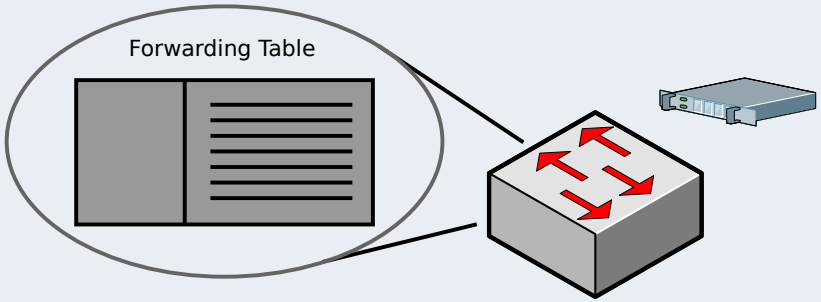
Routing and State Distribution Tradeoffs in Software Defined Networks

Nadi Sarrar, Dan Levin
{nadi,dan}@net.t-labs.tu-berlin.de

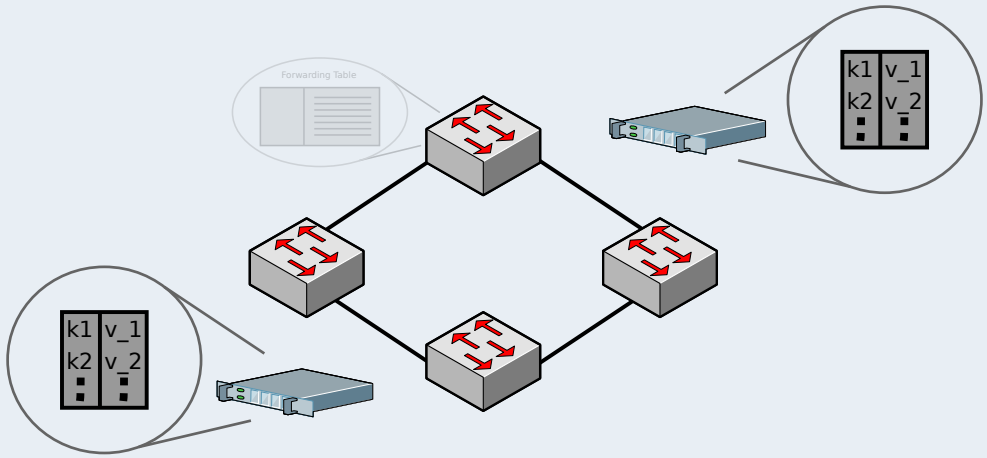
Technische Universtität Berlin / T-Labs
Berlin, Germany

Outline: From Micro to Macro

1: Locality-aware FIB Aggregation

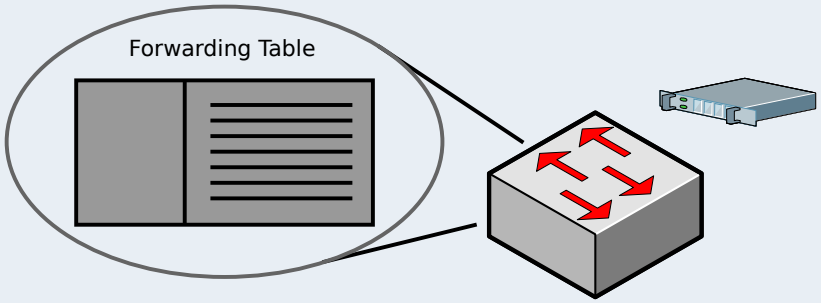


2: Control State Distribution Trade-offs



Motivation

Use-case: IP routing



Challenges when implementing routing on top of SDN

- Limit number of flow entries
- Limit number of flow table modifications
- Use FIB aggregation?

Motivation

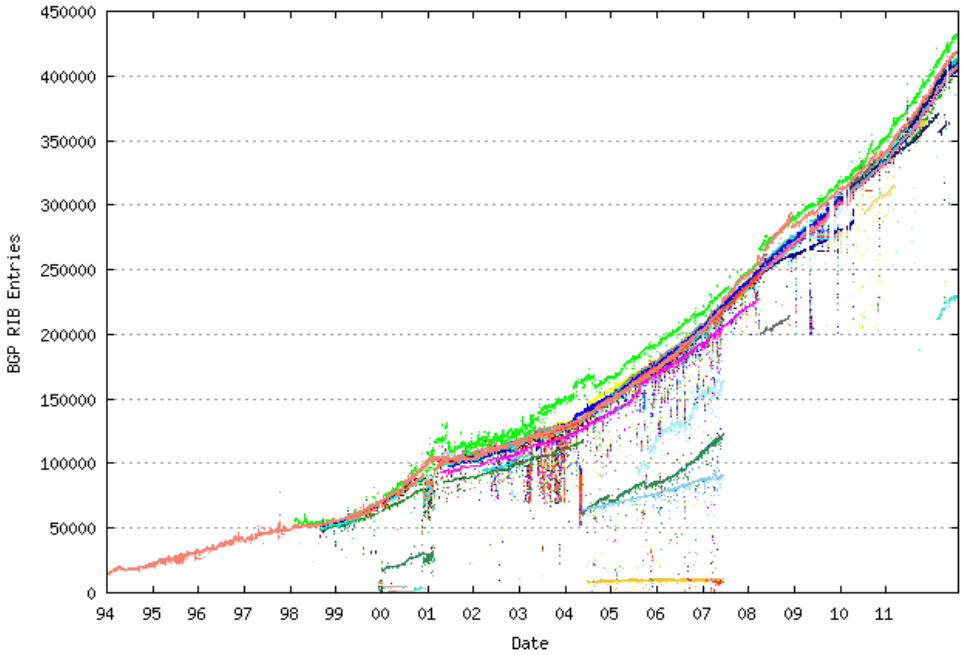
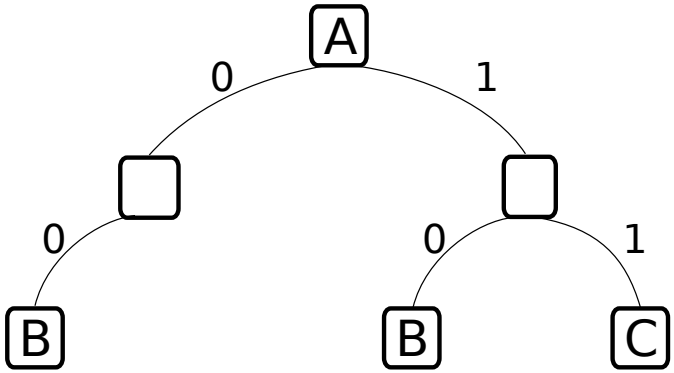


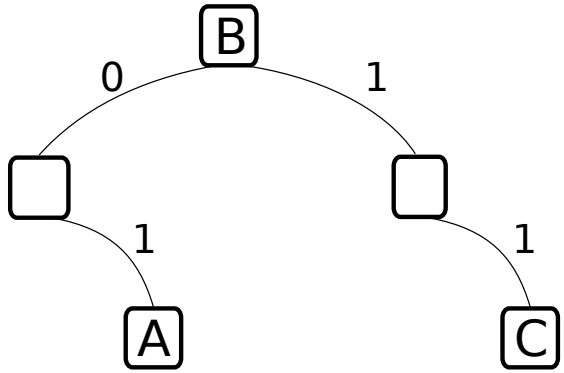
Figure: Geoff Huston, bgp.potaroo.net

Growth of the Internet routing table

Motivation



Original Routing Table (OT)



Aggregated Routing Table (AT)

FIB aggregation: lookup equivalence in OT and AT.

- Existing algorithms: ORTC, SMALTA, ...
- Reduces FIB memory requirements.
- Can increase update cost.

Our approach: treat "churny" regions differently.

Motivation

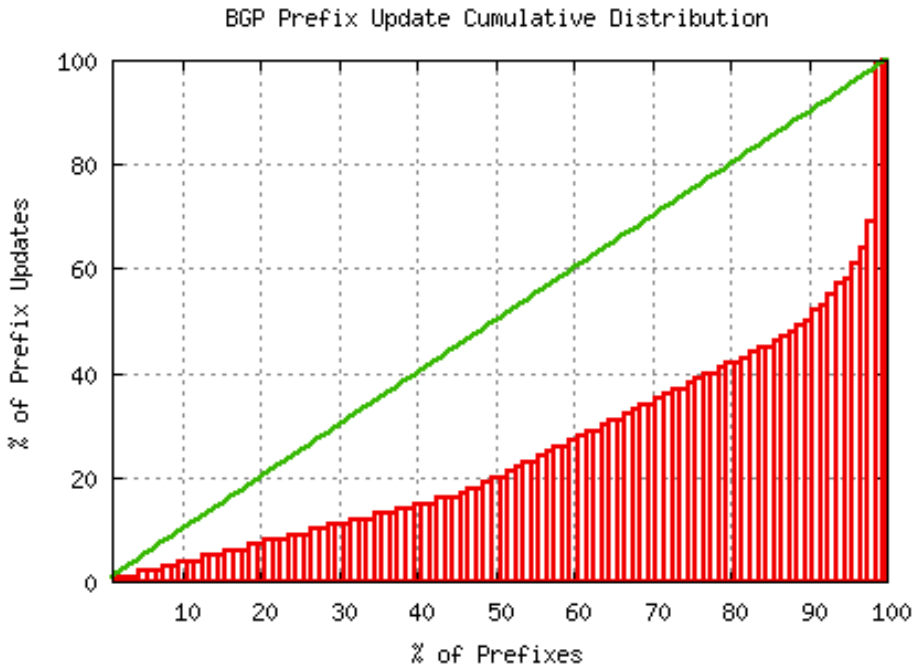
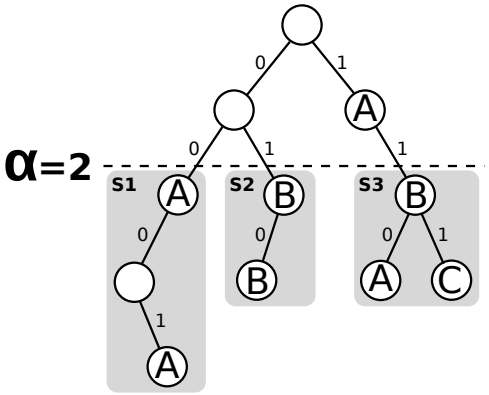


Figure: Geoff Huston, bgp.potaroo.net

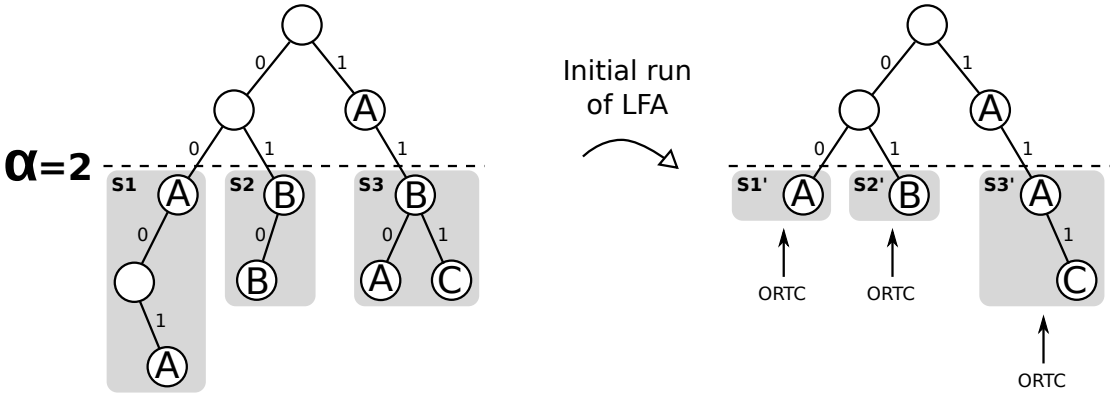
Unequal update rate per IP prefix.

