Robust fault-recovery in Software-Defined Networks

IBSDN: IGP as a Backup in SDN

June 2014 - Olivier Tilmans

Agenda

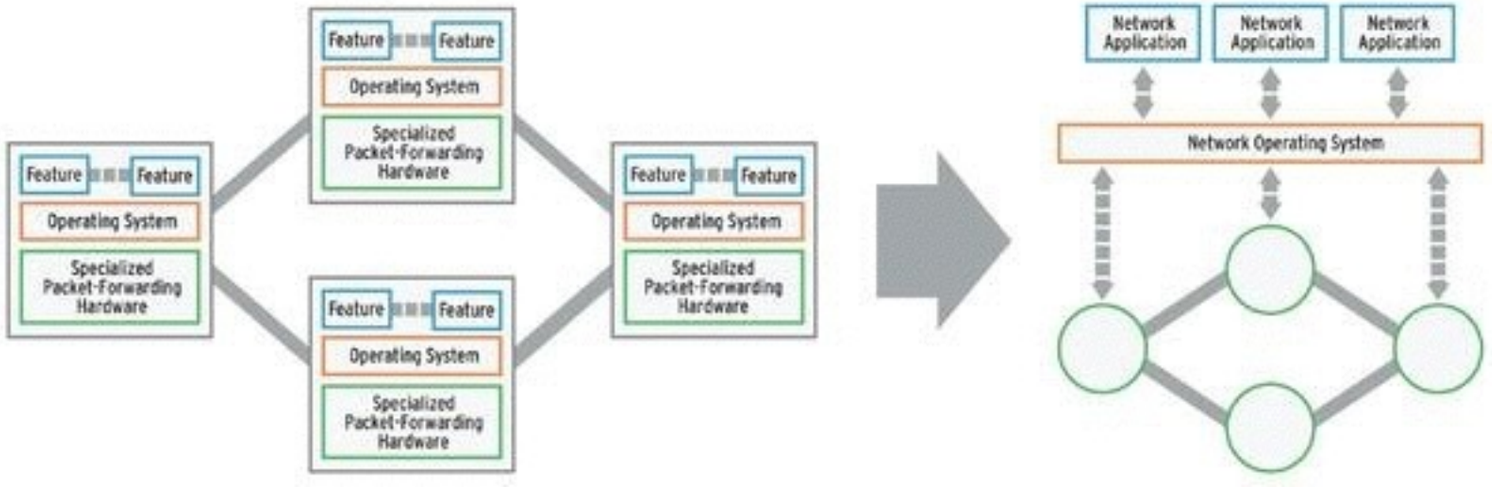
1. Software-Defined Networking
2. IBSDN
3. Evaluation
4. Discussion
5. Summary

