# An Improved Analytical Model for Wormhole Routed Networks with Application to Butterfly Fat-Trees

Ronald Greenberg
Mathematical and Computer Sciences
Loyola University of Chicago Lee Guan

Electrical Engineering
University of Maryland

# OUTLINE

- General performance model for wormhole routing
- Analysis of butterfly fat-tree (an area-universal network)
- Empirical results and comparison to model

# Wormhole Routing

Messages (worms) composed of flits (flow control digits).

Worms snake through network one flit after another.

Constant no. of flits stored in an intermediate node at any time.

# Wormhole Routing Models

- **Prior work**
  - **Dally [90]: $k$-ary $n$-cubes using $e$-cube routing and virtual channel techniques to avoid deadlock. Strong simplifying assumptions.**
  - **More accurate models: Hady [93], Kim and Das [94], Draper and Ghosh [94], Kim and Chien [95]**
- **Drawbacks: All focus on $k$-ary $n$-cubes, especially hypercubes; none apply when messages have a choice among outgoing links from a node. Some models complex.**

# Modeling Assumptions

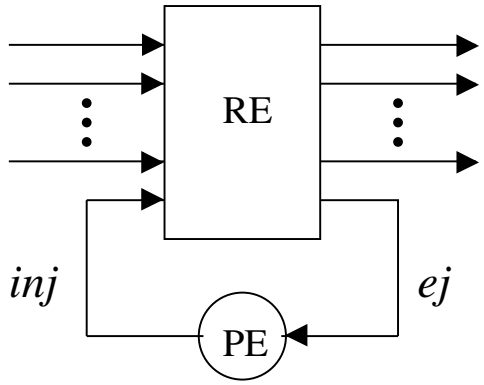
- Arrivals at each source node are governed by a Poisson process, and destinations are uniformly random.
- Messages have fixed length and are longer than the diameter of the network.
- Contentions at incoming links to a node are resolved according to First-Come First-Served (FCFS) scheduling.
- Messages arriving at destinations are immediately consumed at the rate of one flit per time step, i.e., no blocking is encountered at destinations.

# Wormhole Performance Model - Latency

**The latency $L_j$ for a message injected at node $j$ is**

$$L_j = W_{inj,j} + x_{inj,j} + D - 1 \text{ , with}$$

1. $W_{inj,j}$ **= waiting time for the injecting channel.**
2. $x_{inj,j}$ **= service time at the injecting channel.**
3. $D - 1$ **= additional time to traverse the remaining channels.**




**A general node model.**

# Performance Model - Service Time

**At any RE, the service time at an incoming channel $x_i^{\text{in}}$ depends on the service times and waiting times at all possible outgoing channels.**

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

- $R_{i|j}$ = probability a message from incoming channel $i$ is routed to outgoing channel $j$,
- $x_j$ = service time for outgoing channel $j$,
- $w_{i|j}$ = waiting time for outgoing channel $j$ of messages from incoming channel $i$.

# Performance Modeling - Waiting Times

**Start with a well-known queueing model that has been employed to analyze store-and-forward routing:**

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

**and an approximation of Hokstad for multiple servers, e.g.,**

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

**where $\rho = \lambda\overline{x}/m$, $\lambda$ is arrival rate, $\overline{x}$ is mean service time, and $C_b^2 = \frac{\sigma_b^2}{\overline{x}^2}$, where $\sigma_b^2$ is the variance of the service time distribution and can be estimated as $(\overline{x} - s/f)^2$.**

