A Method for Applying Double Scheme Dynamic Reconfiguration over InfiniBand

Timothy Mark Pinkston, Bilal Zafar, and Jose Duato
SMART Interconnects Group and GAP
University of Southern California, USA
Technical University of Valencia, Spain

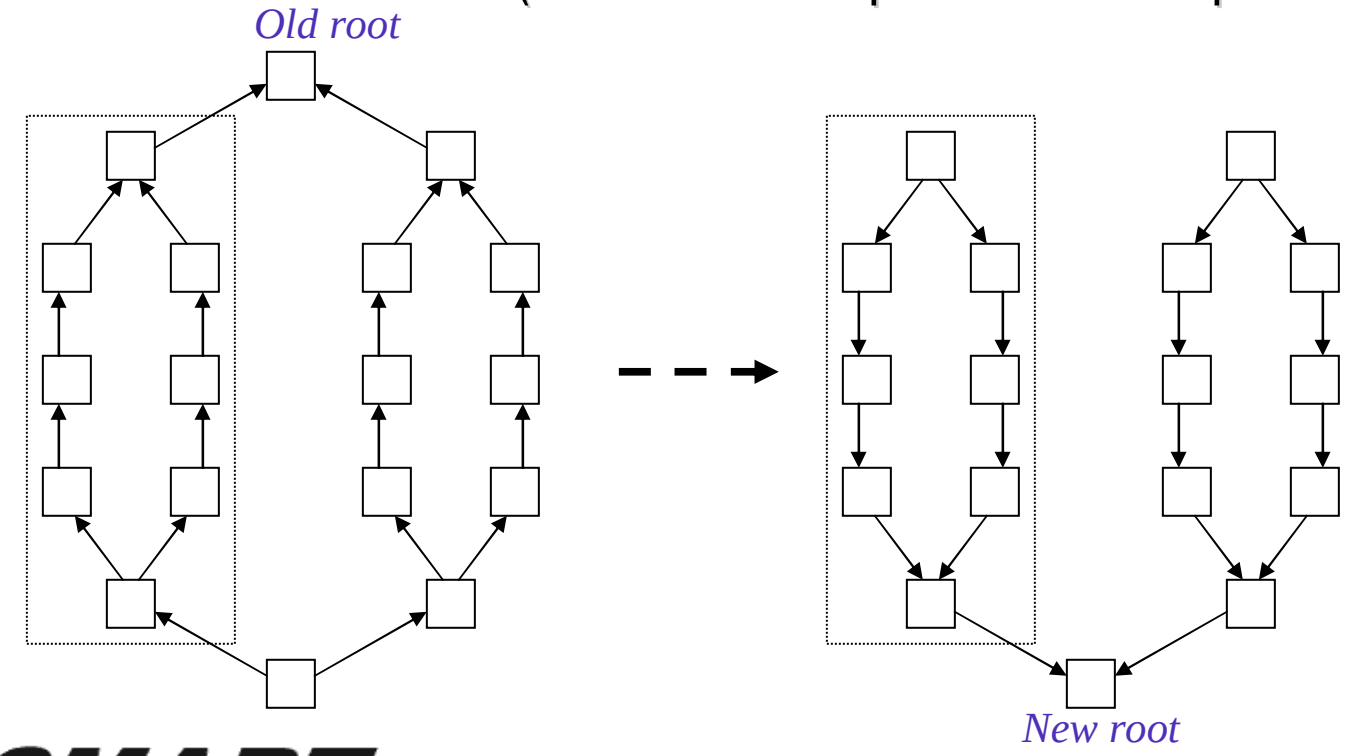
Why Is Reconfigurability Needed, Difficult?

- High-performance interconnection networks need *flexibility* in providing incremental expansion capability:
  - Hop-swapping of components, re-mapping of links/nodes, failure or addition of links/switches, activation or deactivation of hosts/switches, etc.
- Require the ability to reconfigure routing algorithm to reflect changes in network.
  - LANs, NOWs, SANs, STANs, Server-I/O, IPC networks
- What makes dynamic reconfiguration difficult?
  - Deadlocks and unroutable packets!
  - Possible even though the routing function is updated from one deadlock-free algorithm to another.
  - Reason: residual *ghost dependencies*
- Although InfiniBand Architecture (IBA) has various mechanisms useful for configuring the network, no strategy or procedure is specified for ensuring deadlock freedom during dynamic network reconfiguration

Example of Reconfiguration-Induced Deadlock

The network is routed with up*/down* routing algorithm.

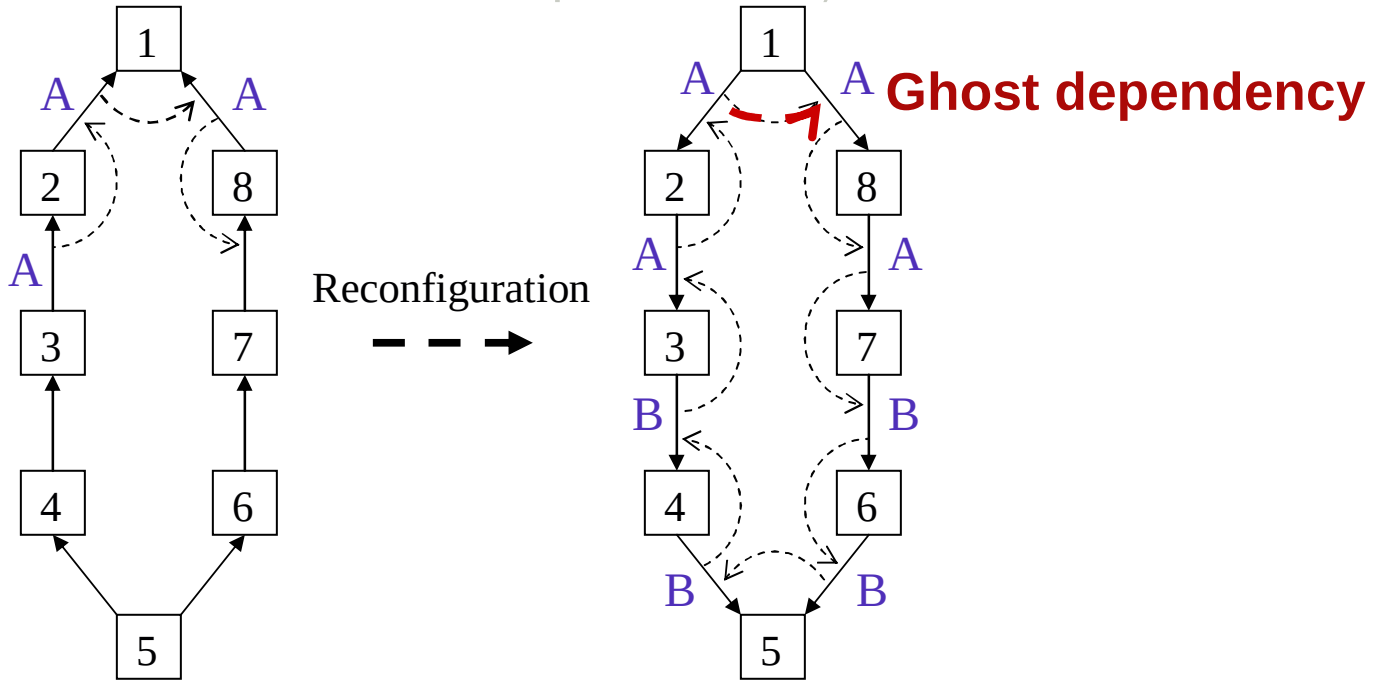
Root node is removed (arrows show up direction in up*/down* tree):



