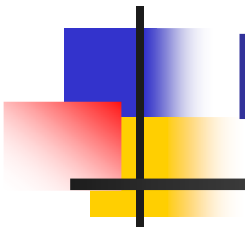
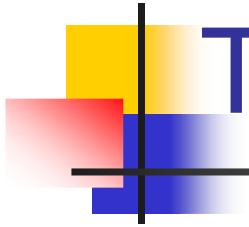


Communication Costs in Parallel Machines

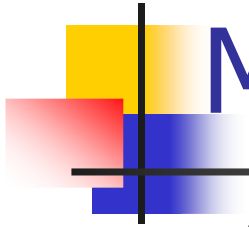
Alexandre David
B2-206



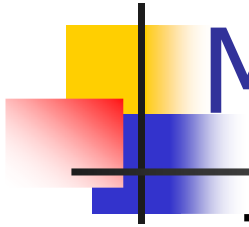
Topics

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

Communication Costs in Parallel Machines

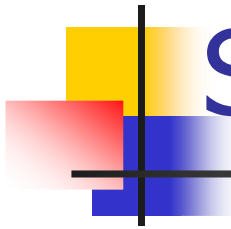


- Sources of overhead in parallel programs:
  - Idling.
  - Contention.
  - Communication costs.
- Costs depend on
  - Programming model.
  - Network topology.
  - Routing…



Message Passing Costs

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

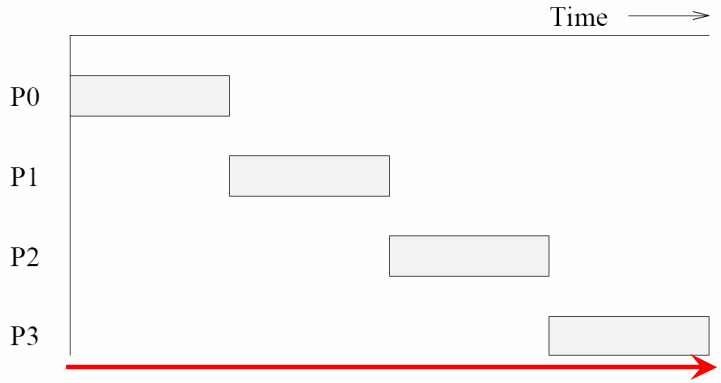


Store-and-Forward Routing

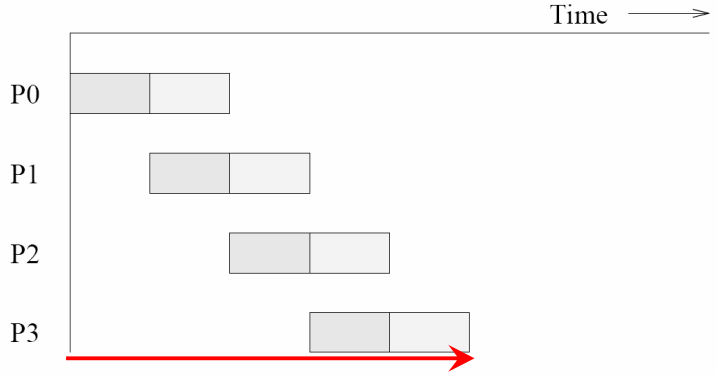
- Intermediate nodes
  - *store* the whole message.
  - *forward* the whole message.
- Message size m traversing l links:
  - $t_{com} = t_s + (m*t_w + t_h)*l$
  - Pay $m*t_w$ for every link.
  - In practice $t_{com} = t_s + m*t_w*l$

Packet Routing

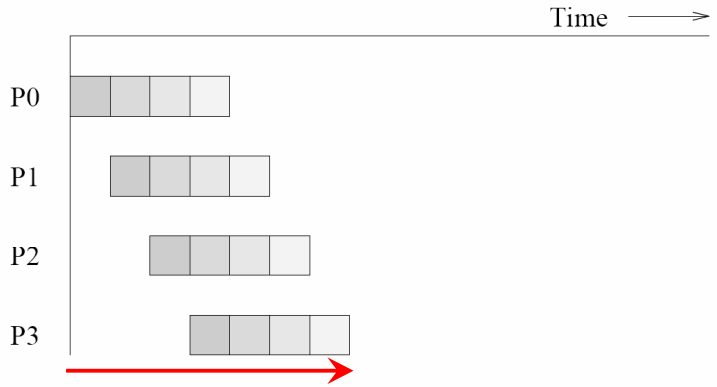
- Cut the message in smaller parts.
- Advantages:
  - Lower overhead for errors (less retransmission).
  - More robust routing (different paths for packets, avoid congestion).
  - Better error correction.
  - Better resource utilization (like pipeline).
  - But… more complex protocol.



