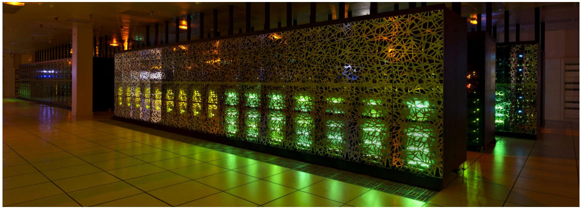
Transitively Deadlock-Free Routing Algorithms

Jean-Noël Quintin: *jean-noel.quintin@atos.net*
Pierre Vignéras: *pierre.vigneras@atos.net*

2016-03-12

Bull
atos technologies

HPC platforms



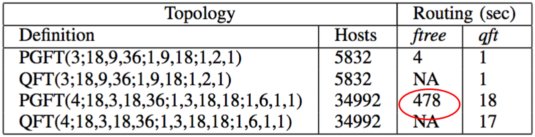
Curie 2011
$>$5 000 nodes
Tera1000 2017
$>$8 000 nodes

- Examples from the Top500 list:
  - K Computer 2011: 82 944 nodes.
  - Sequoia 2013: 98 304 nodes.

OpenSM Solution

- Behavior:
  - discover the topology,
  - select the routing algorithm,
  - compute new routing table,
  - distribute the routing tables.
- Routing computation time above one minute [1]:

TABLE I. ROUTING ALGORITHMS RUNTIME



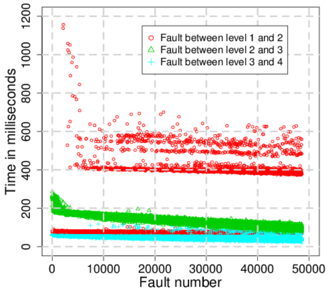
Topology		Routing (sec)	
Definition	Hosts	*ftree*	*qft*
PGFT(3;18,9,36;1,9,18;1,2,1)	5832	4	1
QFT(3;18,9,36;1,9,18;1,2,1)	5832	NA	1
PGFT(4;18,3,18,36;1,3,18,18;1,6,1,1)	34992	478	18
QFT(4;18,3,18,36;1,3,18,18;1,6,1,1)	34992	NA	17

[1] Zahavi et. al. ”Quasi Fat Trees for HPC Clouds and Their Fault-Resilient Closed-Form Routing.” HOTI 2014

BXI Routing Solution

- Behavior:
  - compute and distribute routing table patches.

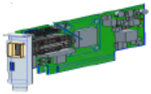
Computes patches under 2 seconds for 64 800 nodes[2]:



[2] Quintin, Vignéras; Fault-Tolerant Routing for Exascale Supercomputer: The BXI Routing Architecture. HiPINEB'15

BXI Hardware Features

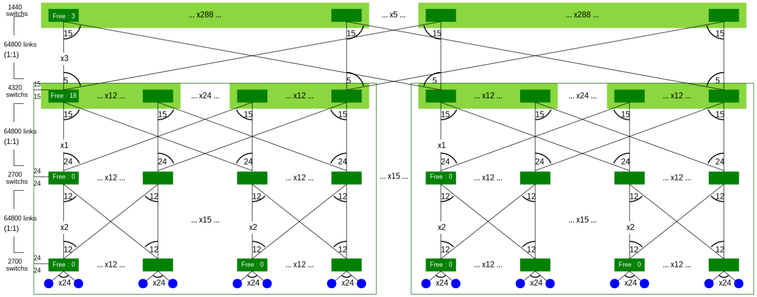
- Scales up to 64k destination
- Network Interface Controler:
  - hardware implementation of Portals4,
  - offload and os bypass (MPI, SHMEM and PGAS),
  - 100Gb/s BXI port to the switch.
- Switch:
  - low latency, non-blocking 48 ports crossbar,
  - out-of-band management.
- Wormhole distributed table-based adaptive routing:
  - 16 virtual channels,
  - **one routing table per port**,
  - **up to 48 adaptive routes**.



