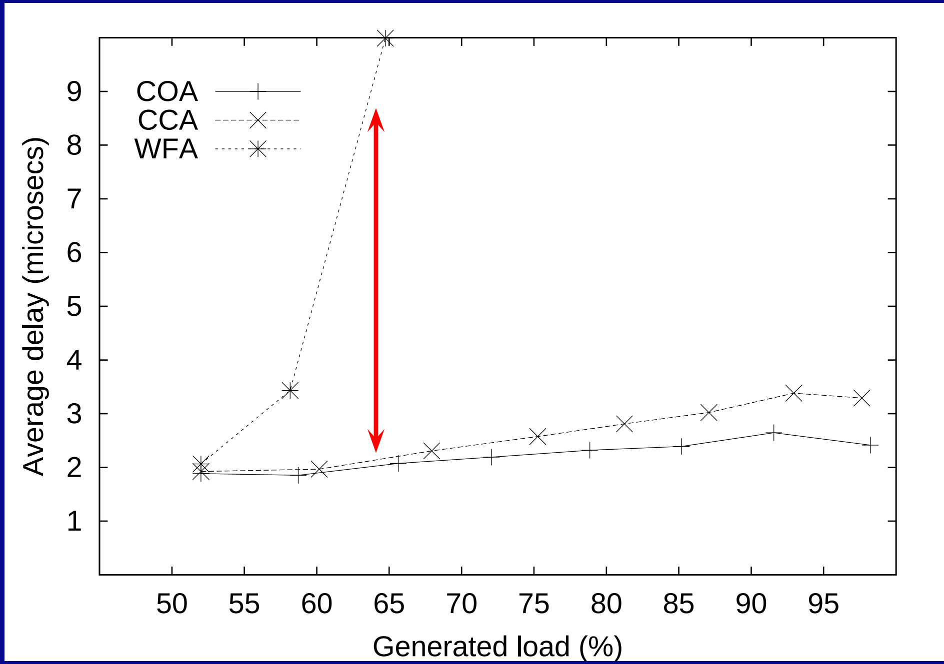
Performance Analysis (cont.)