# Performance Modeling - Waiting Times

**Need to correct waiting time for wormhole routing, since arrivals aren't really Poisson:**

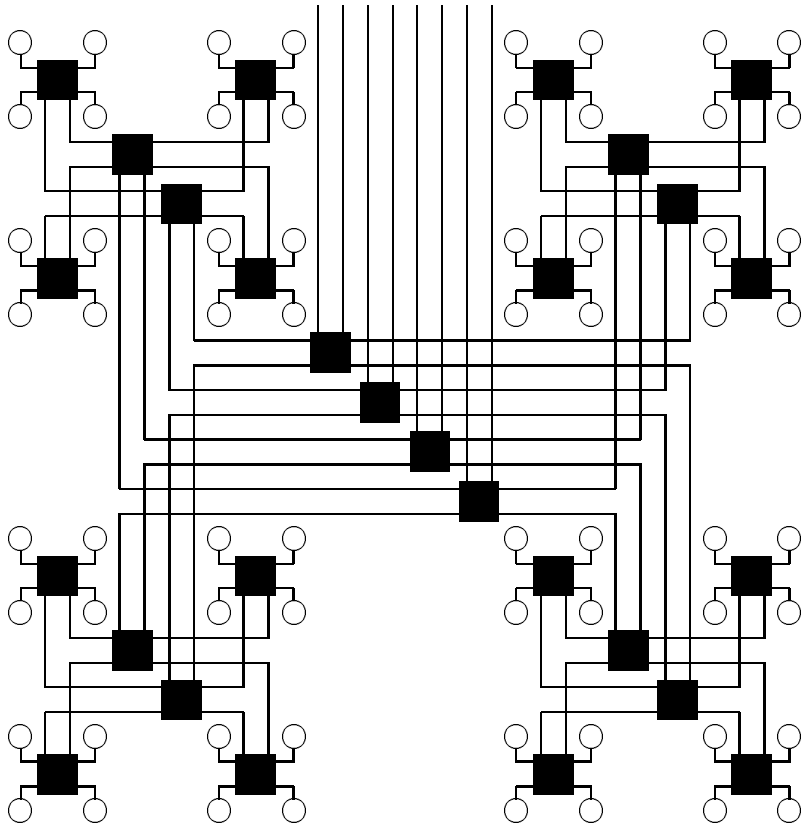
$$w_{i|j} = P_{i|j} W_j$$

**where $P_{i|j}$ = probability that a message from incoming link $i$ deemed to be blocked under M/G/m model is blocked by messages from $m$ distinct other links. A simple approximation:**

$$P_{i|j} = 1 - m \frac{\lambda_i^{\text{in}}}{\lambda_j} R_{i|j} \text{ , where}$$

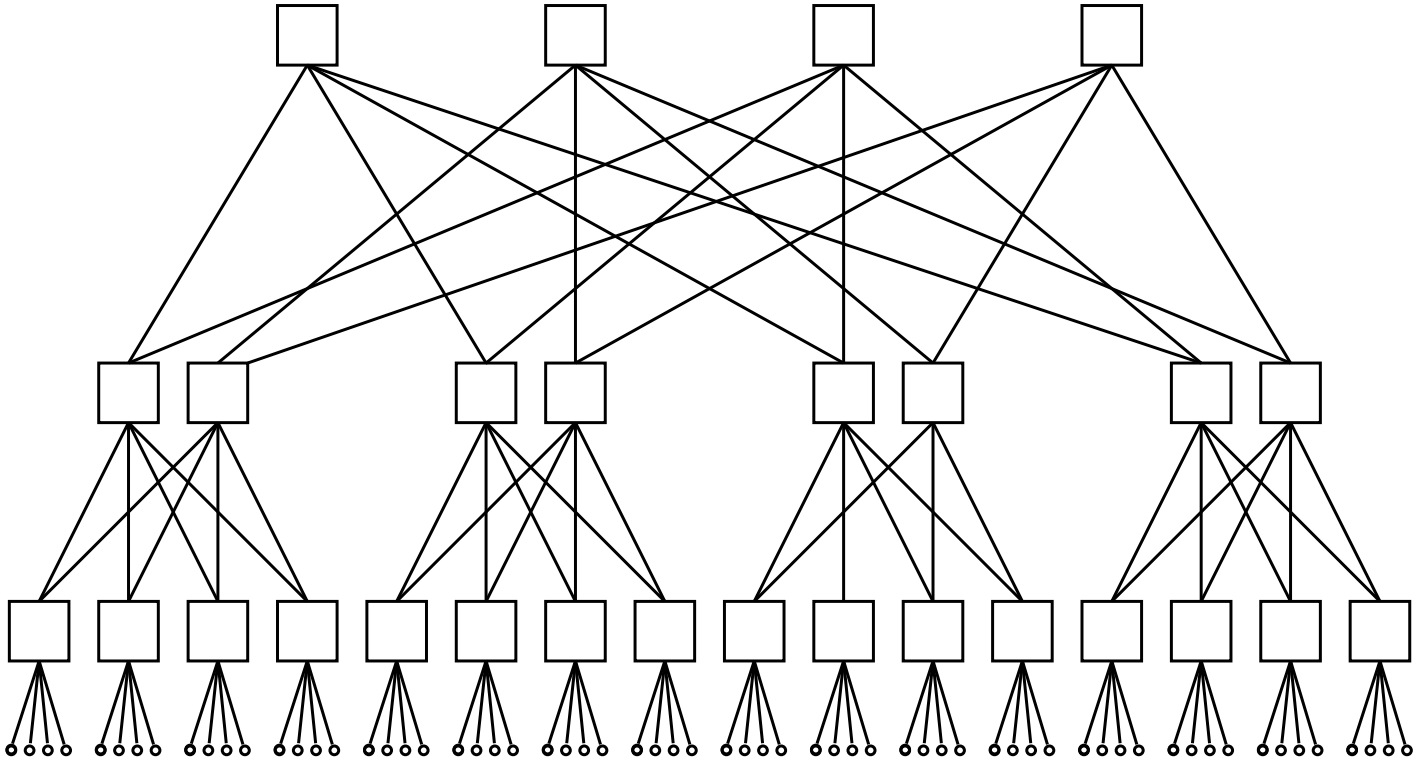
- $\lambda_i^{\text{in}}$ = total message rate on incoming channel $i$
- $\lambda_j$ = total message rate on outgoing channel $j$
- no. of servers, $m$, is less than no. of incoming links.