Example of Reconfiguration-Induced Deadlock

Deadlock during reconfiguration of a wormhole network:

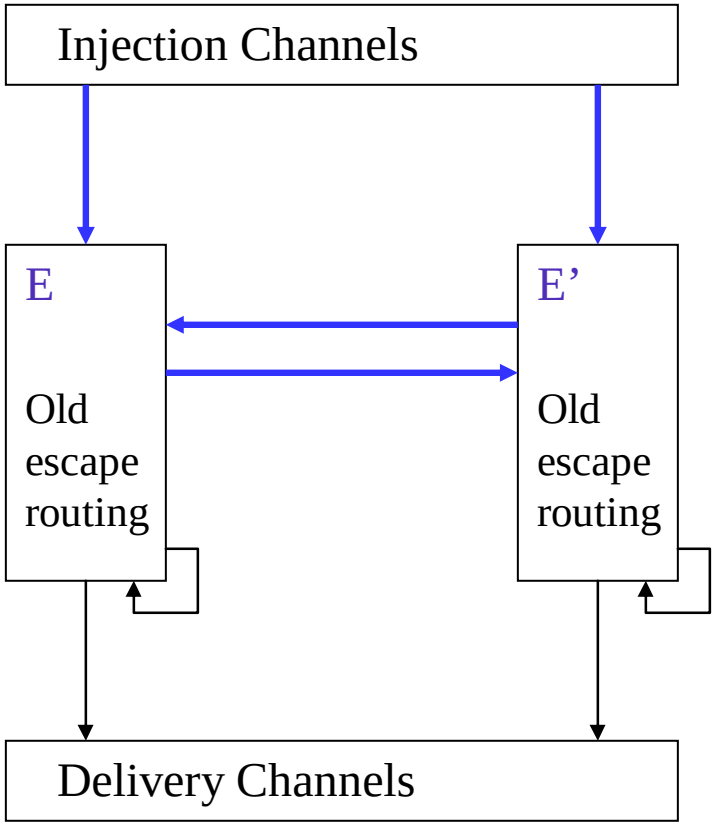
Packet A (from 3 to 6) and Packet B (from 7 to 2) form a deadlock
(dashed arrows show channel dependencies)



The Double Scheme: A Review

- Background: in current networks, each physical channel usually contains several virtual channels: one as an escape channel to solve deadlocks and the others as adaptive channels.
- Double Scheme originally proposed in ICPP'00
- Basic idea:
  - ➔ Network is deadlock-free if the escape virtual network (which is composed of escape virtual channels) is deadlock-free.
  - ➔ During reconfiguration, eliminate harmful ghost dependencies and non-routable packets only on one escape virtual network; all other virtual networks are reconfigured to maintain re-establish connectivity and/or increase performance.