LFA: Locality-aware FIB Aggregation



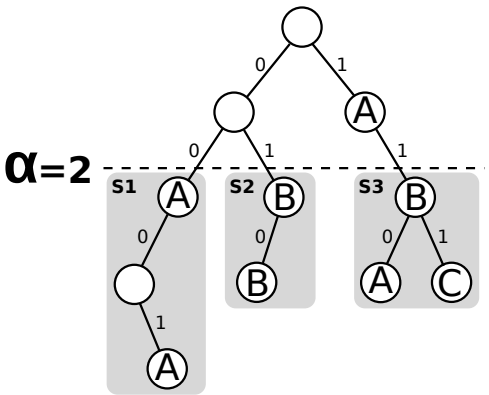
(1) Original FIB

LFA: Locality-aware FIB Aggregation

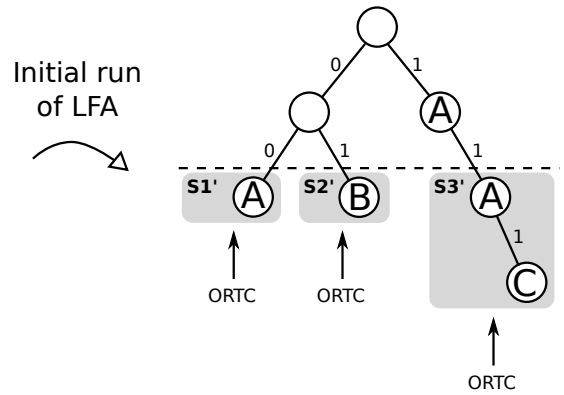


(1) Original FIB *(2) Snapshot*

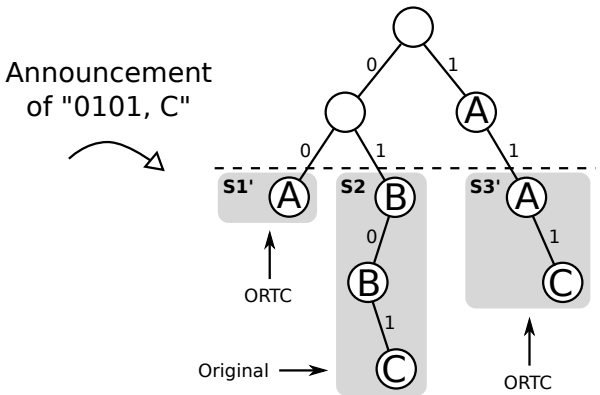
LFA: Locality-aware FIB Aggregation



(1) Original FIB

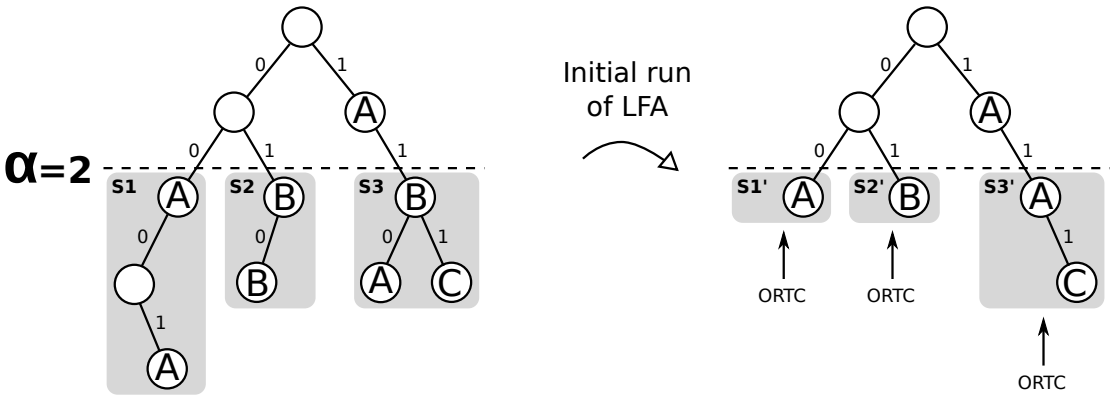


(2) Snapshot

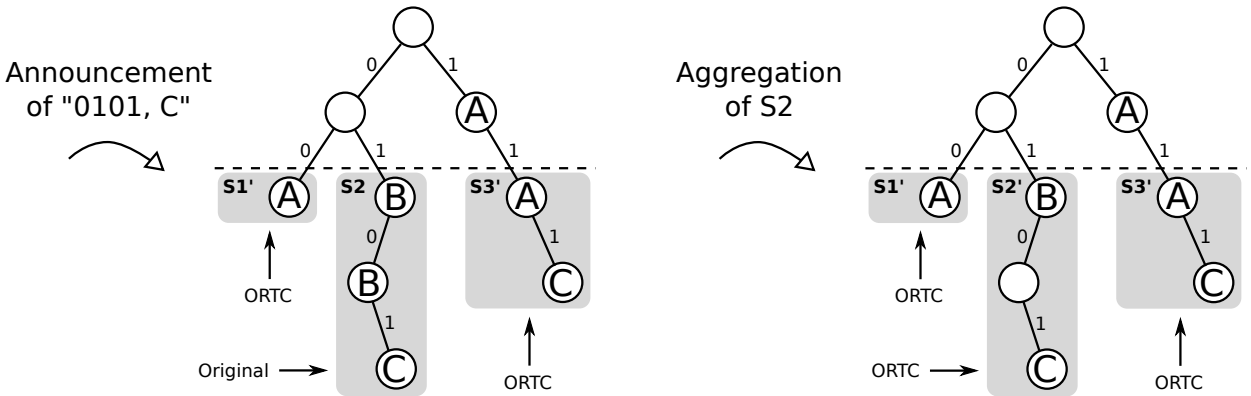


(3) After update

LFA: Locality-aware FIB Aggregation

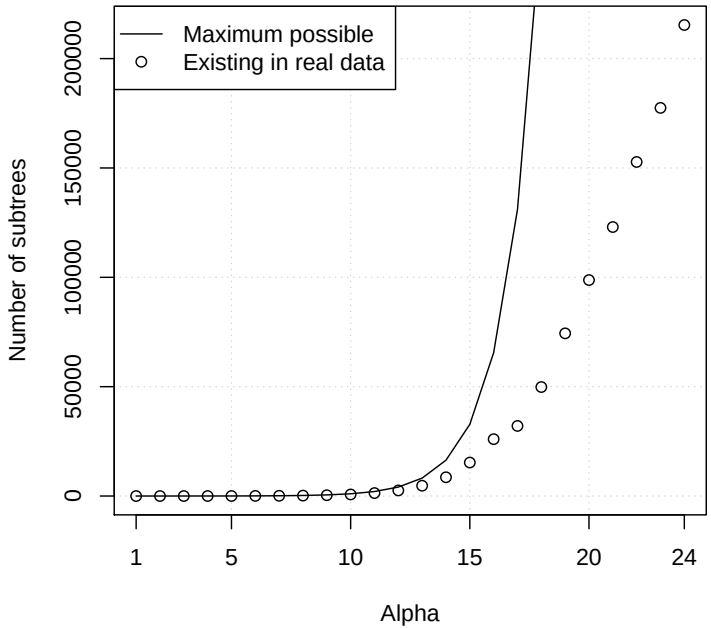


(1) Original FIB *(2) Snapshot*

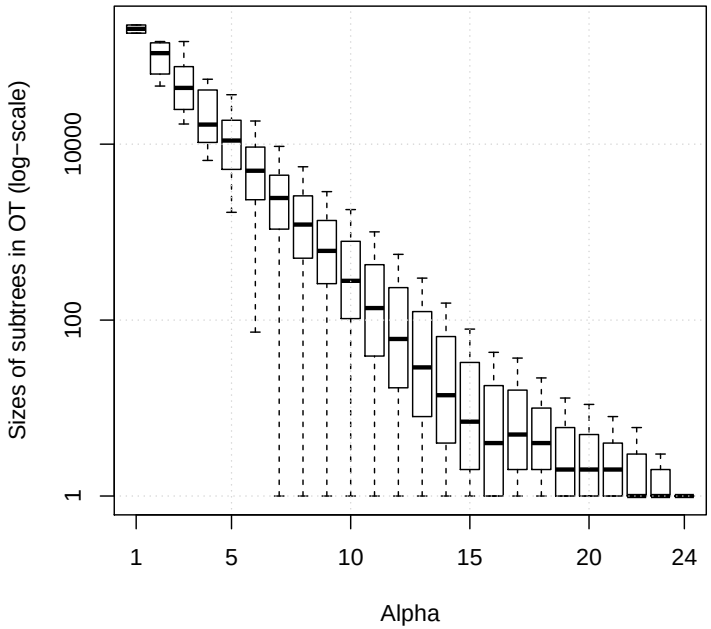


(3) After update *(4) After β seconds*

LFA: Results on table snapshot

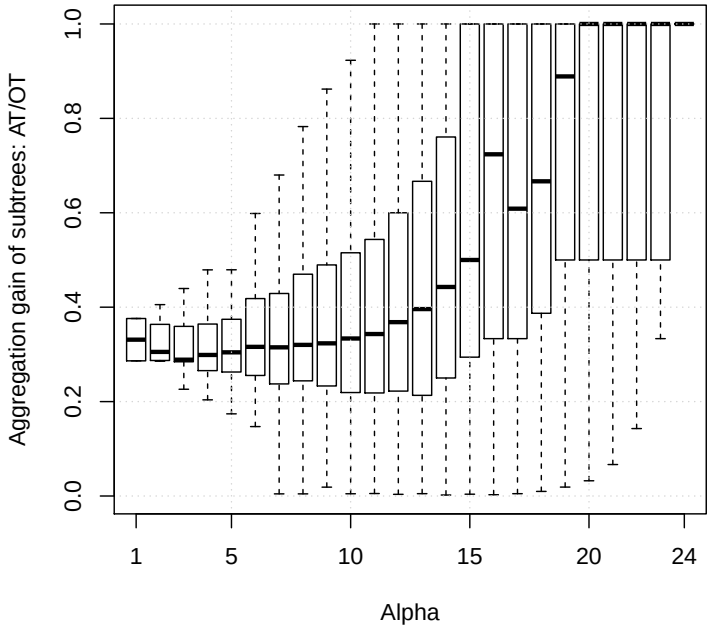


(a) Number of existing subtrees as a function of α.

