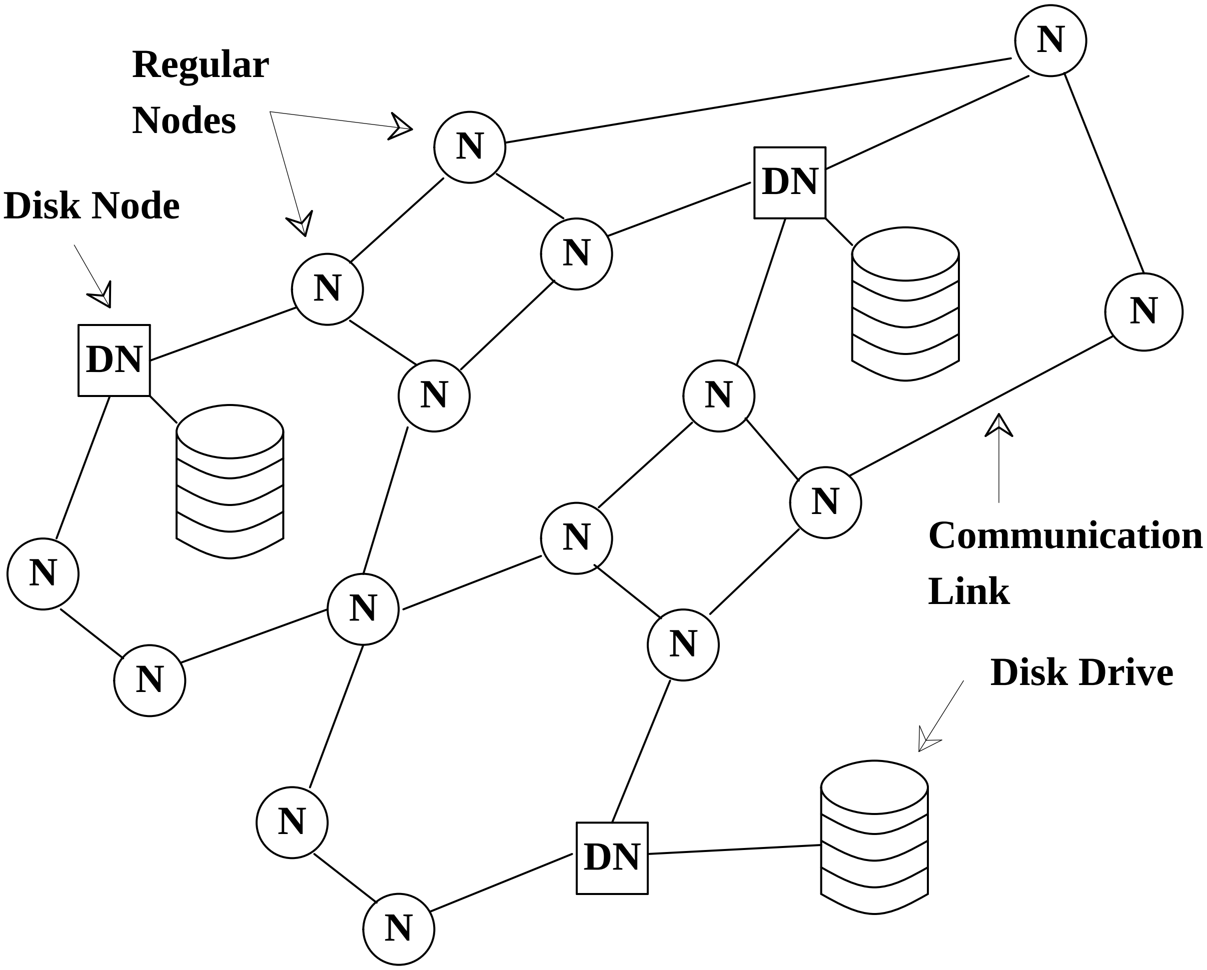
Connection-Based Adaptive Routing Using Dynamic Virtual Circuits