# Butterfly Fat-Tree



**A 64-node butterfly fat-tree.**

# Butterfly Fat-Tree



**A 64-node butterfly fat-tree.**

# Butterfly Fat-Tree Analysis - Message Rates

- $4^n$ **processors at level 0**
- **Switches at levels** $1, 2, \ldots, n$
- **Probability a message goes up from level** $l$ **is**

$$P_l^{\uparrow} = \frac{4^n - 4^l}{4^n - 1} \ ,$$

**and the probability that a message goes down is**

$$P_l^{\downarrow} = 1 - P_l^{\uparrow} \ .$$

- **Total message rate up from level $l$ is $P_l^{\uparrow} 4^n \lambda_0$, distributed among $4^n/2^l$ links, giving a rate on each upgoing link of**

$$\lambda_{l,l+1} = \lambda_0 \frac{4^n - 4^l}{4^n - 1} 2^l \ .$$

- **Downgoing rate from level $l+1$ same as upgoing rate from level $l$. But rate must be doubled when considering upgoing channels, since each is comprised of 2 links between which message may choose.**

# Butterfly Fat-Tree Analysis - Waiting and Service Times

**Resolve service times in reverse order from last downgoing channel back through upgoing channels.**

- **Last downgoing:**

$$x_{1,0} = s/f \text{ and } \overline{W}_{1,0} = \overline{W}_{M/G/1}(\lambda_{1,0}, x_{1,0}) \ .$$

- **Other downgoing:**

$$\overline{x}_{l+1,l} = \overline{x}_{l,l-1} + \left(1 - \frac{1}{4}\frac{\lambda_{l+1,l}}{\lambda_{l,l-1}}\right)\overline{W}_{l,l-1} \ ,$$

**and**

$$\overline{W}_{l+1,l} = \overline{W}_{M/G/1}(\lambda_{l+1,l}, \overline{x}_{l+1,l}) \ .$$

- **Upgoing to root:**

$$\begin{aligned}\overline{x}_{n-1,n} &= \overline{x}_{n,n-1} + \left(1 - \frac{\lambda_{n-1,n}}{\lambda_{n,n-1}}\frac{1}{3}\right)\overline{W}_{n,n-1} \\ &= \overline{x}_{n,n-1} + \frac{2}{3}\overline{W}_{n,n-1} \ ,\end{aligned}$$