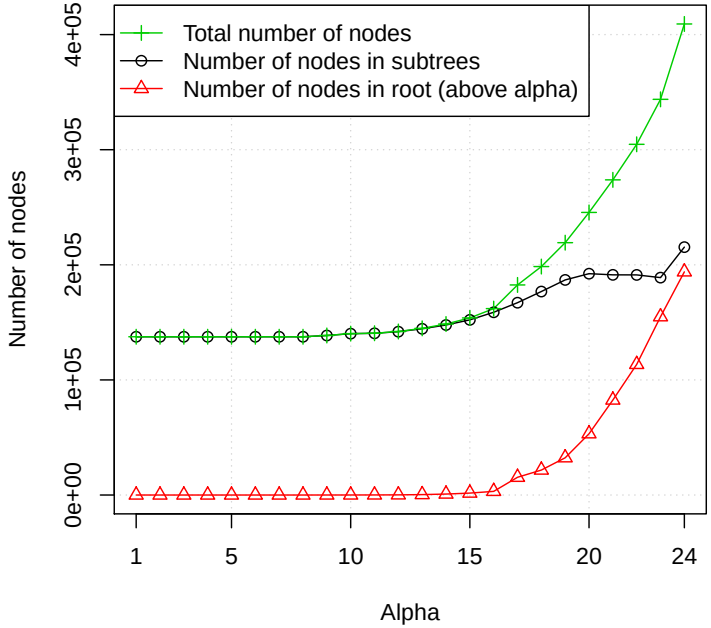


(b) Distribution of subtree sizes in OT as a function of α.

LFA: Results on table snapshot



(c) Per-subtree aggregation gain as a function of α.



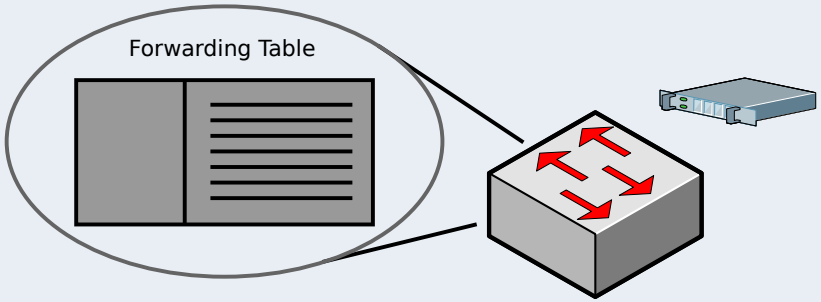
(d) Number of nodes in AT subtrees and root as a function of α.

Next steps

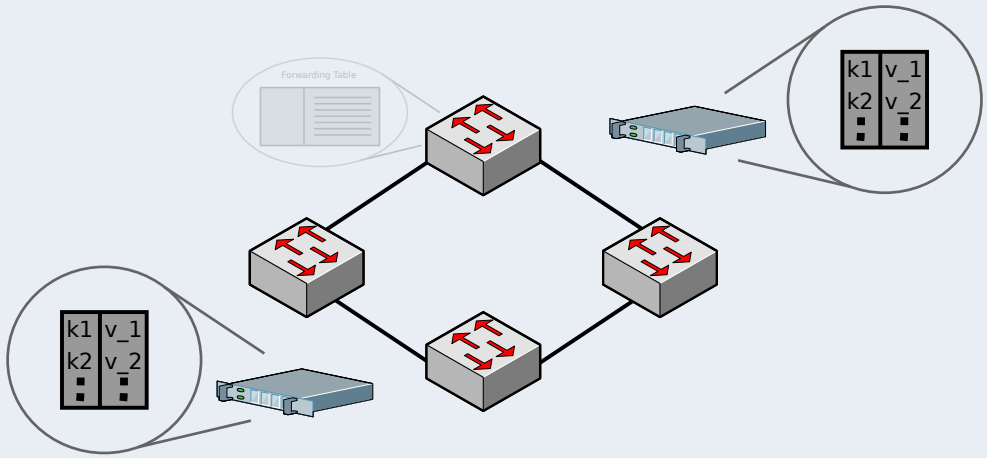
- Online experiments with LFA:
  - What fraction of subtrees is aggregated on average?
  - What is the impact of α and β?
- Impact of FIB aggregation on traffic offloading

Outline: From Micro to Macro

1: Locality-aware FIB Aggregation

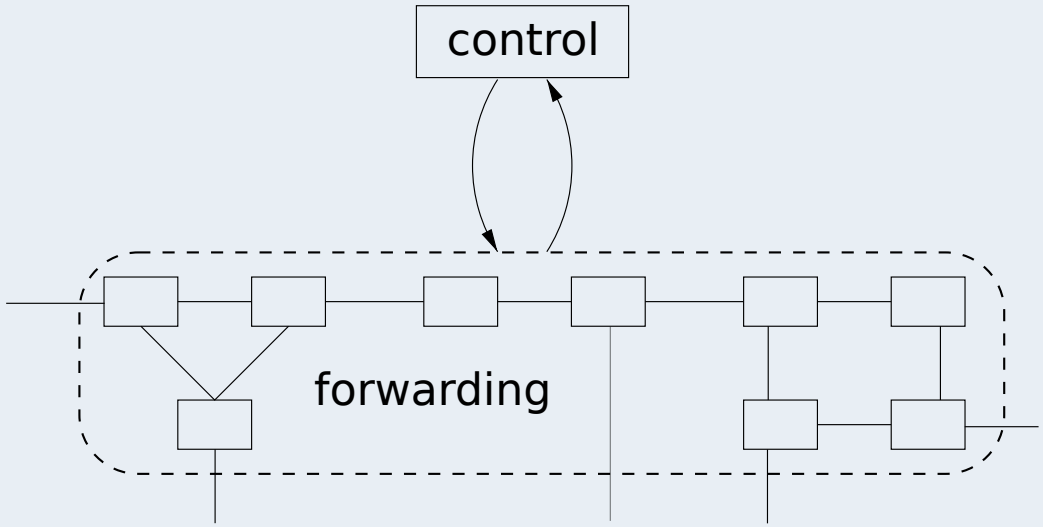


2: Control State Distribution Trade-offs



Motivation

A Key Motivation behind SDN: Control Logic Centralization

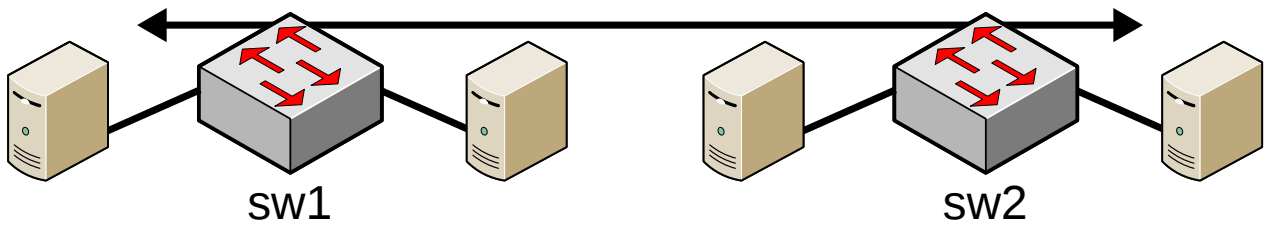


Benefits have been argued [1]:

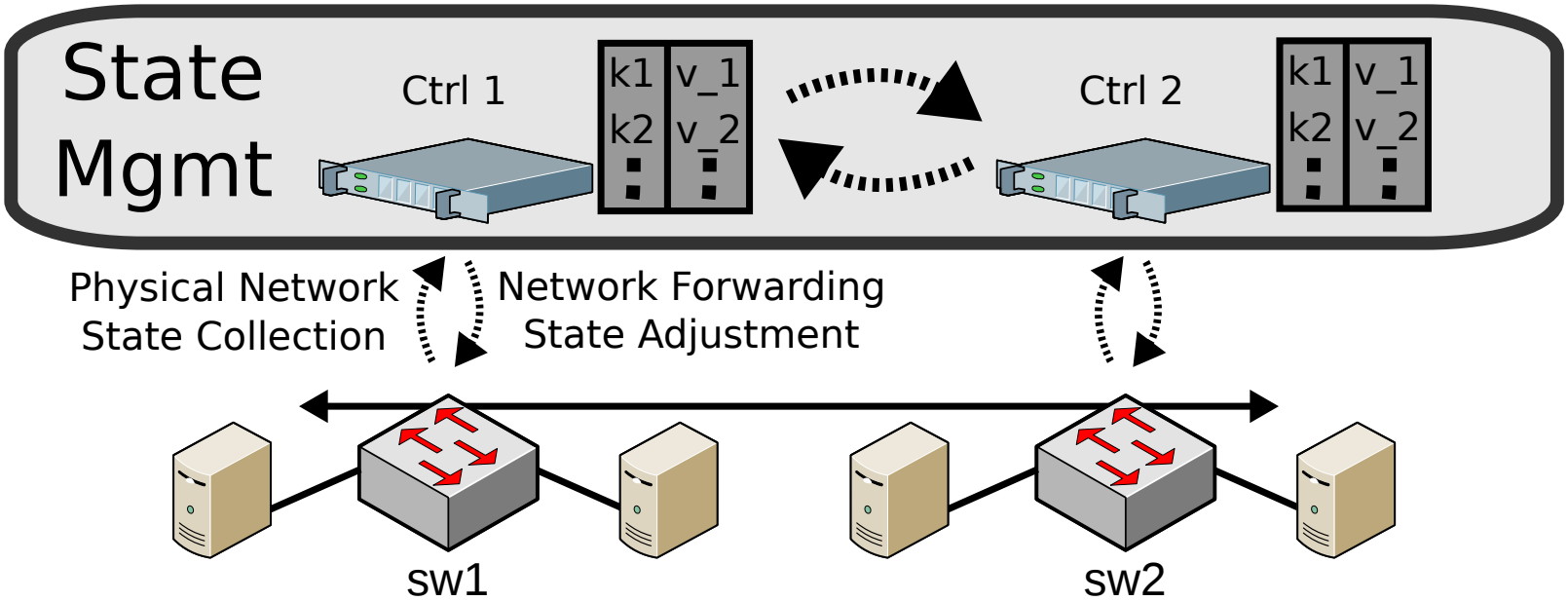
- More Deterministic Behavior
- Better Correctness Guarantees