Yoshio F. Turner and Yuval Tamir

Concurrent Systems Laboratory

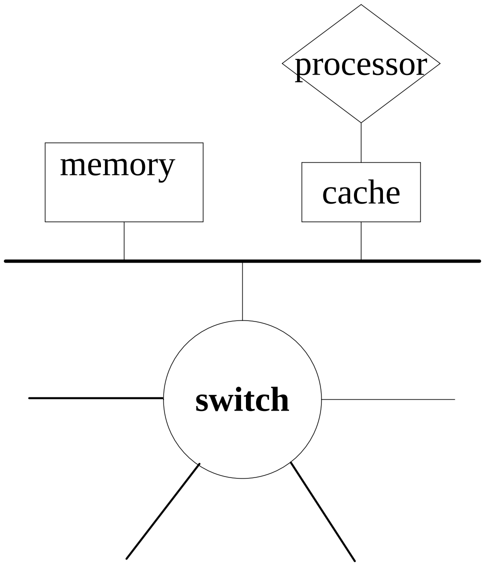
UCLA Computer Science Department

System Model

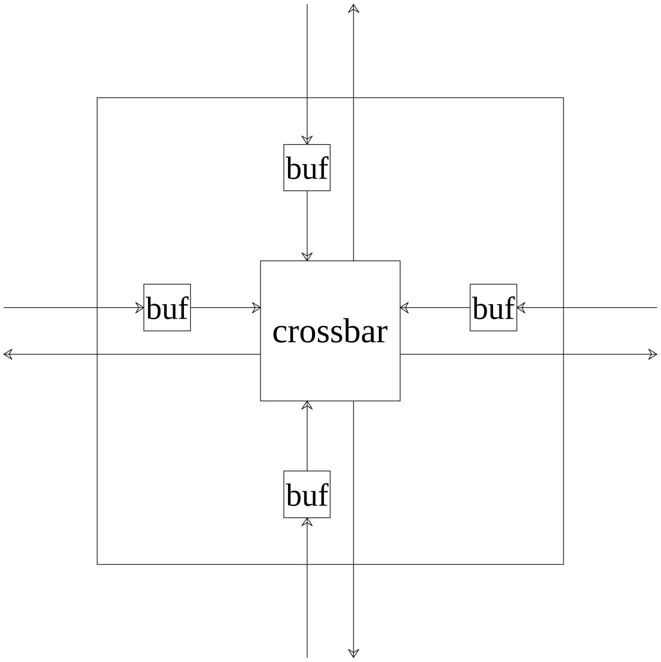


Network with up to thousands of nodes

Node



Switch



Input-buffered, virtual cut-through, N × N switch.

Connection-Based Routing

- Defn: In *connection-based routing*, network resources are reserved in advance of communication.

- Potential Advantages
  - ∗ Reduced processing for packet interpretation/routing.
  - ∗ Reduced addressing/control information transmitted with each packet.
  - ∗ Efficient network resource utilization.

- Connection setup must be fast or infrequent relative to communication.

→Examples: circuit switching, static virtual circuits.

Static Virtual Circuits

- Virtual Circuit (connection) establishment:

  Set up source-destination path —

  entries in routing tables along the path.

  Routing Virtual Channel (RVC): entry in routing table.

- Packet Header: routing table entry (RVC #) in next node.

- After circuit setup, data packets are routed (mapping table lookup) along the virtual circuit’s path.

- Reserved RVCs are released upon termination of the static virtual circuit.

- Each physical link is demand time multiplexed among the active virtual circuits using the link.

- FIFO ordering of packets on a connection.

Problems with static virtual circuits

- Once established, a virtual circuit’s path is fixed — cannot adapt to changes in the system.

- During its lifetime, a virtual circuit cannot release resources to other virtual circuits.

  - New circuits may be prevented from being established.

  - Idle circuits consume RVCs.

  - Circuits may permanently occupy resources if processes terminate without disestablishing their circuits.

Outline

I. Dynamic Virtual Circuits (DVCs): Support for connection-based adaptive routing.

II. Original DVC Approach: Deadlock Resolution

III. New DVC Approach
    A. Deadlock Avoidance
    B. Maintaining FIFO Ordering

IV. Potential performance (bounds, limit cases)

V. Conclusion/Future Work

Dynamic Virtual Circuits (DVCs)

Distributed mechanism for tearing down and re-routing circuits on demand.

→ Support for re-routing due to congestion.

- Circuits can be torn down from intermediate nodes without involving the source nodes.
- Circuit establishment is guaranteed by freeing needed resources.
- Resources held by idle circuits are reclaimed before those held by active circuits.

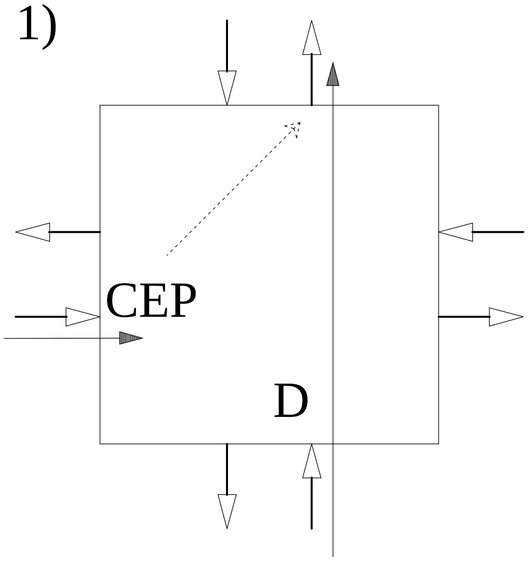
DVC example (CEP, no free RVCs)

