Static Virtual Channel Allocation in Oblivious Routing

Keun Sup Shim, Myong Hyon Cho, Michel Kinsy, Tina Wen, Mieszko Lis
G. Edward Suh (Cornell)
Srinivas Devadas

MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA, USA
Outline

• Motivation

• Static Virtual Channel Allocation Strategy

• Bandwidth-Sensitive Oblivious Routing with Minimal Routes (BSORM)

• Router Architecture

• Evaluation

• Conclusion
Motivation

- Virtual Channels (VCs)
  - Quality-of-Service Guarantees
  - Deadlock Avoidance
  - Head-of-line Blocking Mitigation
  - Hardware complexity ($\leq 16$ VCs)

How are we going to allocate flows to VCs?
Dynamic and Static VC Allocation

- VC allocation can be done either in a *dynamic* way at runtime, or by pre-computed *static* assignment

<table>
<thead>
<tr>
<th>Dynamic Allocation</th>
<th>Static Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime Decision</td>
<td>Pre-determined</td>
</tr>
<tr>
<td>Any VC for any flow</td>
<td>One VC for each flow</td>
</tr>
<tr>
<td>VC utilization ↑</td>
<td>Head-of-line blocking ↓</td>
</tr>
</tbody>
</table>

*Performance Tradeoffs between Dynamic and Static*
Head-of-line Blocking in Street Network

This is why we have such lanes as “straight only” or “left turn only”
Head-of-line Blocking (Dynamic Allocation)

- Four uncorrelated flows with the same demand
- XY routing with four VCs & Link BW = 1 flit/cycle
Head-of-line Blocking (Dynamic Allocation)

- Four uncorrelated flows with the same demand
- XY routing with four VCs & Link BW = 1 flit/cycle

Congested when injection rates are high; Approximately one-third of the link BW for each flow
Head-of-line Blocking (Dynamic Allocation)

- Four uncorrelated flows with the same demand
- XY routing with four VCs & Link BW = 1 flit/cycle

Congested when injection rates are high; Approximately one-third of the link BW for each flow

Flow A is also limited to 1/3 of link BW when flow A is held up by flow B
Separation of Flows by Static Allocation

- Statically assign flows A and B to separate virtual channels

Flow A can proceed regardless of flow B slowing down

The full link bandwidth can be utilized
Mitigating the Head-of-line Blocking Effects

- Through judicious separation of flows, static allocation schemes can enhance throughput

How to determine the separation of flows in oblivious routing?

![Diagram showing the separation of flows](image)

Throughput (flits/cycle)

- Flow A
- Flow B
- Flow C
- Flow D
Static VC Allocation Framework

• A pair of flows is said to be entangled if the flows share at least one VC across all the links used by both flows

Reduce the number of distinct entangled flow pairs

Given a link and a flow $F$ using it

Repeat for each flow at the given link, and move on to the next link

(Prior to channel assignment, no pairs of flows are entangled)

Use for all oblivious routing methods (DOR, ROMM, BSORM)
Static Allocation under ROMM and Valiant

- Source-to-intermediate and intermediate-to-destination sub-routes should not share the same VCs to avoid deadlock.

Static allocation (same as DOR) only within the particular set
Turn Model

- A systematic way of generating deadlock-free routes
- Deadlock-free if routes conform to ONE of the turn models (acyclic channel dependence graph)

We use the combination of these two turn models in order to partition minimal routes for bandwidth-sensitive oblivious routing

* The third turn model, Negative-First, does not induce a flow partition and so is not shown
Minimal Routes

A Set of Minimal Routes

The eight different two-turn minimal routes

Four one-turn routes

* Minimal routes with a single turn or no turns can be ignored as special cases of two-turn routes
Minimal Routes

A Set of Minimal Routes

North-Last

The eight different two-turn minimal routes

Four one-turn routes

* Minimal routes with a single turn or no turns can be ignored as special cases of two-turn routes
Minimal Routes

A Set of Minimal Routes

North-Last

West-First

The eight different two-turn minimal routes

Four one-turn routes

* Minimal routes with a single turn or no turns can be ignored as special cases of two-turn routes
Minimal Routes

A Set of Minimal Routes

North-Last

West-First

Both

The eight different two-turn minimal routes

Four one-turn routes

Not deadlock-free yet...

* Minimal routes with a single turn or no turns can be ignored as special cases of two-turn routes
THEOREM
Given a router with $\geq 2$ VCs, and an arbitrary set of minimal routes over an $n \times n$ mesh, it is possible to statically allocate VCs to each flow to ensure deadlock freedom.