(a) A single message sent over a store-and-forward network



(b) The same message broken into two parts and sent over the network.



(c) The same message broken into four parts and sent over the network.

Figure 2.26 Passing a message from node P_0 to P_3 (a) through a store-and-forward communication network; (b) and (c) extending the concept to cut-through routing. The shaded regions represent the time that the message is in transit. The startup time associated with this message transfer is assumed to be zero.

Packet Routing

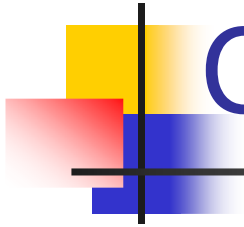
- Message size m traversing l links:
  - $t_{com} = t_s + t_h*l + t_w*m$
  - with $t_w = t_{w1} + t_{w2}(1+s/r)$

packing

overhead

Packet = | s | r |

Message = | r | r | r | r | r |



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*.

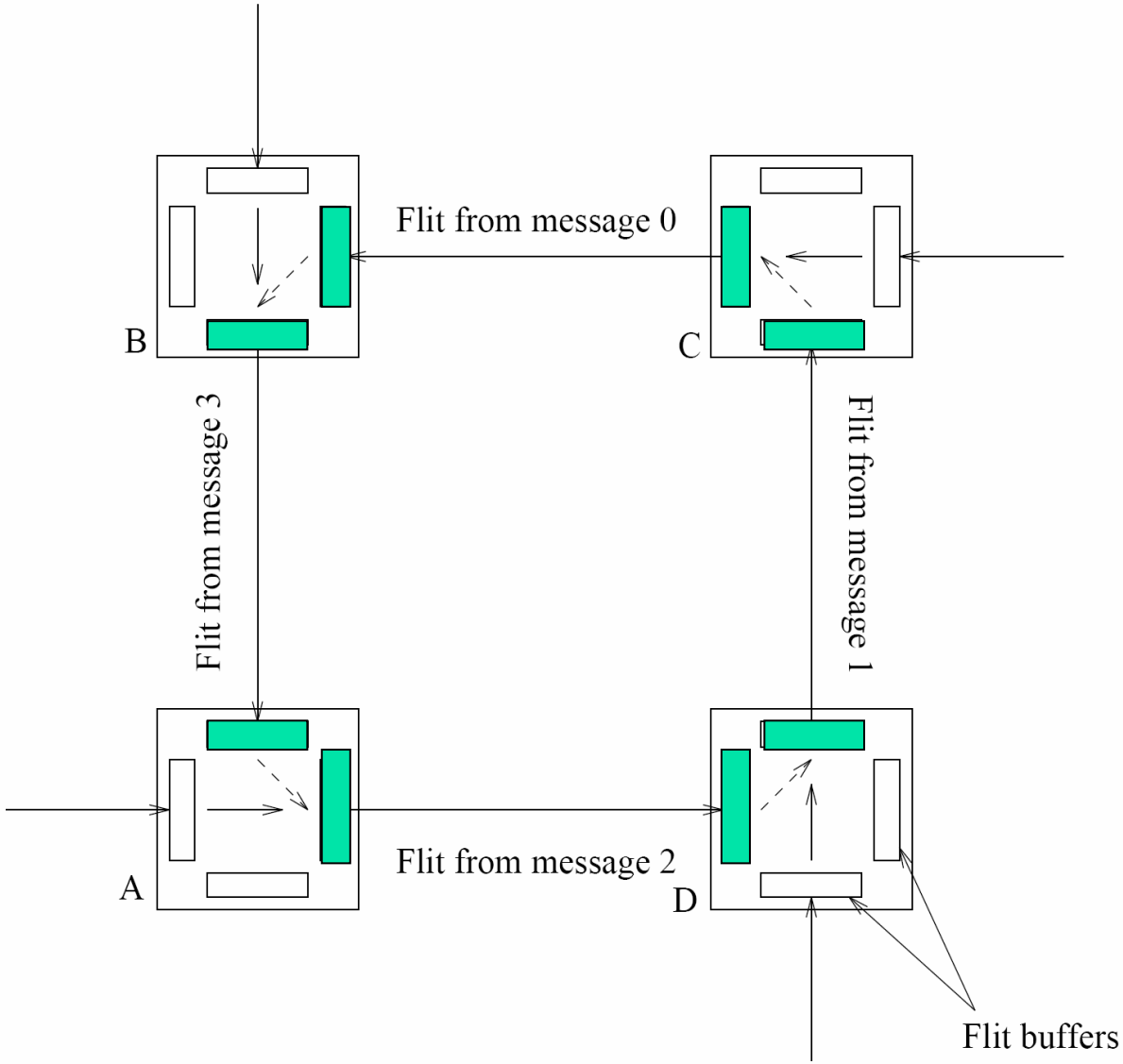
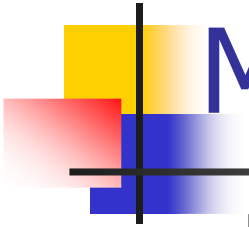


