Abstractions for Reconfigurable Hybrid Network Update and A Consistent Update Approach

Weitao Wang, Sushovan Das, T. S. Eugene Ng
Rice University

Reconfigurable Optical Switches

Diverse optical circuit switching technologies

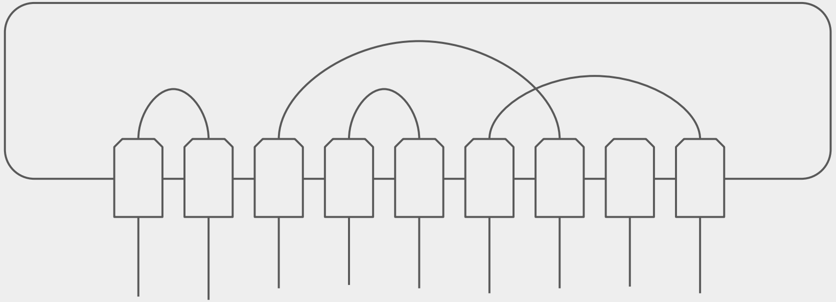
MEMS switches: mirror rotation
AWGR switches: diffraction grating

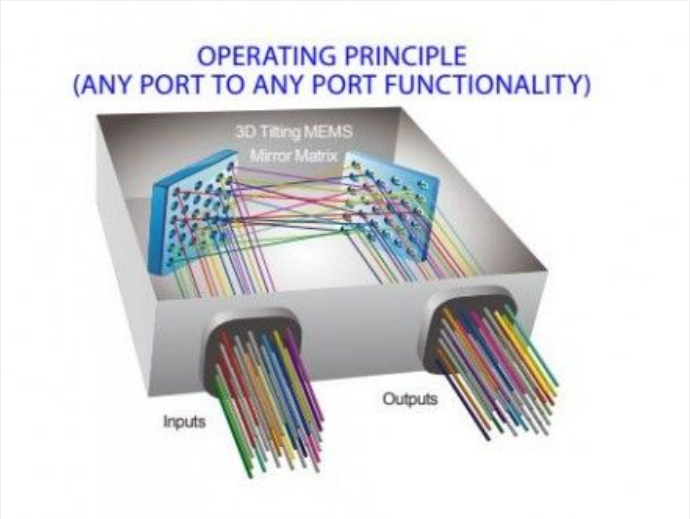
Pros:

➢ Low power consumption
➢ Data rate agnostic
➢ No buffering

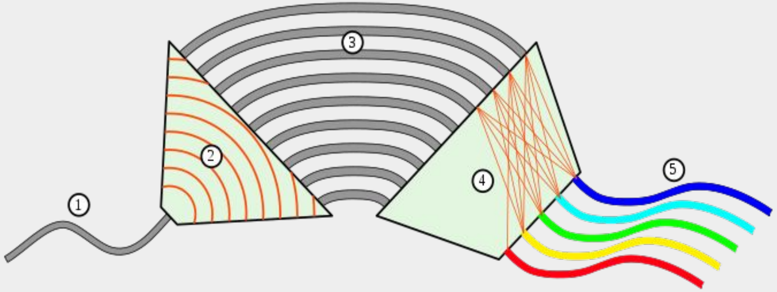
Cons:

➢ **Limited routing flexibility**





MEMS



AWGR

What is the Reconfigurable Hybrid Network (RHN)?

Reconfigurable hybrid networks are networks that consist of both **packet switches** and the **reconfigurable optical switches**.

Example systems like Helios, C-Through, OSA, REACToR, Firefly, ProjecToR, RotorNet, Flat-Tree, Sirius.