**and**

$$\overline{W}_{n-1,n} = \overline{W}_{M/G/2}(2\lambda_{n-1,n}, \overline{x}_{n-1,n}) \ .$$

- **Other upgoing:**

$$\overline{x}_{l-1,l} = \left[\overline{x}_{l,l+1} + \left(1 - \frac{\lambda_{l-1,l}}{\lambda_{l,l+1}}P_l^{\uparrow}\right)\overline{W}_{l,l+1}\right]P_l^{\uparrow} + \left[\overline{x}_{l,l-1} + \left(1 - \frac{P_l^{\downarrow}}{3}\right)\overline{W}_{l,l-1}\right]P_l^{\downarrow} \ ,$$

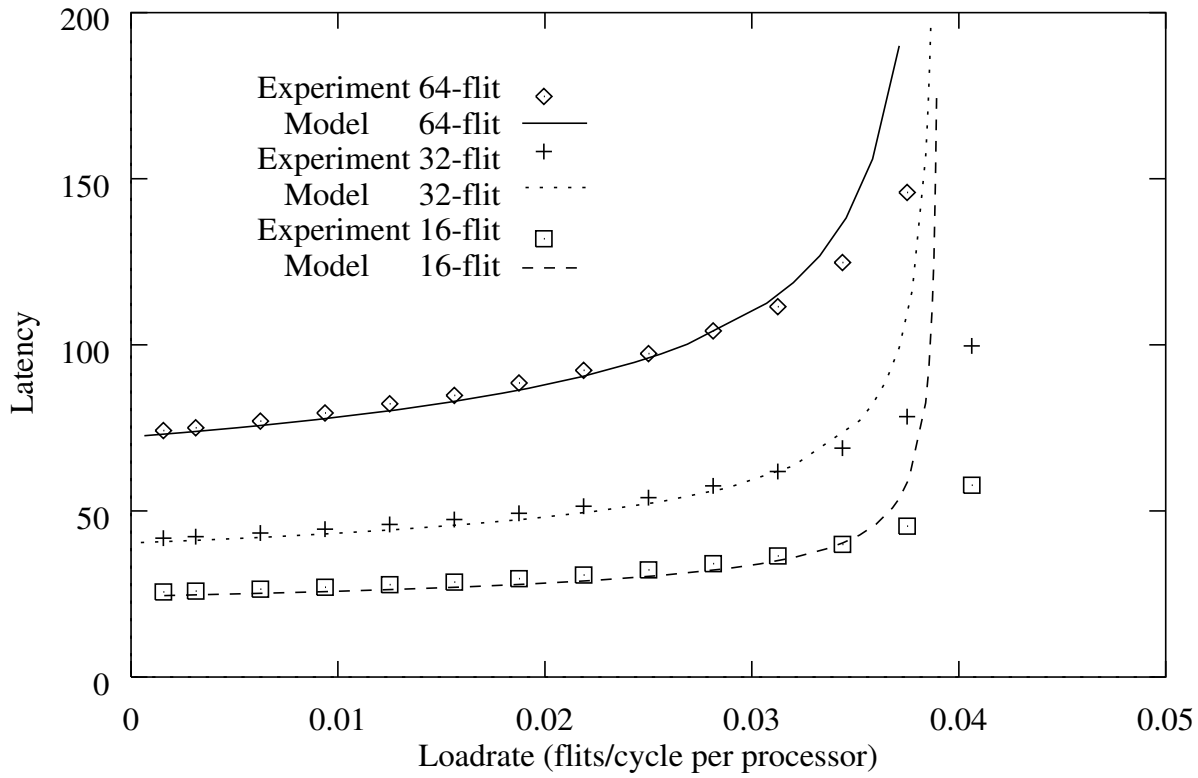
**and**

$$\overline{W}_{l-1,l} = \overline{W}_{M/G/2}(2\lambda_{l-1,l}, \overline{x}_{l-1,l}) \ .$$

- **First upgoing:**

$$\overline{W}_{0,1} = \overline{W}_{M/G/1}(\lambda_{0,1}, \overline{x}_{0,1}) \ .$$

# Experimental Validation




**Comparisons of latency and throughput between model and simulation for 1024-processor network**