Flows with two-turn and single-turn routes that conform to **West-First**

Flows with two-turn and single-turn routes that conform to **North-Last**

Flows with single-turn or zero-turn routes that conform to **Both**

* The case of 2 VCs
Deadlock Freedom by Static VC Allocation

**THEOREM**

Given a router with \( \geq 2 \) VCs, and an arbitrary set of minimal routes over an \( n \times n \) mesh, it is possible to statically allocate VCs to each flow to ensure deadlock freedom.

- Flows with two-turn and single-turn routes that conform to **West-First**
- Flows with two-turn and single-turn routes that conform to **North-Last**
- Flows with single-turn or zero-turn routes that conform to **Both**

*The case of 2 VCs*
THEOREM
Given a router with $\geq 2$ VCs, and an arbitrary set of minimal routes over an $n \times n$ mesh, it is possible to statically allocate VCs to each flow to ensure deadlock freedom.

- Flows with two-turn and single-turn routes that conform to West-First
- Flows with two-turn and single-turn routes that conform to North-Last
- Flows with single-turn or zero-turn routes that conform to Both

* The case of 2 VCs
THEOREM

Given a router with $\geq 2$ VCs, and an arbitrary set of minimal routes over an $n \times n$ mesh, it is possible to statically allocate VCs to each flow to ensure deadlock freedom.

- Flows with two-turn and single-turn routes that conform to **West-First**
- Flows with two-turn and single-turn routes that conform to **North-Last**
- Flows with single-turn or zero-turn routes that conform to **Both**

*The case of 2 VCs*

Deadlock-free !!
Bandwidth-Sensitive Oblivious Routing with Minimal Routes (BSORM)

Estimates of flow bandwidths

Dijkstra’s weighted shortest-path algorithm

\[ w(e) = \frac{1}{\tilde{c}(e) - d_i} \] \[ \tilde{c}(e) \leq d_i, \quad w(e) = \infty \]

\( \tilde{c}(e) \): Residual capacity of link \( e \), \( d_i \): Demand of flow \( i \)

Replace the path with one of the four one-turn routes if the same minimum weight holds

Greater freedom for the static VC allocation

Weights are updated, and moves on to the next flow

Minimal routes that attempt to minimize maximum channel load

Static VC Allocation to avoid deadlock
Static VC Allocation under BSORM

- Local flexibility in the ratio of VCs across the two sets
  - Reduce wasted VCs

2 flows in the first set
6 flows in the second set
8 VCs

Static allocation within each set using the “entanglement” framework
Typical VC Router Architecture

The ONLY change required

→ Routing module needs “Table-based Routing”
Two ways of Table-Based Routing

- **Source routing**
  - Eliminates the routing step, but results in longer packets
- **Node-table routing**
  - Each module contains a routing table, which is looked up at every hop
Static Virtual Channel Allocation

- Router can obtain both route and VC information in the routing step (RC)
  - Simplifies the VC allocation step
  - P·V to 1 arbitration for each VC (about 20% faster than P·V to V arbitration, which is required by dynamic allocation)

- Assume 256 entries (256 flows) per each routing table
  - 2 bits (output port) + 8 bits (next table index)
  - Additional bits for VC info. (3 bits for 8 VCs)
  - Each routing table only takes a few KB

Accessible in a single cycle
Evaluation

• Standard synthetic benchmarks (3):
  - transpose, bit-complement, and shuffle
  - all flows with the same average bandwidth demands

• Application benchmark (1):
  - a set of flows with the traffic pattern of an H.264 decoder
  - flow bandwidths derived through profiling

• Simulator:
  - a cycle-accurate network simulator
  - 8 X 8 2-D mesh network with 1, 2, 4 or 8 VCs per port
  - Fixed packet length : 8 flits
  - Per-hop latency : 1 cycle
  - Flit buffer size per VC : 16 flits
  - Simulation for 100,000 cycles after 20,000 cycles of warm-up
Throughput for DOR (static and dynamic, 2 VCs)
ROMM and Valiant (static and dynamic, 4 VCs)
BSORM and XY (static and dynamic, 8 VCs)
Conclusion

- Static VC allocation often outperforms dynamic VC allocation for existing oblivious routing schemes
  - Effectively mitigating head-of-line blocking effects

- BSORM provides better performance than existing oblivious routing schemes
  - If head-of-line blocking effects are small, maximum channel load serves as a dominant factor in determining the performance of a given route

- VC allocation step in routers can be slightly simpler with static allocation, and static allocation assures in-order packet delivery
Thank you!