Simulation Results - CBR Traffic

Average flit delay since generation



64 Kbps connections

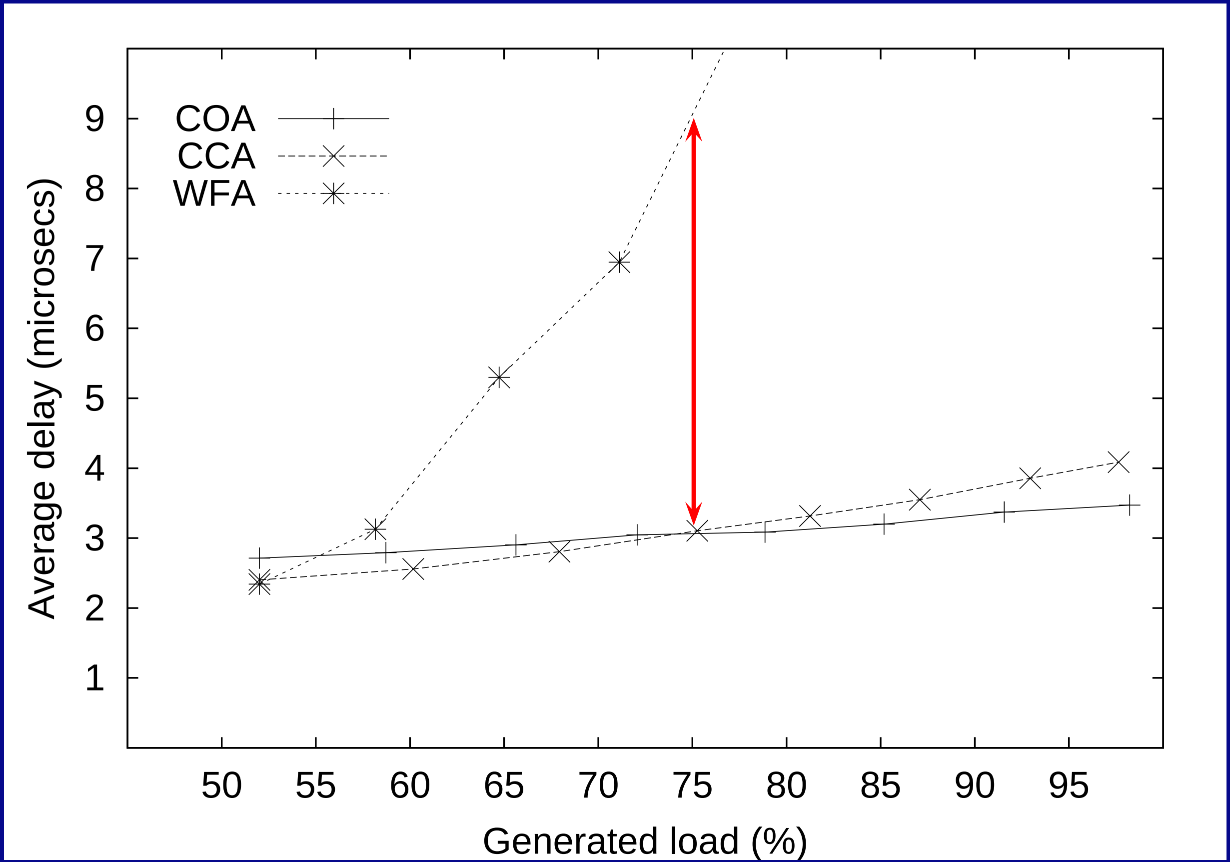


55 Mbps connections

Performance Analysis (cont.)

SIMULATION RESULTS - VBR TRAFFIC

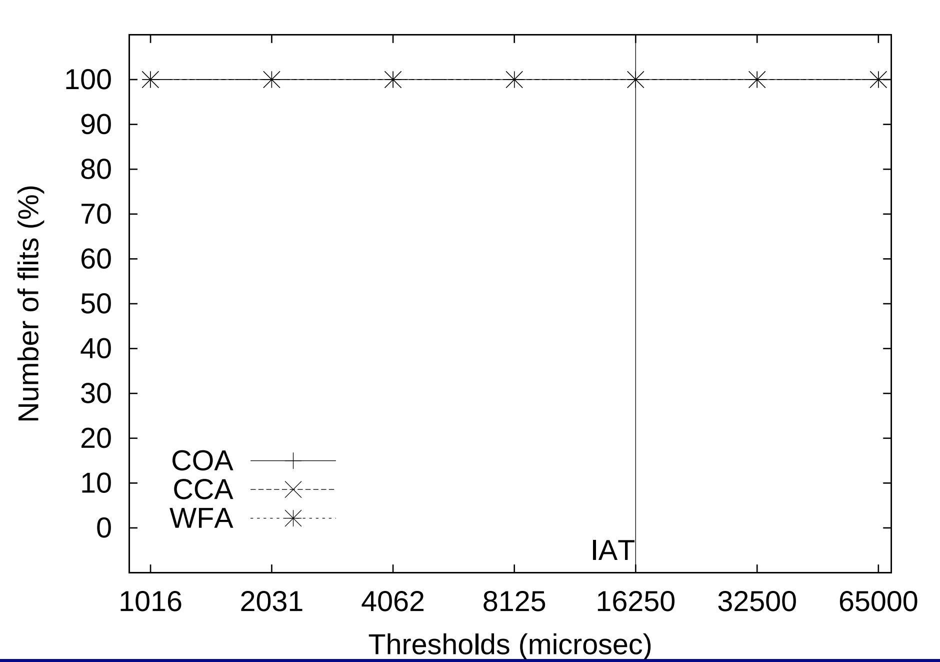
Average frame delay since generation



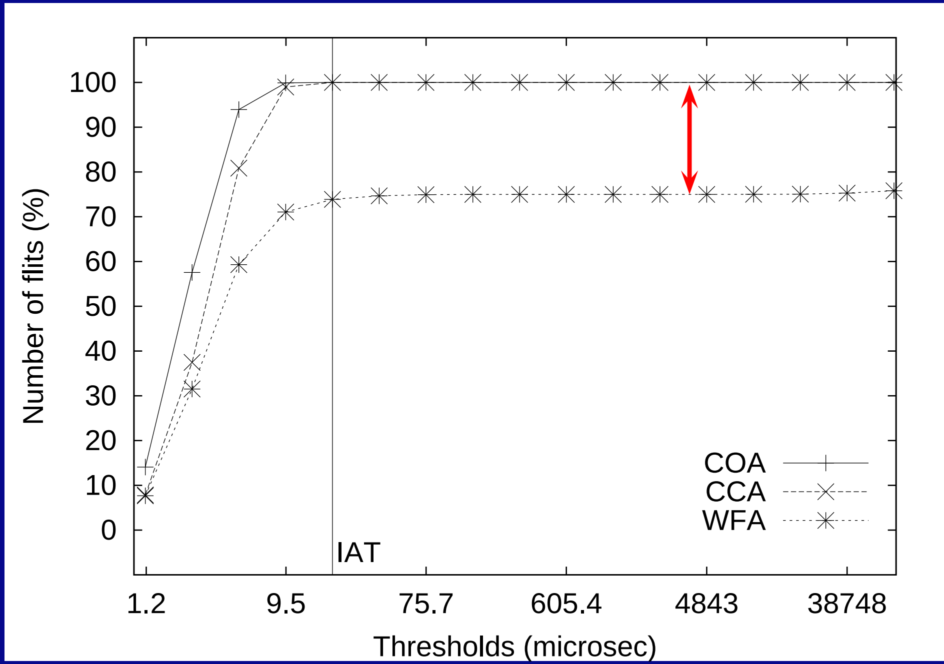
Performance Analysis (cont.)

