Slide 0

PP-MESS-SIM: A Simulator for Evaluating Interconnection Networks for Message-Passing Parallel Machines

Jennifer Rexford, James Dolter, Wu-chang Feng, and Kang Shin

April 11, 1995

Real-Time Computing Laboratory
The University of Michigan
Ann Arbor, Michigan

28th Annual Simulation Symposium

Slide 1

Overview

1. Research Goals and Design Issues
2. Design of PP-MESS-SIM (Point-to-Point MESSage SIMulator)
3. Experiments and Results

Real-Time Computing Laboratory
The University of Michigan

Slide 2

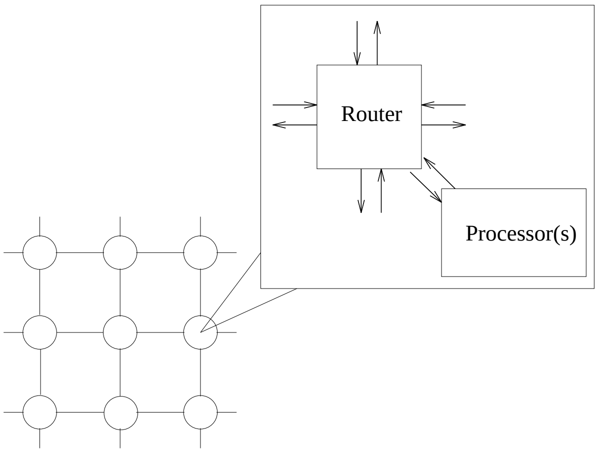
Research Goals

- Perform complex multi-factor experiments involving router architectures and application workloads to optimize network performance.
- Build a better router

Real-Time Computing Laboratory
The University of Michigan

Slide 3

Sample Network and Router



Real-Time Computing Laboratory
The University of Michigan

Slide 4

Design Trade-offs for Point-to-Point Multicomputer Networks

- **Topology**: Determines the interconnection between nodes in the network
- **Node Architecture**: Determines the internal policies, speed, and the amount of resources within the router.
- **Routing**: Determines the path message takes between two nodes. (Minimal, Non-minimal)
- **Switching**: Determines how the message flows through the nodes. (Packet Switching, Virtual Cut-Through, Wormhole)
- **Application workloads**: Message interarrival, packet lengths, destination node distributions.

Slide 5

Why PP-MESS-SIM?

- Network performance depends on complex interactions of multiple factors. Necessitates a simulation environment.
- Simulator must be flexible and easily extensible to allow for a wide range of experiments to be performed.
- For example: Simulating networks which support multiple routing/switching schemes simultaneously.

Slide 6

Design of PP-MESS-SIM

- Separate functionality into different modules.
- Decouple modules by defining clean, powerful interfaces between them.
- Extend each module independently to provide a variety of features.
- Create mechanism for the user to compose a wide range of experiments using these features.

Slide 7

PP-MESS-SIM

Object-oriented discrete-event simulator written in C++

- **Node** : Handles all event flow in simulation.
- **Ralg** : Executes higher level routing-switching scheme.
- **Net** : Encapsulates all functionality related to topologies.
- **Workload** : Models application behavior (message generation) and collects message based statistics.
- **Spec** : Initializes modules according to user specification.

Slide 8

Node (Overview)

Handles all of the details of event flow in simulation.

- Pull out details of Net, Ralg, and Workload from Node. Concentrate solely on capturing queuing, arbitration for resources, and flow control policies within router.
- Separation of policy (Ralg) from mechanism (Node model)
- Collects router specific statistics (buffer/bus utilization)
- Currently support 2 models: SPIDER and v-router
  - Detail of simulation
  - Internal timing
  - Flexibility of model

Slide 9

Node Interfaces

Interfaces to other modules.

- Workload: Generates messages for Node and post-processes messages upon reception.
- Net: Identifies adjacent node and adjacent link id for Node as message travels from one Node to another.
- Ralg: Issues commands (routing-switching instructions) to Node. Node returns status information of instruction back to Ralg.

Slide 10

Ralg (Overview)

Determines where to send a packet when it enters Node. Ralg embodies the intelligence (policy) while Node models the mechanism.

- Insulated from Net and Node. Executes without detailed knowledge either the Net module or underlying Node module.
- Allows Ralgs to be interchangeable.

Switching	Routing
Packet switching	Minimal oblivious Minimal adaptive
Virtual cut-through	Minimal oblivious Minimal adaptive Non-minimal adaptive
Wormhole	Minimal oblivious Minimal adaptive Non-minimal adaptive
Hybrid (h-hop)	Minimal oblivious

Slide 11

Ralg Interfaces (Node)

- Interacts with Node with Routing-Switching Instructions
  - Candidate switching decision (buffer, stall, drop, cut-through)
  - Candidate channels to attempt
- Node returns status after attempting to execute instruction.
- Node model captures internal timing of instruction execution.
- Node and Ralg iterate until a final decision is made.

Slide 12

Ralg Interfaces (Net)

- Interacts with Net to get a list of candidate directions.
- Net returns an ordered list of directions which can be minimal or non-minimal
- Ordering is done according to a selection function which determines the order in which candidate directions are considered.

Selection Function	Description
Dimension order	Favors links in lower dimensions of the topology
Random	Generates all rankings with equal probability
Hops remaining	Favors directions with more remaining hops
Minimum congestion	Favors links with fewer allocated virtual channels

Slide 13

Net

Encapsulates all functionality related to topologies.

