



Communication Costs in Parallel Machines

Alexandre David


1.2.05



Topics

- Communication Costs in Parallel Machines (2.5)
 - MPI
 - Shared Address Space
- Routing Mechanisms for Interconnection Networks (2.6)

Communication Costs in Parallel Machines

- Sources of overhead in parallel programs:
 - Idling – nothing to do.
 - Contention – blocked.
 - Communication costs. 
- Costs depend on
 - Programming model.
 - Network topology.
 - Routing...

Communication cost is one of the major overheads.



Message Passing Costs

- Total time for communication =
 - Startup time (t_s) – *only once per message*
 - + Per-hop time (t_h) – *between directly connected nodes (aka node latency)*
 - + Per-word transfer time (t_w) – *1/bandwidth.*

27-02-2008

Alexandre David, MVP'08

4

The total time to transfer a message over a network comprises of the following:

- *Startup time* (t_s): Time spent at sending and receiving nodes (executing the routing algorithm, programming routers, etc.).
- *Per-hop time* (t_h): This time is a function of number of hops and includes factors such as switch latencies, network delays, etc. Also known as **node latency**.
- *Per-word transfer time* (t_w): This time includes all overheads that are determined by the length of the message. This includes bandwidth of links, error checking and correction, etc.



Store-and-Forward Routing

- Intermediate nodes
 - *store* the whole message.
 - *forward* the whole message.
- Message size m traversing l links:
 - $t_{com} = t_s + (m * t_w + t_h) * l$
 - Pay $m * t_w$ for every link.
 - In practice $t_{com} = t_s + m * t_w * l$

27-02-2008

Alexandre David, MVP'08

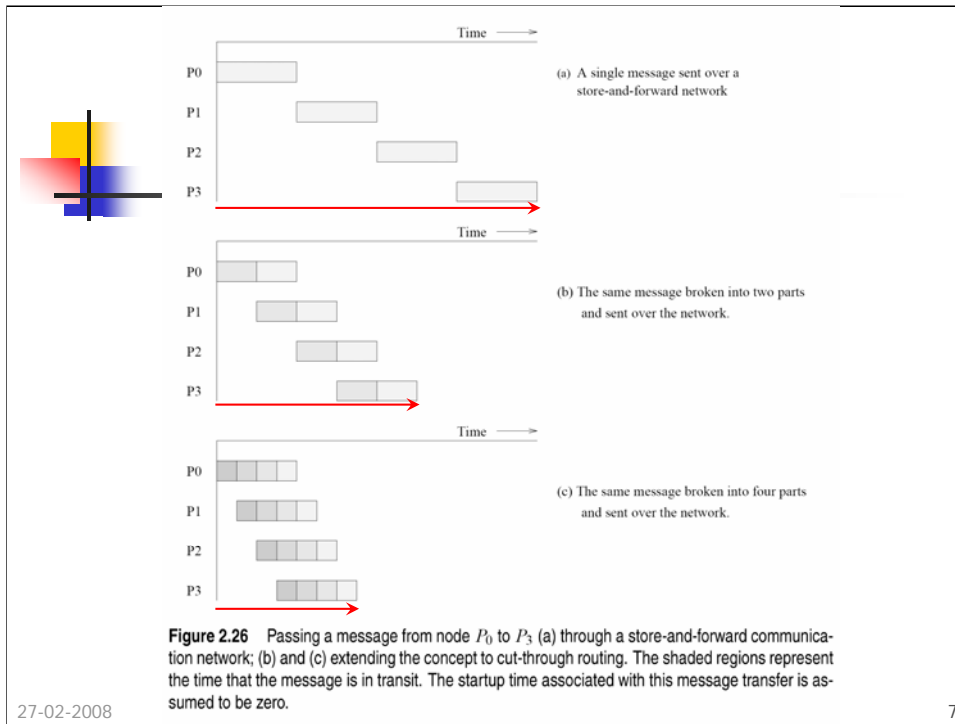
5

Simplify in practice because t_h is small.
Obviously inefficient for long messages!