SIMULATION RESULTS - CBR TRAFFIC

Distribution of flit delay since generation



64 Kbps connections

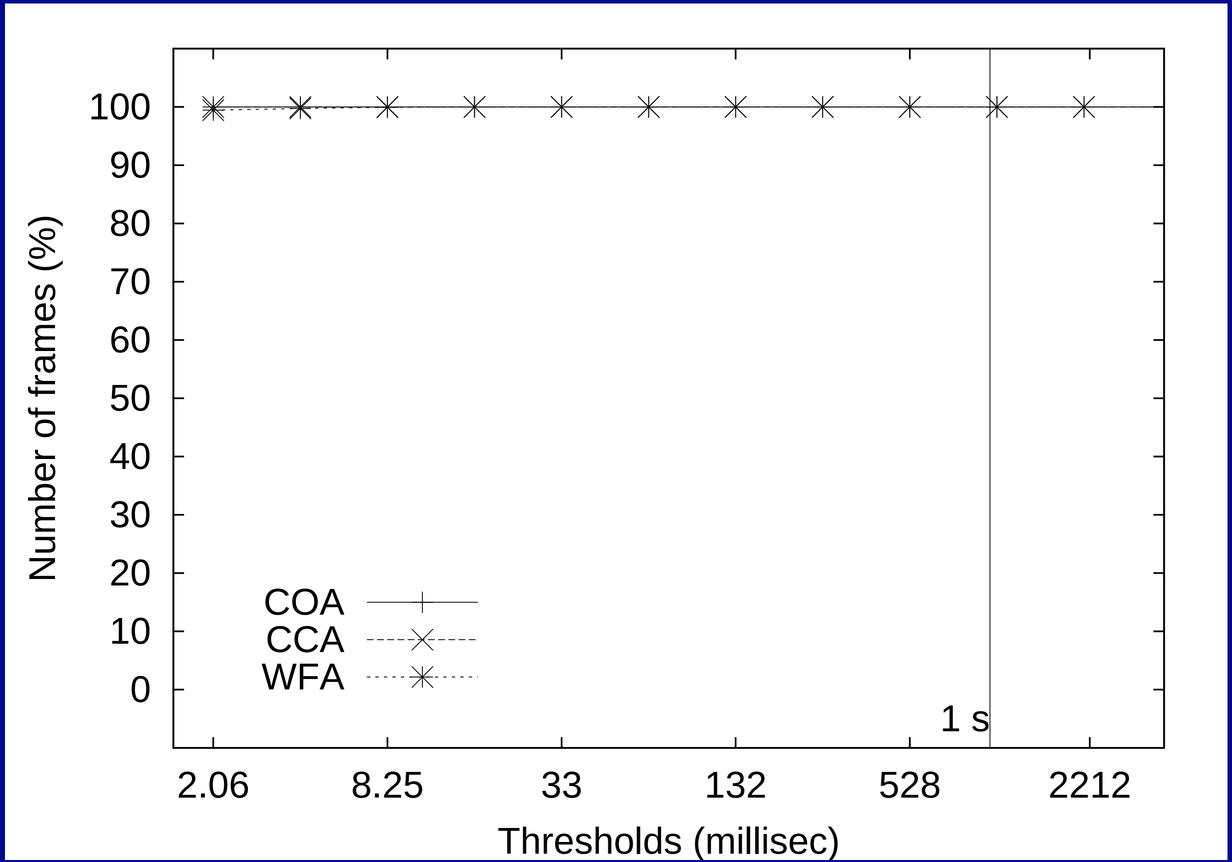


55 Mbps connections

Performance Analysis (cont.)