CEP = Circuit Establishment Packet

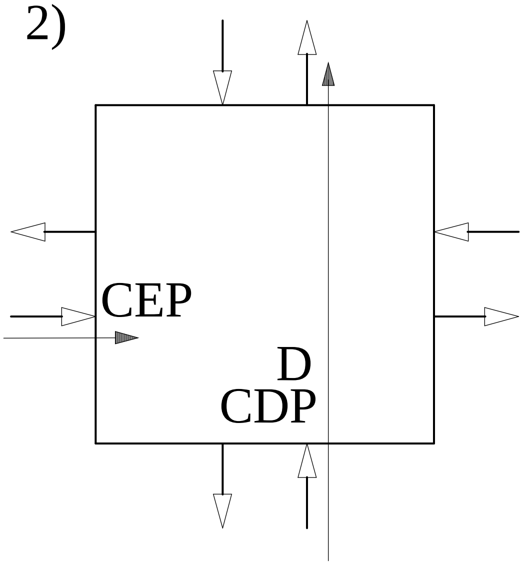
CDP = Circuit Destruction Packet

D = Data Packet

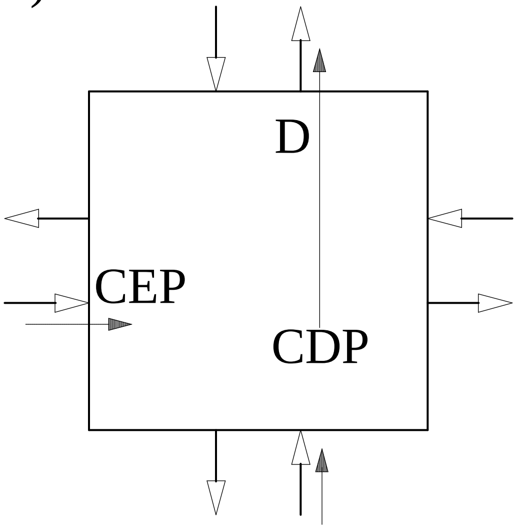
1)



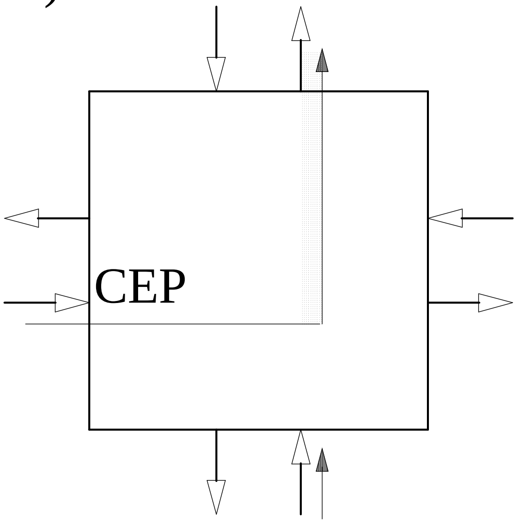
2)



3)



4)



DVC Challenges

- Low per-hop overhead.
- Deadlock-free fully-adaptive routing:
  - ∗ Standard packet buffer deadlock problem.
  - ∗ Complication: deadlocks involving RVCs.
    - → RVC allocation depends on data packet transmission.
    - → Data packet transmission depends on RVC allocation at a neighbor switch.

First Approach: Deadlock Resolution

Motivation: Avoid constraints on routing and buffer utilization.

1. Detect deadlock
2. Determine cycle of resource requests
3. Resolve deadlock by careful introduction of additional resources.

Jaffe, Sidi, *Algorithmica*, **4**(5), 1989

- Detection — idle full buffer
- Cycle determination — control messages used to determine cycle of full buffers, auxiliary buffer
- Auxiliary buffers used to ‘‘rotate’’ packets around cycle

– Deadlock cycles resolve too slowly to prevent formation of new cycles.

Alternative: Deadlock Avoidance

J. Duato, ‘‘A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks,’’ *IEEE Tr. Par. & Dstr. Sys.*, **7**(8), 1996.

Approach: restrict buffer utilization.

Buffering Virtual Channels (BVCs): separately flow-controlled buffers. Called elsewhere ‘‘virtual channels.’’

Sufficient Condition:

- A set *C* of BVCs can be reached in one hop by all packets.
- Routing in set *C* reaches all destinations from all nodes.
- The set *C* is free of buffer dependency cycles.

DVC Deadlock Avoidance

Two BVCs (Buffering Virtual Channels) per link for data packets.

1. *Primary BVC:* fully-adaptive routing. Associated with *primary buffer*, large DAMQ/FIFO/etc.
2. *Diversion BVC:* restricted routing with no buffer dependency cycles. Associated with small *diversion buffer*.