[1] Source: Urs Hölzle, Google, ONF Summit 2012

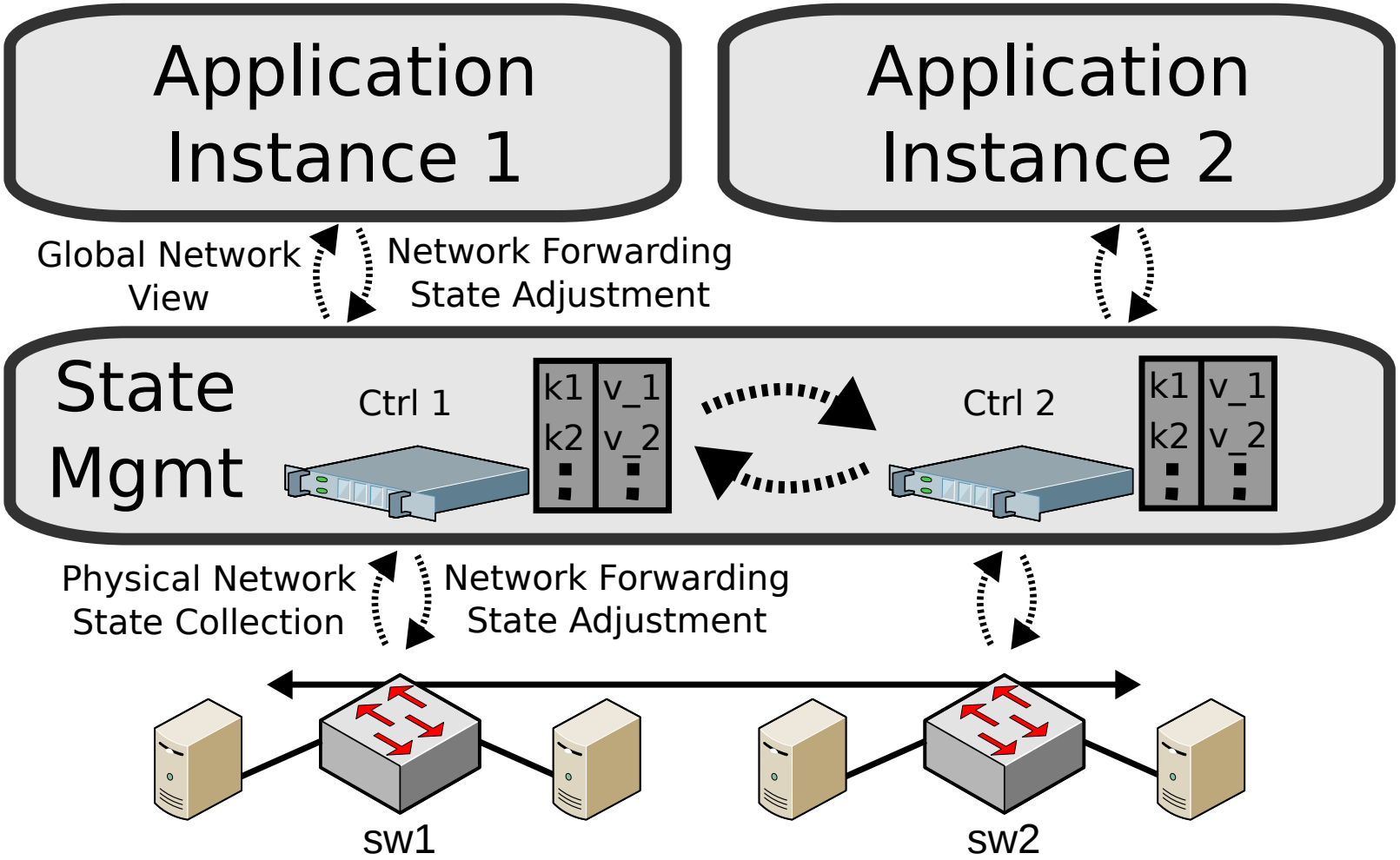
Problem – Logically Centralized, Physically Distributed



Problem – Logically Centralized, Physically Distributed



Problem – Logically Centralized, Physically Distributed



Problem – Logically Centralized, Physically Distributed

Design Choices

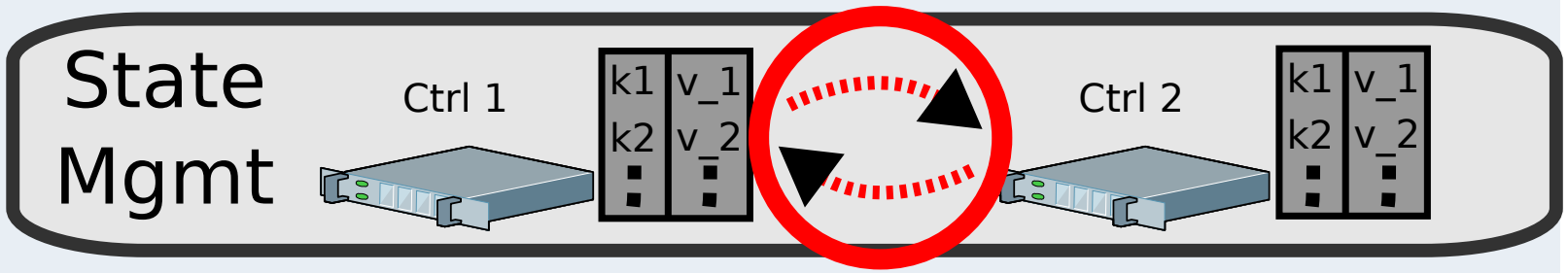
- We may choose among **consistency models**
- Safety Properties: Loop-free, not oversubscribed...
- Liveness Properties: Forwarding updates in timely fashion

Consequences for centralized control logic design

- Impact of **inconsistent** distributed state
- Impact of **application state awareness** on network performance

