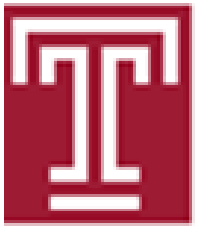
Chronus: Consistent Data Plane Updates in Timed SDNs

Jiaqi Zheng

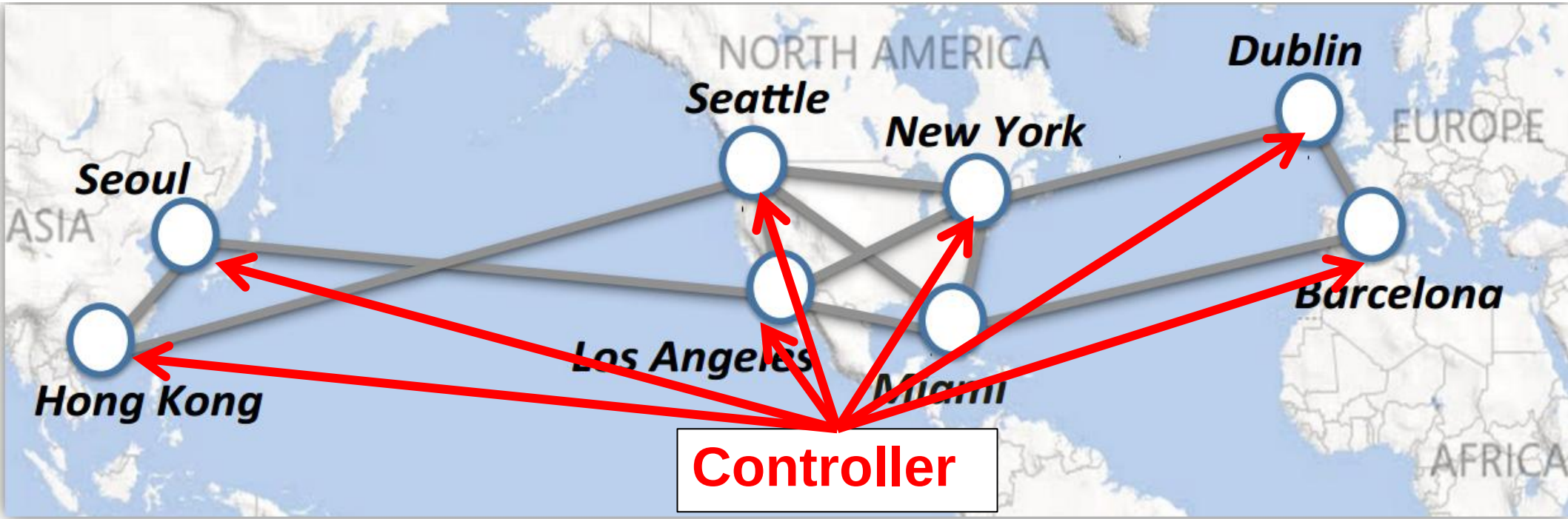
with Guihai Chen, Stefan Schmid, Haipeng Dai and Jie Wu

Nanjing University
Shanghai Jiao Tong University
Aalborg University
Temple University



SDN at a glance

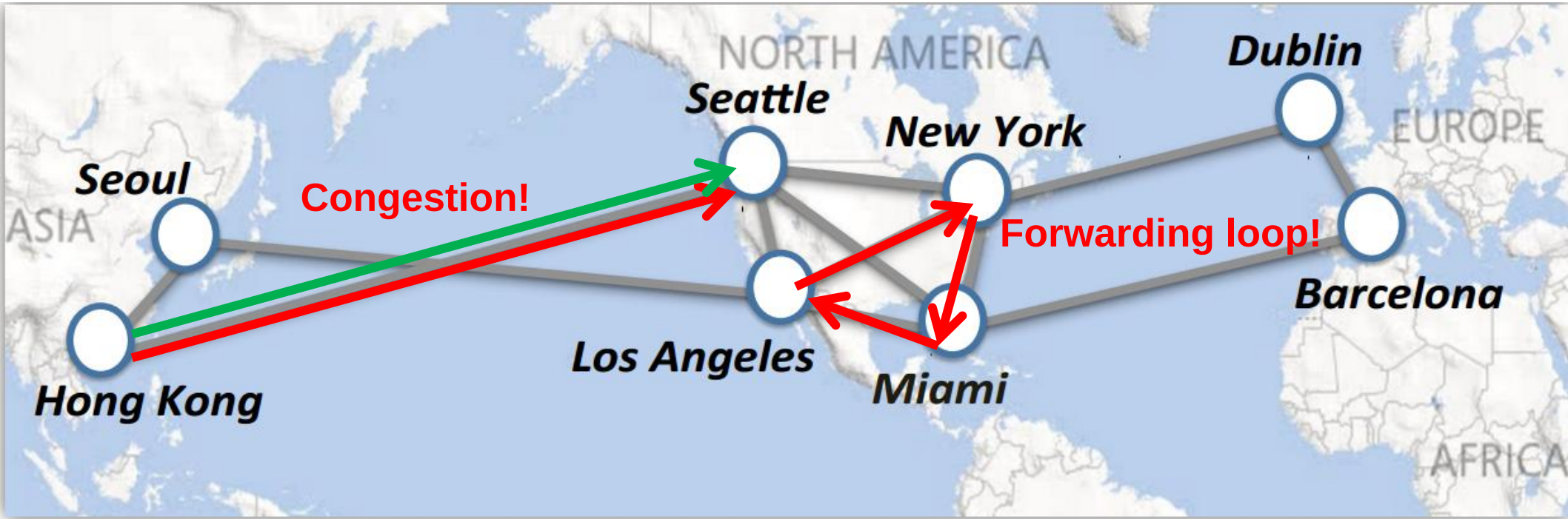
- The logically centralized controller periodically enforce new policies by installing, modifying, or deleting forwarding rules in switch flow tables through southbound APIs such as Openflow.



What is consistent data plane update?