- The set of Diversion BVCs forms the ‘‘*diversion network*’’ — the set *C* of the sufficient condition.
- After timeout, blocked packet in primary buffer may be diverted.

Control Packets

PROBLEM: control packets must not be diverted.

⇒ control packet deadlock cycles.

SOLUTION OUTLINE:

1. Impose a restriction that limits the demand for buffer space by control packets.

2. Provide sufficient buffer space to eliminate inter-switch control packet blocking. *Dedicated buffers for control packets.*

   Based on enumeration of all possible buffer requirements by control packets.

3. Remove the new dependency created by the implementation of the imposed restriction.

Limiting Buffer Space Demand

Restriction: only one unmapped data packet per input.

All sequences of enqueued packets with RVC i:

1. No data packets.

empty

(tail) [CDP_1] CEP_1 (head)

(tail) CDP_1 (head)

(tail) [CDP_2] CEP_2 CDP_1 (head)

2. Mapped data packets (M).

[CDP_2] CEP_2 CDP_1 $\mathrm{M}_n \cdots \mathrm{M}_1$

3. Unmapped data packet (U).

[CDP_3] CEP_3 CDP_2 U $\mathrm{CEP}_2\mathrm{CDP}_1$

4. Mapped (M) and unmapped (U) data packets.

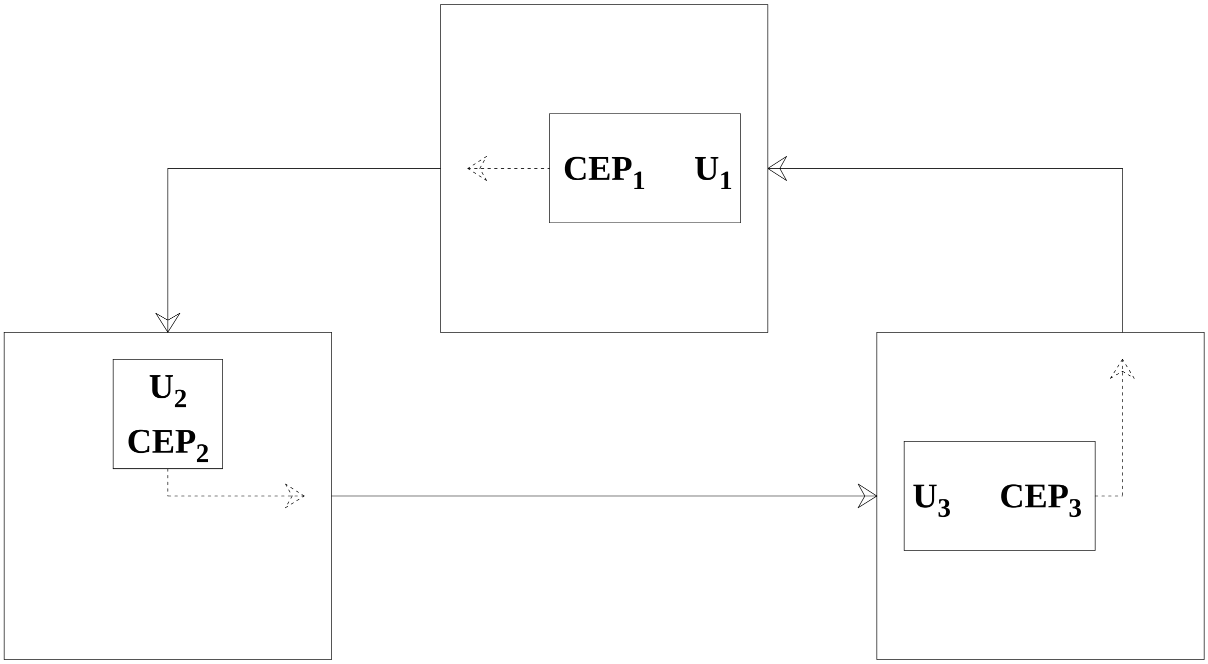
[CDP_3] CEP_3 CDP_2 U $\mathrm{CEP}_2\mathrm{CDP}_1$ $\mathrm{M}_n \cdots \mathrm{M}_1$

$\boxed{\text{R RVCs} \Rightarrow \text{R CDPs + R CEPs + 1 CEP + 1 CDP.}}$

One Unmapped Data Packet per Input: Implementation and Deadlock Avoidance

Enforce restriction: block flow to primary BVC upon arrival of an unmapped data packet.

PROBLEM: Cyclic dependencies between control and unmapped packets.



SOLUTION: Introduce *Control BVC*.

→ Separates control and data packets, breaks inter-node dependencies.

Maintaining FIFO Ordering

Diversion and re-routing violate FIFO ordering.
Goal: restore FIFO ordering without stamping all packets with sequence numbers.