Figure 2.27 An example of deadlock in a cut-through routing network.

Simplified Cost Model

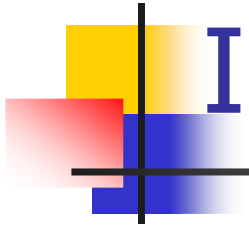
- $t_{com} = t_s + t_h*l + t_w*m$
- Optimize:
  - Communicate in bulk (fewer t_s).
  - Minimize volume (smaller m).
  - Minimize number of hops (smaller l), but difficult.
- Almost same time between any pair = like a completely connected network.

Costs in Shared Address Space Machines

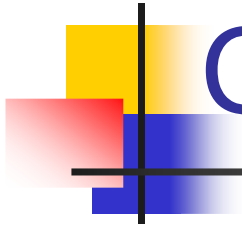


- Difficult to have accurate models because:
  - Memory layout depends on the system.
  - Limitations with caches.
  - Invalidate/update overheads difficult to estimate (cache coherence protocols).
  - Spatial locality difficult to estimate.
  - Prefetching plays its role.
  - False sharing may be a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

Routing Mechanisms for Interconnection Networks



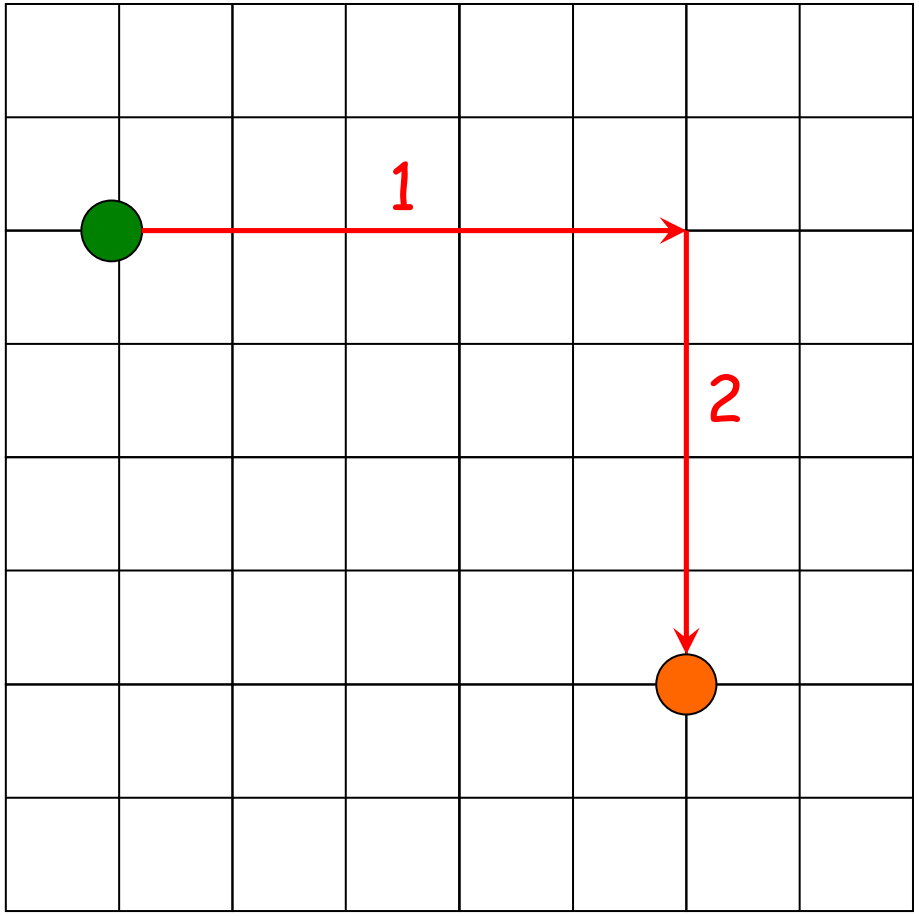
- Goal: find a path from src to dest.
- Types:
  - Minimal: selects shortest path, progress at every hop – prone to congestion
  - vs. non-minimal: may use longer path to avoid *congestion*.
  - Deterministic: finds a unique path
  - vs. adaptive: use current state to find a path.