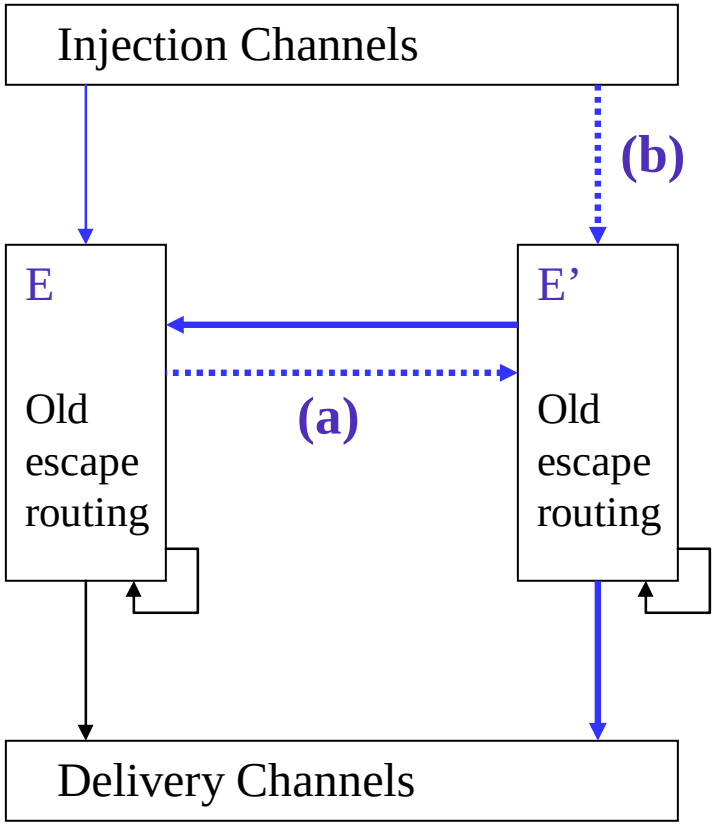
Enhanced Basic Double Scheme



1. Before Reconfiguration.

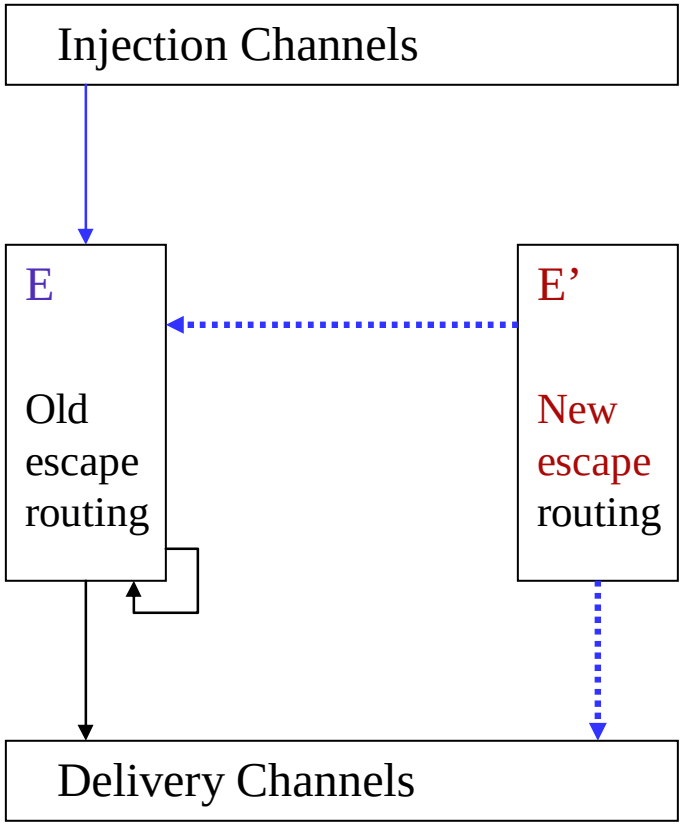
Basic idea:
Both E and E' are used for escape routing when there is no reconfiguration.

Enhanced Basic Double Scheme



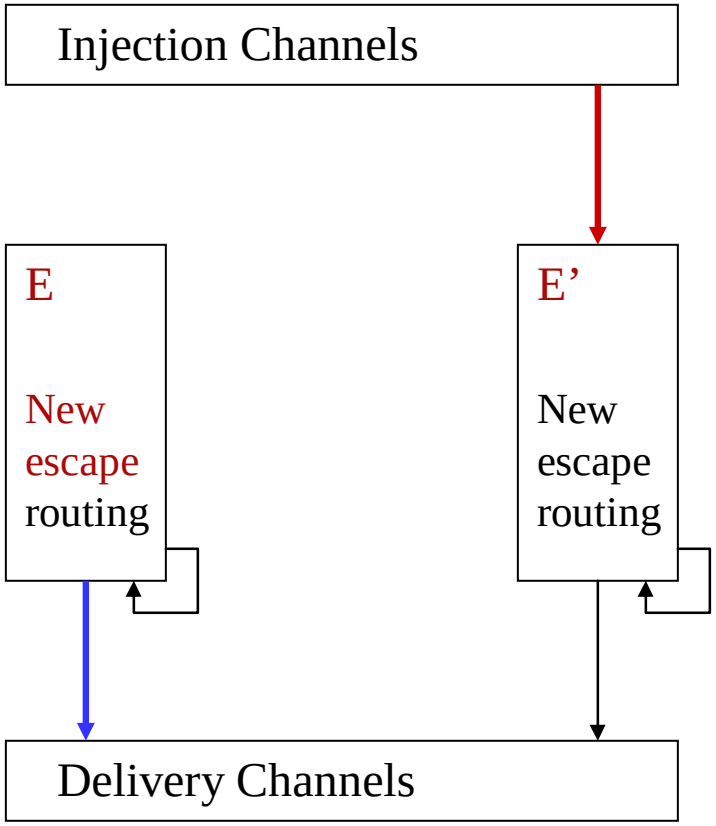
1. Before Reconfiguration.
2. Reconfiguration is initiated. Stop packet injection into subset of escape channels (a) from network channels and (b) from source nodes. This subset will be used to implement the new escape routing function.

Enhanced Basic Double Scheme



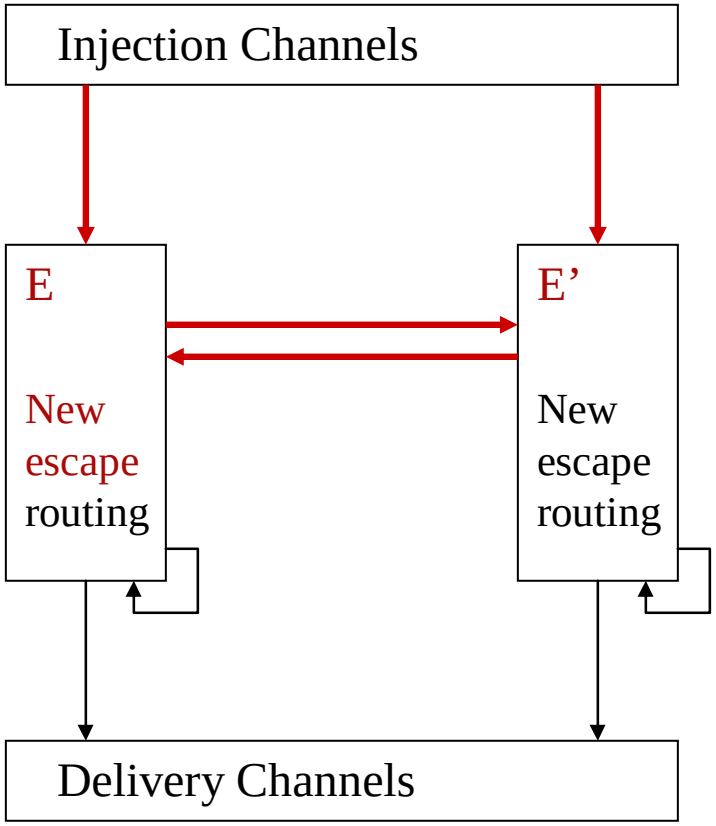
1. Before Reconfiguration.
2. Reconfiguration is initiated. Stop packet injection into E'channels.
3. All packets are drained from E'. Update E' to the new escape routing function.