- Base class contains generic member functions.
- Derived classes implement functionality specific to the topology.
- Easily extensible to other topologies
- Topologies supported include: k-ary n-cube, C-wrapped Hexagonal Mesh, and Square Mesh
- Interfaces to Workload by providing destination node ids given a particular destination node distribution.

Slide 14

Workload

Implemented as independent tasks mapped onto nodes of the network.

Policies are specified per task for flexibility.

- Routing/Switching scheme
- Interarrival time distribution
- Packet length distribution
- Destination node distribution
- Data Collection Metrics

Slide 15

Interarrival Time and Packet Length Distributions

Distribution	Definition
Negative exponential (λ)	Exponential distribution with mean λ
Uniform (a, b)	Select integers between a and b with equal probability
Discrete ($\{p_i, \ell_i\}$)	Select ℓ_i with probability p_i
Normal (μ, σ)	Normal distribution with mean μ and standard deviation σ
Two-stage normal ($p, \mu_1, \sigma_1, \mu_2, \sigma_2$)	Select from normal distribution (μ_1, σ_1) with probability p; otherwise, select from normal distribution (μ_2, σ_2)

Slide 16

Destination Node Distributions

Distribution	Description
NodeUniform	Uniform random selection of destination node
HopUniform ($\{p_i\}$)	Select a destination i hops away with probability p_i
Discrete ($\{n_i, p_i\}$)	Select destination node n_i with probability p_i
DimensionReversal	Source $(w, x, \ldots, z)$ selects destination $(z, \ldots, x, w)$
BitComplement	Destination node id is the bit-complement of source
BitReversal	Destination node id is the bit-reversal of source

Slide 17

Data Collection Support

Data Collection

- Warm-up period
- Data collection period
- Drainage period

Performance Metric	Description
Latency	Mean, min, max, and variance of packet latency
Histogram(a, b, c)	Histogram of packet latency with c bins over range $[a, b]$
Cut-through statistics	Histogram of packet cut-through history
Null	No data collection

Slide 18

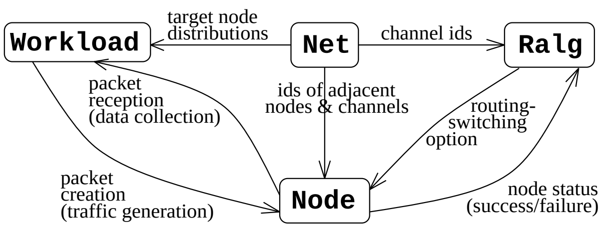
Spec

Initializes modules according to user specification.

- Specification language allows for the selection of parameters at a higher level.
- Mechanism for user to select fine-grained parameters. Allows for specification of heterogeneous networks.
- GUI for easily generating input files for experiments (Tcl/Tk).

Slide 19

Structure



Real-Time Computing Laboratory
The University of Michigan

Slide 20

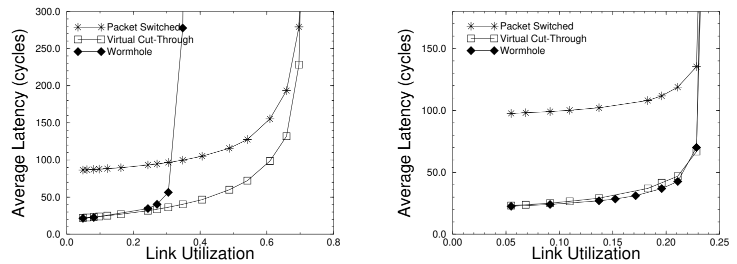
Experiments and Results

Experimental Setup

- 8x8 square mesh with v-router nodes
- 16 flit (long word) packets
- Poissonian arrival times of packets
- Vary routing/switching policy and destination node distribution

Real-Time Computing Laboratory
The University of Michigan

Slide 21

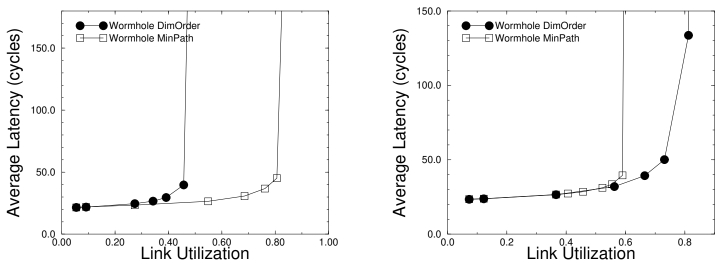


(a) Uniform Traffic (b) Transpose Traffic

Figure 1: Comparing switching schemes under dimension-order routing

Real-Time Computing Laboratory
The University of Michigan

Slide 22



(a) Transpose Traffic (b) Bit Complement Traffic

Figure 2: Comparing routing algorithms under wormhole switching

Real-Time Computing Laboratory
The University of Michigan

Slide 23

Future Work

Experiments:

- Continue experiments across other factors.
- Supporting different policies simultaneously.

Extensions:

- Extending Workload to capture higher level, application task semantics.
- Extending Ralg with other routing/switching schemes.
- Modularizing Node modules to separate queuing policies, arbitration policies, and flow control policies from each other.

URL to download source:

- ftp://rtcl.eecs.umich.edu/outgoing/ppsim/

Real-Time Computing Laboratory
The University of Michigan

Slide 24

Simulation Techniques

Event Queue

- Selected from one of many priority queue implementations in libg++
- Augmented further into a two level priority queue: zero time queue and a non-zero time priority queue

Random Number Generation

- Single random number stream using the additive congruential generator developed by Knuth and implemented in libg++
- Extended to provide separate streams for each stochastic process by carving the original stream into consecutive chunks and assigning chunks to different streams.

Real-Time Computing Laboratory
The University of Michigan