Bull
atos technologies

Topology and Routing table Example

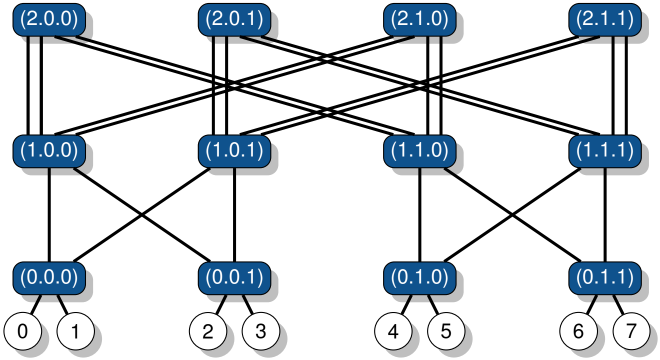
- Topology: 4-level rearrangeable non-blocking fat-tree
  - 64 800 nodes, 11 160 switches, 194 400 inter-switches links
  - **≈50 GB of routing tables**
  - In production, faults may happen on a daily basis (link failures, human mistakes, ...)



Outline

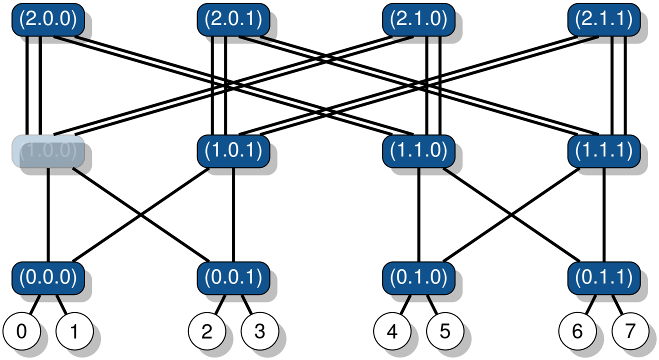
Handling Faults

- Recomputing all routing tables each time is not an option:
  - topology structure is considered immutable



Handling Faults

- Recomputing all routing tables each time is not an option:
  - topology structure is considered immutable
  - a fault is only a missing equipment.



BXI Routing Architecture

- Defines two distinct modes of operation:
  - offline mode:
    - computes routing tables from scratch,
    - archives routing tables on storage,
    - analyses quality of routing tables,
  - online mode:
    - computes routing table **patches**.
    - uploads routing tables on switches,
    - archives patches on disk,
    - analyses quality of online routing tables.

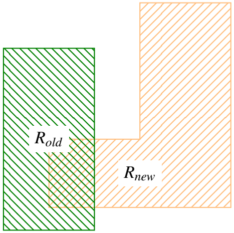
BXI Routing Architecture

- Defines two distinct modes of operation:
  - offline mode:
    - computes routing tables from scratch,
    - archives routing tables on storage,
    - analyses quality of routing tables,
  - online mode:
    - computes routing table **patches**.
    - uploads routing tables on switches,
    - archives patches on disk,
    - analyses quality of online routing tables.
- How to ensure routing tables updates are deadlock-free?

Outline

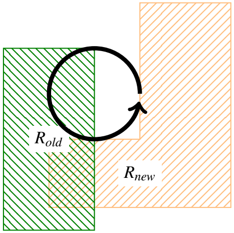
Transitions Between Two Routing Functions

- Transition from R_{old} and R_{new}:
  - not atomic, even on a switch,
  - each switch updates routing tables separately,



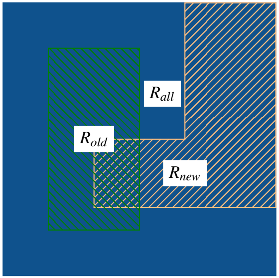
Transitions Between Two Routing Functions

- Transition from R_{old} and R_{new}:
  - not atomic, even on a switch,
  - each switch updates routing tables separately,
  - ghost dependencies remain.



Deadlock-free Transitions

- Transition between two routing functions is deadlock-free if both are included within a deadlock-free routing function.
- Each routing table entry can be applied **in any order**:
  - new routes are included within enclosing routing function,
  - removing routes is safe.



Transitively Deadlock-Free Property

- A Transitively Deadlock-Free routing algorithm:
  - computes patches to switch between routing functions,
  - selects only routes under an enclosing routing function,
  - the enclosing function must be deadlock-free.