Consistent network update

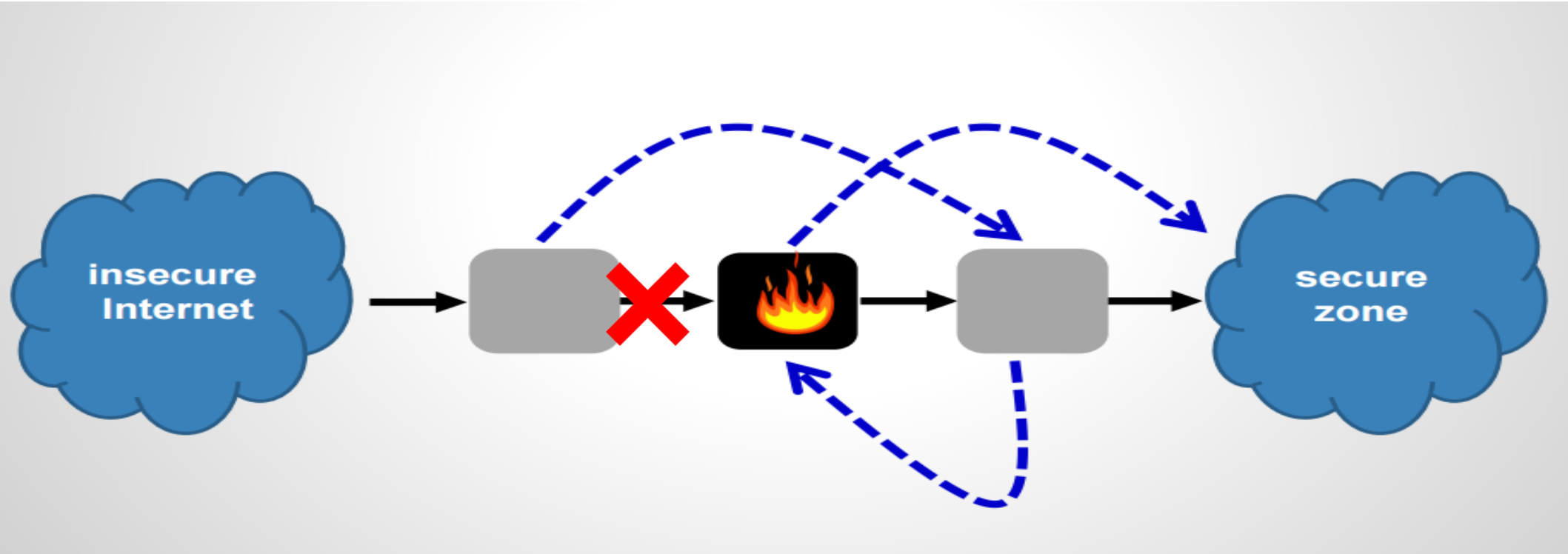
- Connectivity consistency: no blackhole, no forwarding loops.
- Policy consistency: firewalls, proxies, etc.
- Performance consistency: congestion-free



Why do we need data plane update?

Network update is important

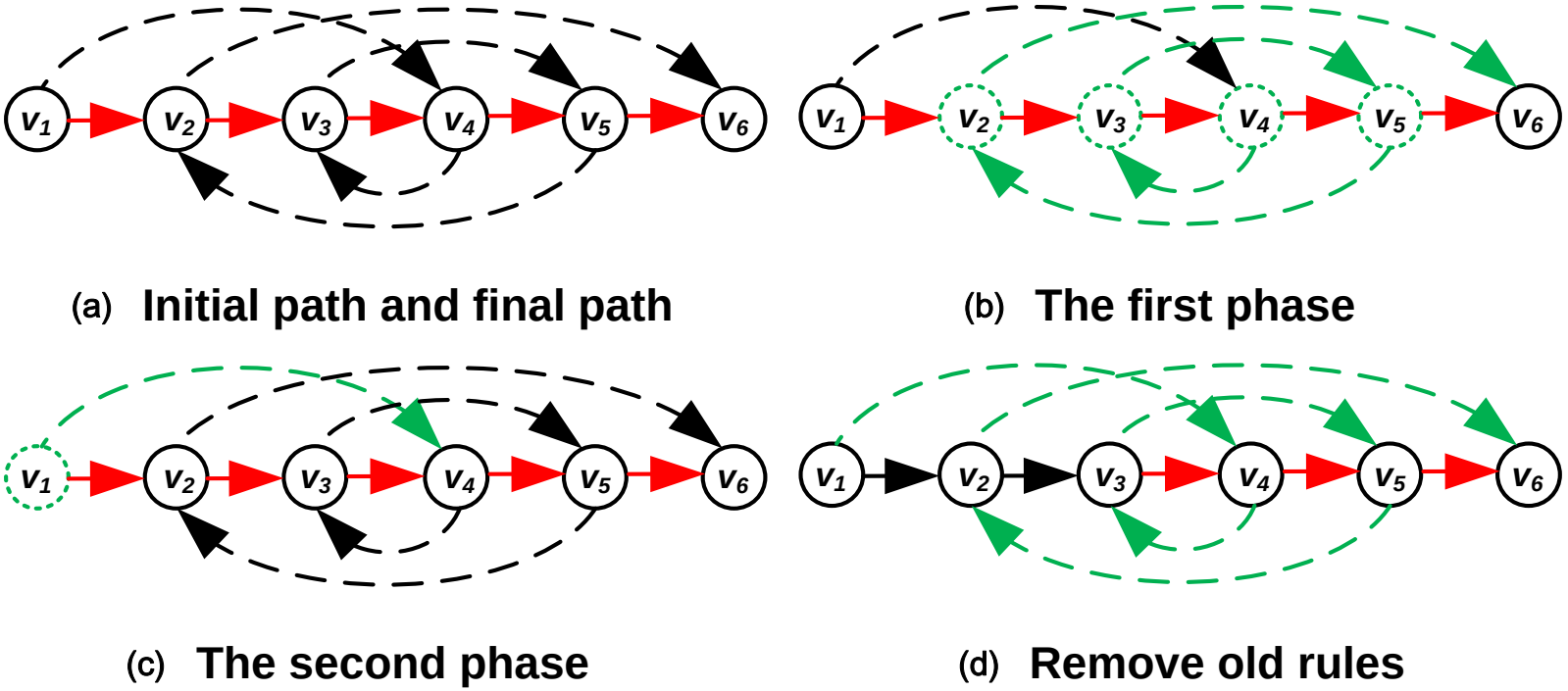
- Changes in security policies.
- Traffic engineering
- Network maintenance
- Failure recovery



How do the existing solutions work?

Two-phase update protocols [SIGCOMM'12,'13,'14]
and node ordering protocols [HotNets'14,SIGMETRICS'16]