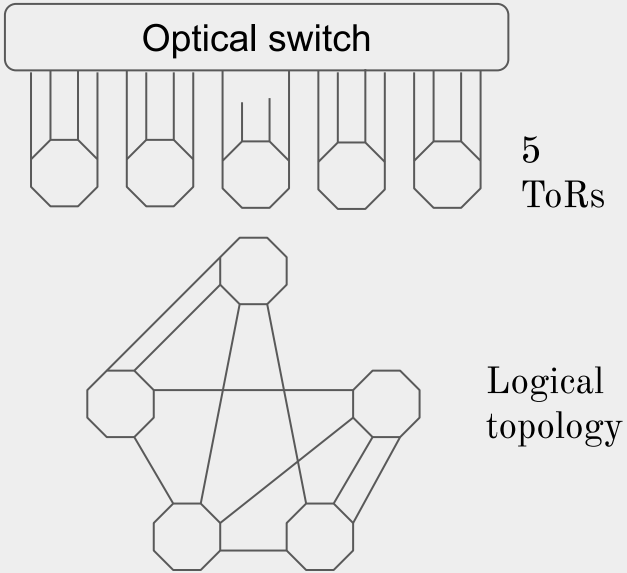
Benefit of reconfigurable topology:

- **Higher bandwidth** by adapting the topology according to the traffic pattern
- **Lower cost** to provide similar performance as a fixed network

Two Example Reconfigurable Hybrid Networks

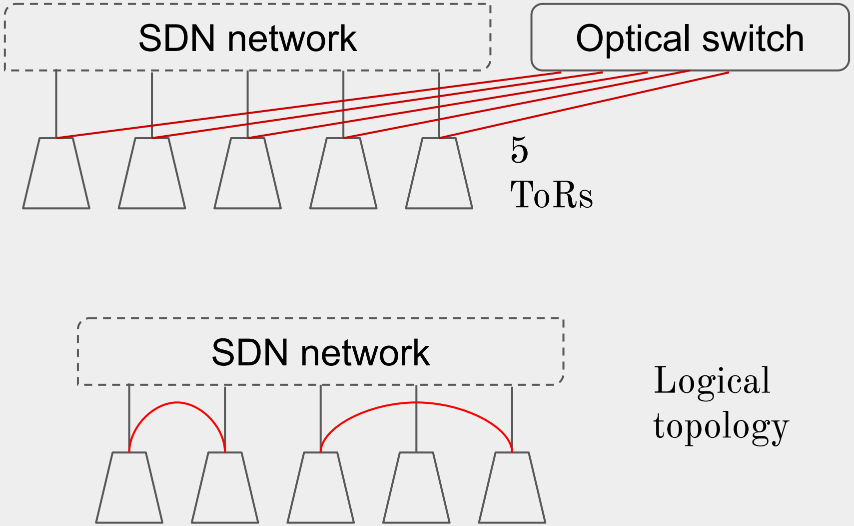
Spineless network core

- More bandwidth between heavily communicated ToRs
- Less bandwidth or even only 2-hop path between barely communicated ToRs



Additional optical paths

- Provide additional path for heavily communicated ToRs



RHN Update Consist of Optical Update and Policy Update

A reconfigurable hybrid network update usually has two tasks:

1. Configure optical links
2. Configure SDN policies
   a. Routing policy
   b. ACL policy
   c. ...

Those two tasks are usually conducted by two separate controllers:

➢ The optical switch controller
➢ The commodity switch controller

Existing Methods: RHN Architectures' Solution

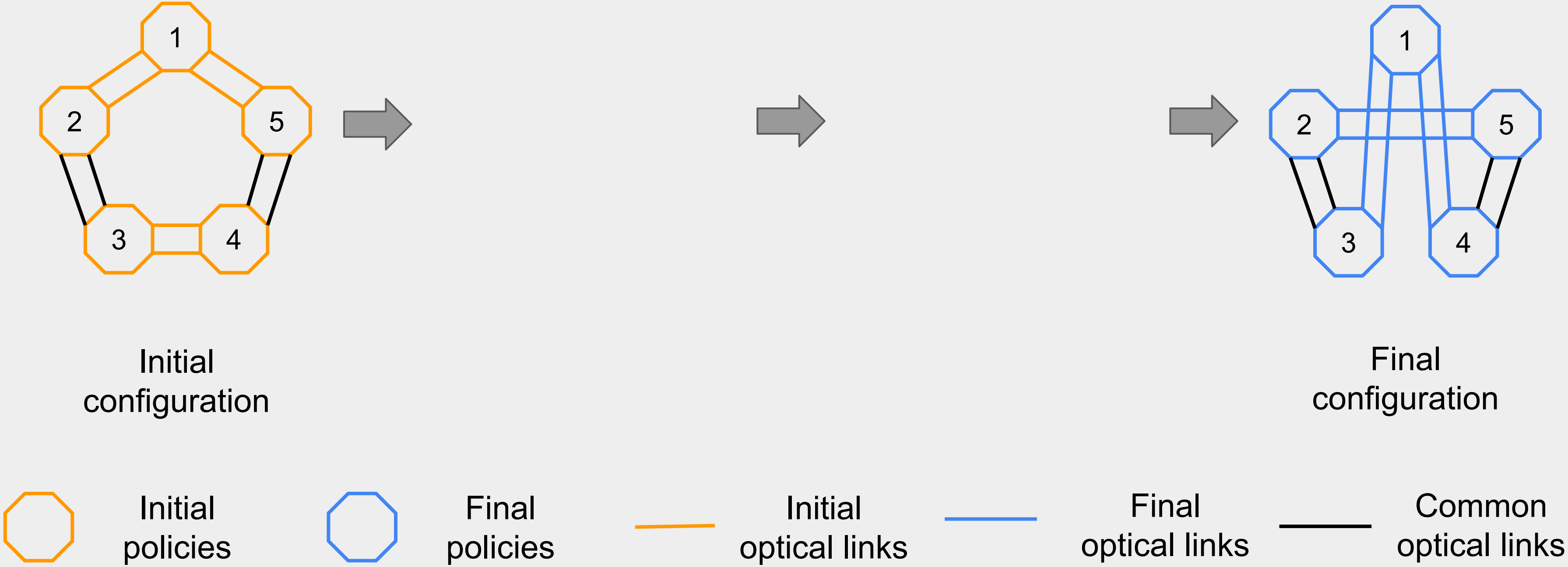
Typical method:

1. Reconfigure the optical part and the SDN part simultaneously
2. Rely on the fast retransmission and ignore the transient states

Pros: Minimize the update time

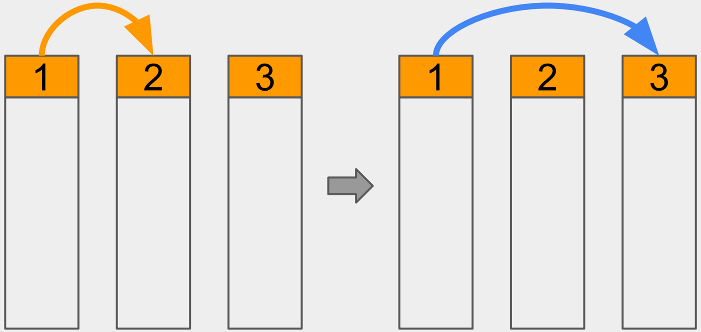
Cons: Leads to per-packet inconsistency

RHN's Solution Leads to Per-packet Inconsistency



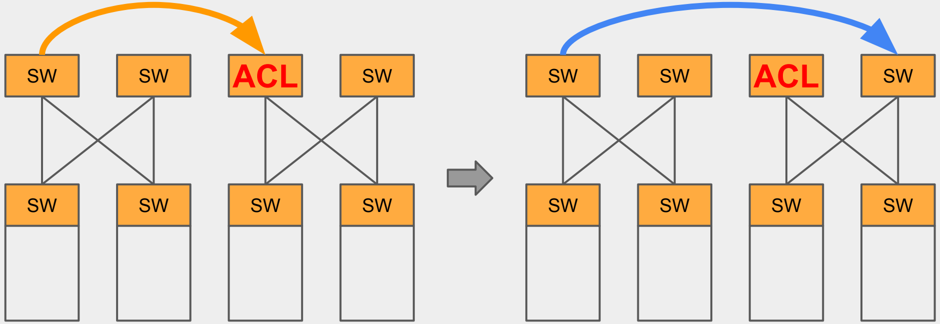
The SDN policies and the optical links are not configured consistently.