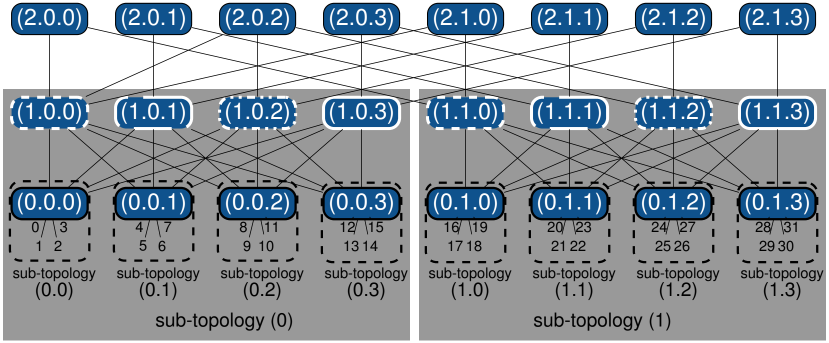
Transitively Deadlock-Free Property

- A Transitively Deadlock-Free routing algorithm:
  - computes patches to switch between routing functions,
  - selects only routes under an enclosing routing function,
  - the enclosing function must be deadlock-free.
- Since the initial routing function is deadlock-free and is within the same deadlock-free routing function.

Outline

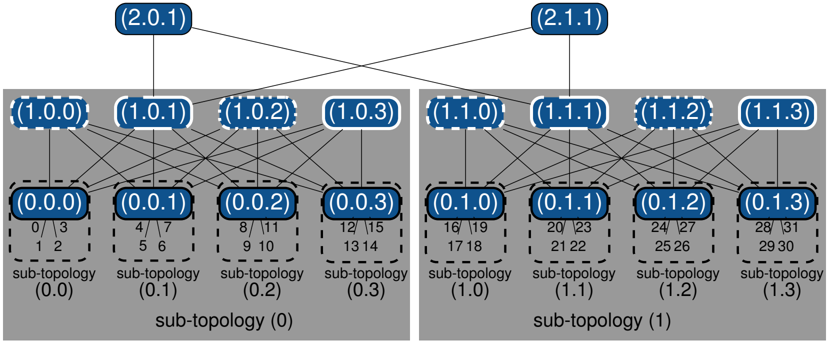
Notation on Fat-Tree

- Twins are all switches at same relative location in other sub-topology.



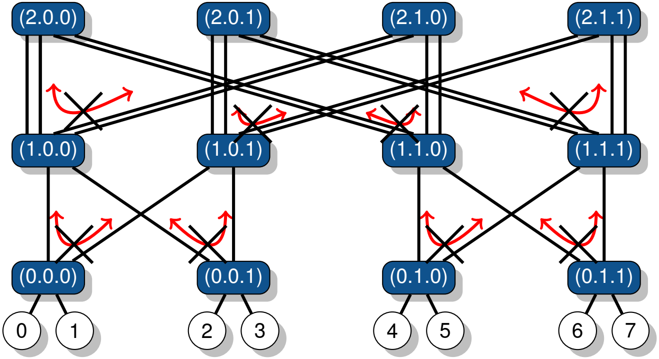
Notation on Fat-Tree

- Twins are all switches at same relative location in other sub-topology.