Existing work: two-phase update protocols



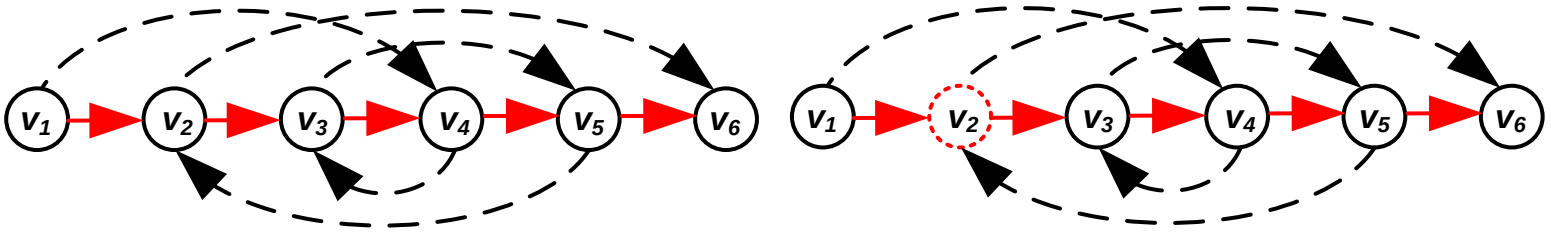
(a) **Initial path and final path**

(b) **The first phase**

(c) **The second phase**

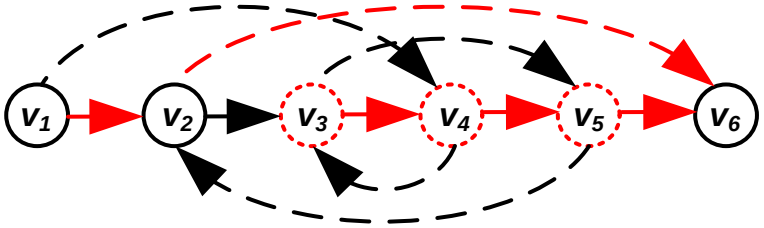
(d) **Remove old rules**

Existing work: node ordering protocols

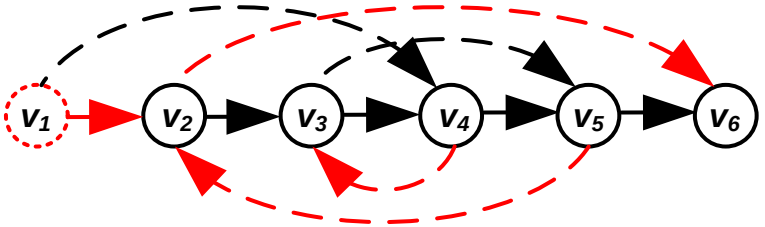


(a) **Initial path and final path**

(b) **Round 1**



(c) **Round 2**



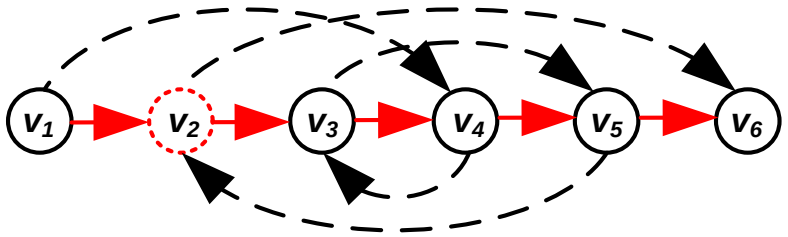
(d) **Round 3**

Any solutions to overcome the drawbacks of both?

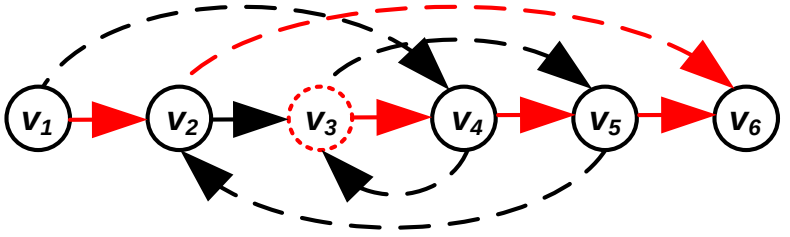
Chronus

Synchronized clocks among all the switches

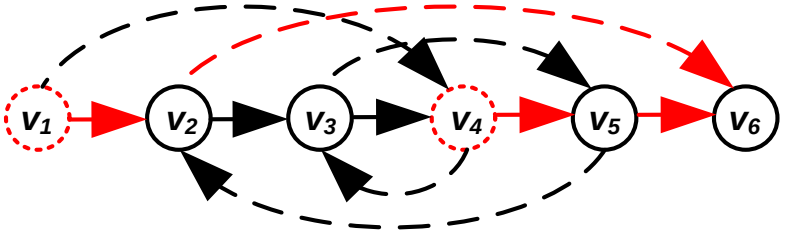
Chronus overview: a timed update sequence



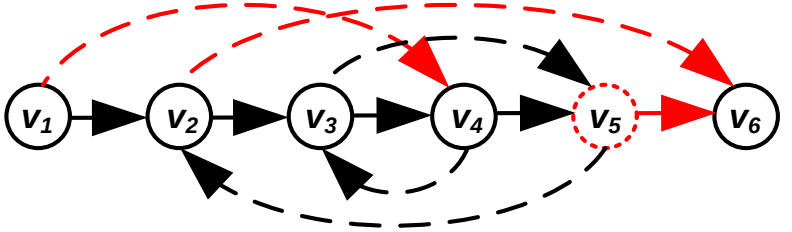
(a) **Time t_0**



(b) **Time t_1 (t_1=t_0+1)**



(c) **Time t_2 (t_2=t_1+1)**



(d) **Time t_3 (t_3=t_2+1)**

Challenge 1:

How to synchronize all the switches and schedule the timed updates?

Precision Time Protocol

Precision Time Protocol (PTP)
[IEEE 1588 2008]
~1µsec accuracy


Mobile backhaul


Power substations


Industrial automation

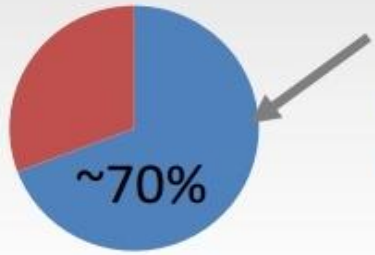

Financial applications

China Mobile: over 1,000,000 PTP-enabled base stations
https://mailarchive.ietf.org/arch/attach/tictoc/pdfsY1ADO.pdf

But what about

Software Defined Networks?

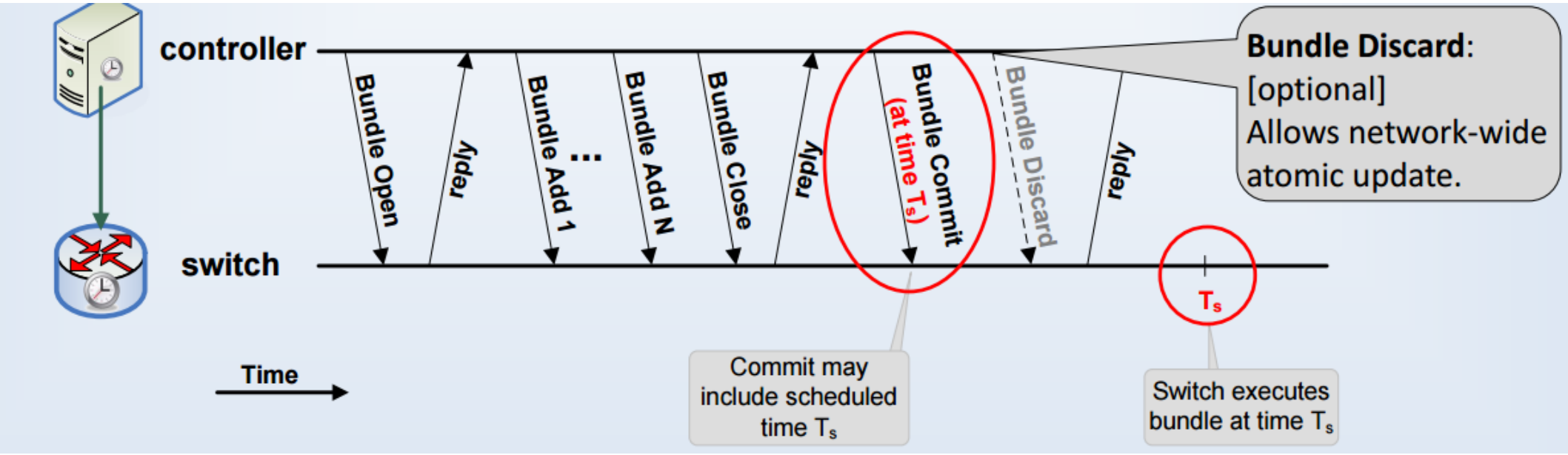
Open Networking Foundation (ONF) SDN product directory



~70% of the SDN-enabled silicons have native PTP support.

