PDCS '97

Modeling and Comparison of Wormhole Routed Mesh and Torus Networks

Ronald I. Greenberg
Mathematical and Computer Sciences, Loyola University

Lee Guan
ACE COMM Corporation

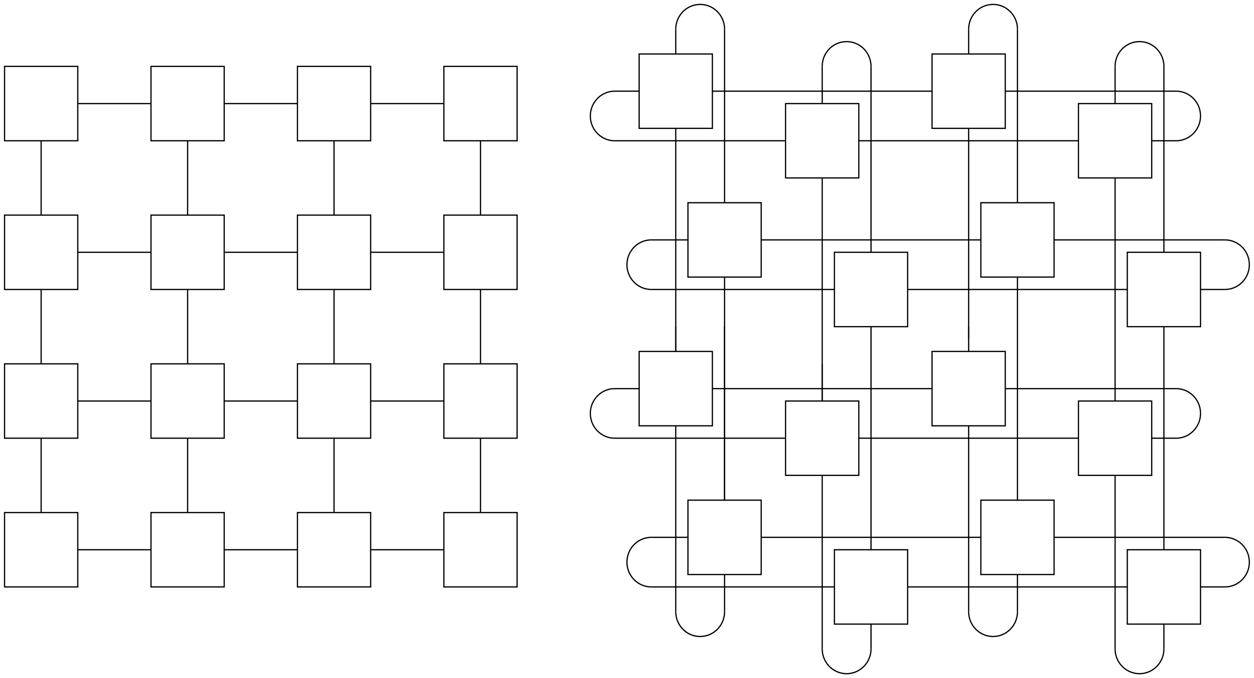
OUTLINE

- **BACKGROUND**
  - — Multiprocessor Networks
  - — Routing Protocols
- **WORMHOLE ROUTING PERFORMANCE MODEL**
  - — A Performance Model for Wormhole Routed Networks
- **COMPARISON OF MESH AND TORUS**
- **CONCLUSIONS**

BACKGROUND - Multiprocessor Networks

- **Multiple processors to achieve increased computation power**
- **Key factors for performance:**
  - — Network topologies
  - — Message routing protocols
- **Leading candidates for multiprocessor networks:**
  - — Bus and ring networks
  - — Fully-connected network and crossbar
  - — k-ary n-cubes
  - — Fat-tree and fat-pyramid
- **Routing Protocols**
  - — Packet routing, wormhole routing