Enhanced Basic Double Scheme



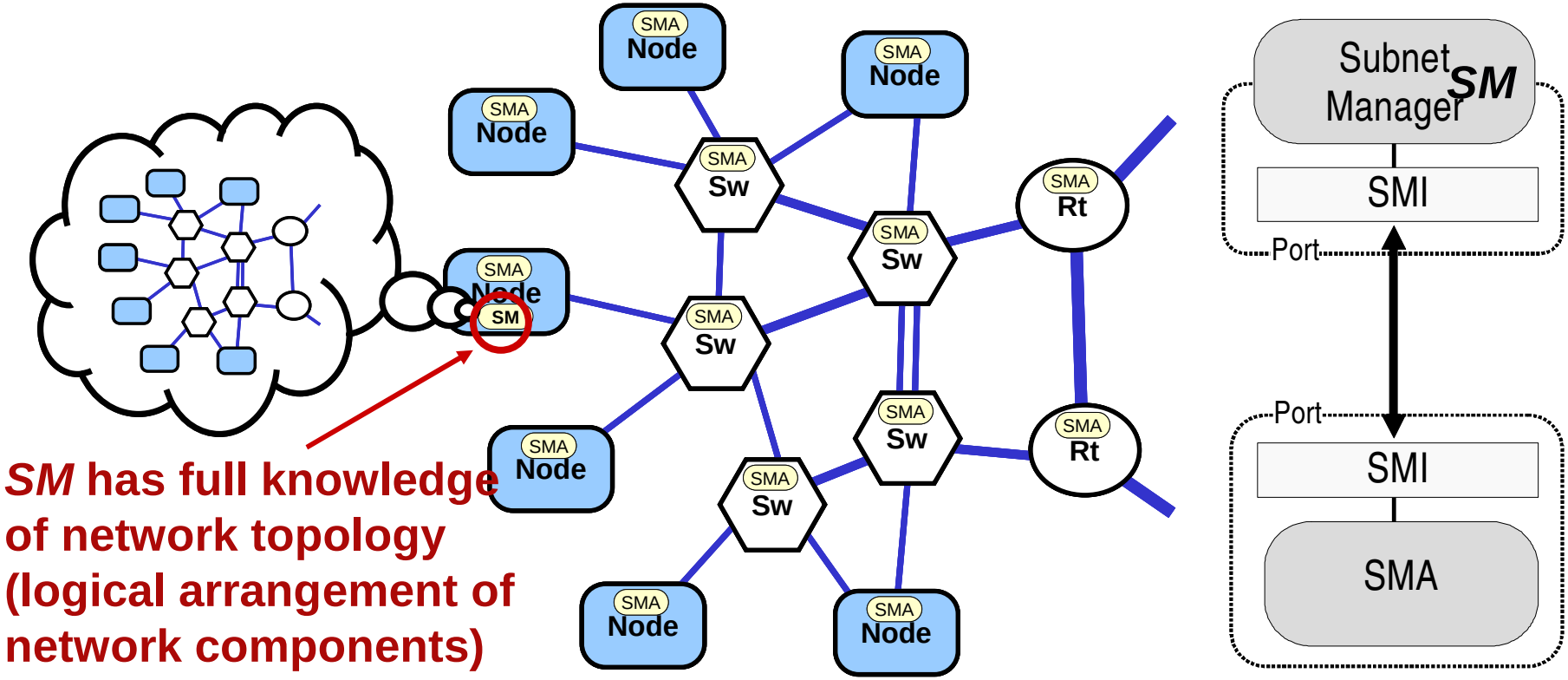
1. Before Reconfiguration.
2. Reconfiguration is initiated. Stop packet injection into E'channels.
3. All packets are drained from E'. Update E' to the new escape routing function.
4. All packets are drained from E. Update E to the new escape routing function and allow injection into E'.

Enhanced Basic Double Scheme



1. Before Reconfiguration.
2. Reconfiguration is initiated. Stop packet injection into E'channels.
3. All packets are drained from E'. Update E' to the new escape routing function.
4. All packets are drained from E. Injection allowed into E'.
5. Allow injection into E and between both sets of escape resources.

InfiniBand Management Model



***SM* has full knowledge of network topology (logical arrangement of network components)**

InfiniBand Management Model

- Subnet Manager (SM)
  - At least one per subnet (master SM)
- Subnet Management Agents (SMA)
  - Present in switches, routers, and CAs
  - Processes *control packets* sent by the SM and configures local components accordingly
- Subnet Management Interface (SMI)
  - Interface between the hardware and management entities
  - Connected to the network through logical port 0 in the switches and physical port in CAs
- *Control Packets*
  - Subnet Management Packets (SMPs)
  - Methods: Get, Set, GetResp, Trap, and TrapRepress
  - Attributes: NodeInfo, SwitchInfo, PortInfo
  - Directed Routing and LID routing used to transfer them
  - Use exclusively VL15. Priority over data Virtual Lanes 0-14

Applying Double Scheme over InfiniBand

- Reconfiguration Procedure
  - Step 1: Reconfiguration is initiated
  - Step 2: Modify SL-to-VL mapping table and Forwarding Table
  - Step 3: Modify PathRecord and GUID-to-LID mapping
  - Step 4: Detect old VL drainage
  - Step 5: Final modification of the SL-to-VL mapping table to restore all VLs to data packets

Step 1: Initiate Reconfiguration

- SM periodically polls the subnet to gather information about topology changes
  - ➔ SM sends SubnGet(SwitchInfo) SMP to detect changes for each subnet switch
  - ➔ Sweeping rate is dynamically computed using the “subnet sweeping time” (global attribute) and an additional factor value
- On recognizing a change in the topology, and if global flag “reconfig” is set (during first bootup), the Discoverer sub-process spawns the Rebuilder sub-process to start Double Scheme reconfiguration