Agenda

1. Software-Defined Networking
2. IBSDN
3. Evaluation
4. Discussion
5. Summary

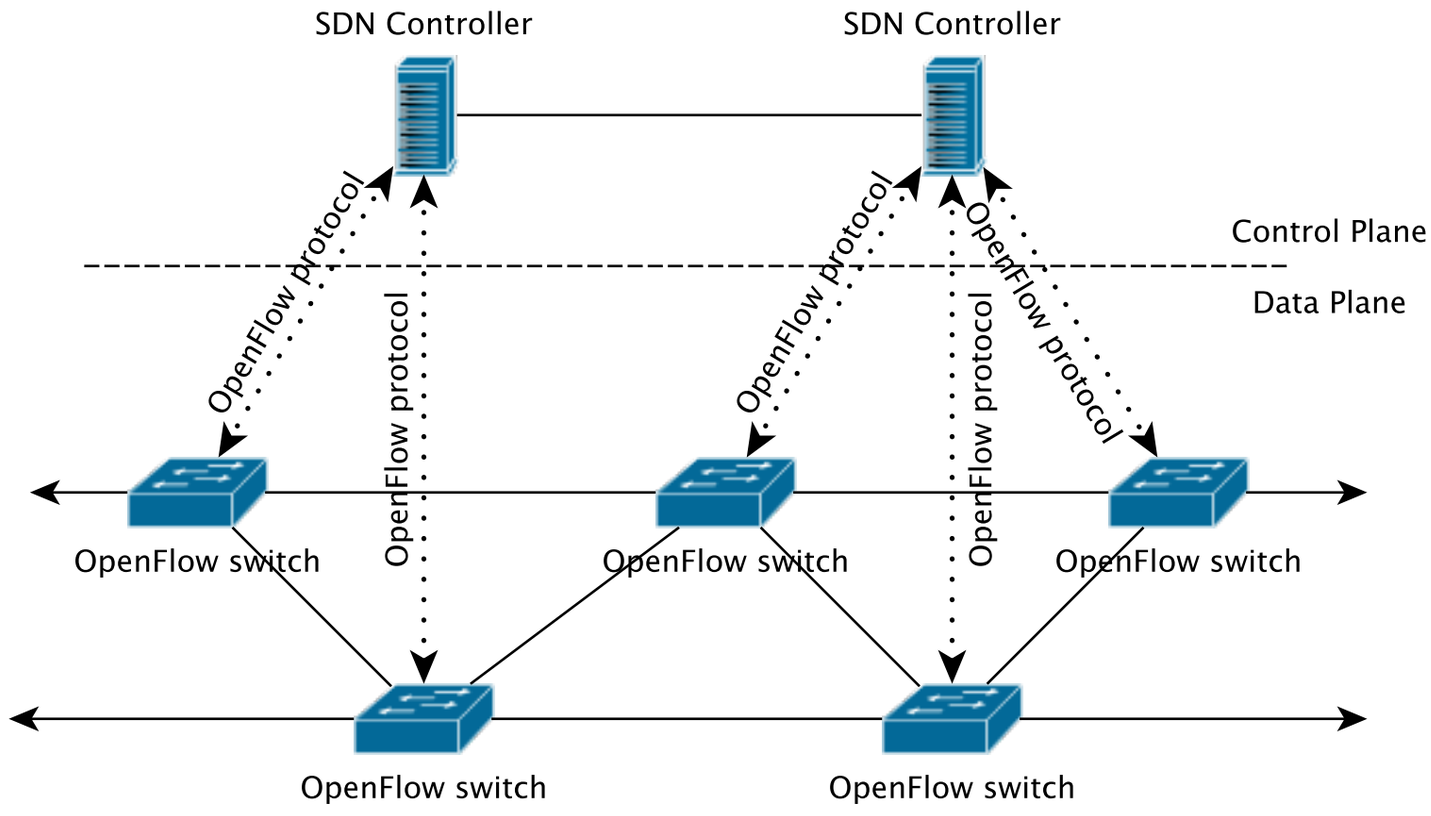
Software-Defined Networking

- Decouples the control-plane from the data-plane
- Brings programmability in networks



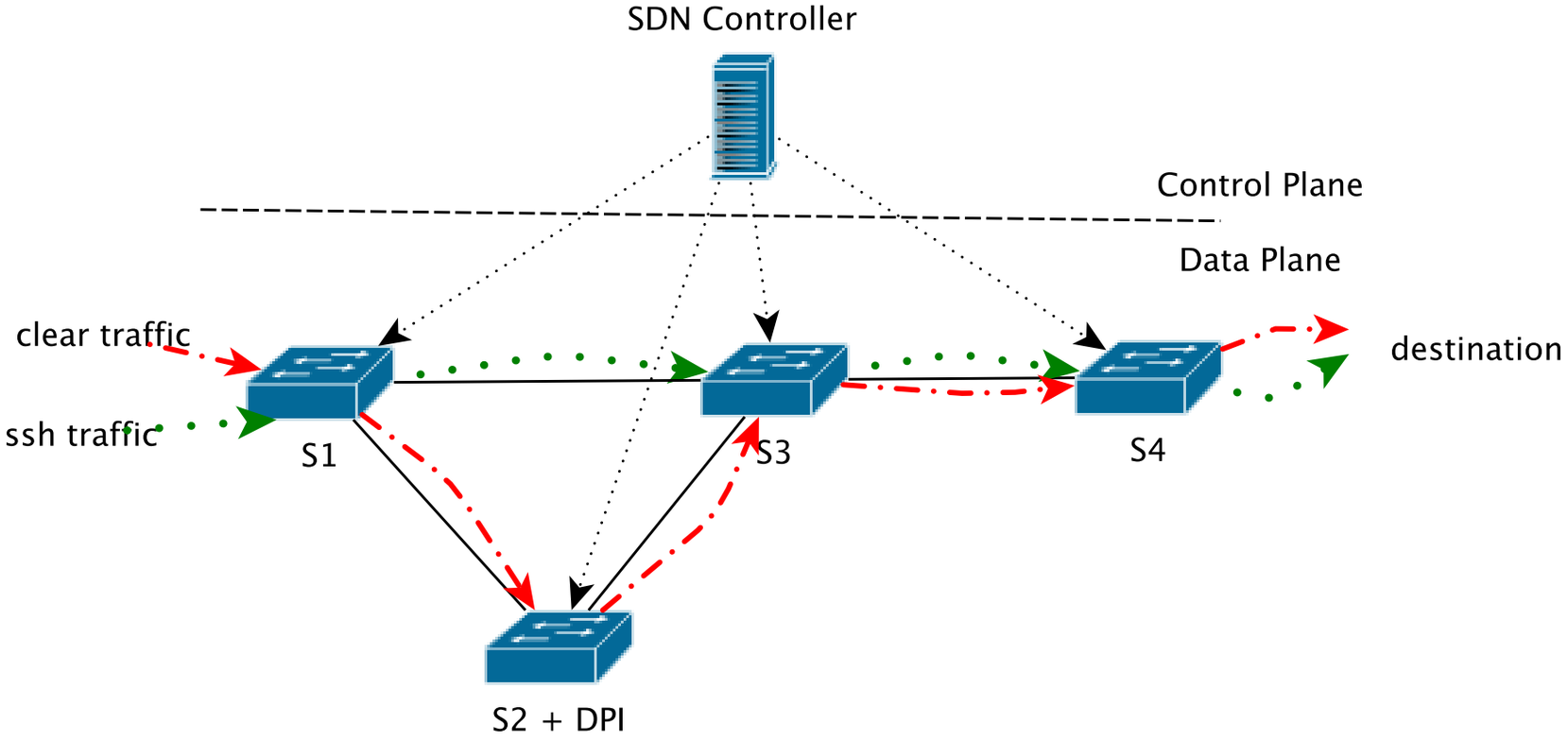
OpenFlow concepts

- Well-defined protocol and switch specifications



OpenFlow concepts (cont'd)