- Sequence number field in RVC Mapping Table.
- Attach sequence number to diverted packets and subsequent data packet.
- Subsequent data packet updates the table field at each hop.
- Destination host interface reconstructs FIFO ordering from packets with sequence numbers and consecutive ordering of normal data packets.

Hardware Costs

Assume: 32 RVCs per input port, 16-bit SRC/DST/SEQ.

State For Static Virtual Circuits

RVC Mapping Table Fields:

valid (1)	output port (3)	output RVC (5)

⇒ 36 bytes per port.

Additional State For DVCs:

- For Each Input Port:
  - + One diversion buffer per port: 40 bytes
  - + One CEP: 8 bytes
- For Each RVC:
  - * Frequently Accessed
    - + Sequence number: 2 bytes
    - + ‘‘Out of Sequence’’ flag: 1 bit
    - + Victim selection bits: 3 bits
    - ⇒ 80 bytes per port.
  - * Infrequently Accessed
    - + Diversion/Reroute Info: 4 bytes
    - + Victim selection bits: 11 bits
    - + Control BVC Storage:

      Control packet sequence (encoded): 3 bits

      One CEP (minus header): 6 bytes
    - ⇒ 376 bytes per port.

Performance Potential: Limit Studies

Goal: Evaluate potential for reducing network contention by choosing low latency paths for DVCs.

- Comparison: routed circuits versus Dimension-Order Routing (DOR).

- Details: 8×8 mesh, packet length 32 phits, cut-through routing, DAMQ primary buffers, diversion buffer capacity = 32 phits, FIFO crossbar priority.

- Stable traffic patterns:

  1. Uniform. DOR performs well.

  2. Transpose. Source row i col j transmits to destination row j col i. Poor DOR performance.

  3. Bit-Reversal: Source $x_{i-1}x_{i-2} \cdots x_0 y_{i-1} y_{i-2} \cdots y_0$ transmits to destination $y_0 y_1 \cdots y_{i-1} x_0 x_1 \cdots x_{i-1}$. Poor DOR performance.

Uniform Traffic

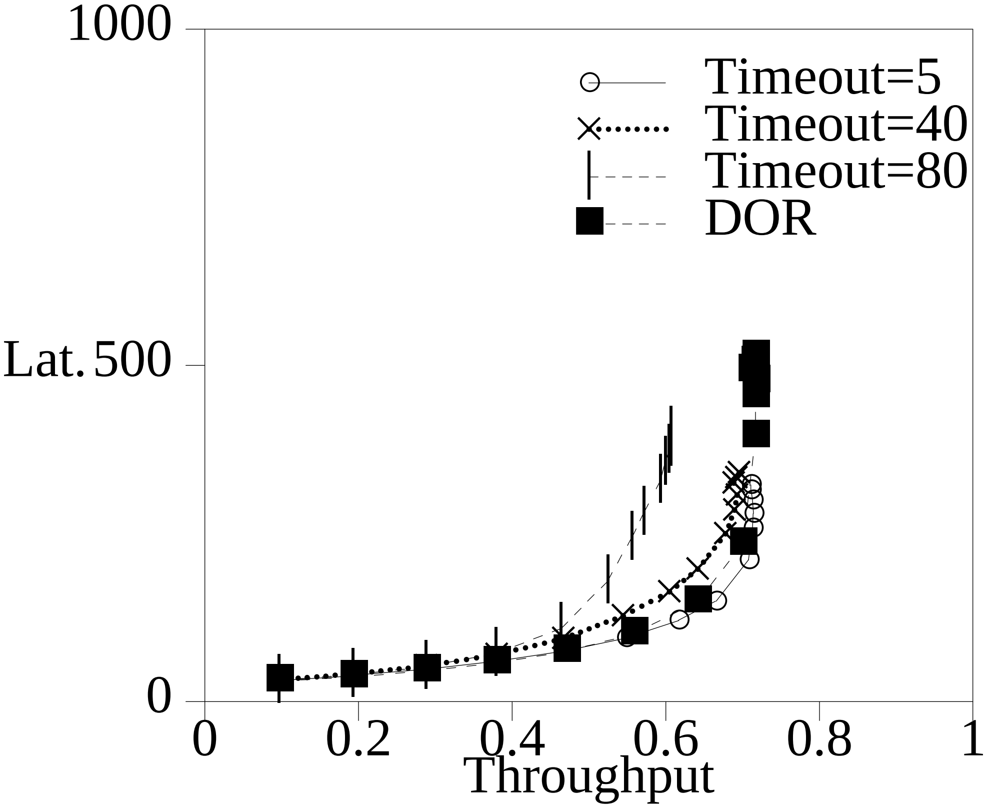


FIGURE 1: LATENCY VS. NORMALIZED THROUGHPUT. Total input buffer capacity = 64 phits. Uniform traffic pattern.