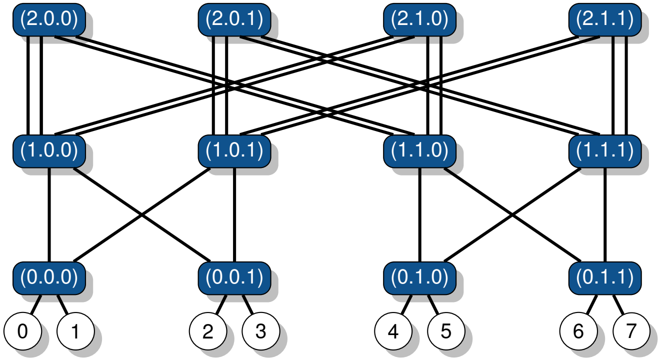


Deadlock-Free Routing Algorithm for Fat-Tree

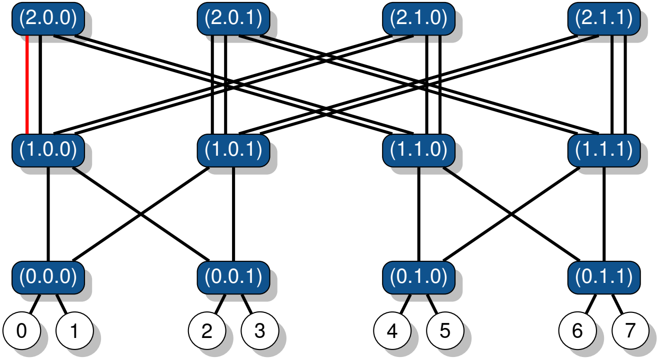
- Set Restrictions between up-ports.



Online Routing Algorithm for Fat-Tree

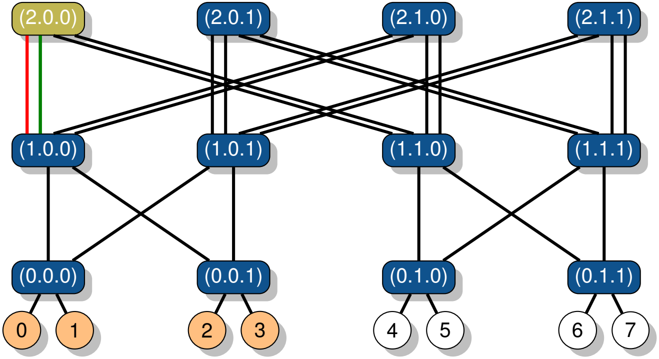


Online Routing Algorithm for Fat-Tree



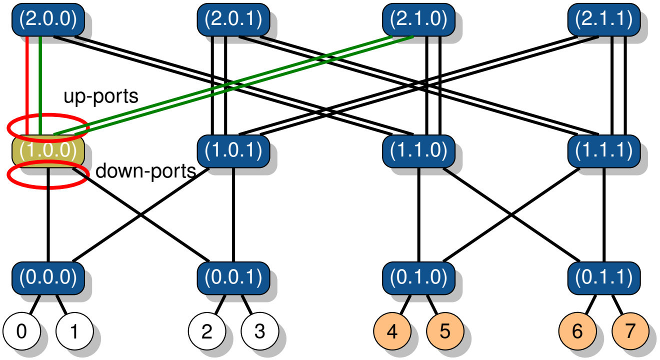
Online Routing Algorithm for Fat-Tree

- On switch (2.0.0):
  - routing tables to reach [0-3] are patched.

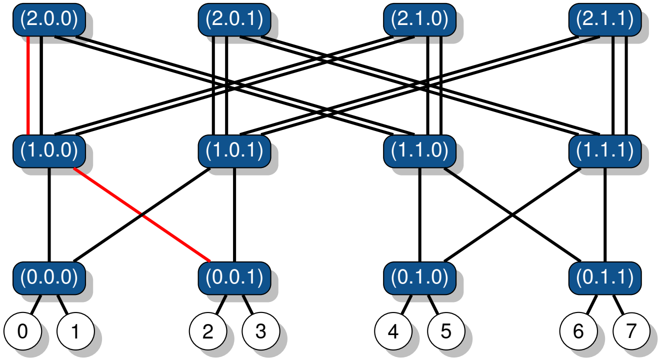


Online Routing Algorithm for Fat-Tree

- On switch (1.0.0):
  - down-port routing table entries for nodes [4-7] are patched,
  - up-port routing tables are not patched.