Notably, 9 out of the 13 SDN-capable switch silicons listed in the Open Networking Foundation (ONF) SDN Product Directory have native IEEE 1588 support.

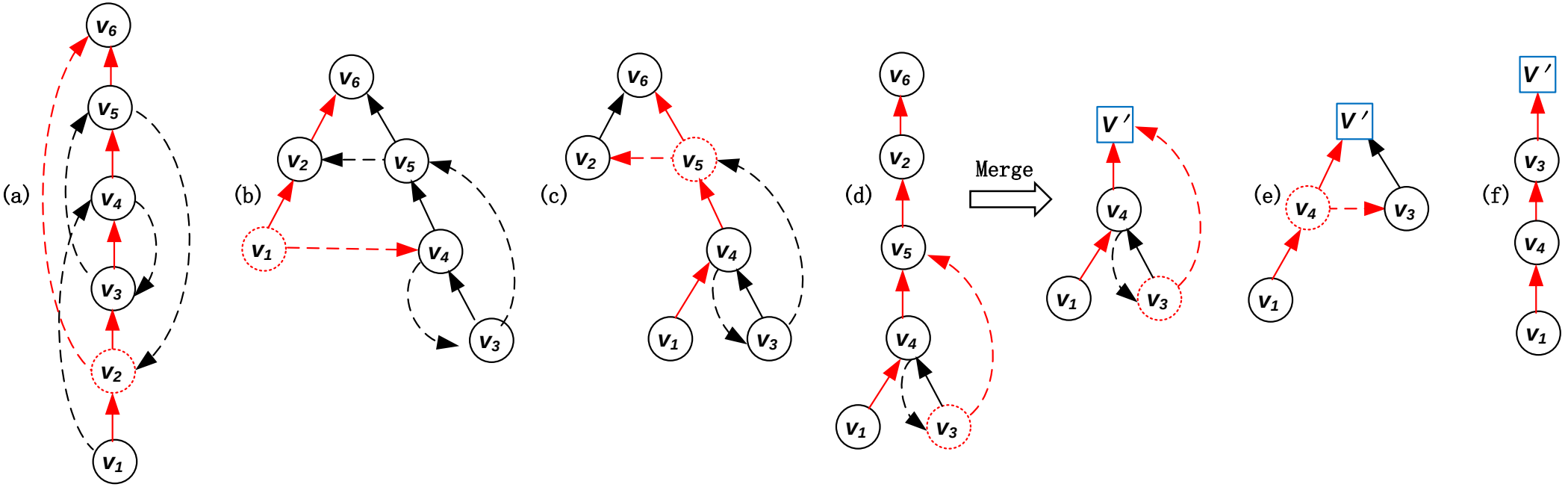
Openflow1.5 scheduled bundles feature



Challenge 2:

How to find a feasible solution such that congestion- and loop-free conditions hold in any moment in time?

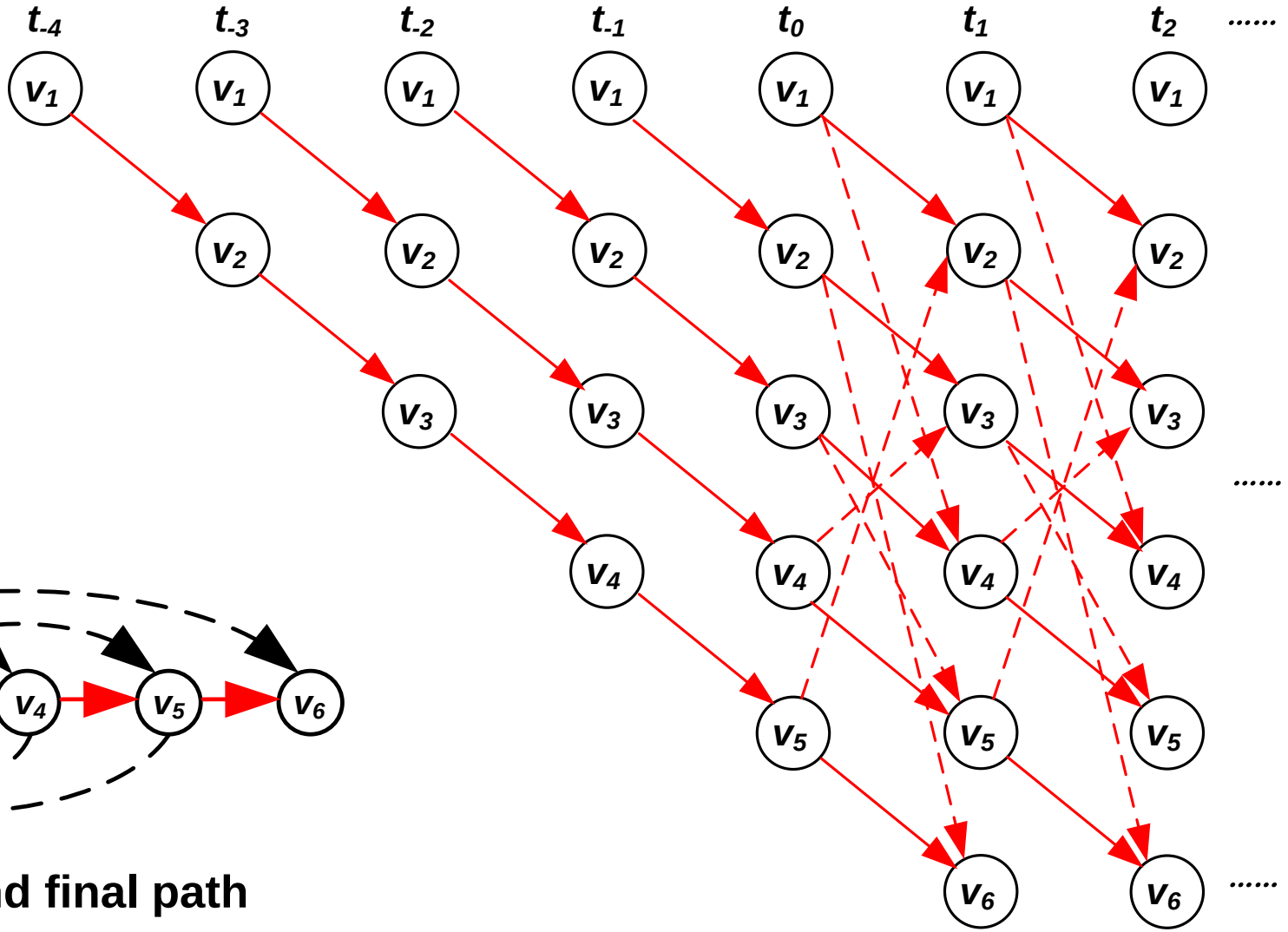
A tree-based algorithm

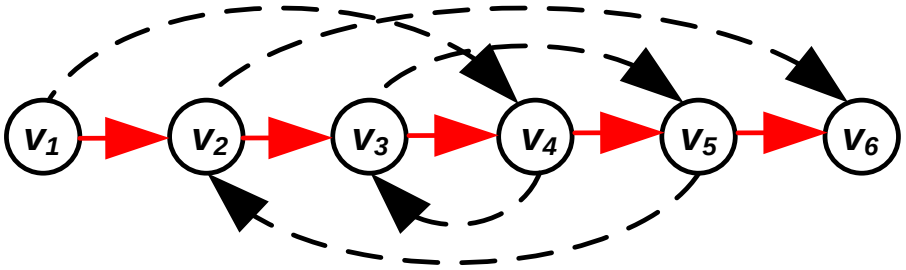


Challenge 3：

How to calculate a update sequence with minimum update time?