Simulation Results - VBR Traffic

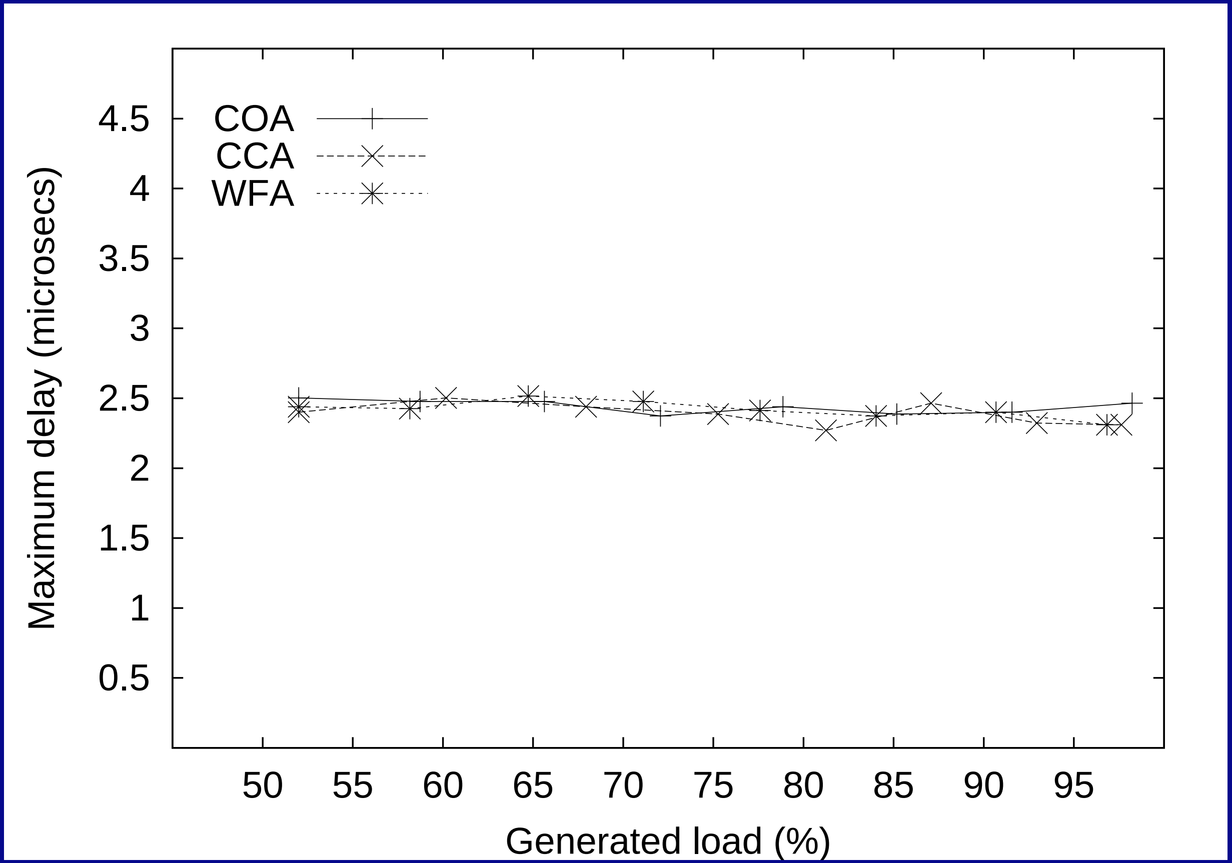
Distribution of frame delay since generation



Performance Analysis (cont.)