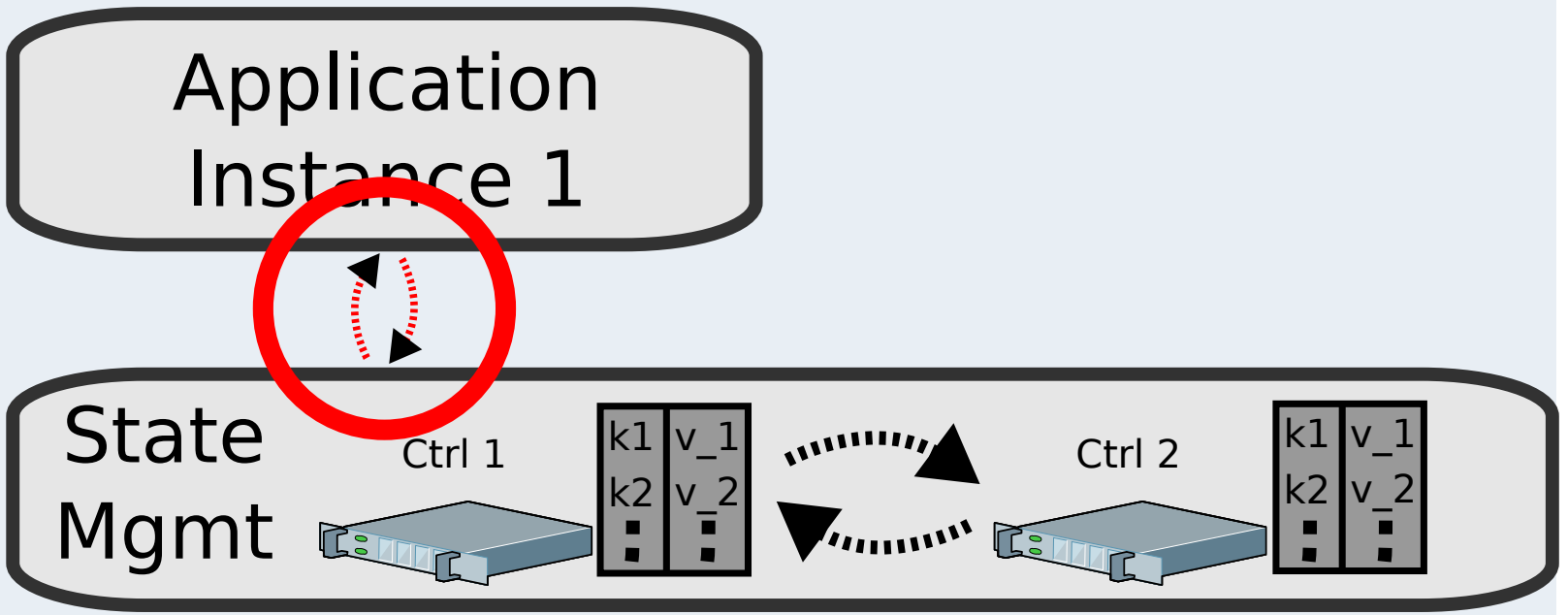
Approach – Distributed State Trade-offs

Trade-off 1: Consistency Model vs. Optimality



Approach – Distributed State Trade-offs

Trade-off 2: Application State Awareness vs. Inconsistency Robustness

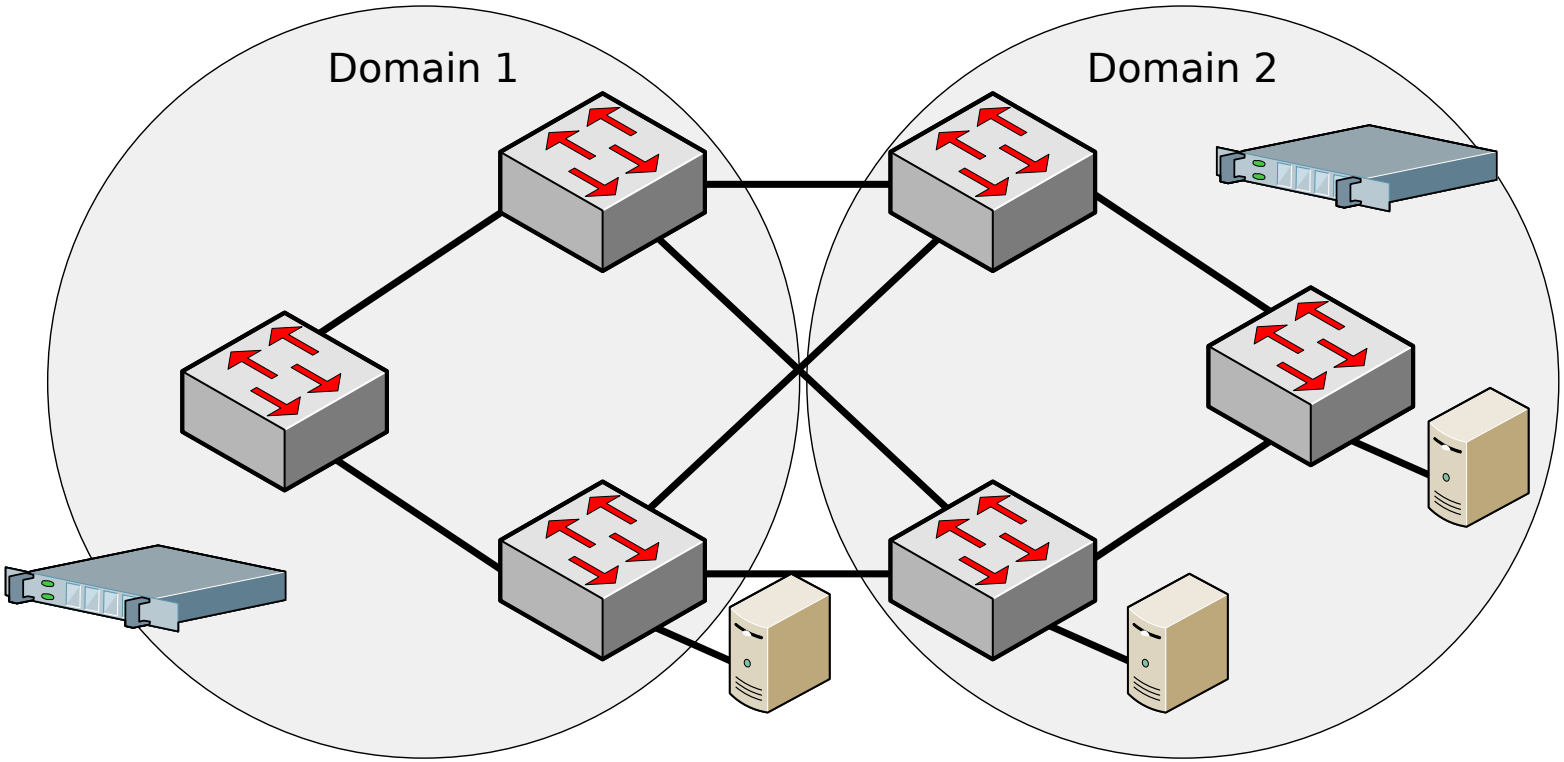


Simulation: Sensitivity Experiment

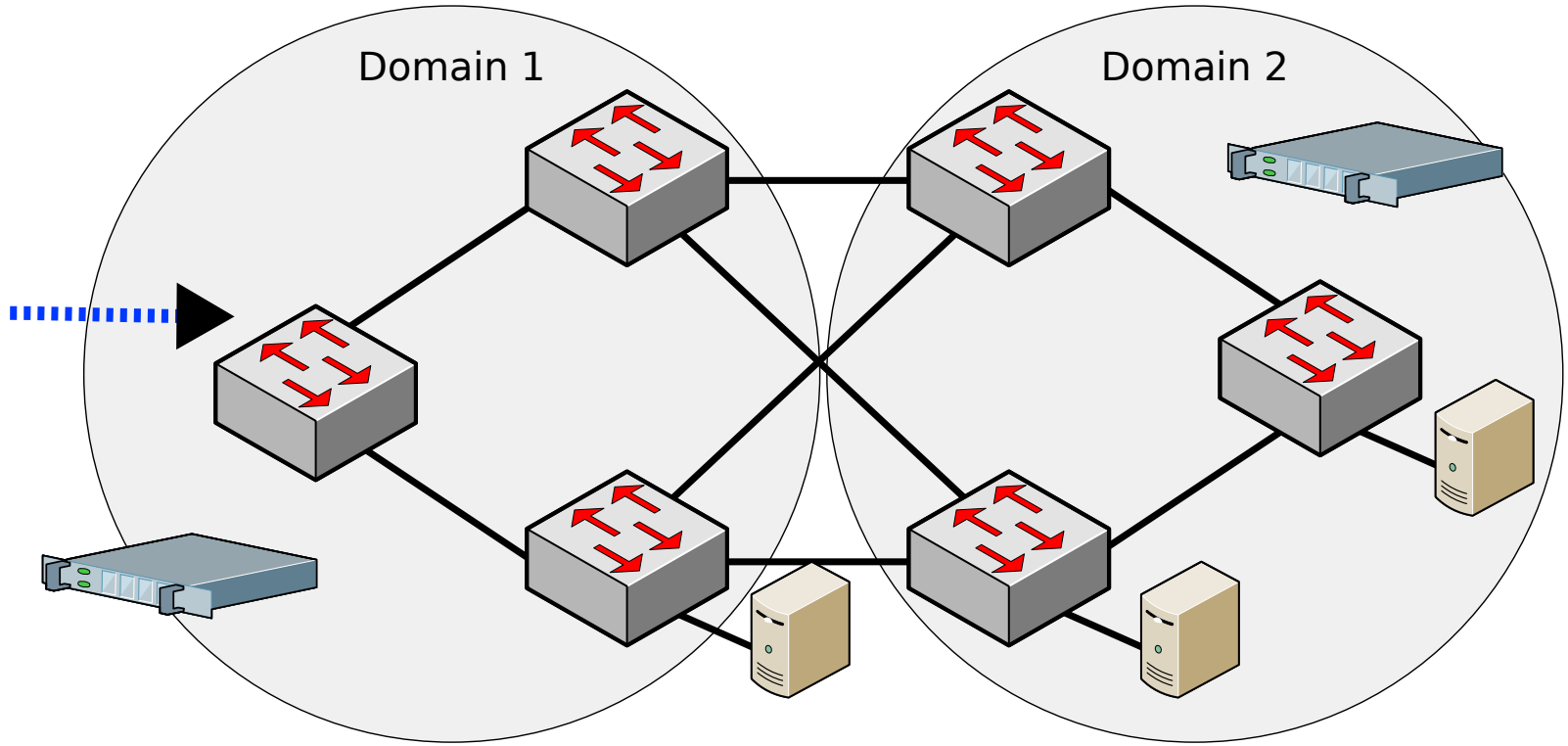
We examine Trade-off 1 and 2 subject to:

1. SDN Load Balancer Application
2. Topology and Controller Domains
3. Traffic Workload
4. Metric: Link Utilization RMSE

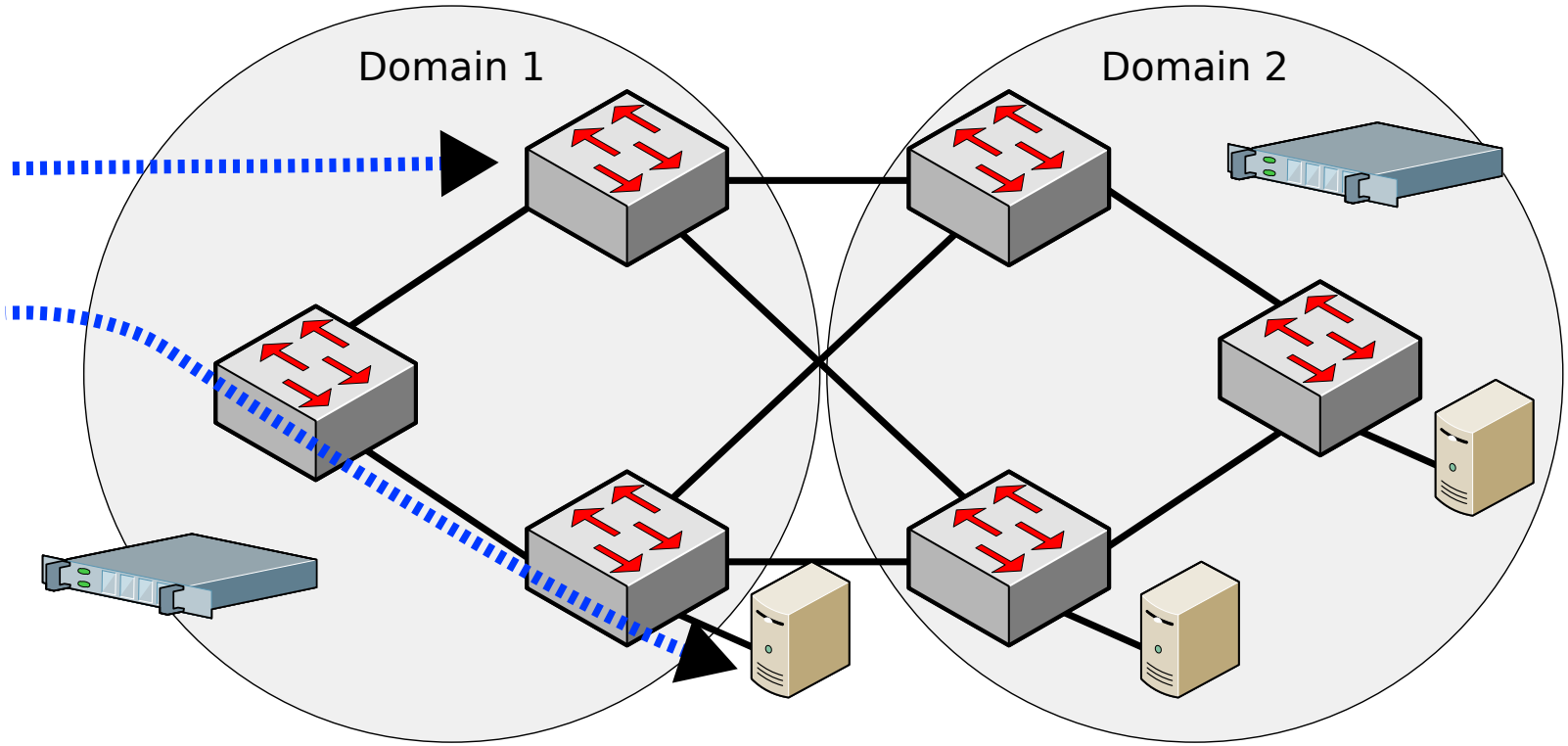
Application – Service + Network Load Balancer



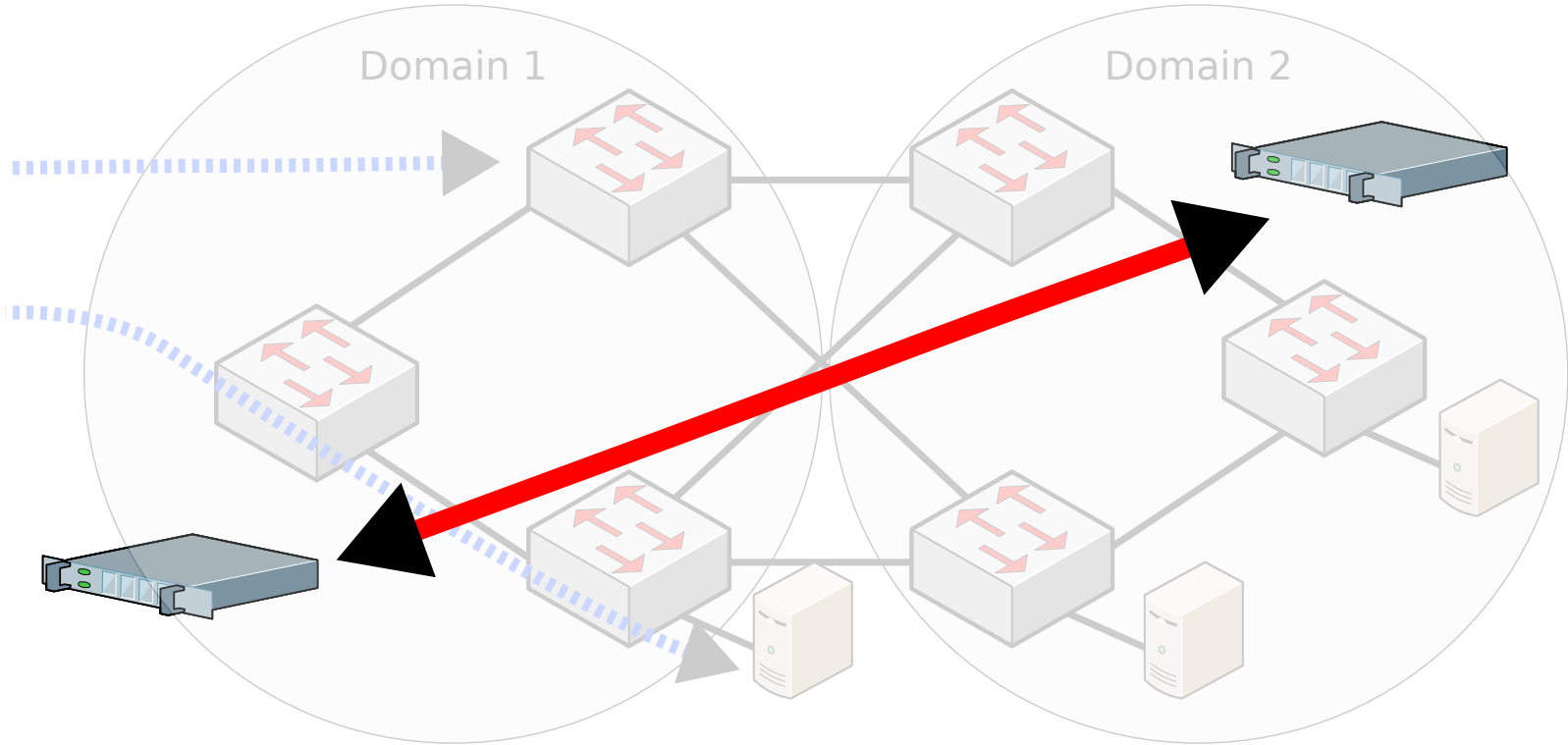
Application – Service + Network Load Balancer