Step 2: SL-to-VL Mapping Update

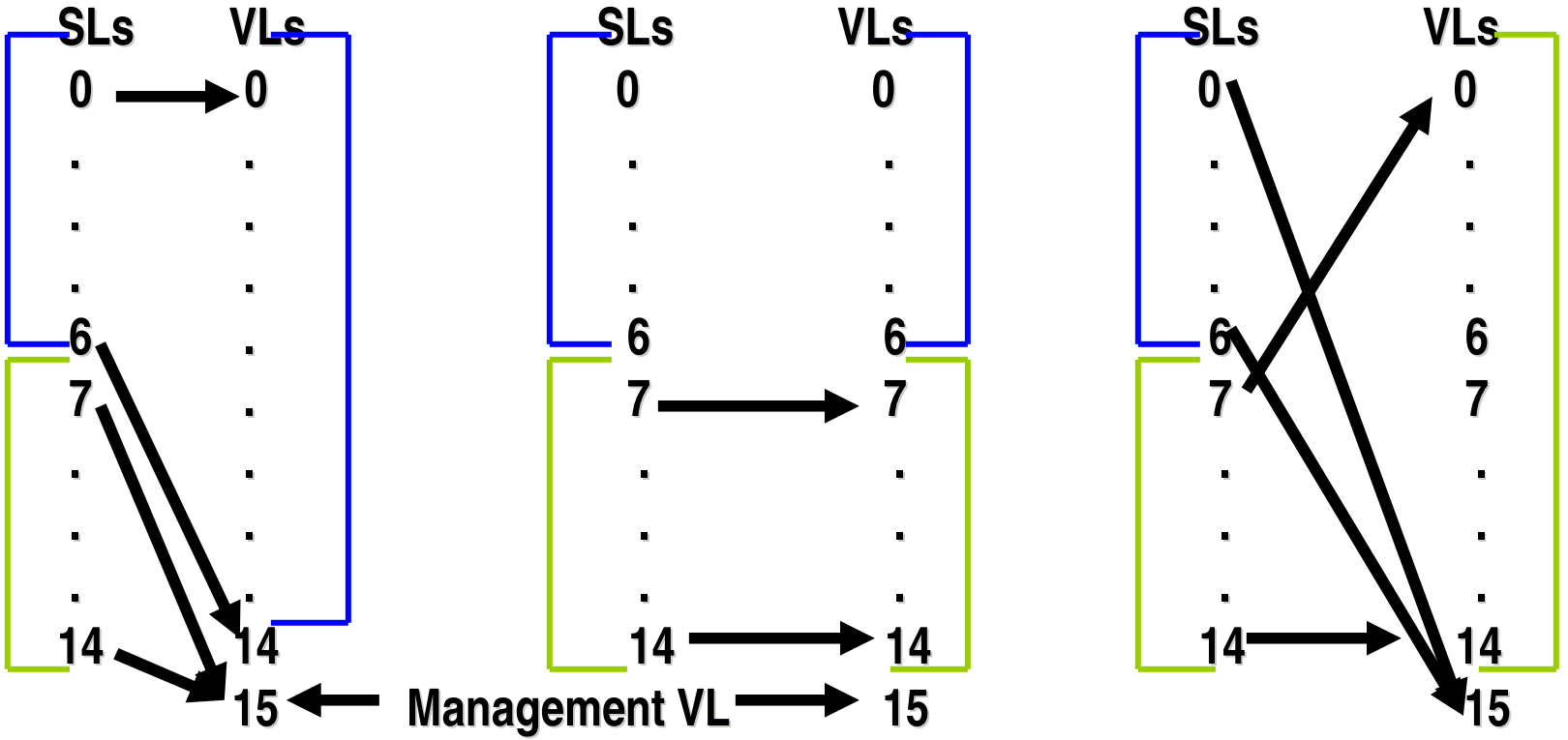
- *During normal operation*, only half of the Service Levels (SLs) are available to packets, but they are mapped to all the available data virtual lanes (VLs).
- *During reconfiguration*, when both routing functions exist, packets routed by old routing function use one set of VLs and those routed by new routing function use second set of VLs. *Dictated by SL-to-VL mapping:*
  - ➔ SM uses subnet management method SubnGet() to read the SL-to-VL mapping table associated with each CA or switch port
  - ➔ It then uses SubnSet() to configure the SL-to-VL mapping table
  - ➔ One half of SLs map to one set of VLs, and the other half maps to the other set of VLs. Each half used by one routing function.

Step 2: SL-to-VL Mapping Update

Old Routing Function: Blue

New Routing Function: Green

Normal Operation *During Reconfiguration* *After Reonfiguration (Step 5)*



Step 2: Computing and Updating Forwarding Tables

- Restricted Up*/Down* Routing [Lopez, Flich, Duato ICPP'01]
  - ➔Remove routing alternatives that can lead to deadlock
  - ➔Criterion: for a given destination if any output port in the down direction, then ignore all the routing options that imply the use of a link in the up direction. Finally, choose only one output port (which gives shortest route) for each destination.