BACKGROUND - Multiprocessor Networks

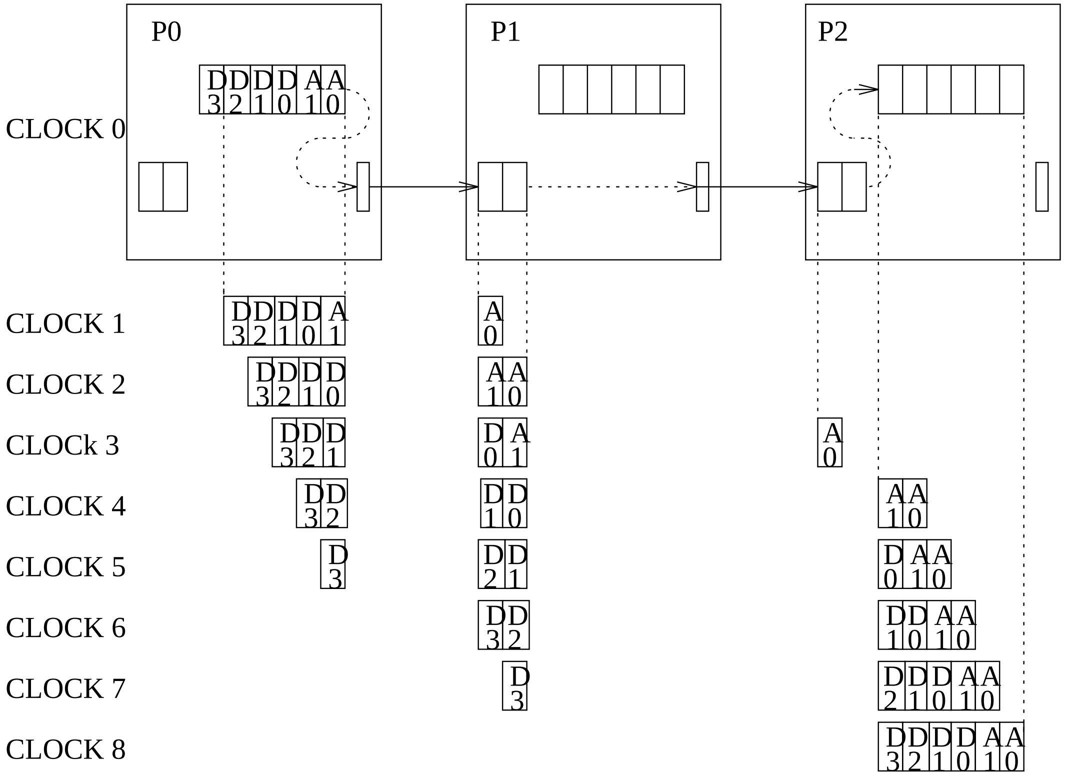


A 4 × 4 mesh and a folded 4 × 4 torus (4-ary 2-cube) system. Bidirectional mesh and unidirectional torus have equal bisection widths.

BACKGROUND - Routing Protocols

- **Packet routing**
  - — Traditional, large packets, each contains routing information.
- **Wormhole routing**
  - — Worms consist of flits (flow control digits)
  - — Forward as soon as enough flits received to tell destination
  - — If a message is blocked, the whole message is blocked in place.

BACKGROUND - Routing Protocols



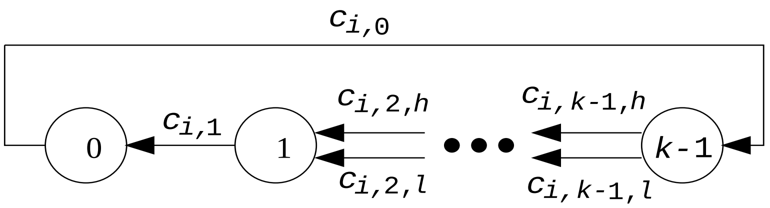
A worm snakes through a network. (Address length = 2 flits, data length = 4 flits)

BACKGROUND - Routing Protocols

- **Deadlock in wormhole routing and its prevention**

— *E-cube routing*: Route through dimensions in the order specified by scanning the address bits from MSB to LSB. (Suffice for mesh and hypercube).