Transpose Pattern

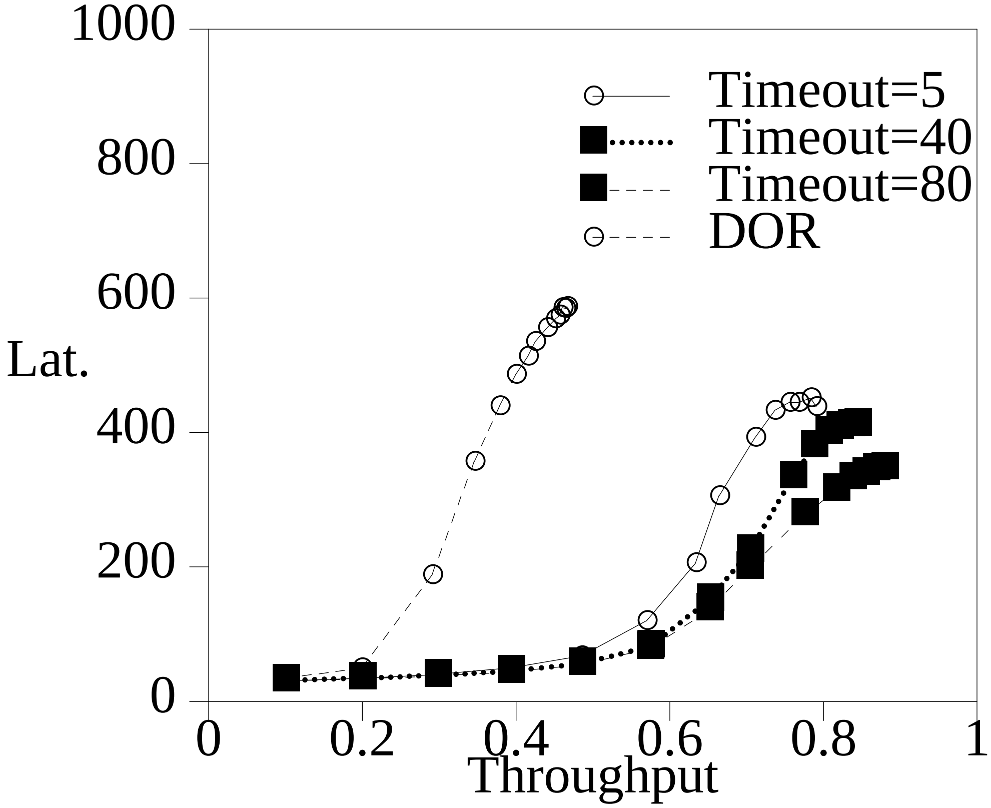


FIGURE 2: LATENCY VS. NORMALIZED THROUGHPUT. DAMQ buffer capacity = 64 phits. Transpose traffic pattern.

Transpose (continued)