- Can put ports in the ARMED state during table update:
  - ➔Incoming packets are buffered and processed when port returns to ACTIVE state, instead of being dropped
  - ➔Also, ARMED state allows updates to appear atomic

Step 3: Modify PathRecord

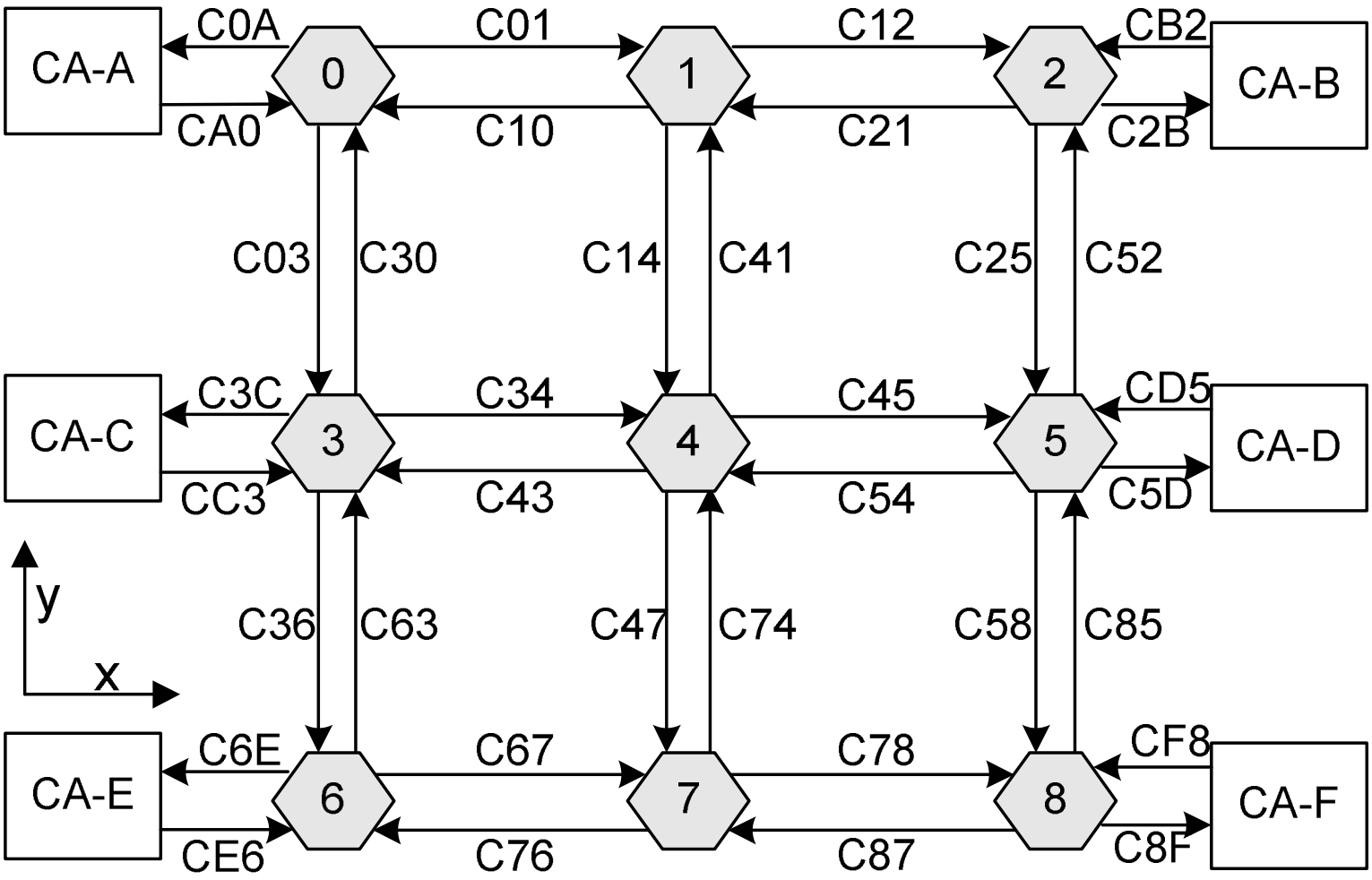
- SM modifies the PathRecord information for each port such that it now supplies the set of SLs that were previously unused.
  - ➔ New packets injected into the network use valid VLs (emptied set)
- Resources and network paths used by each routing function are spatially separated by assigning two sets of LID addresses for each GUID, using the LID Mask Control (LMC) feature:
  - ➔ One set of LIDs correspond to old routing function; other to new
  - ➔ GUID-to-LID mapping maintained by SM
  - ➔ IBA uses *source-based and LID destination routing*
- Ports switched to ARMED state so that modification of PathRecord and supply of LID addresses to sources appears “atomic”
- Possible implementation: maintain two tables corresponding to each set of LIDs for each port and select between them on the basis of a routing function flag maintained locally in the SM.