— *Virtual Flow Control*: all but two channels in a unidimensional cycle are broken into pairs of virtual channels, implemented by time-multiplexing onto the associated physical channel (essential for other k-ary n-cubes).



Virtual channels in the ith dimension of a k-ary n-cube.

WORMHOLE ROUTING PERFORMANCE MODEL

- **Previous Works**

  — Dally [90] studied the k-ary n-cubes using e-cube routing and virtual channel techniques to avoid deadlocks. Simplified assumption does not hold for high-dimension networks and high traffic rates.

  — Kim and Das [94] improved the accuracy based on Dally's work; but still specific to k-ary n-cube and very involved.

  — Draper and Ghosh [94], Kim and Chien [95] start to employ the M/G/1 model to estimate the waiting time, still k-ary n-cubes.

- **Motivations**

  — A general model for networks other than k-ary n-cubes.

  — Performance comparison of mesh versus torus networks. networks.

WORMHOLE ROUTING PERFORMANCE MODEL

- **Assumptions**

  — Arrivals at each source node are governed by a Poisson process, and destinations are uniformly random.

  — Messages have fixed length and are longer than the diameter of the network.

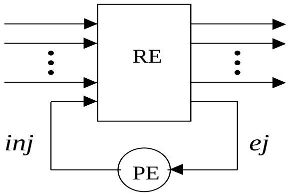
  — Contentions at incoming links to a node are resolved according to First-Come First-Served (FCFS) scheduling.

  — Messages arriving at destinations are immediately consumed at the rate of one flit per time step, i.e., no blocking is encountered at destinations.

WORMHOLE ROUTING PERFORMANCE MODEL

- **Latency**:
  — A message is from node j it encounters:
    1) A waiting time $W_{inj,j}$ for the injecting channel.
    2) A service time $x_{inj,j}$ at the injecting channel.
    3) An additional time (D-1) to traverse the remaining channels.
  — The latency L_j for the message injected at node j is

$$L_j = W_{inj,j} + x_{inj,j} + D - 1 \ . \tag{1}$$



A general node model.

WORMHOLE ROUTING PERFORMANCE MODEL

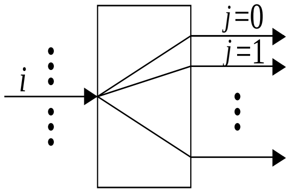
- **Service Times**:

  — The service time at the injecting channel $x_{inj,j}$ depends on the service time and waiting time of the subsequent channels,

$$x_i^{\text{in}} = \sum_j (x_j + w_{i|j}) \cdot R_{i|j} \ , \qquad (2)$$

  where x_j is the service time for the outgoing channel j, and $w_{i|j}$ is the waiting time for outgoing channel j of messages from incoming channel i.

  — Service times are resolved in the reverse order of the channels traversed, from the last channel (ejecting channel) backwards to the injecting channel.



WORMHOLE ROUTING PERFORMANCE MODEL

- **Waiting Time**

— The mean waiting time $w_{i|j}$ is caused by contention for the outgoing channel j.

— When a message is blocked, it must wait for the message that is holding the outgoing channel to be fully serviced. This motivates us to take advantage of well-known queuing models that have been employed to analyze store-and-forward routing:

$$\overline{W}_{M/G/1} = \frac{\rho \overline{x}}{2(1-\rho)}(1 + C_b^2) \ , \tag{3}$$

and

$$\overline{W}_{M/G/2} = \frac{\rho^2 \overline{x}}{2(1-\rho^2)}(1 + C_b^2) \tag{4}$$

where $\rho = \lambda \overline{x}$, λ is message rate, $\overline{x}$ is the mean service time, and $C_b^2 = \frac{\sigma_b^2}{\overline{x}^2}$, where σ_b^2 is the variance of the service time distribution.