Simulation Results - Control Traffic

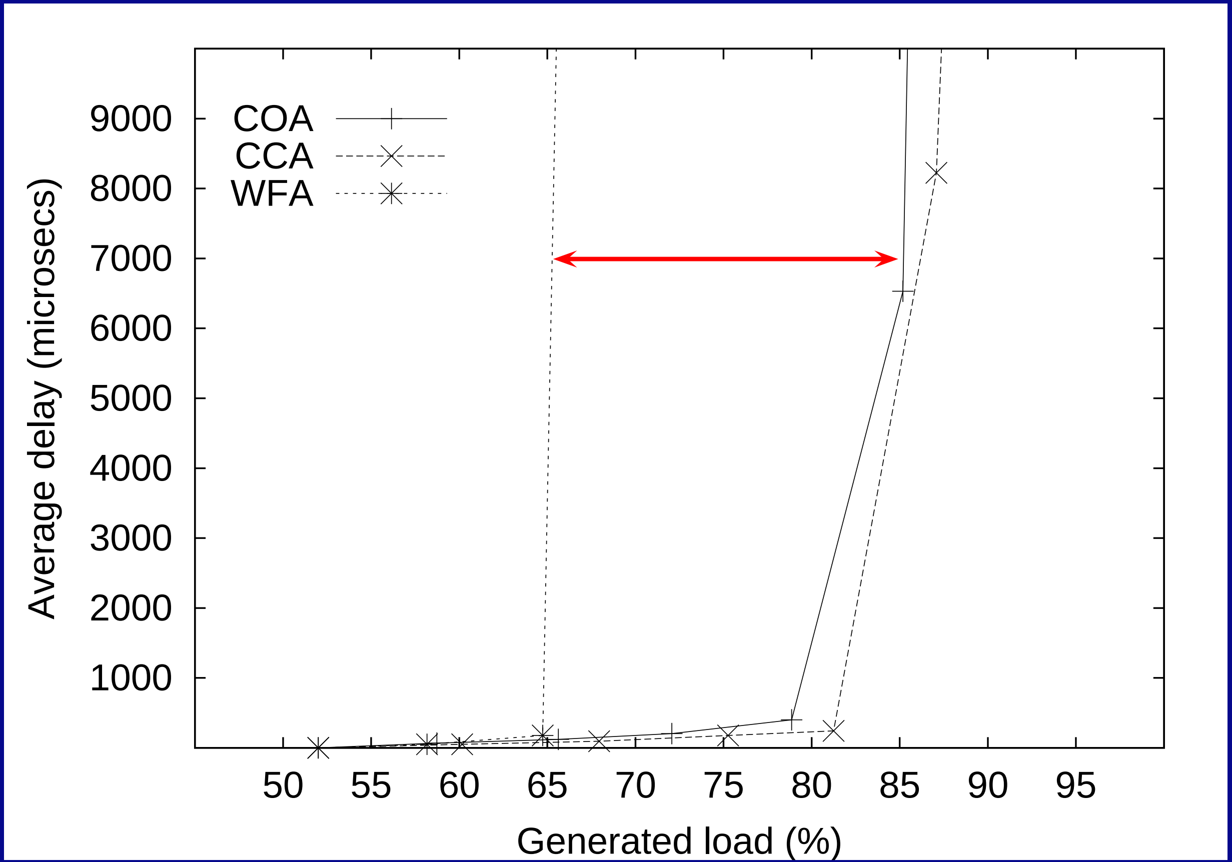
Maximum flit delay since generation



Performance Analysis (cont.)

Simulation Results - Best-effort Traffic

Average flit delay since generation



Contents

- Introduction
- The Multimedia Router Architecture
- Scheduling Algorithms
- Performance Analysis
- Conclusions

Conclusions

- Two switch scheduling algorithms developed for the MMR, have been evaluated
  - They consider the QoS needs of the applications when taking scheduling decissions
- Experiments have been carried out with a mixed workload :
  - CBR and VBR multimedia traffic
  - Best-effort and control traffic
- Conventional non QoS-aware WFA switch scheduler for reference purposes

Conclusions (cont.)

- A switch scheduler with QoS support is needed to provide the connections with an adequate support
  - The most demanding connections experience heavy QoS degradations when using a non QoS-aware switch scheduler
  - The rest of multimedia connections also suffer some performance degradation
- When COA or CCA are used, every multimedia connection gets its reserved bandwidth in spite of the presence of best-effort traffic
- Control messages achieve a constant delay in any cases, independently of the presence of other traffic

Conclusions (cont.)

- The proposed scheduling algorithms can be easily adapted to other routers with input buffers and multiplexed crossbar
- Open issue :
  - More complex topologies should be analyzed in order to assess the conclusions obtained in this work

A Solution for Handling Hybrid Traffic in Clustered Environments: The MultiMedia Router MMR

B. Caminero C. Carrión F.J. Quiles[1] J. Duato[2] S. Yalamanchili[3]

[1] Dept. of Computer Science - Univ. de Castilla-La Mancha (Spain)

[2] Dept. of Computer Engineering (DISCA) - Univ. Politecnica de Valencia (Spain)

[3] School of Electrical and Computer Engineering - Georgia Institute of Technology (U.S.A.)

Work partially supported by the Spanish CICYT under Grant TIC2000-1151-C07,

and by the Junta de Comunidades de Castilla-La Mancha (regional government) under Grant PBC-02-008.

Index