Packet Routing

- Cut the message in smaller parts.
- Advantages:
 - Lower overhead for errors (less retransmission).
 - More robust routing (different paths for packets, avoid congestion).
 - Better error correction.
 - Better resource utilization (like pipeline).
 - But... more complex protocol.



Comparison of store & forward with cutting the message in to 2 and 4 packets.



Packet Routing

Packet =

s	r
---	---

Message m = (m/r)*r

- Message size m traversing l links:
 - $t_{com} = t_s + t_h * l + t_w * m$
 - with $t_w = \underbrace{t_{w1}}_{\text{packing}} + \underbrace{t_{w2}(1+s/r)}_{\text{overhead}}$

Compared to

$$t_{com} = t_s + m * t_w * l$$

(store and forward)

$$t_{com} = t_s + t_h * l + t_{w1} * m + t_{w2} * m + (m/r - 1) t_{w2} * m$$

27-02-2008
Alexandre David, MVP'08
8

Approximation here.

Goal is not to remember a bunch of formulas but to understand how to model communication costs.



Cut-Through Routing

- Simplified packet routing:
 - Packets take the same path (1x routing information).
 - In sequence packet delivery (no sequencing).
 - Error detection at message level, cheap detection (for good networks).
 - Fixed size unit for packets = flow control digits (flits).
 - Same cost model with smaller s .

27-02-2008

Alexandre David, MVP'08

9

It is an optimization for interconnection networks of parallel machines since error rates are very low (dedicated network).

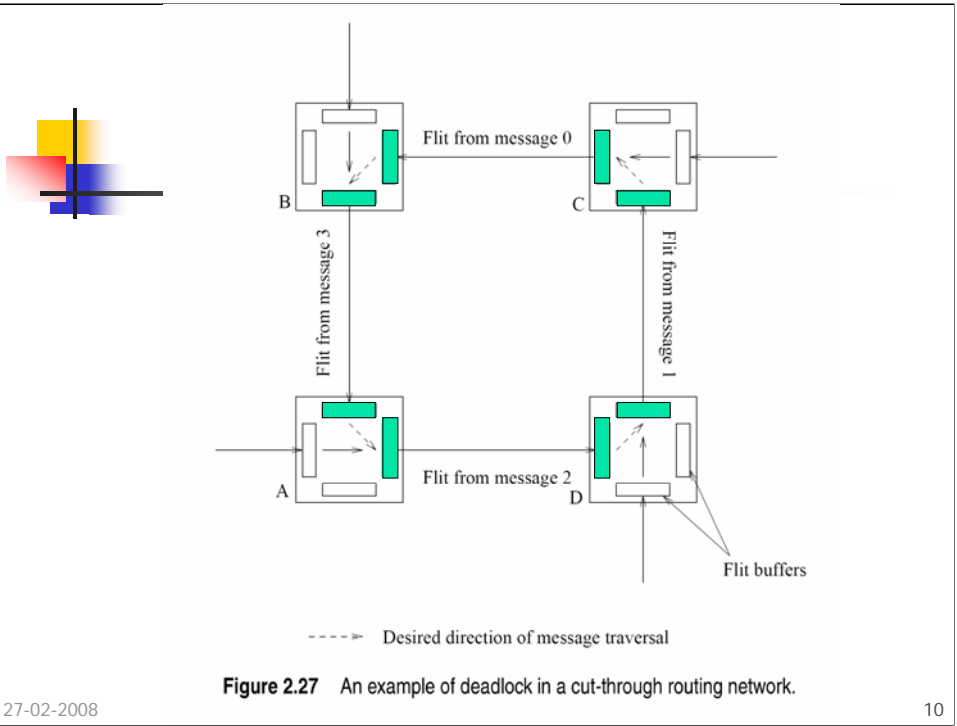


Figure 2.27 An example of deadlock in a cut-through routing network.