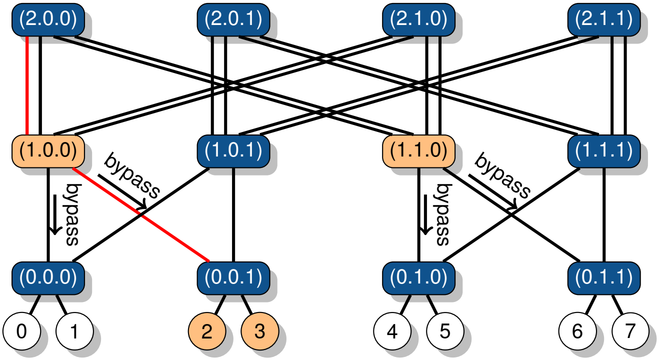


Online Routing Algorithm for Fat-Tree

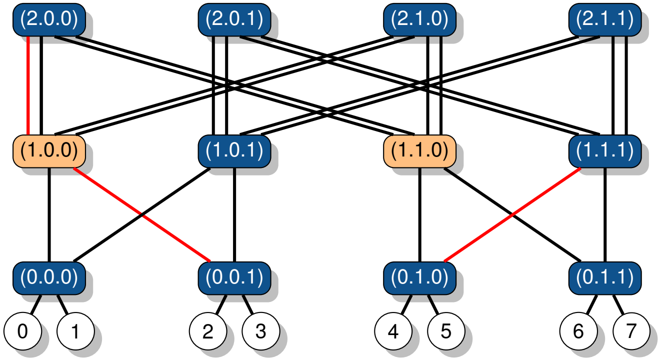


Online Routing Algorithm for Fat-Tree

- (1.0.0) and its twin(s) (1.1.0), are bypassed to reach [2-3].
- On (0.0.0) (0.1.0) and (0.1.1), down-port routing tables are patched to reach [2-3].

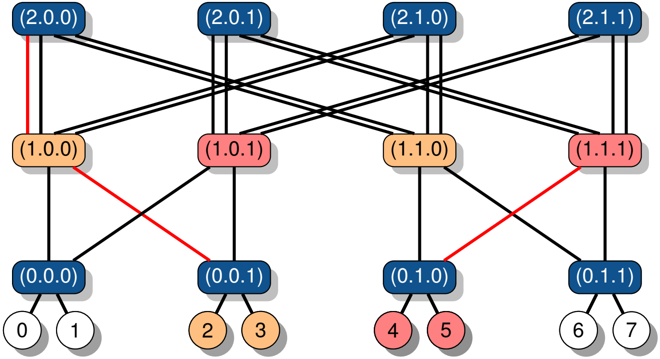


Online Routing Algorithm for Fat-Tree



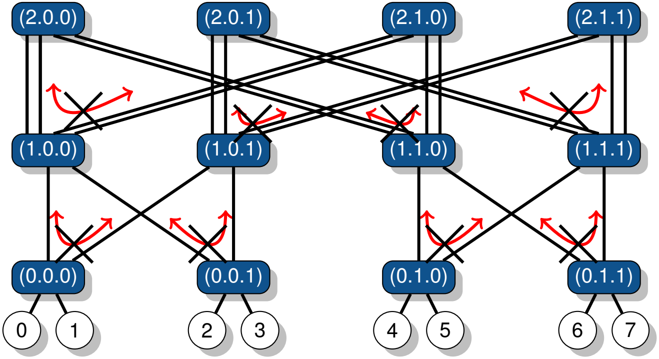
Online Routing Algorithm for Fat-Tree

- (1.1.1) and its twin(s) (1.0.1), are bypassed to reach [4-5]:
  - nodes [4,5] and [2-3] cannot exchange messages,
  - for the routing function, the topology is no more connected.



Transitively Deadlock-Free Algorithm for Fat-Tree

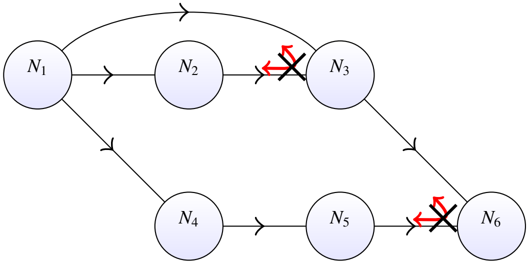
- Enclosing routing algorithm R_{all} provides:
  - down-port(s) for each destination within the sub-topology,
  - all up-ports for each destination outside the sub-topology.



Outline

Transitively Deadlock-Free Agnostic Routing Algorithm

- Enclosing Routing function:
  - computes all routes following the up-down rule:
    - Messages cannot go upward after going downward,
  - returns for any port a set with all ports leading to the destination.



Methodology for New Online Routing Algorithms

- Creates an enclosing routing algorithm.
- Computes restrictions to remove deadlocks.
- All unrestricted routes mustn't introduce deadlock.
- Computes offline routing tables.
- Provides restrictions to online agnostic routing algorithm.