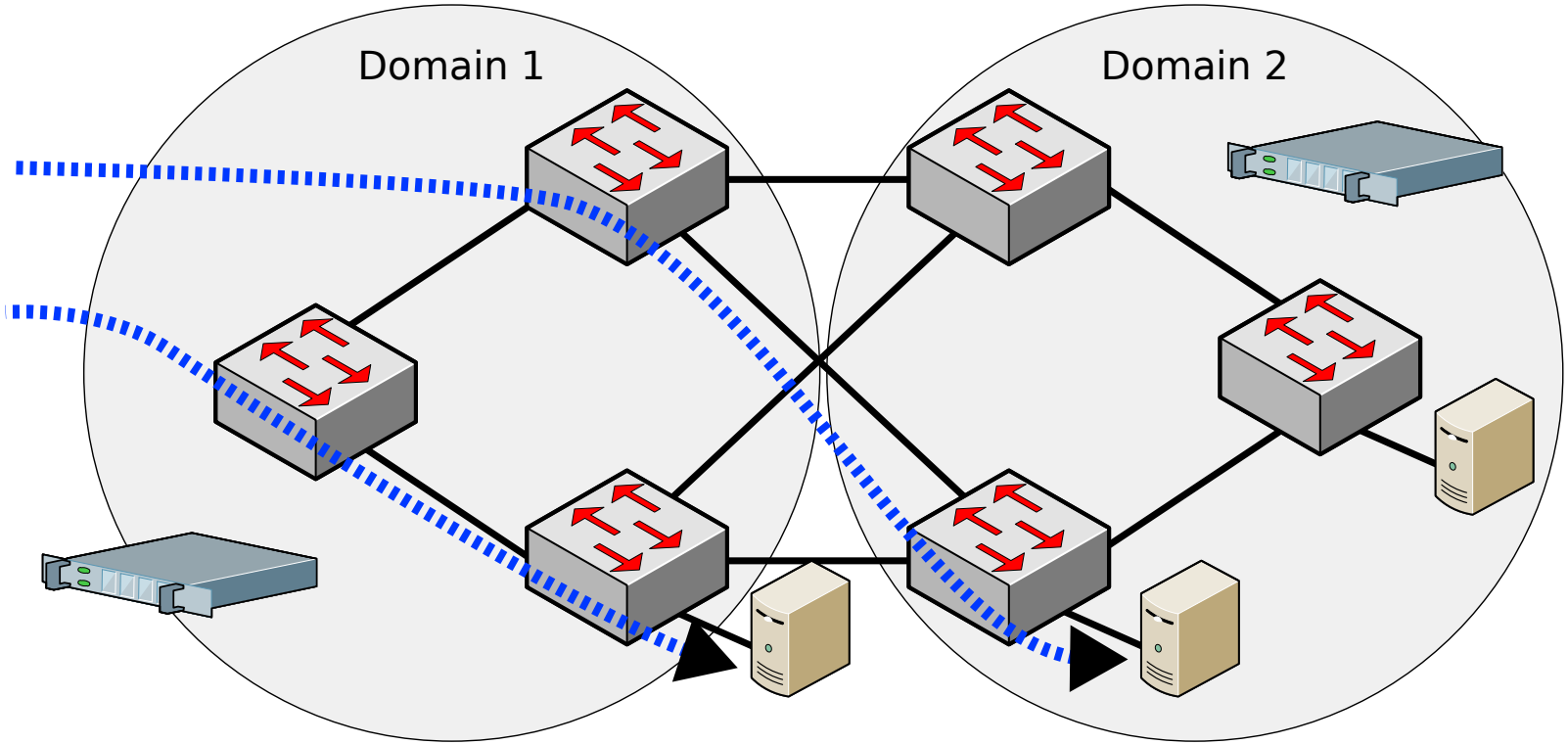
Application – Service + Network Load Balancer



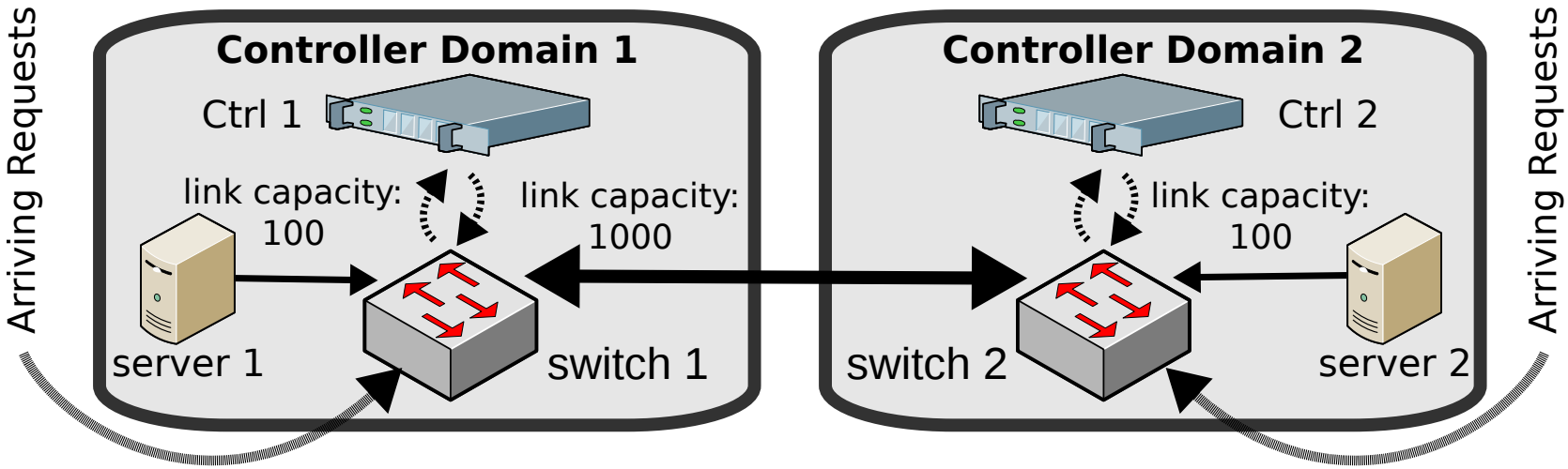
Application – Service + Network Load Balancer



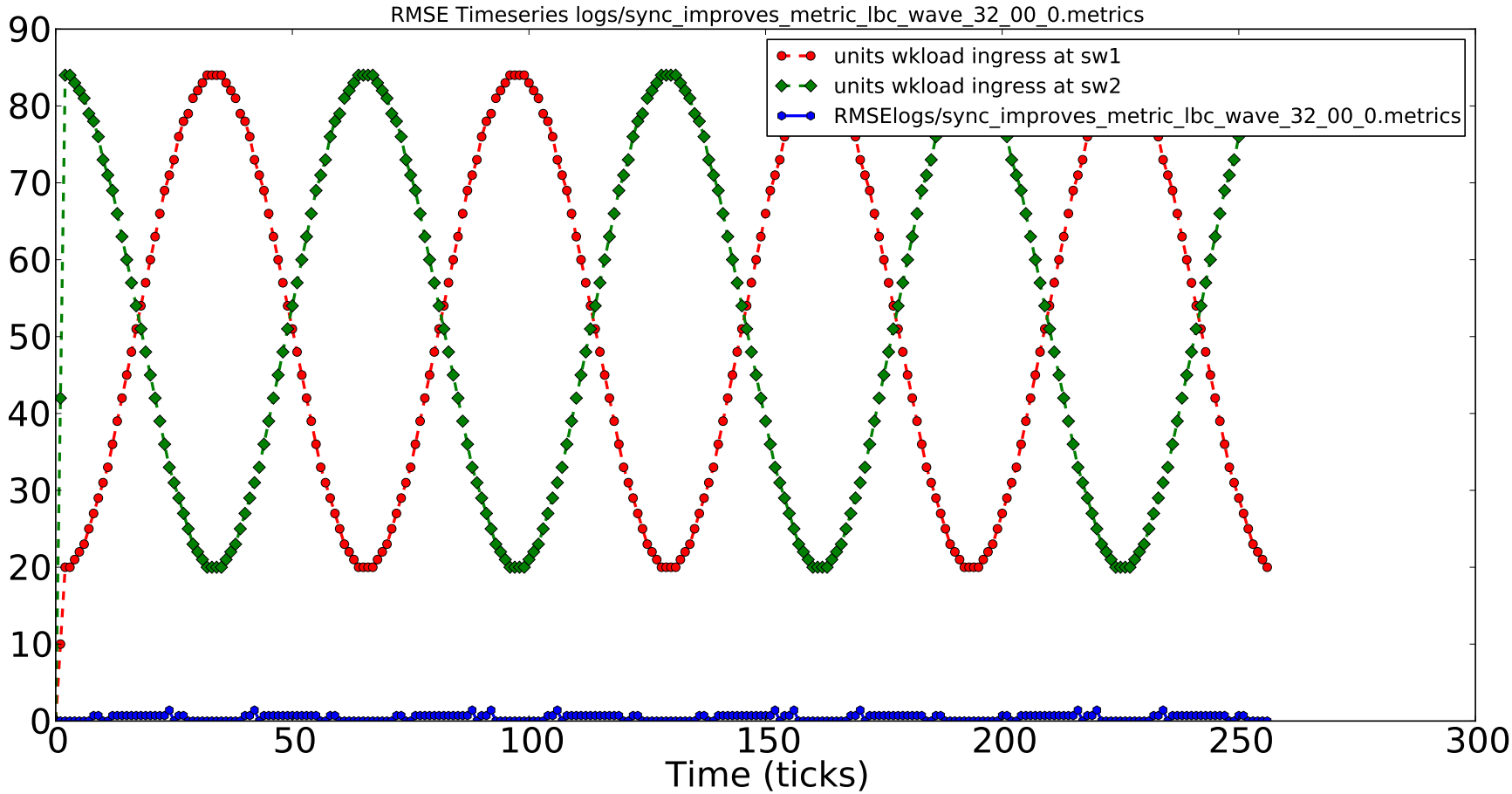
Application – Service + Network Load Balancer