— C_b^2 can be estimated as $C_b^2 = \frac{(\overline{x} - s/f)^2}{\overline{x}^2}$.

WORMHOLE ROUTING PERFORMANCE MODEL

- **Waiting Time - Wormhole Routing Adjustments**

— In general, waiting time corrected by a blocking probability,

$$w_{i|j} = P_{i|j} W_j \ . \tag{5}$$

— $P_{i|j}$ should reflect the probability that messages from incoming link i are blocked by messages from *other links*,

$$P_{i|j} = 1 - \frac{\lambda_i^{\text{in}}}{\lambda_j} R_{i|j} \tag{6}$$

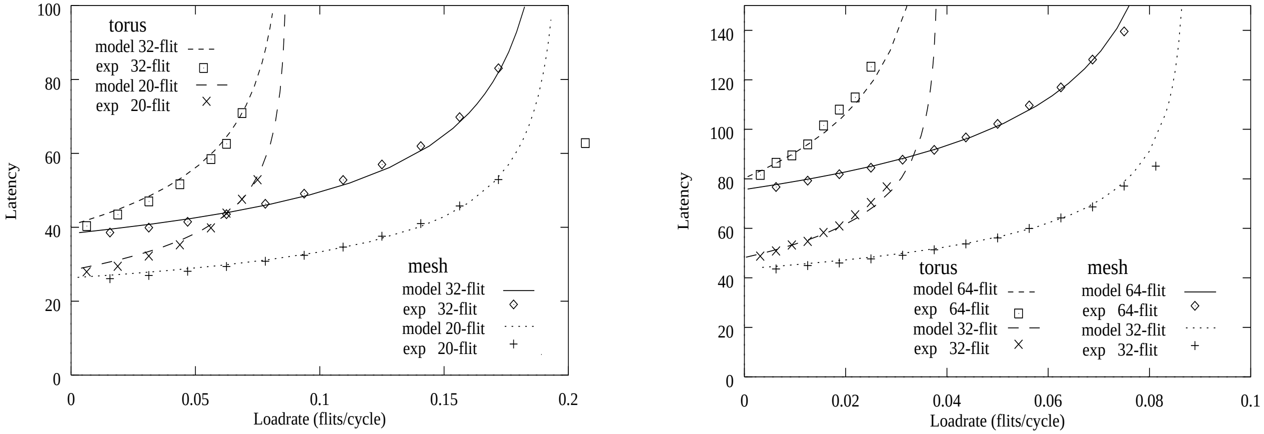
— Service time can then be generally be expressed as

$$x_i^{\text{in}} = \sum_j \left[x_j + (1 - \frac{\lambda_i^{\text{in}}}{\lambda_j} R_{i|j}) W_j \right] R_{i|j} \ . \tag{7}$$

WORMHOLE ROUTING PERFORMANCE MODEL

- **Applying the Model to Mesh and Comparing the Results with Torus**
  - Procedures:
    - 1) Compute message rate to each channel.
    - 2) Compute waiting and service times.
    - 3) Compute average latency and throughput.
  - Resolve x-dimension first, followed by y-dimension.
  - Section 3 in paper provides the gory details.
  - Models validated through comparisons with simulations.
  - Results for Torus are obtained from model by Draper and Gosh[94].

WORMHOLE ROUTING PERFORMANCE MODEL



(a) Comparison of model and simulation results for the mesh and torus with 64 nodes using message lengths of 20 and 32 flits. (b) Comparison of model and simulation results for the mesh and torus with 256 nodes using message lengths of 32 and 64 flits.

CONCLUSIONS

- **Summary**

  — Presented a generalized service time relation between the channels incident to a node and channels going out from the node.

  — An accurate model for wormhole routed 2D-mesh networks.

  — Comparisons with torus show that the 2D-mesh network exhibits better performance.