Step 4: VL Drainage

- VLs for new routing function are drained in a single hop
  - ➔ SL-to-VL mapping forces packets out of VLs that will be used by the new routing function at next switch.
- However, the VLs for the old routing function must be verified across the entire subnet since not completed synchronously:
  - ➔ Step 1: SM computes the channel dependency graph (CDG) for the routing function
  - ➔ Step 2: SM sends control packets to the switches connected to all the source channels (i.e., highest order nodes in the CDG)
  - ➔ Step 3: Every switch forwards control packets along all the directions indicated by arcs in the CDG when no more packets remain in the VLs that are to be drained at a switch
  - ➔ Step 4: The control packet is forwarded back to the SM by the leaf channels (i.e., lowest order nodes in the CDG)
  - ➔ Step 5: When the SM has received all the control packets it sent out, VL drainage for the entire subnet is complete

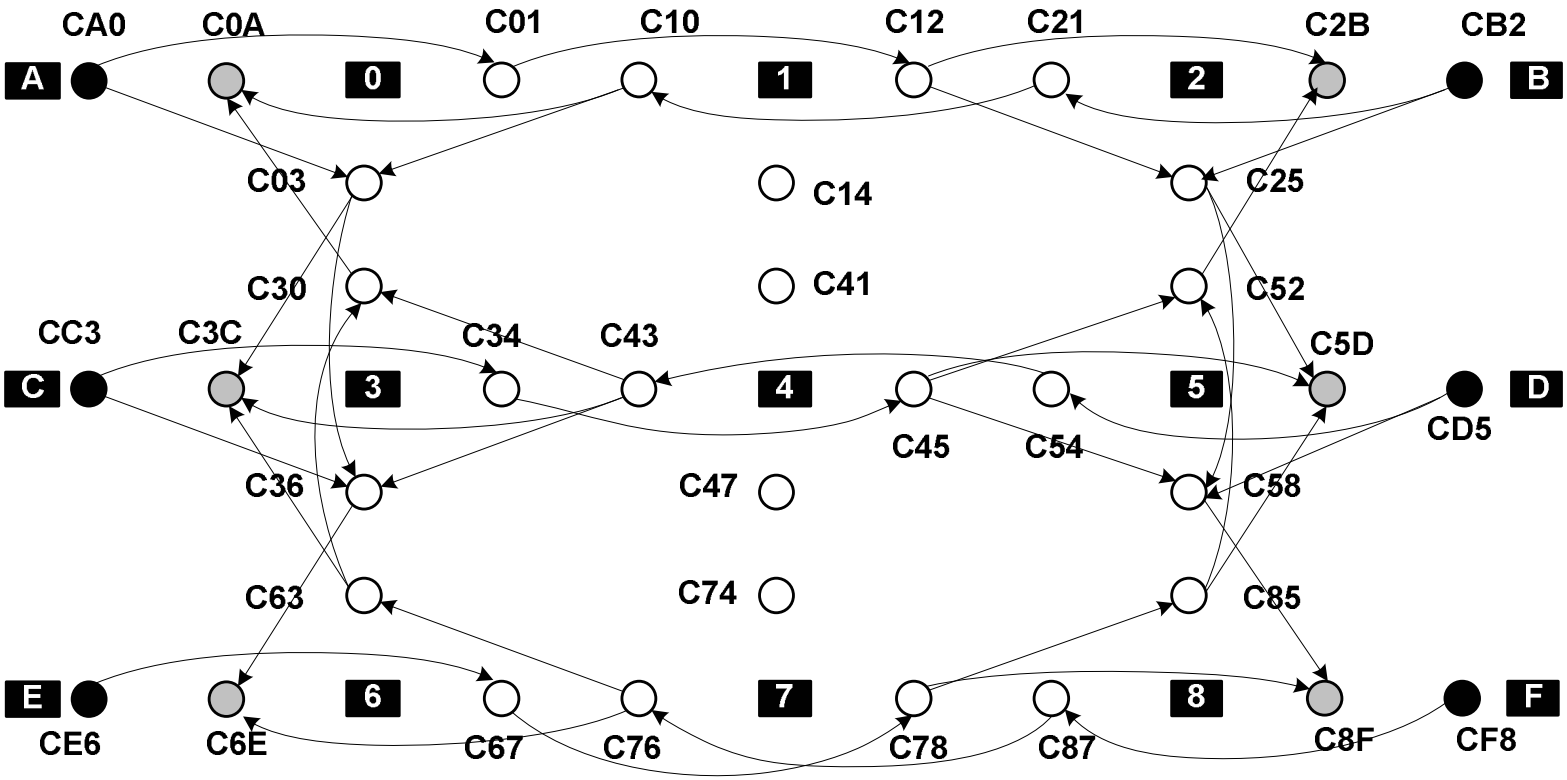
Step 4: VL Drainage Example

Network Graph



Step 4: VL Drainage Example

Channel Dependency Graph

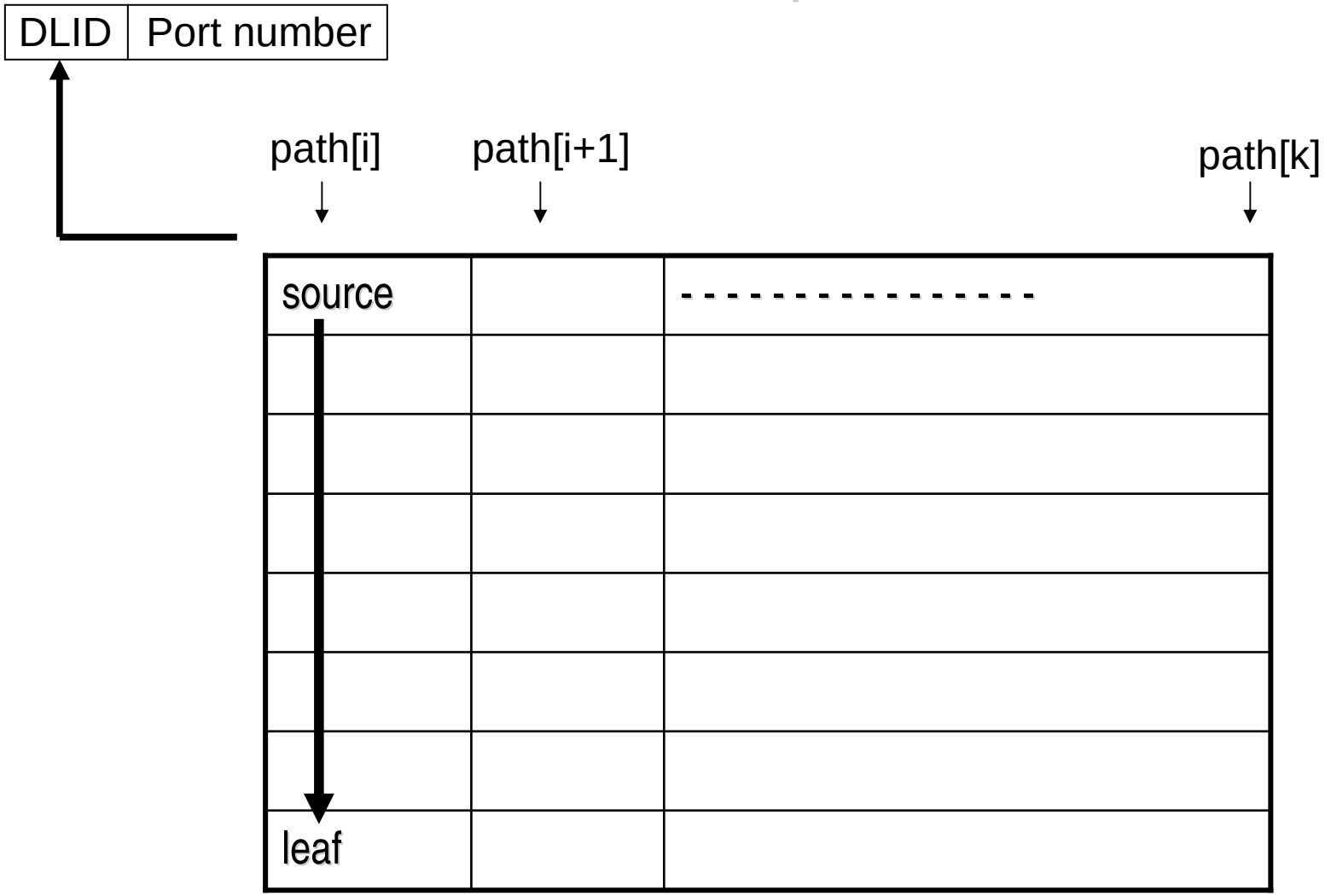


● Source channel ● Leaf channel ○ Non-leaf/non-source channel

Step 4: VL Drainage cont'd…

- Implementation Alternatives:
  - Send a vendor specific SubnGet() SMP to get the VL status
  - The VL status is obtained by making a comparison of the head of the queue and the tail of the queue. If they are same, then the queue is empty and SubnResp() SMP is sent back.
  - In case the queue is not empty then a response is not sent back and the SM timeouts on the SMP and re-polls after a back-off time.