Conclusion

- New architecture based on two modes: Offline/Online.
- Two new online algorithms:
  - handle link faults, up to 25 percent of faulty links,
  - handle link recovery,
  - scalable,
  - formally described,
  - transitively deadlock-free,
- Methodology to create new transitively deadlock-free algorithms.
- Future steps:
  - study routing quality,
  - adapt on events the routing tables:
    - such as computed statistics. . .

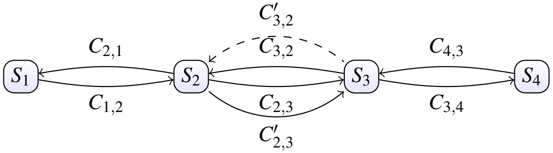
Questions?

10^18
Bull
exascale
program

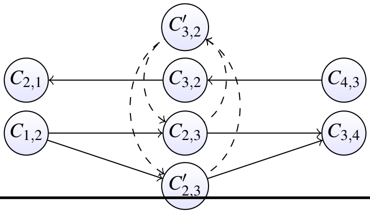
Bull
atos technologies

Dynamic Routing and Deadlock/Livelock

- Network Topology
  - Minimal routing exception: dashed linked is only usable to reach S_4.

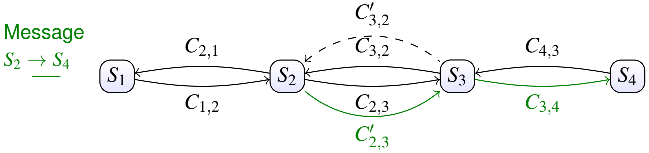


- Dependency channel graph.

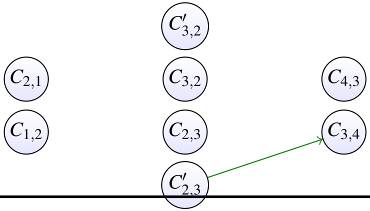


Dynamic Routing and Deadlock/Livelock

- Network Topology
  - Minimal routing exception: dashed linked is only usable to reach S_4.

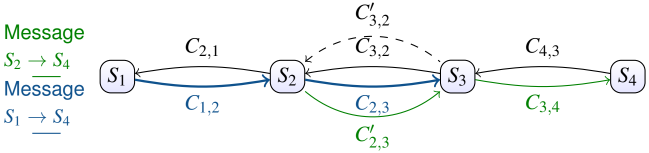


- Waiting buffer graph.

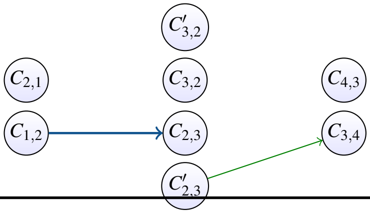


Dynamic Routing and Deadlock/Livelock

- Network Topology
  - Minimal routing exception: dashed linked is only usable to reach S_4.

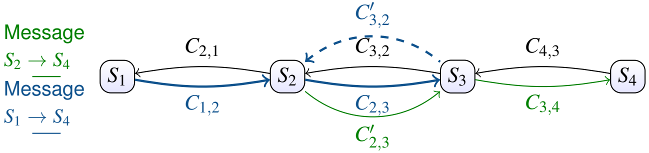


- Waiting buffer graph is dynamic.

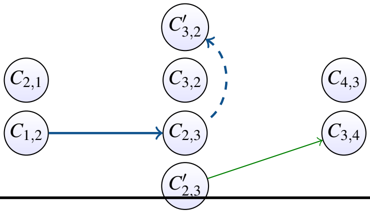


Dynamic Routing and Deadlock/Livelock

- Network Topology
  - Minimal routing exception: dashed linked is only usable to reach S_4.

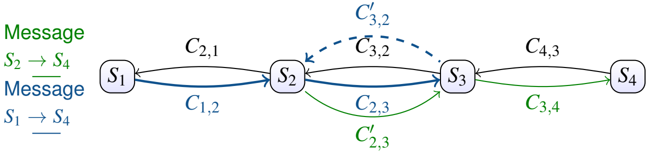


- Waiting buffer graph is dynamic.