Simplified Cost Model

- $t_{com} = t_s + t_h * l + t_w * m$ *uncongested network*
- Optimize:
 - Communicate in bulk (fewer t_s).
 - Minimize volume (smaller m).
 - Minimize number of hops (smaller l), but difficult.
- Almost same time between any pair = **like a completely connected network.**

27-02-2008

Alexandre David, MVP'08

11

Simplification justified by t_s (in practice) and $t_w * m$ (for algorithm we will see) much larger than $t_h * l$.

Point 3 difficult because the program has little control over this parameter that is more architecture bound. Can use proximity with a good process-processor mapping.

Original expression valid for **uncongested** networks. Communication patterns have an impact on congestion. Incorporate congestion: links have to transport more messages (x), so t_w is affected and it takes x messages more time -> talk about **effective bandwidth** to scale down bandwidth (or scale up transmission time)

Costs in Shared Address Space Machines

- Difficult to have accurate models because:
 - Memory layout depends on the system.
 - Limitations with caches.
 - Invalidate/update overheads difficult to estimate (cache coherence protocols).
 - Spatial locality difficult to estimate.
 - Prefetching plays its role.
 - False sharing may be a problem.
 - Contention...

27-02-2008

Alexandre David, MVP'08

12

So, use the same model as before with much smaller t_w (for UMA machine).




Routing Mechanisms for Interconnection Networks

- Goal: find a path from src to dest.
- Types:
 - **Minimal**: selects shortest path, progress at every hop – prone to congestion
 - **vs. non-minimal**: may use longer path to avoid *congestion*.
 - **Deterministic**: finds a unique path
 - **vs. adaptive**: use current state to find a path.

Common issue is congestion.



Good Routing

- Prevents deadlock.
 - Use dimension ordered routing. 
 - XY-routing for 2-D mesh.
 - E-cube routing for hypercubes.
- Avoids hot-spots.
 - Two-step routing may be used.

27-02-2008

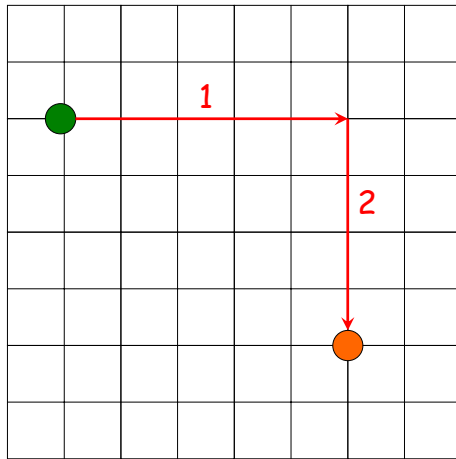
Alexandre David, MVP'08

14

Deterministic minimal routing commonly used: dimension ordered routing.
Use a numbering scheme for channels determined by the dimension.

Two-step routing: 1) choose an intermediate randomly, 2) route.


XY-Routing



Path length =
 $|S_x - D_x| + |S_y - D_y|$



E-Cube Routing

- N-dimension hypercube:
 - Nodes have N neighbors.
 - 2^N nodes.
 - Numbering scheme s.t. change 1 bit along any dimension.
-  Routing: progress towards a goal number.

E-Cube Routing

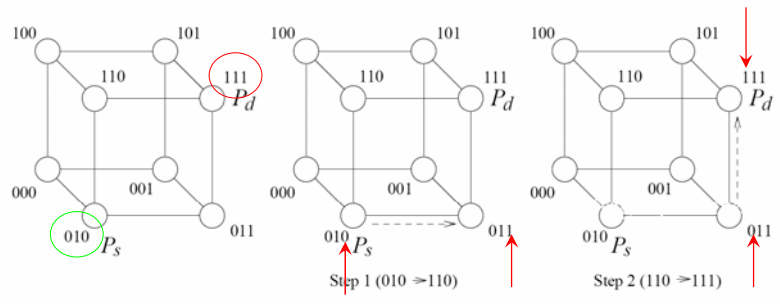


Figure 2.28 Routing a message from node P_s (010) to node P_d (111) in a three-dimensional hypercube using E-cube routing.