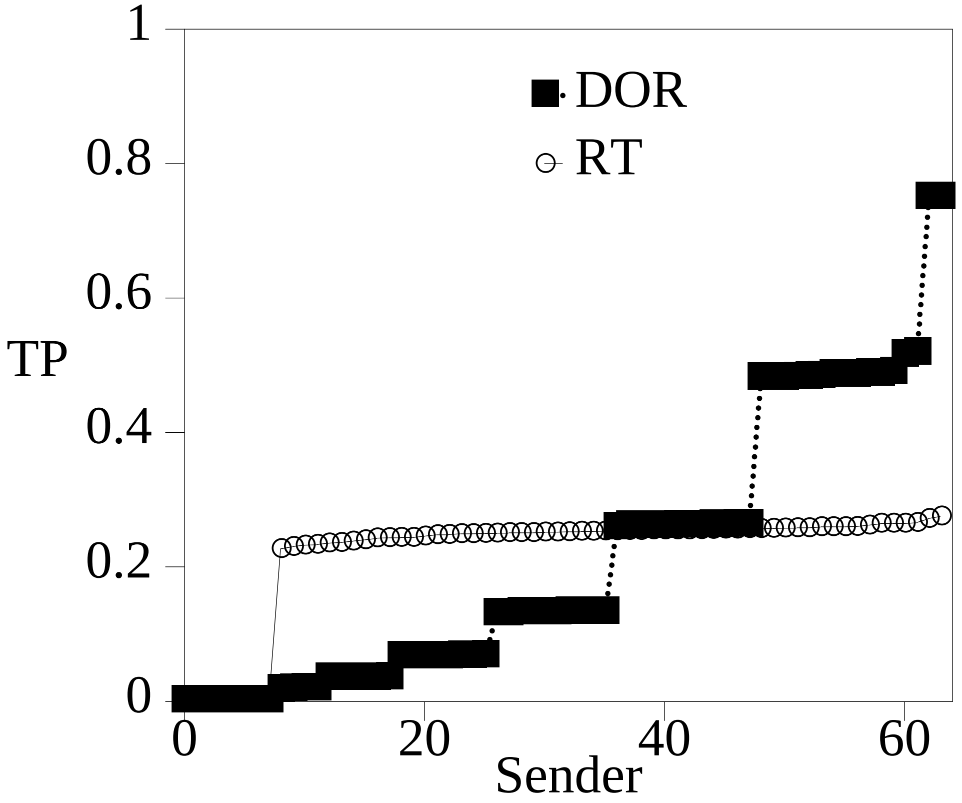


FIGURE 3: THROUGHPUT FAIRNESS. Throughput vs. Sender, sorted. Aggregate raw throughput = 0.233 for DOR, 0.242 for routed virtual circuits.

Transpose (continued)

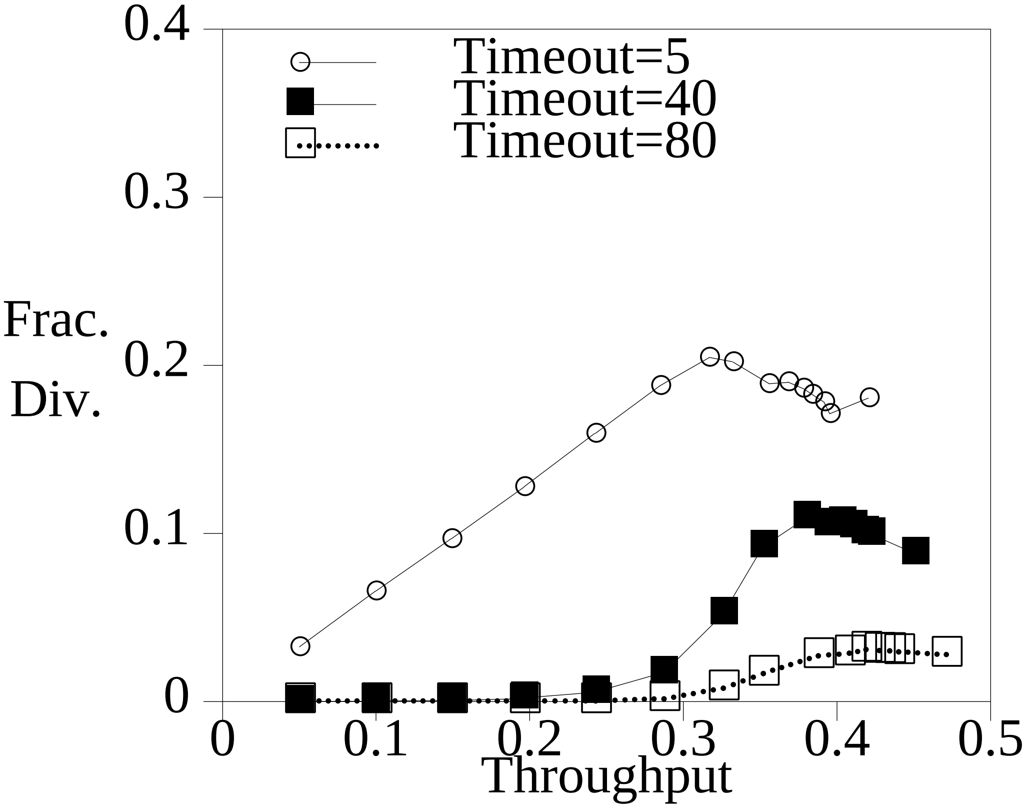


FIGURE 4: FRACTION OF TRAFFIC DIVERTED VERSUS AGGREGATE THROUGHPUT. Throughput is measured as useful phits received per cycle per receiver. DAMQ primary input buffer capacity = 64 phits.