SM Data Structure to keep track of drained VLs

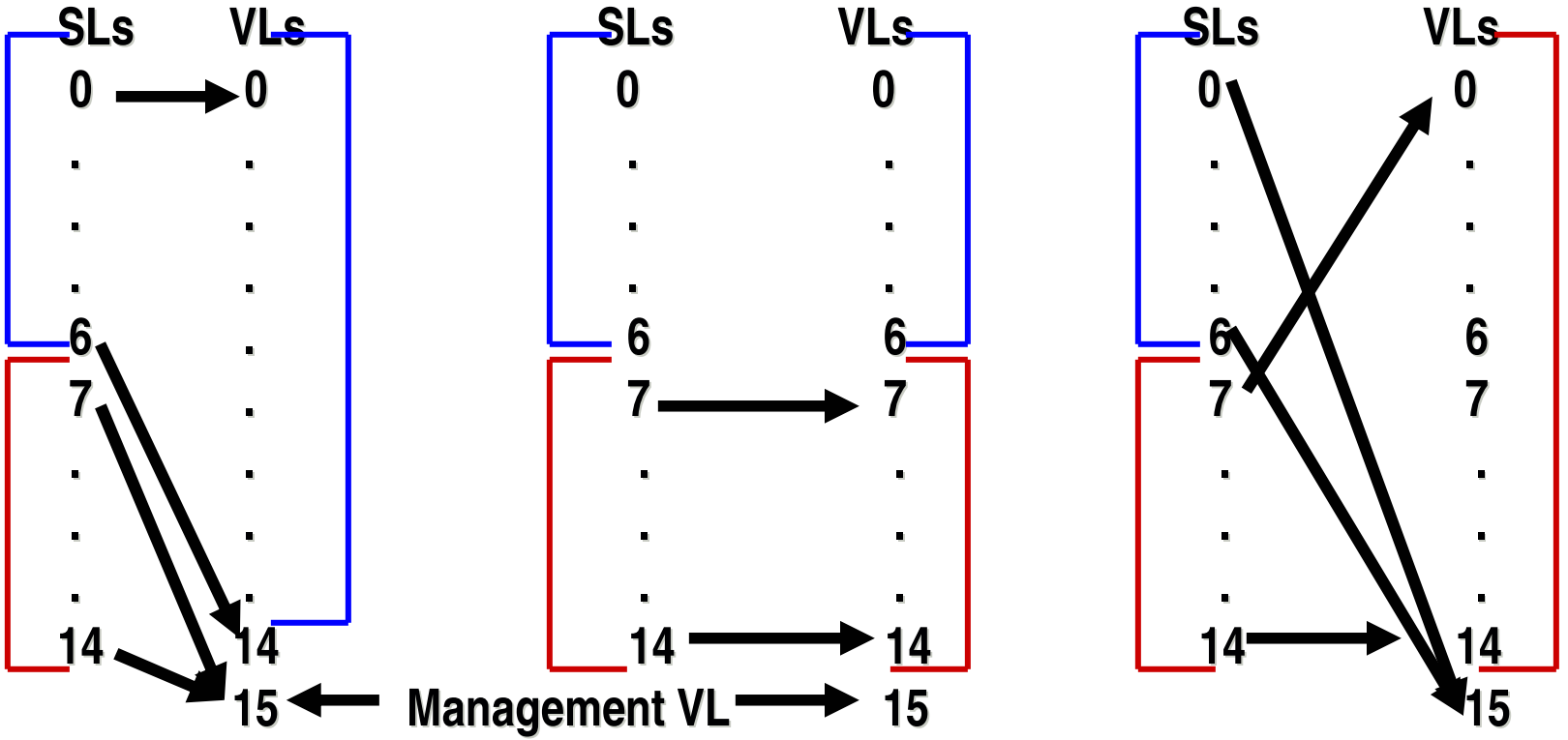


Step 5: SL-to-VL Mapping Update

Old Routing Function: Blue

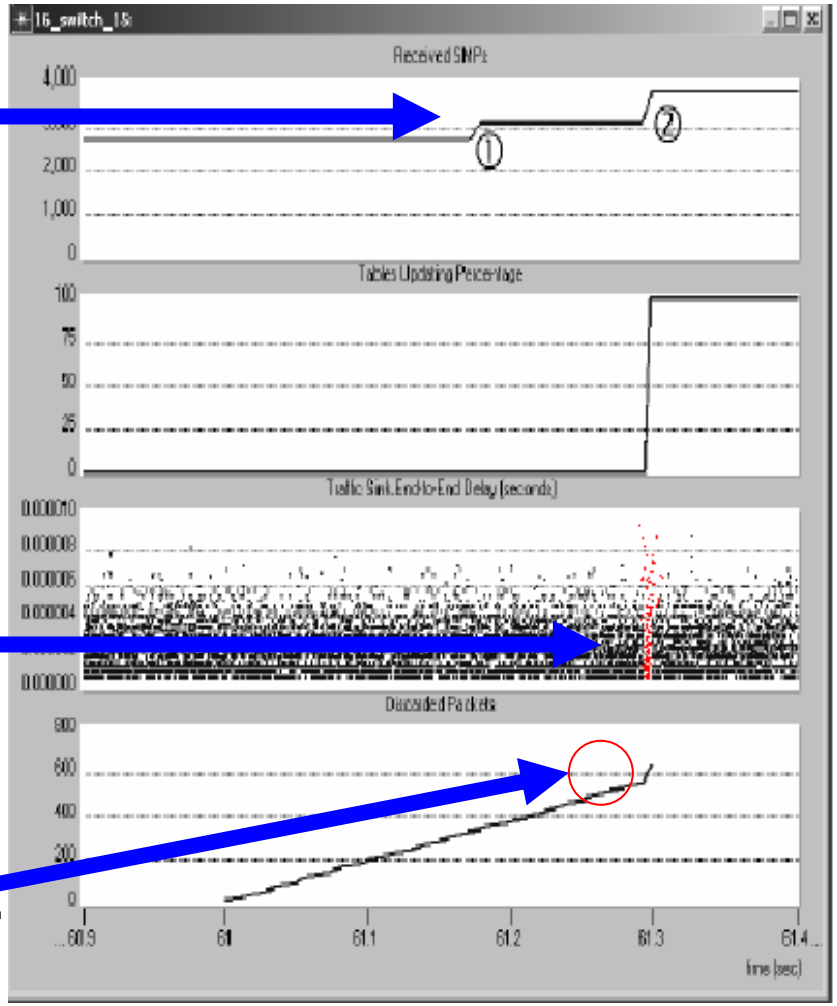
New Routing Function: Red

Normal Operation *During Reconfiguration* *After Reonfiguration (Step 5)*



Results & Analysis

1. **Topology Discovery Process**
2. **Path Distribution**



Since application traffic is stopped, the packet latency increases.
In static reconfiguration, a gap occurs.

Also, discarded packets rises sharply because if ports are in INITIALIZE state.
With Double Scheme on IBA, there is no loss of packets.

Results and Analysis cont'd

For 16 switch topology (14 nodes) assuming no transmission losses

- Subnet Management Packet (SMP) count:
  - SL-to-VL mapping update = 296 * 2 = 592 SMPs
  - Computing FTs = 200 SMPs (same as static reconfig)
  - Distributing FTs = 360 SMPs (same as static reconfig)
  - VL Drainage = 60 SMPs (minimum)
- Time taken to reconfigure network:
  - SL-to-VL mapping update = 0.011 sec
  - Computing FTs = 0.1 sec (same as static reconfig)
  - Distributing FTs = 0.0067 sec (same as static reconfig)
  - VL Drainage = time for 7 network hops

Related Work

- Partial Progressive Reconfiguration (PPR) is another reconfiguration scheme possibly applicable to IBA
  - ➔ After each partial update of the forwarding table it requires each switch to synchronize with some of its neighbors
- Evaluation of InfiniBand Subnet Management Mechanism is ongoing [Bermudez, et al., ICPP'03].
- Detailed simulation models of IBA are under development [Bermudez, et al., SAN'03]
- Efficient subnetwork discovery methods are currently being investigated [Bermudez, et al., unpublished]

Conclusions and Future Work

- A systematic method for applying Double Scheme dynamic reconfiguration to InfiniBand Architecture has been found.
- This technique has potential to improve IBA performance
- Latency of the packets is increased largely due to the time taken for topology discovery and for computing and distributing new forwarding tables. Significantly more work should be done to address this potential bottleneck:
  - IBA specs allow distributed implementation of the SM. With this, it may be possible to parallelize the discovery process
  - IBA allows the use of traps, which also may reduce discovery time
  - It may be possible to confine forwarding table updates to a small "skyline" portion of the network to reduce time for computation and update.