Per-packet Inconsistency Analysis



Packet may be sent to the wrong destination.

Old policy + New link



Packet may violate or bypass the security rules.

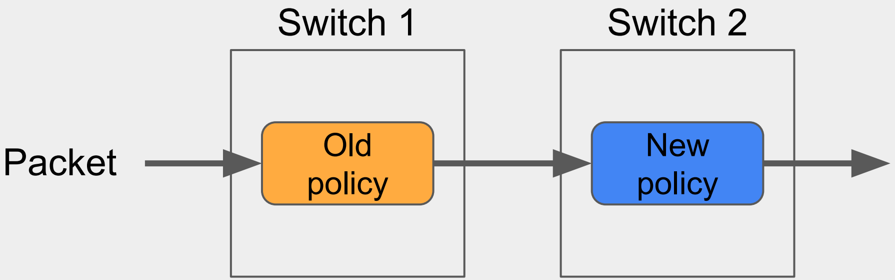
Old policy + New link

RHN Per-packet Consistency:

a packet should only use **old** SDN policies and **old** optical links, or only use **new** SDN policy and **new** optical links, but never a mix of two.

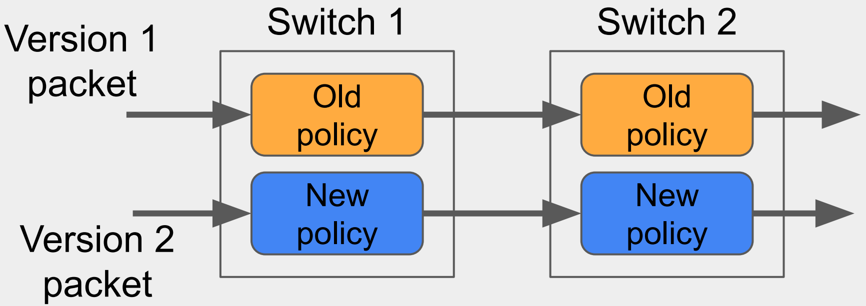
Two-phase Commit Can Preserve Per-packet Consistency

To update multiple SDN switches, a packet may experience **new policy and old policy on different hops** during the transient state.



Hence, two-phase commit is introduced:

1. Add the new SDN policies on all the switches;
2. Change the packet version;
3. Remove the old policies.



It ensures a packet will only experience **new** policies or **old** policies on all the hops, **never a mix of two**.

But two-phase commit require the topology to be fixed.

Existing Methods: SDN Networks' Solution

Thus, to use two-phase commit for RHN Update:

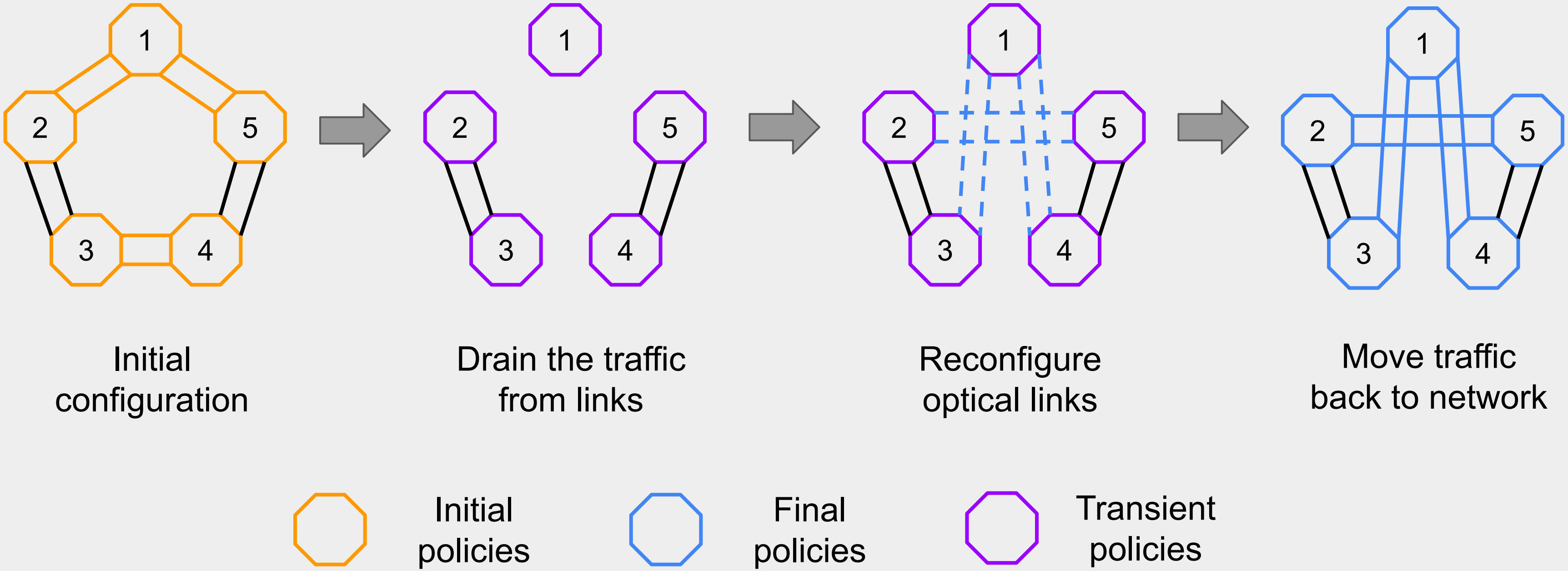
1. Apply two-phase commit to drain the traffic from the links need to be configured;
2. Reconfigure all the optical links safely;
3. Apply two-phase commit to move the traffic back to all the links;

In this way, we can ensure the **topology will not be changed during the two-phase commit**, so that the per-packet consistency is preserved.