Bit Reversal Traffic

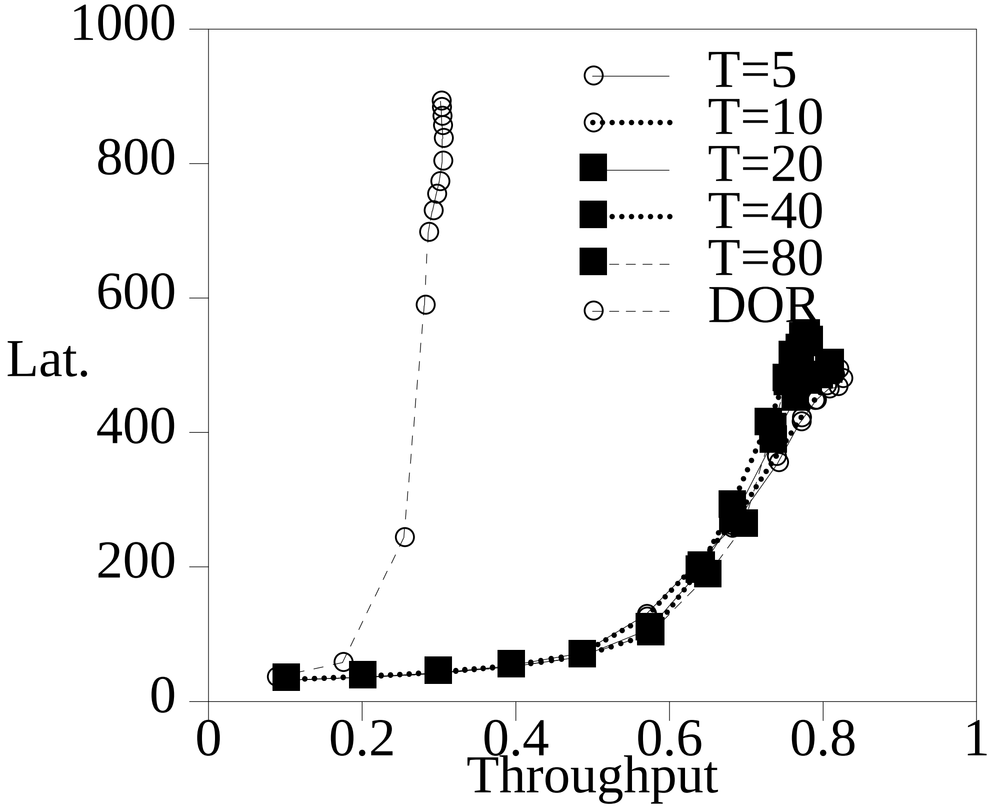


FIGURE 5: LATENCY VS. NORMALIZED THROUGHPUT. DAMQ buffer capacity = 64 phits. Bit reversal traffic pattern. T = Timeout.

Conclusion

- DVCs retain traditional advantages of virtual circuits: low per-packet bandwidth overhead, FIFO delivery, and establishment on paths with low contention.
- DVCs provide adaptive circuit rerouting and efficient circuit establishment even when RVCs are fully allocated.
- Performance results show potential of global routing optimization and demonstrate low frequency of packet diversion.
- Future work: shifting traffic patterns, alternatives for choosing when to reroute circuits, fault tolerance, multicast virtual circuits.

16x16 Mesh Transpose Pattern

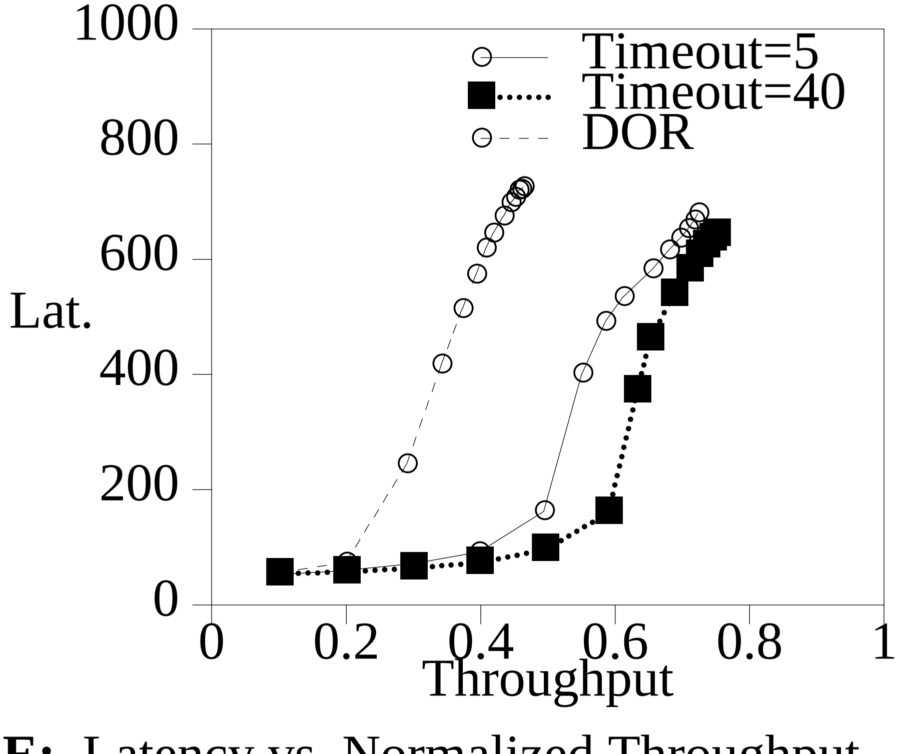


FIGURE: Latency vs. Normalized Throughput. DAMQ buffer capacity = 32 phits. Transpose traffic pattern.