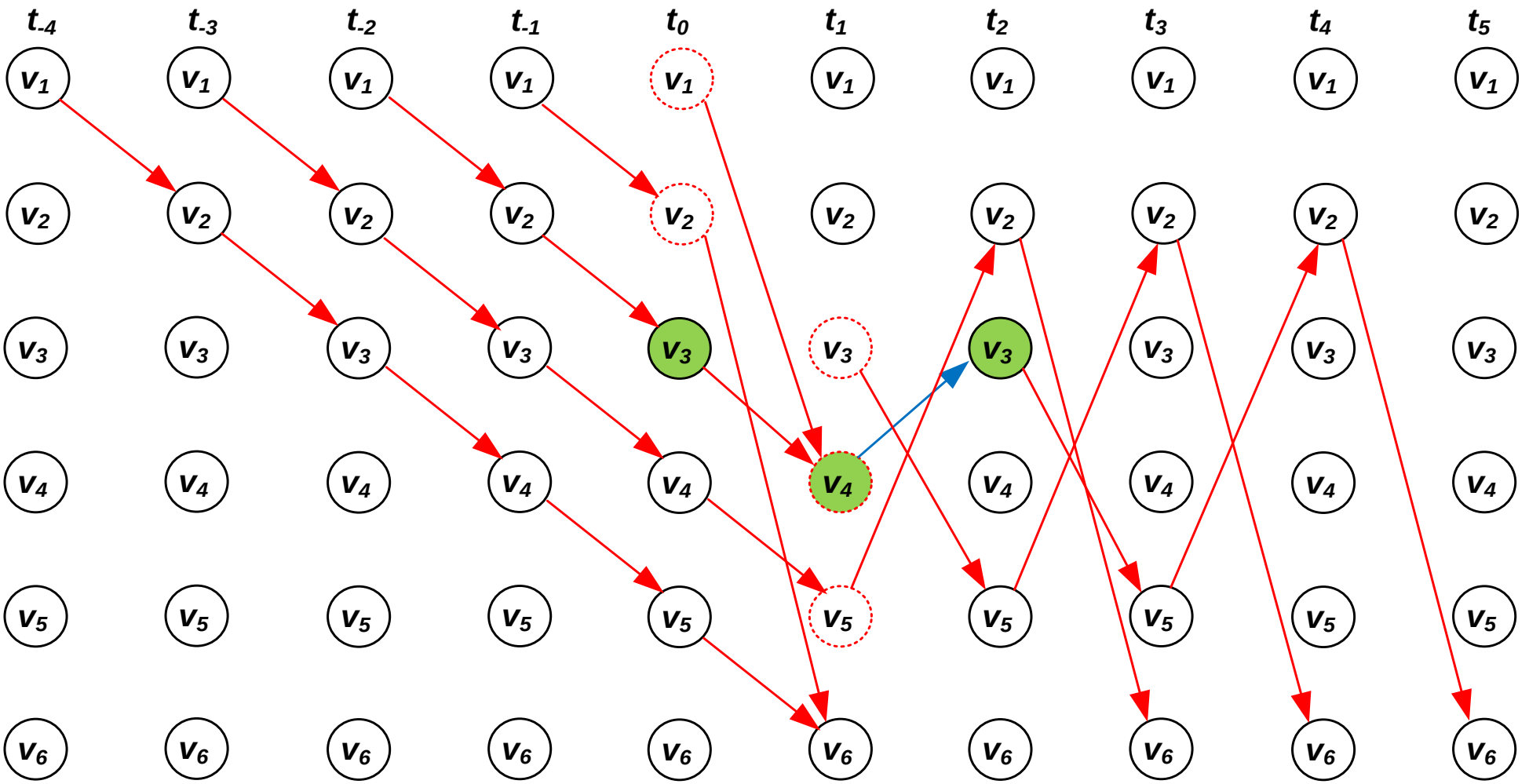
A time-extended network perspective



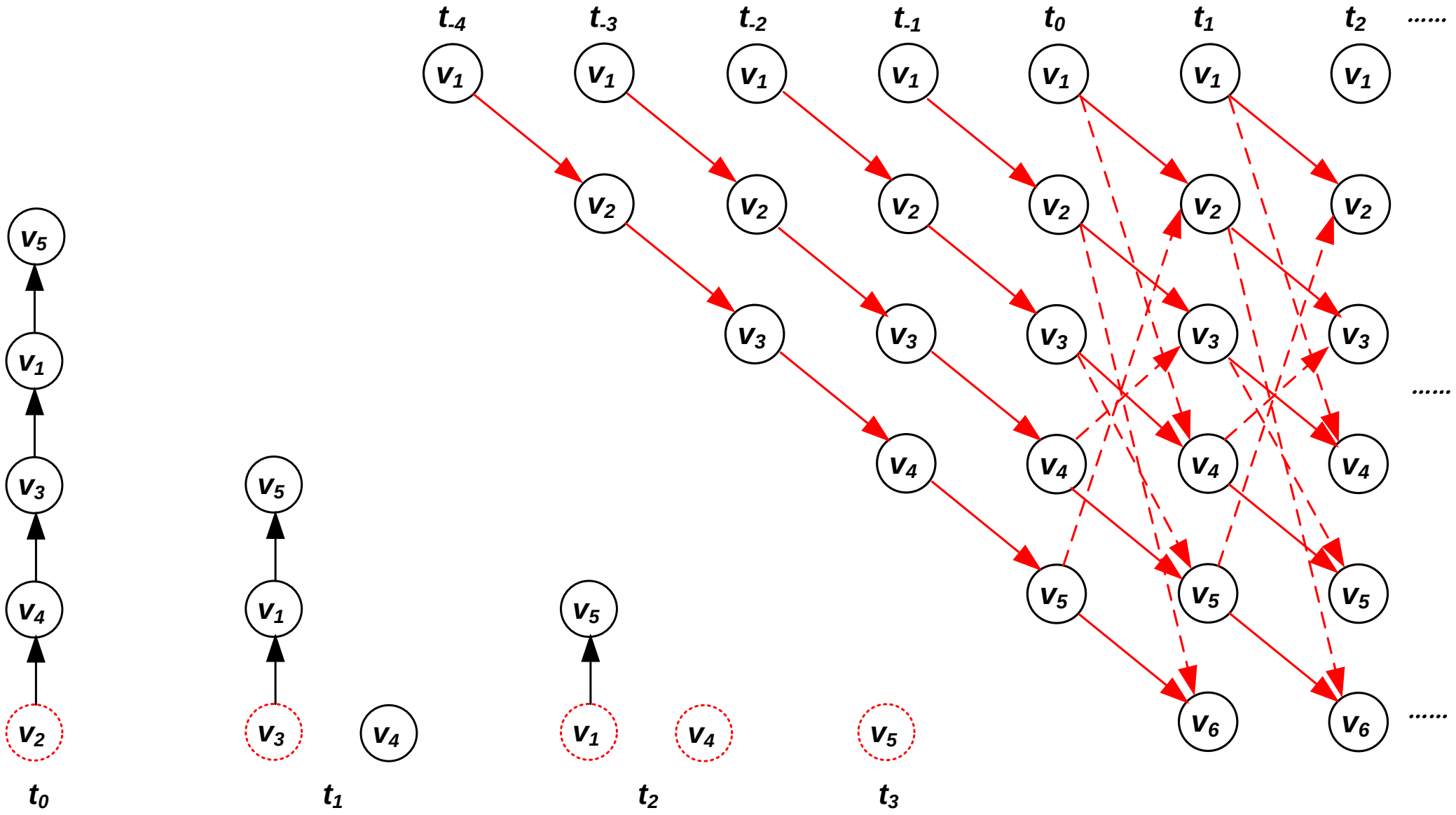


Initial path and final path

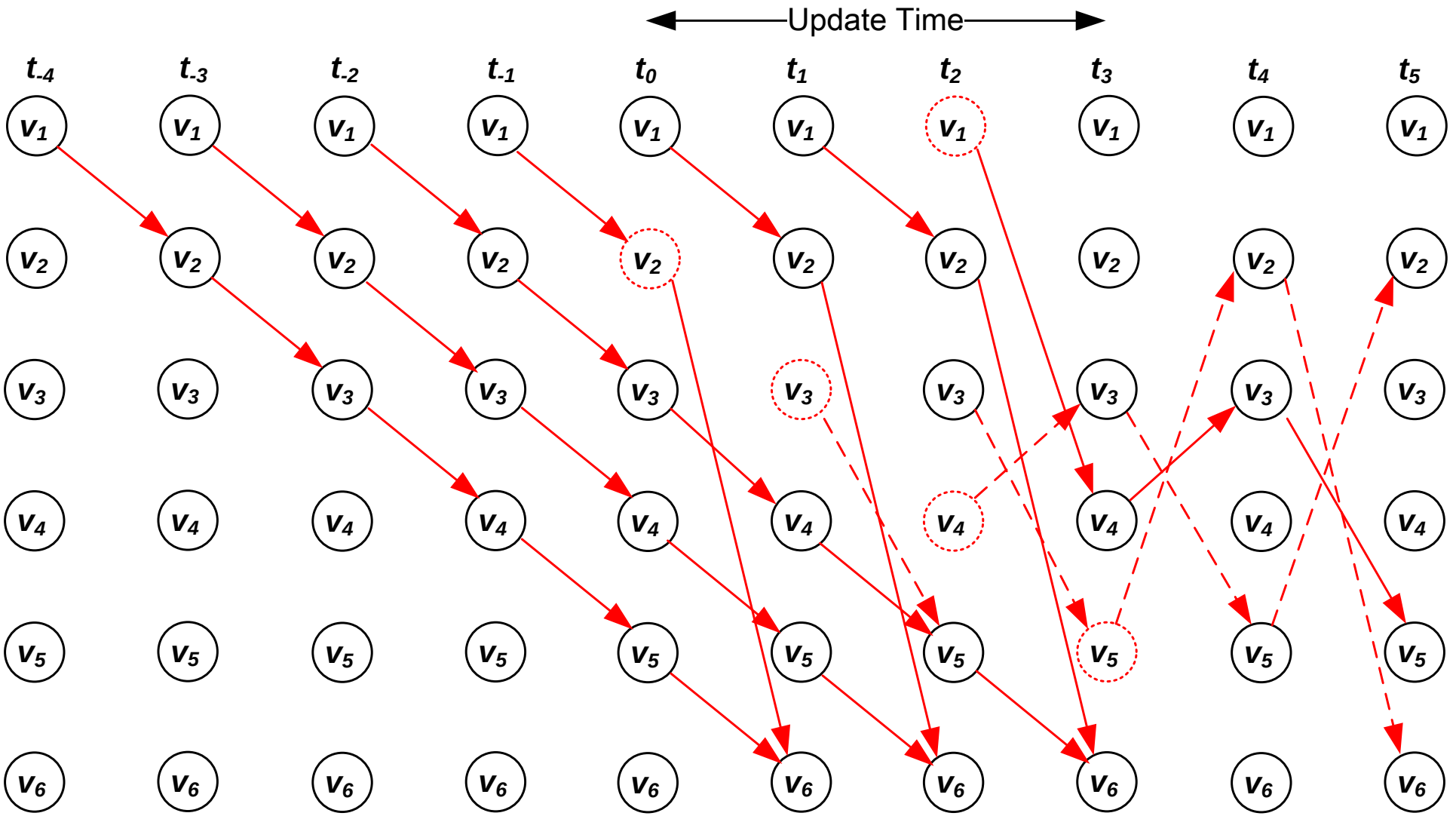
Congestion and forwarding loops happen



Constructing dependency graph



A timed update schedule



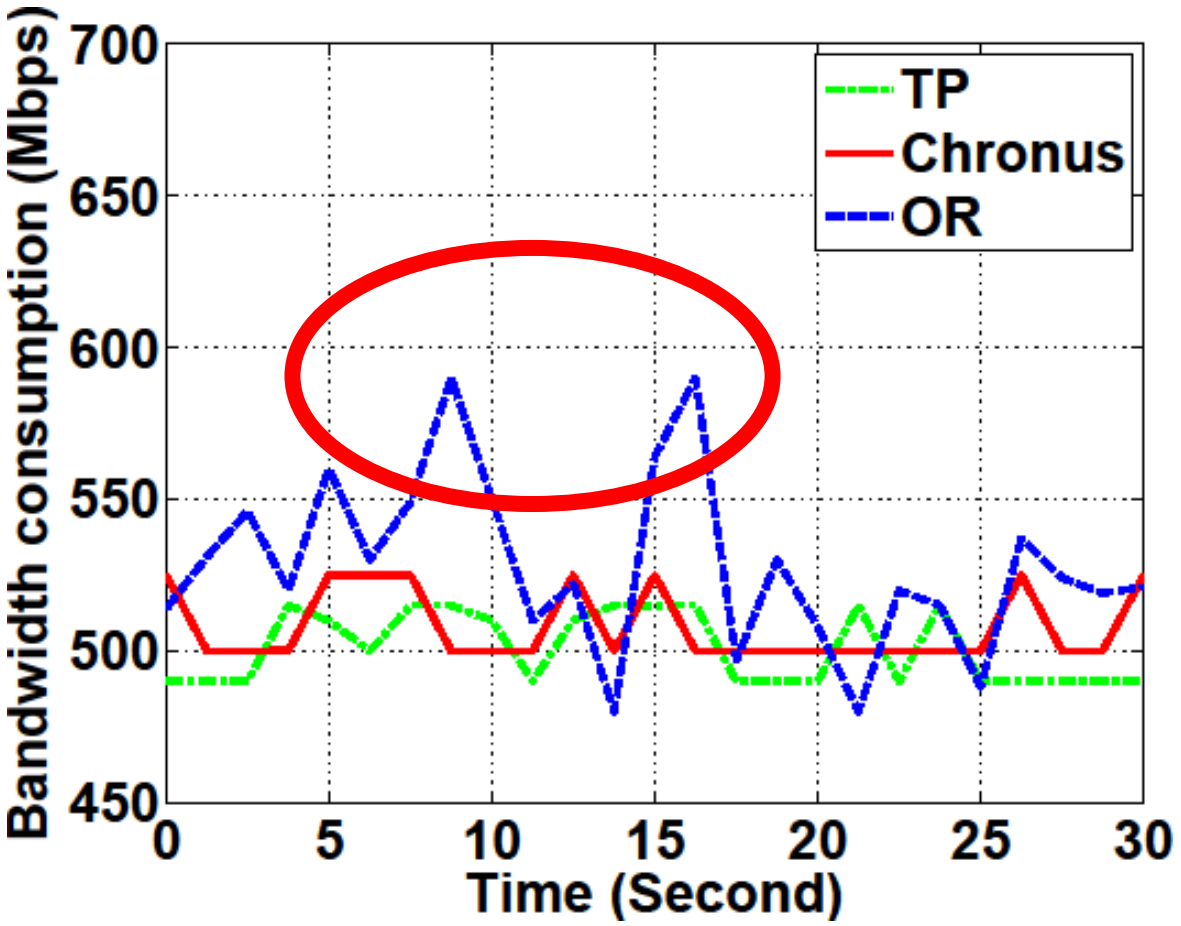
Evaluation：benchmark schemes