Pros: Preserve per-packet consistency

Cons: Transient disconnectivity

SDN's Solutions Suffer From Transient Disconnectivity



Problem: Some ToRs are disconnected until the final configuration

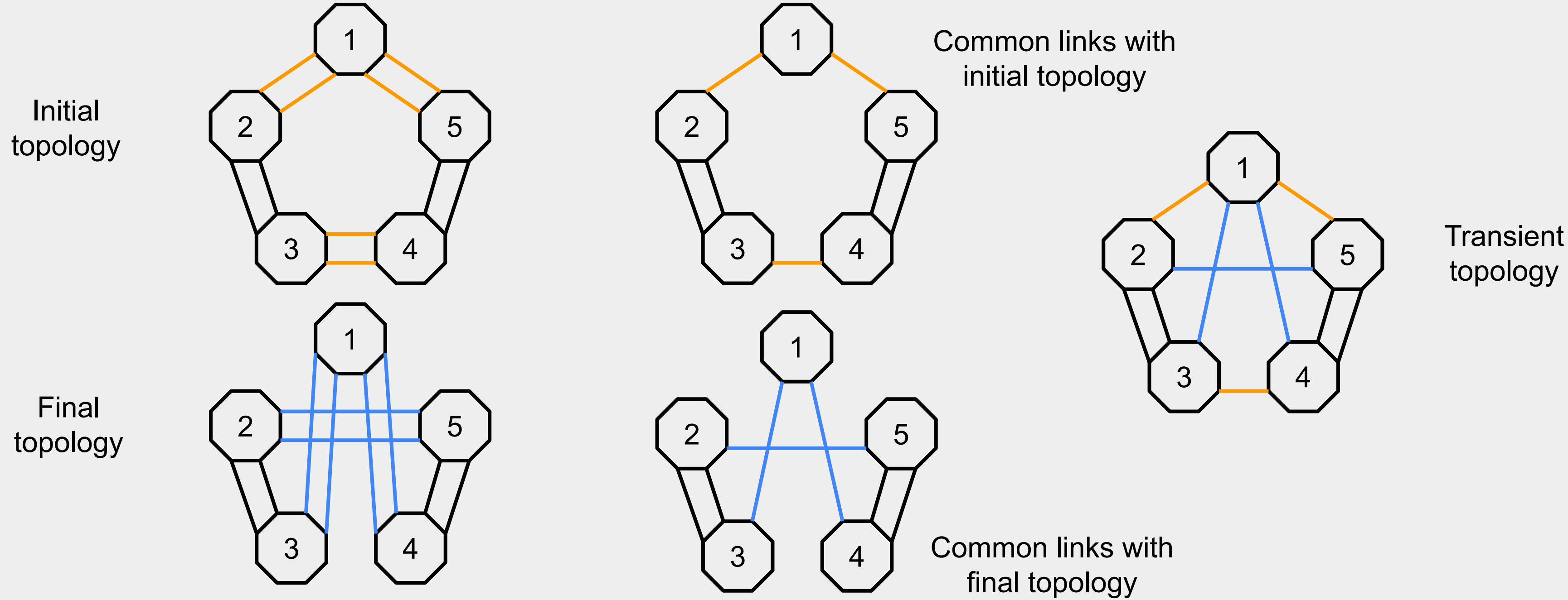
Design Overview

Transtate: multi-phase update engine for RHN

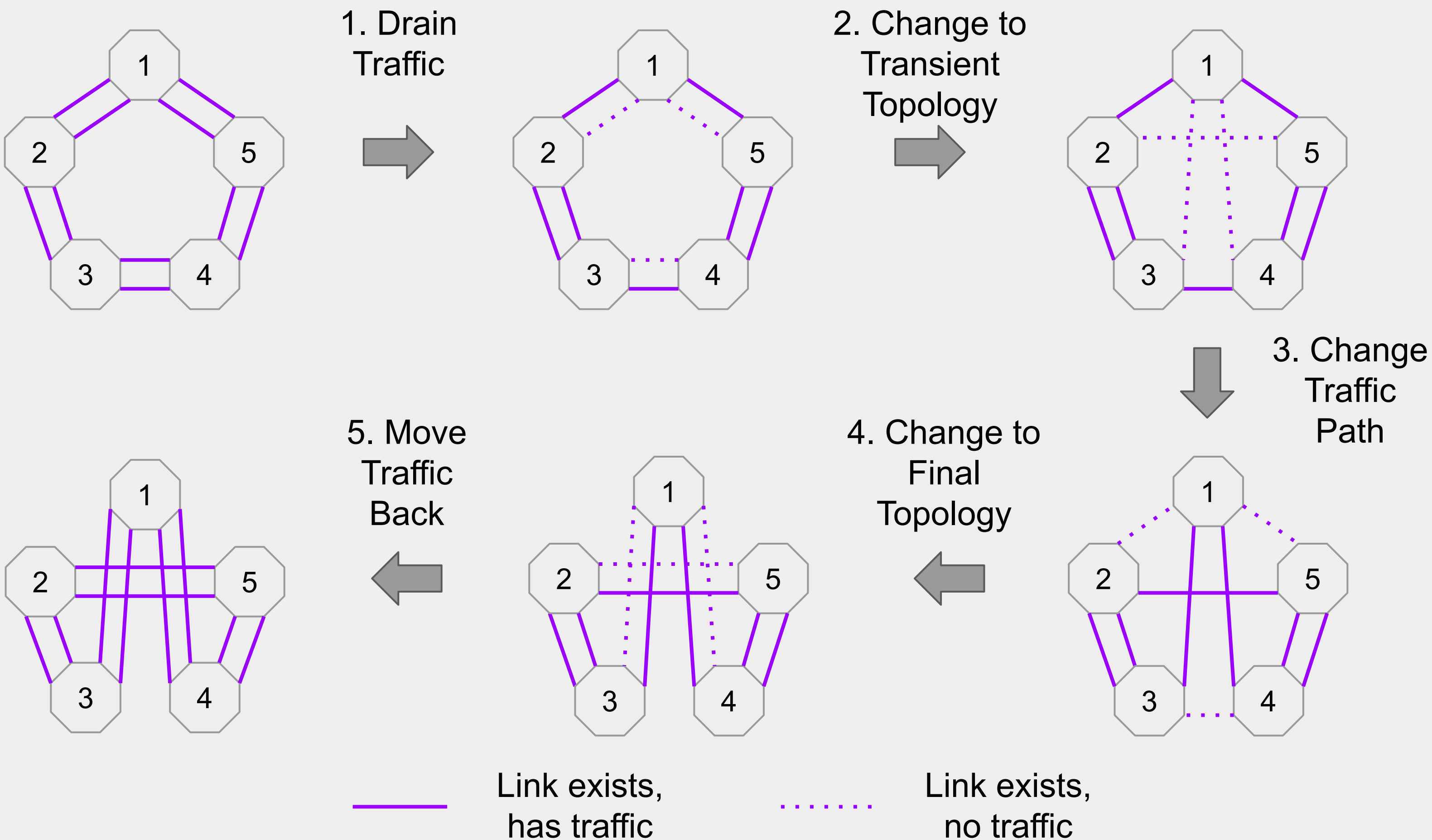
a) Finding optimal transient topologies to preserve all-to-all reachability
   i) Allow multiple-hop path
   ii) Minimize the bandwidth reduction during reconfiguration
b) For each sub-update, move traffic to common links for per-packet consistency
   i) Use two-phase commit to drain the traffic
   ii) Update the optical topology
   iii) Use two-phase commit to move the traffic back