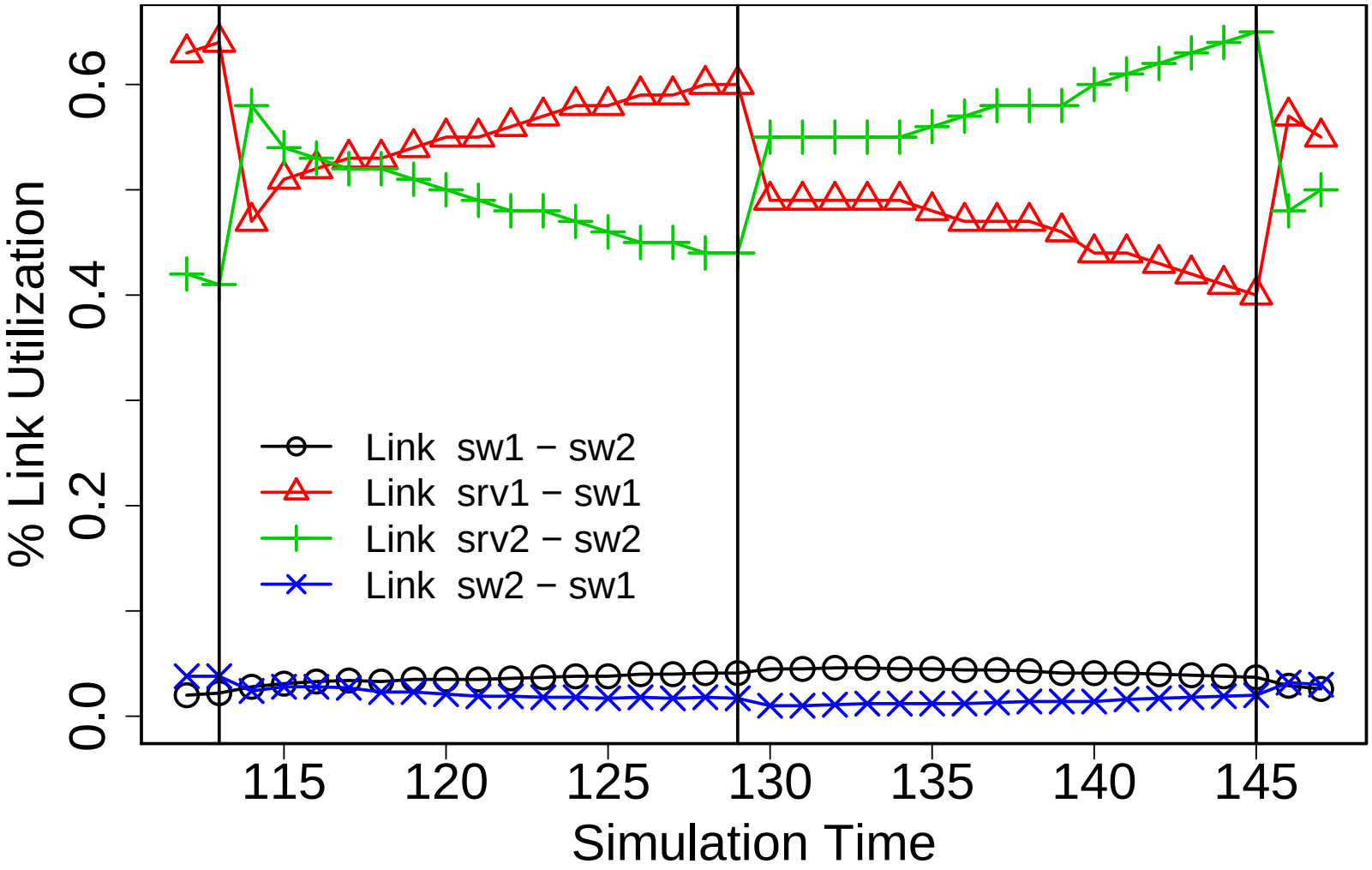
Simulation – Topology



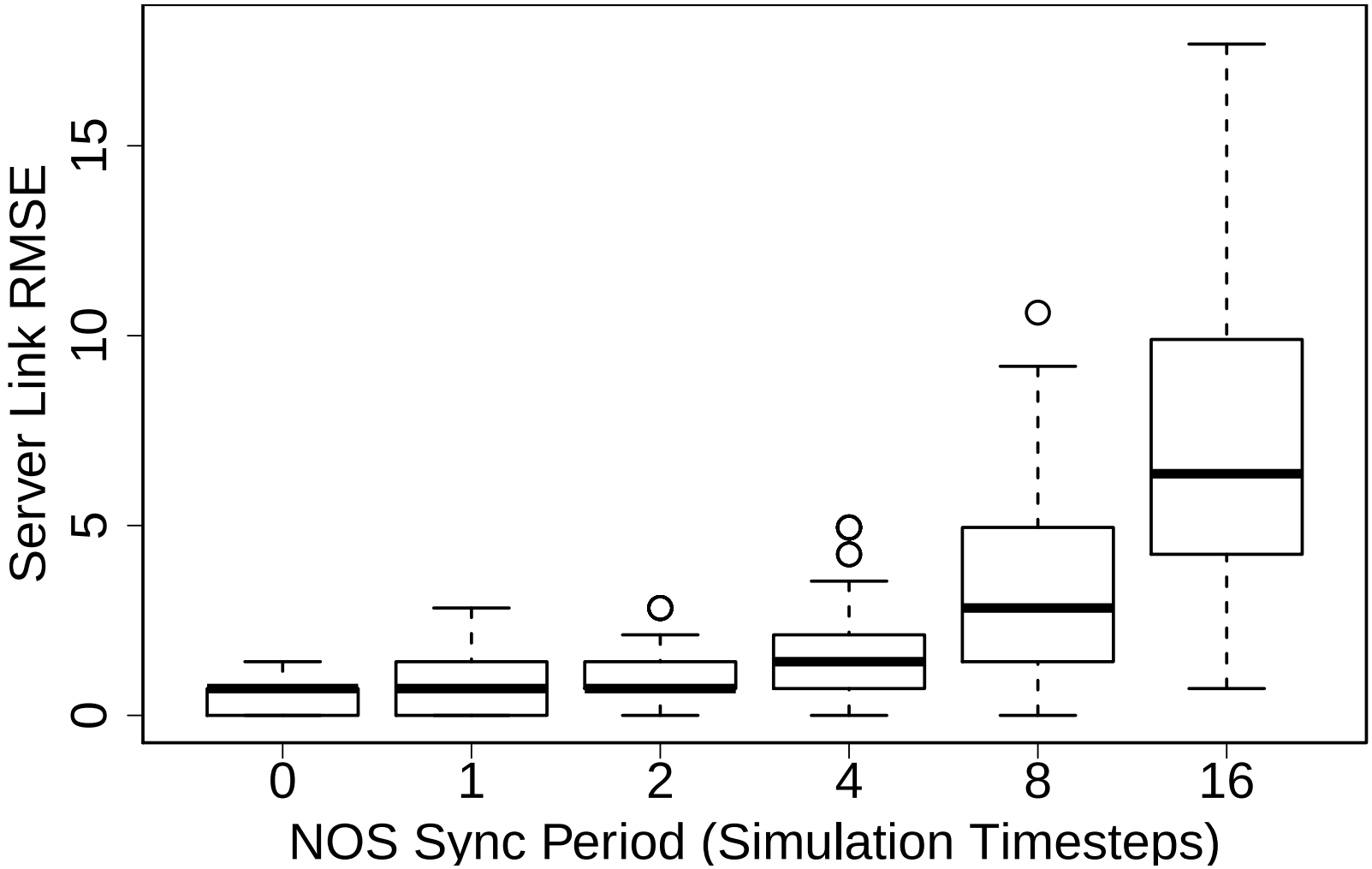
Simulation – Workload: Simple Wave



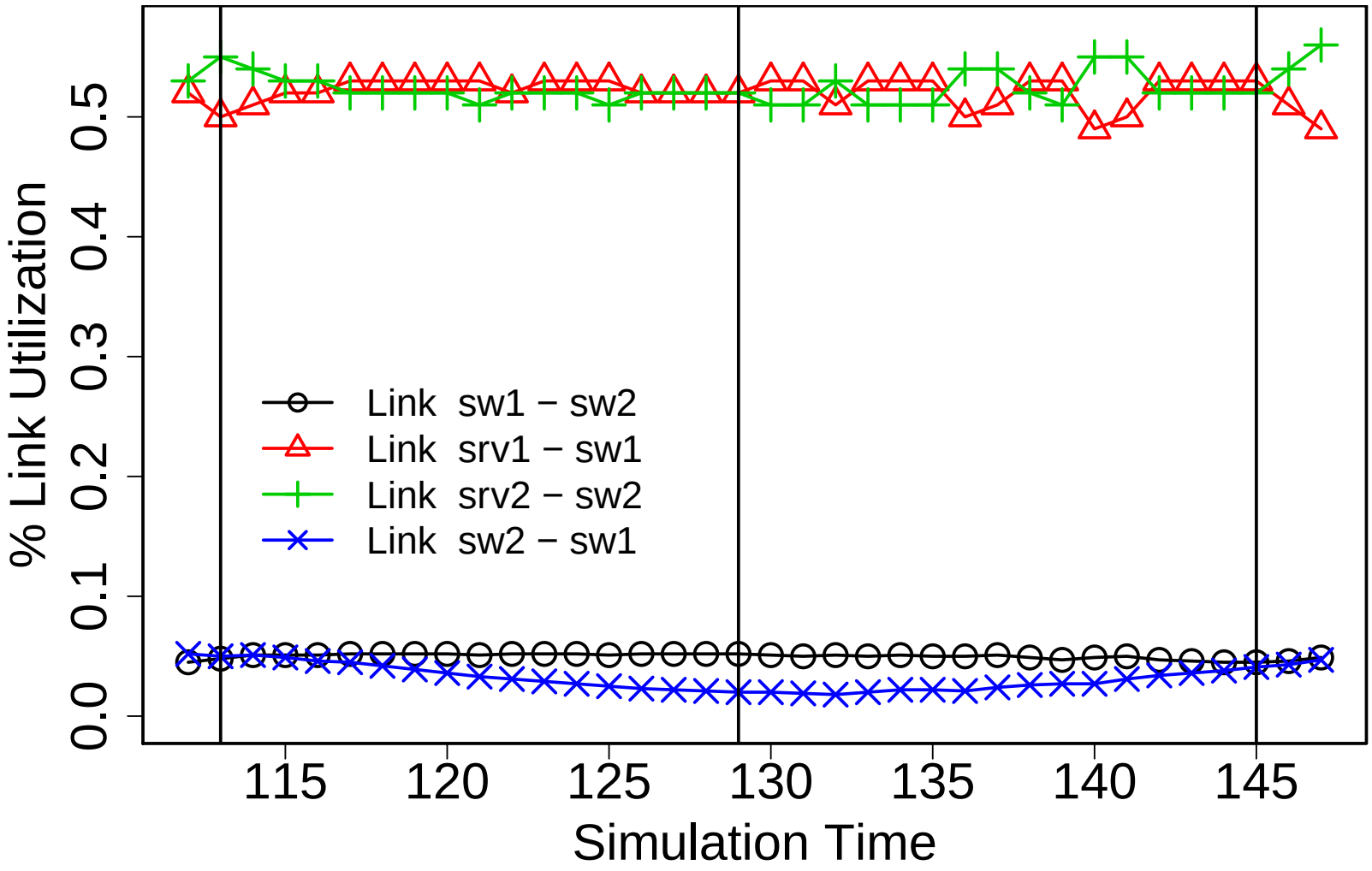
Results – Trade-off: Staleness vs. Optimality