- **OR**：the node ordering protocols that minimize the number of rounds (i.e., the interactions between switches and the controller) and avoid the forwarding loops.
- **TP**：the two-phase updates protocols where we use VLAN IDs as version number in our experiments.
- **OPT**：the optimal solution of integer programming in the formulation.

Evaluation：Mininet setup

- The clock of all switches are synchronized by default in Mininet.
- Switches are emulated by Mininet v2.2.1 and OpenvSwitch version is v2.3.1.
- We adopt a small scale network topology with 10 switches. We set the link capacity to 500 Mbps.

Experiment results

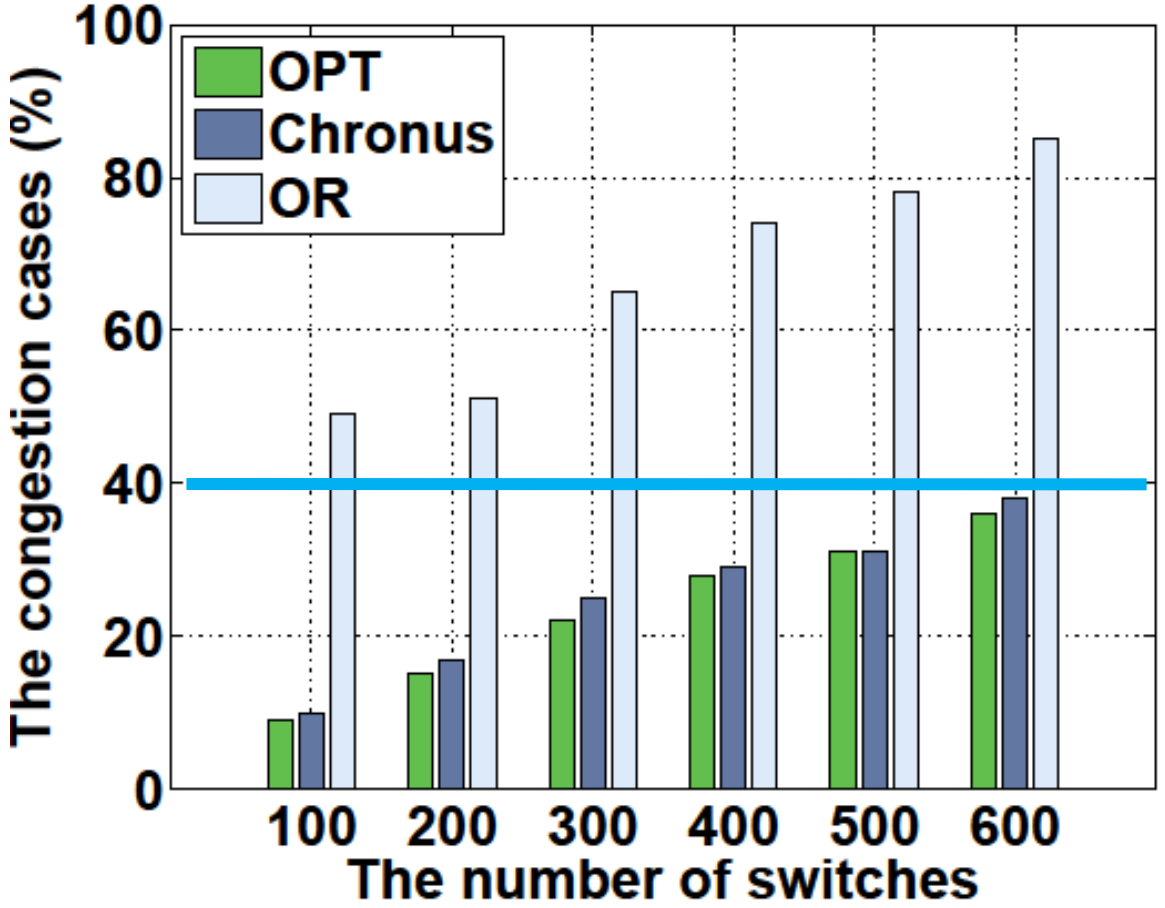


The peak value of OR is around 600 Mbps at 9th and 16th second, whereas the fluctuation of Chronus and TP is relatively stable and changes in the normal range.