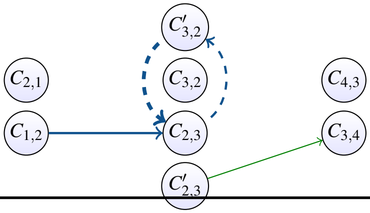


Dynamic Routing and Deadlock/Livelock

- Network Topology
  - Minimal routing exception: dashed linked is only usable to reach S_4.

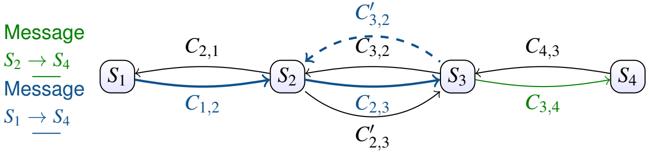


- Waiting buffer graph is dynamic.

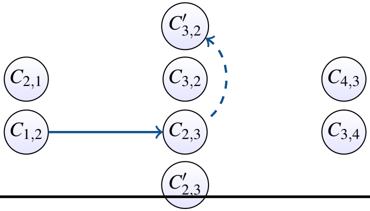


Dynamic Routing and Deadlock/Livelock

- Network Topology
  - Minimal routing exception: dashed linked is only usable to reach S_4.

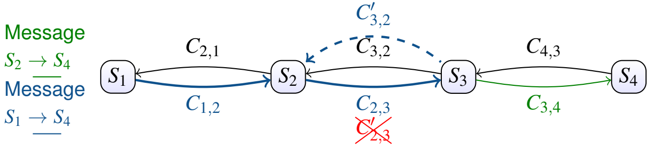


- Waiting buffer graph is dynamic.

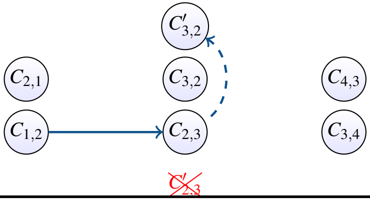


Dynamic Routing and Deadlock/Livelock

- Network Topology
  - Minimal routing exception: dashed linked is only usable to reach S_4.

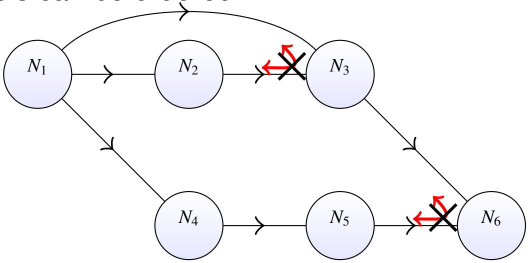


- Waiting buffer graph is dynamic.



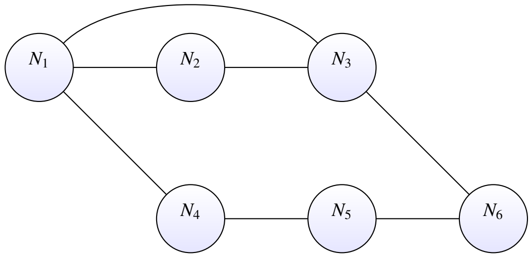
Transitively Deadlock-Free

- Enclosing Routing function:
  - computes all routes following the up-down rule:
    - Messages cannot go upward after going downward,
  - returns for any port a set with all ports leading to the destination.
- Channels can be ordered.



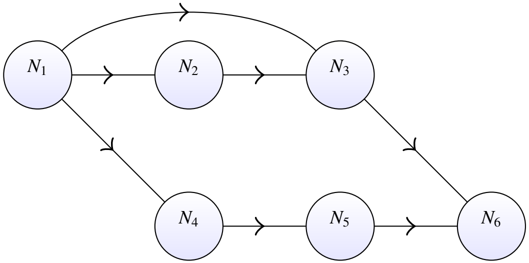
Online Routing Algorithm for Agnostic Topology

- Up*/Down* Algorithm on the network:



Online Routing Algorithm for Agnostic Topology

- Up*/Down* Algorithm on the network:
  - N_1 is the selected root,
  - directions are added to links.


N_1
N_2
N_3
N_4
N_5
N_6