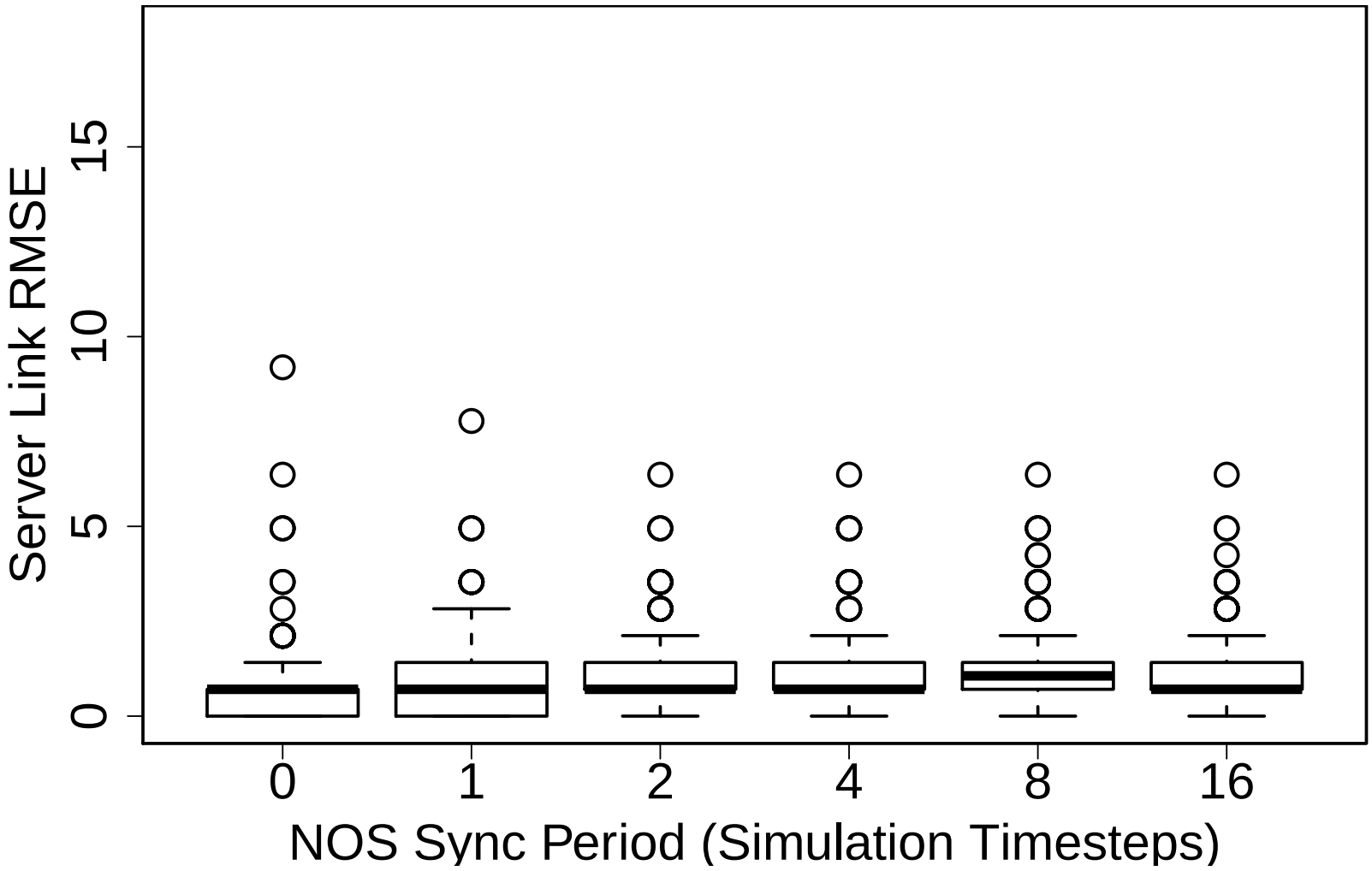
Results – Trade-off: Staleness vs. Optimality



Results – Trade-off: Application State Awareness vs. Inconsistency Robustness



Results – Trade-off: Application State Awareness vs. Inconsistency Robustness



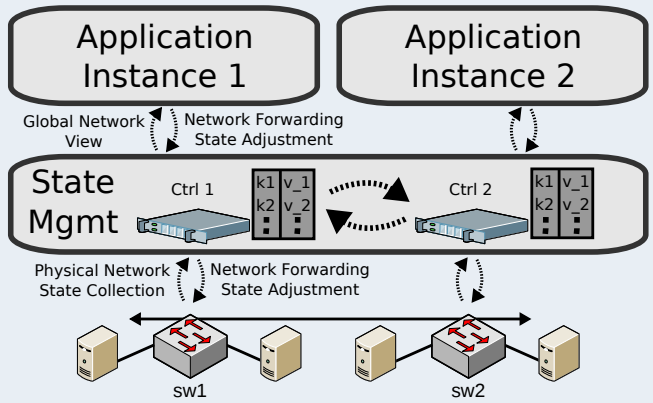
Summary

Trade-off Characterization

Identified SDN state exchange trade-offs

- Control State Consistency vs. Application Optimality
- Application Complexity vs. Robustness to Inconsistency

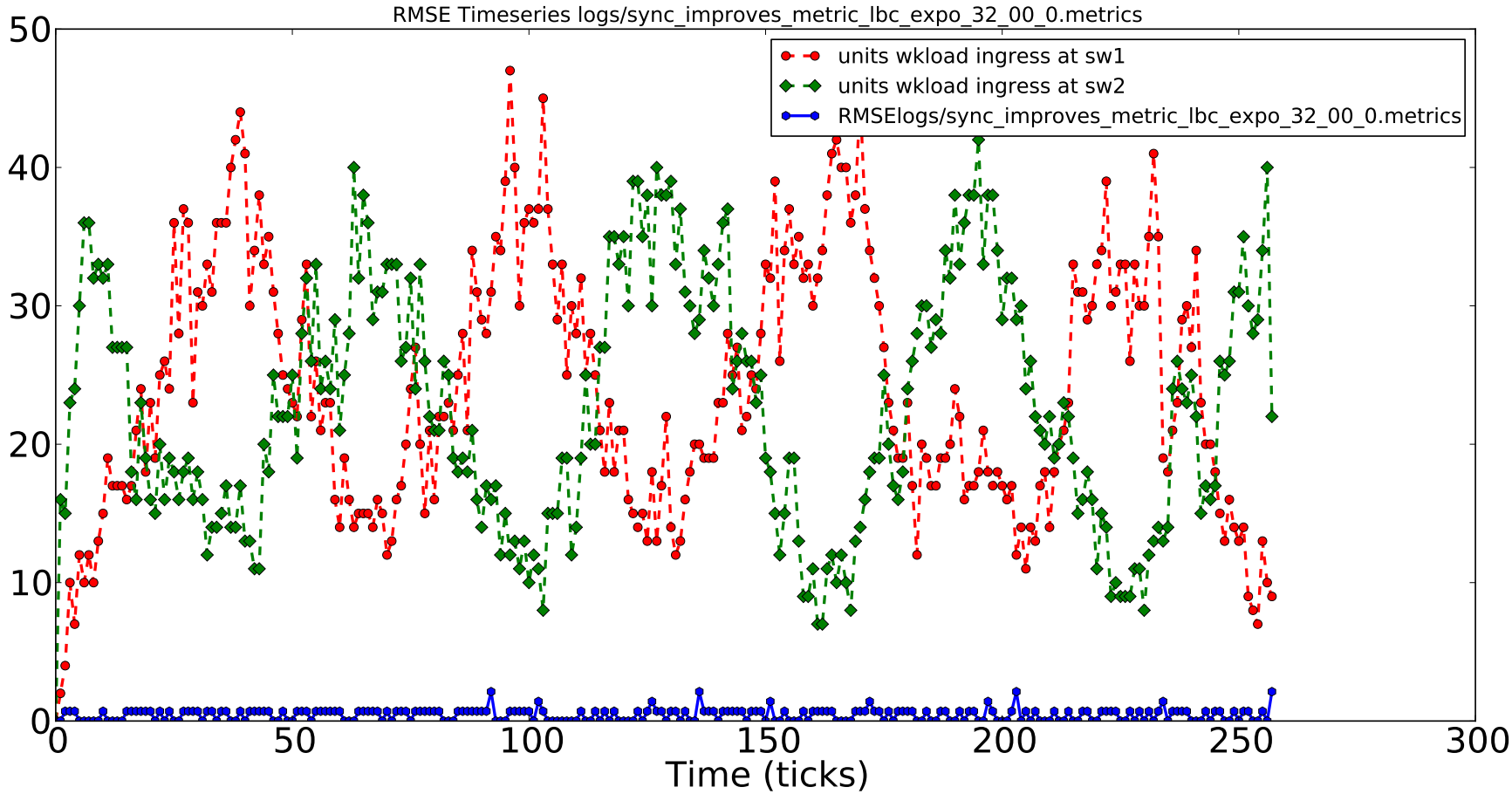
Simulation: sensitivity study



Next Steps

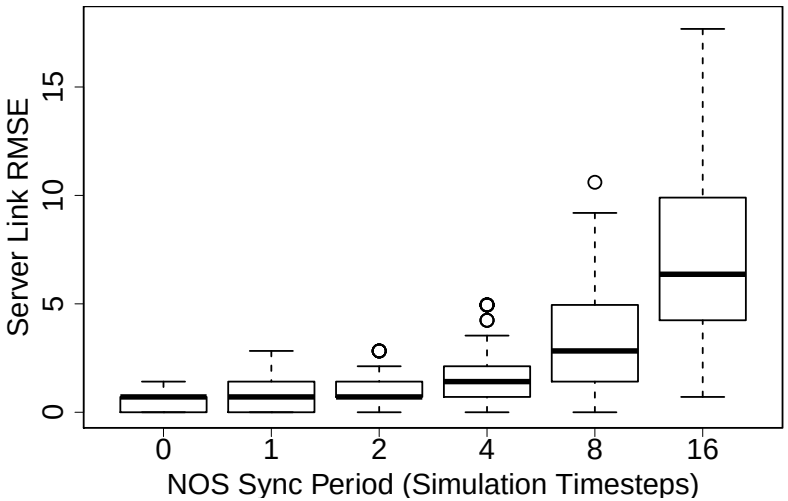
- Towards Improved Quantitative Approaches: TE Measurement
- More Diverse Applications, Topologies, Objectives: Where to choose which consistency model?

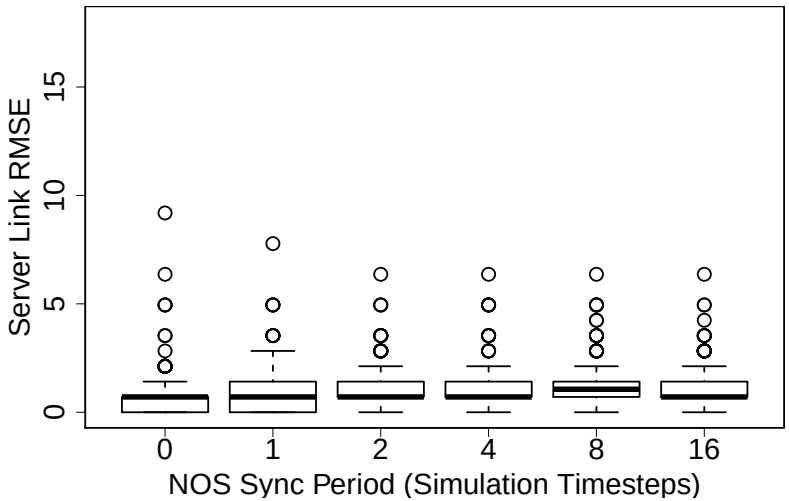
Simulation – Workload: Self-similar



Results – Trade-off 2

Workload 1: Sin wave “diurnal” flow arrivals, uniform duration





Results – Trade-off 2

Workload 2: Exponentially distributed flow arrival times
Weibull distributed flow durations