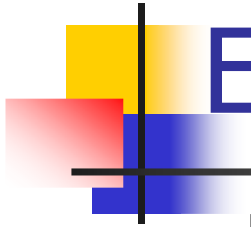
Good Routing

- 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.

XY-Routing



Path length =
|Sx - Dx| + |Sy - Dy|



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

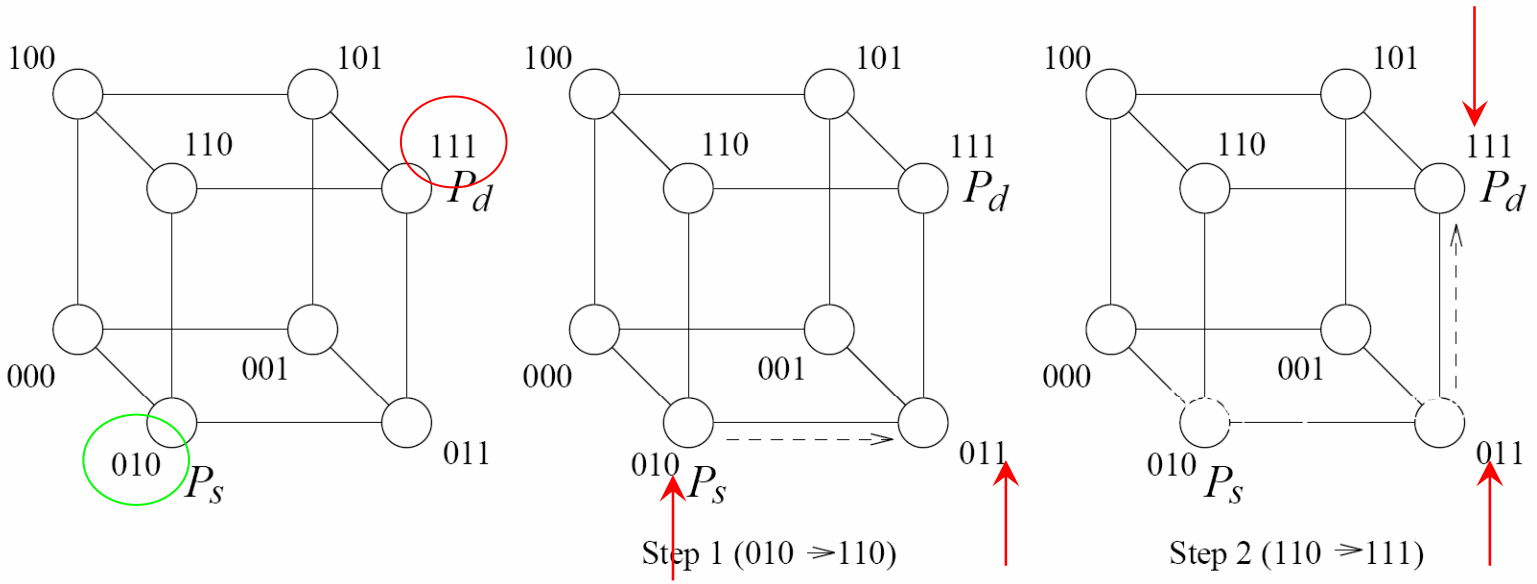


Figure 2.28 Routing a message from node P_s (010) to node P_d (111) in a three-dimensional hypercube using E-cube routing.