Preserve Connectivity with The Transient Topology

Transtate creates a transient topology that has **common links** with both initial topology and final topology. Thus, the all-to-all connectivity can be preserved during the update.



Transtate Preserves both Consistency and Connectivity



Implementation and Evaluation

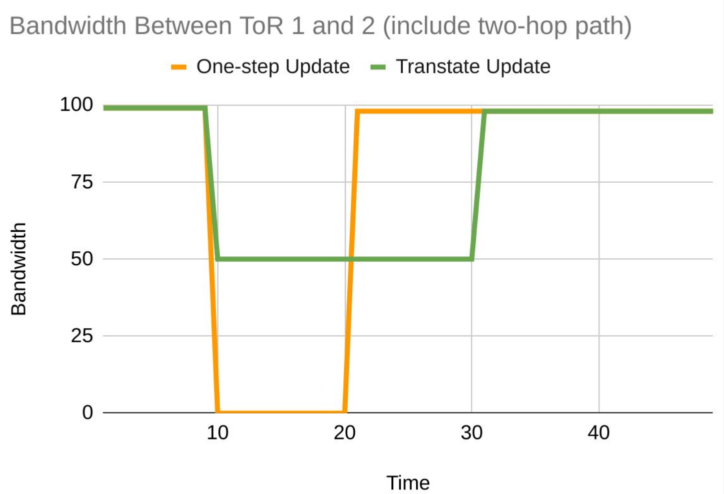
Transtate uses an LP solver to find a valid transient topology for any RHN update. (More details about the solver can be found in the paper)

1. Minimize bandwidth reduction

➢ The bandwidth reduction between ToR 1 and 2 is reduced from 100% to 50%.

2. Short computation time

➢ For optical switches with less than 50 ports
  ○ The update plan can be found within 10 ms
➢ For optical switches with 1000 ports
  ○ The update plan can be found within 1 seconds.



The source code for the solver can be found under
Github repo: github.com/Clark5/RHN_Update

Conclusion

In this slides, we introduces Transtate:

1. RHN Per-packet Consistency;
2. A multi-phase update solution for both consistency and connectivity;
3. A fast LP solver to find a valid update plan given any RHN update.

In the paper, we also cover:

4. A new abstraction to model the hybrid network updates;
5. Detailed analysis and evaluations for the LP algorithm.

Q & A!