Simulation setup

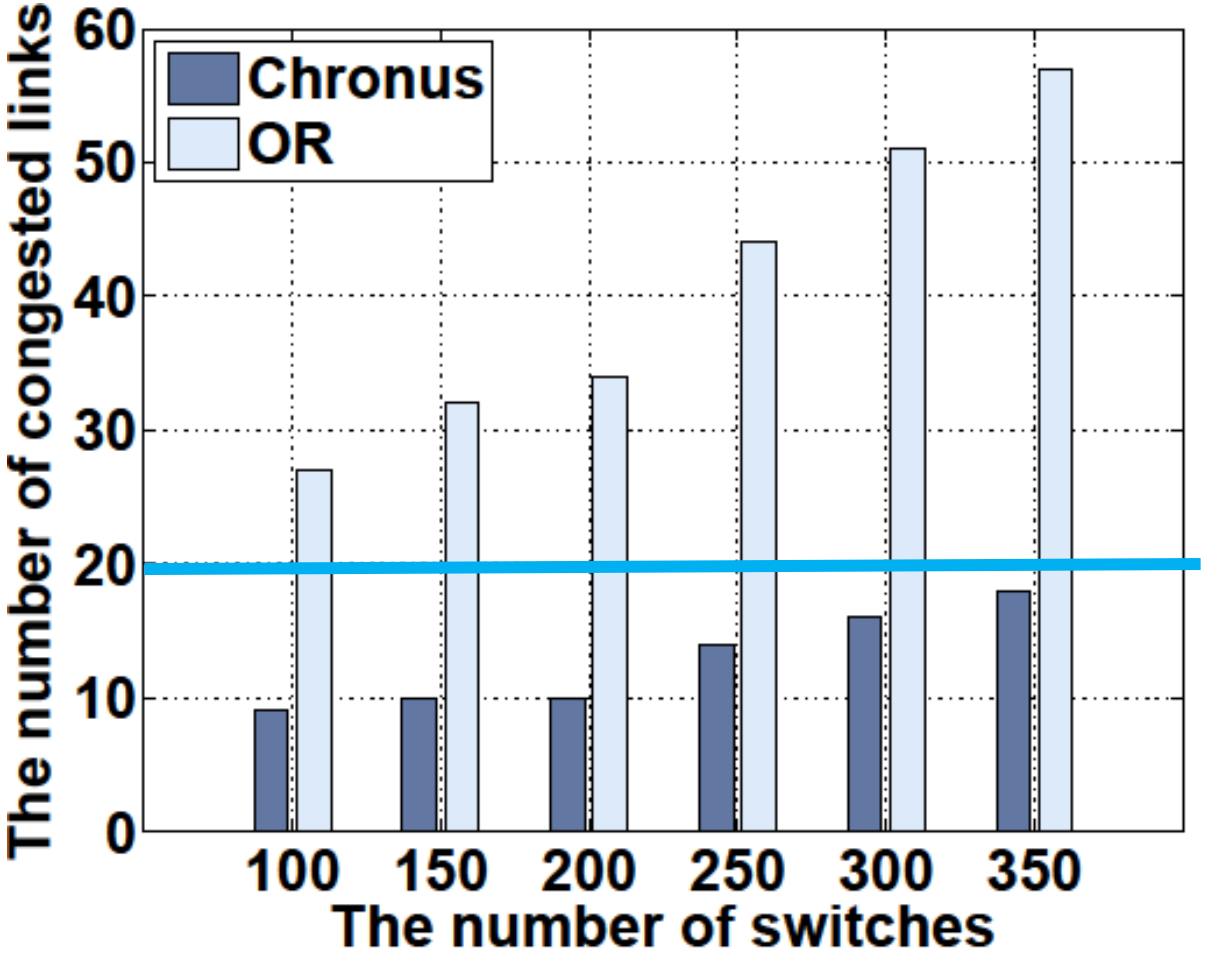
- In addition to the small-scale topology in Mininet, here we use a large-scale network topology. The initial routing path is fixed and the final routing path is chosen randomly
- We run our algorithm on an Amazon EC2 c4.2xlarge instance with 8 CPU and 15 GB memory.

Experiment results



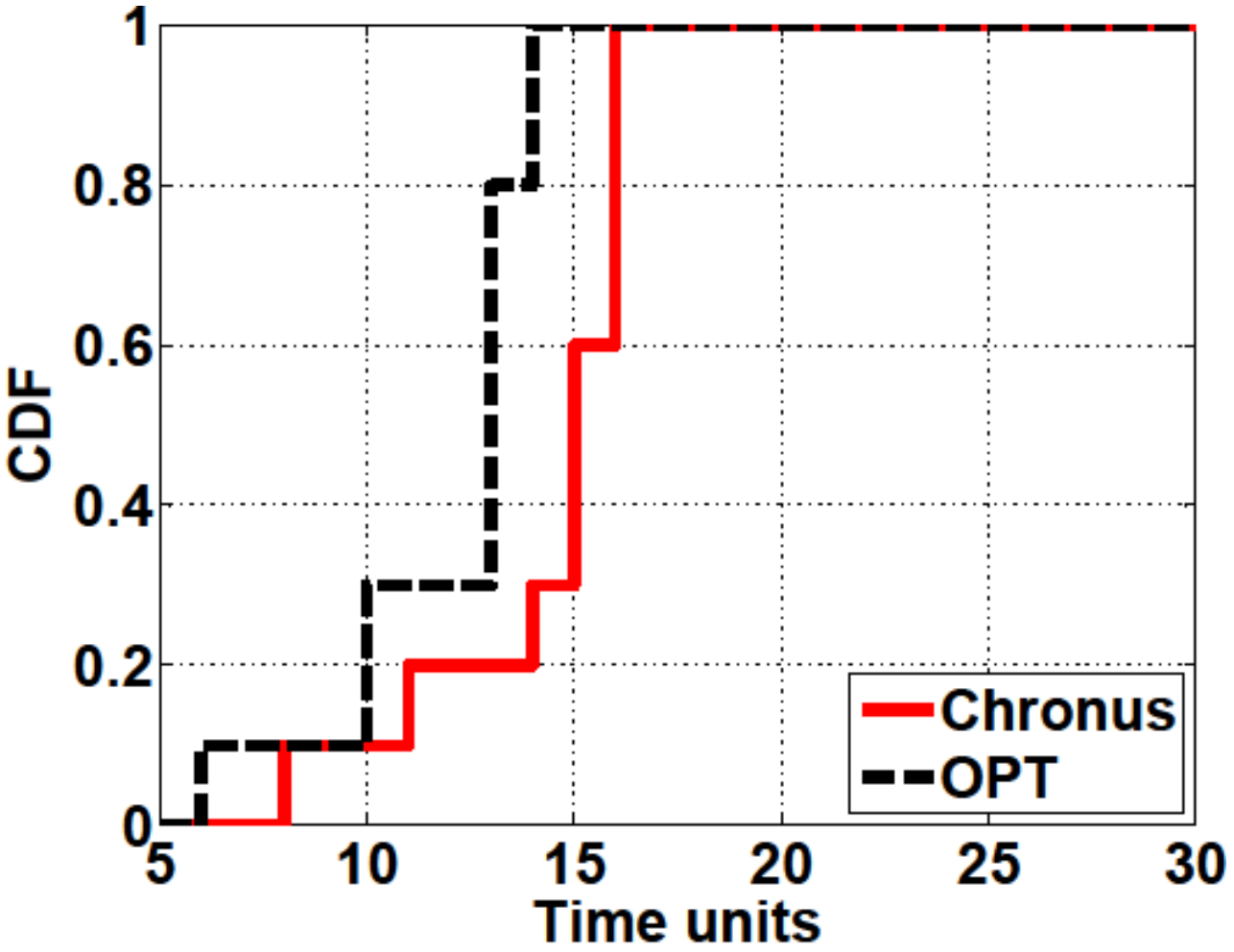
The congestion cases

Experiment results



The number of congested links

Experiment results



Update time

Conclusion

- We presented and formulated the problem of minimizing the route update time in timed SDNs.
- We proposed a tree-based algorithm to check the feasibility of an update schedule in polynomial time and described a greedy algorithm to solve the problem.
- Our evaluation results show that our solutions can reduce transient congestion and save flow table space.

Thanks & Questions?

Email: jiaqi369@gmail.com