- Programmability via flow tables setup by the controller



Challenges

- Of the data-plane
  - Scalability

Challenges

- Of the data-plane
  - Scalability
- Of the controller
  - Consistency
  - Correctness
  - Capacity

Challenges

- Of the data-plane
  - Scalability
- Of the controller
  - Consistency
  - Correctness
  - Capacity
- Robustness

Impact of failures

- 3 sources of failures

Impact of failures

- 3 sources of failures
- Handling failures:
  - Reactively
  - Proactively

Agenda

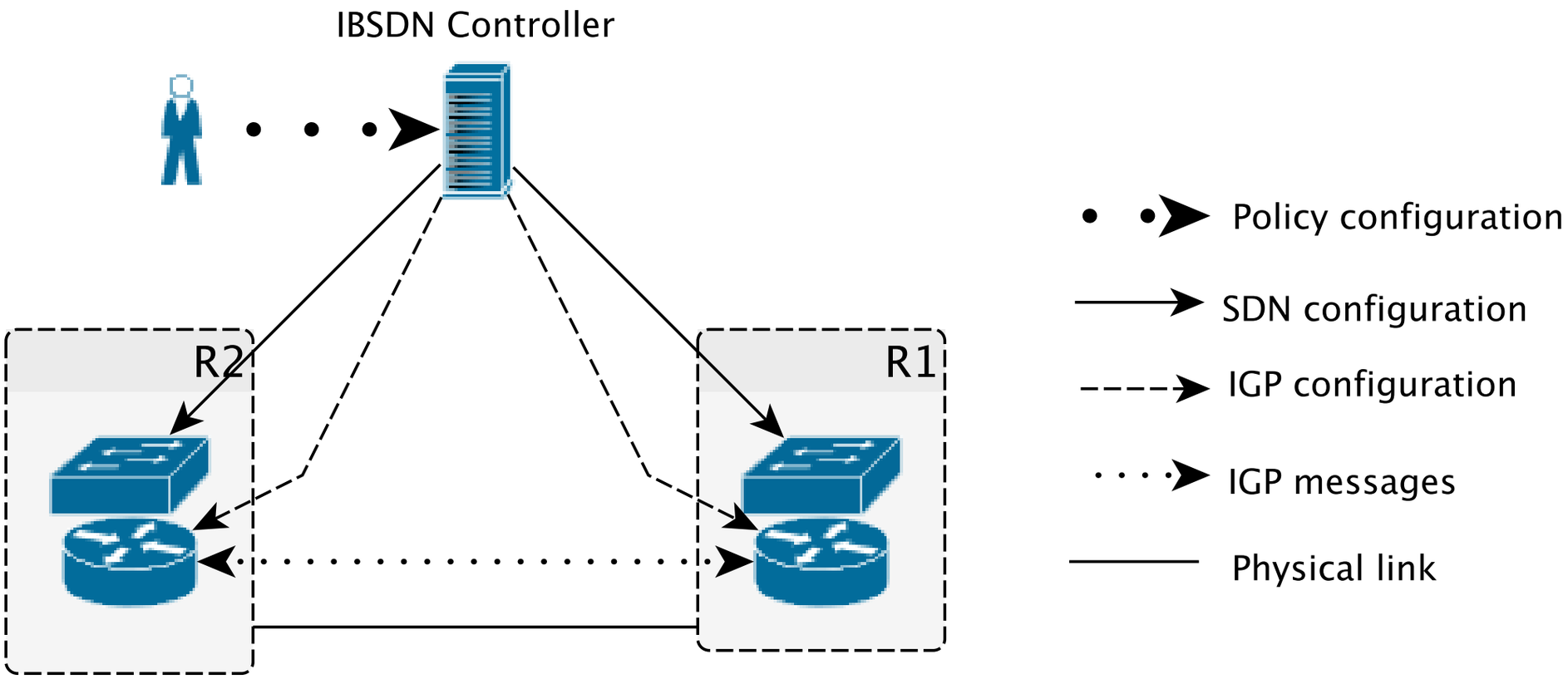
1. Software-Defined Networking
2. IBSDN
3. Evaluation
4. Discussion
5. Summary

Motivation

- Existing recovery schemes have inherent limitations

- Deployment of SDN in service-provider networks will co-exist with legacy hardware

Architecture

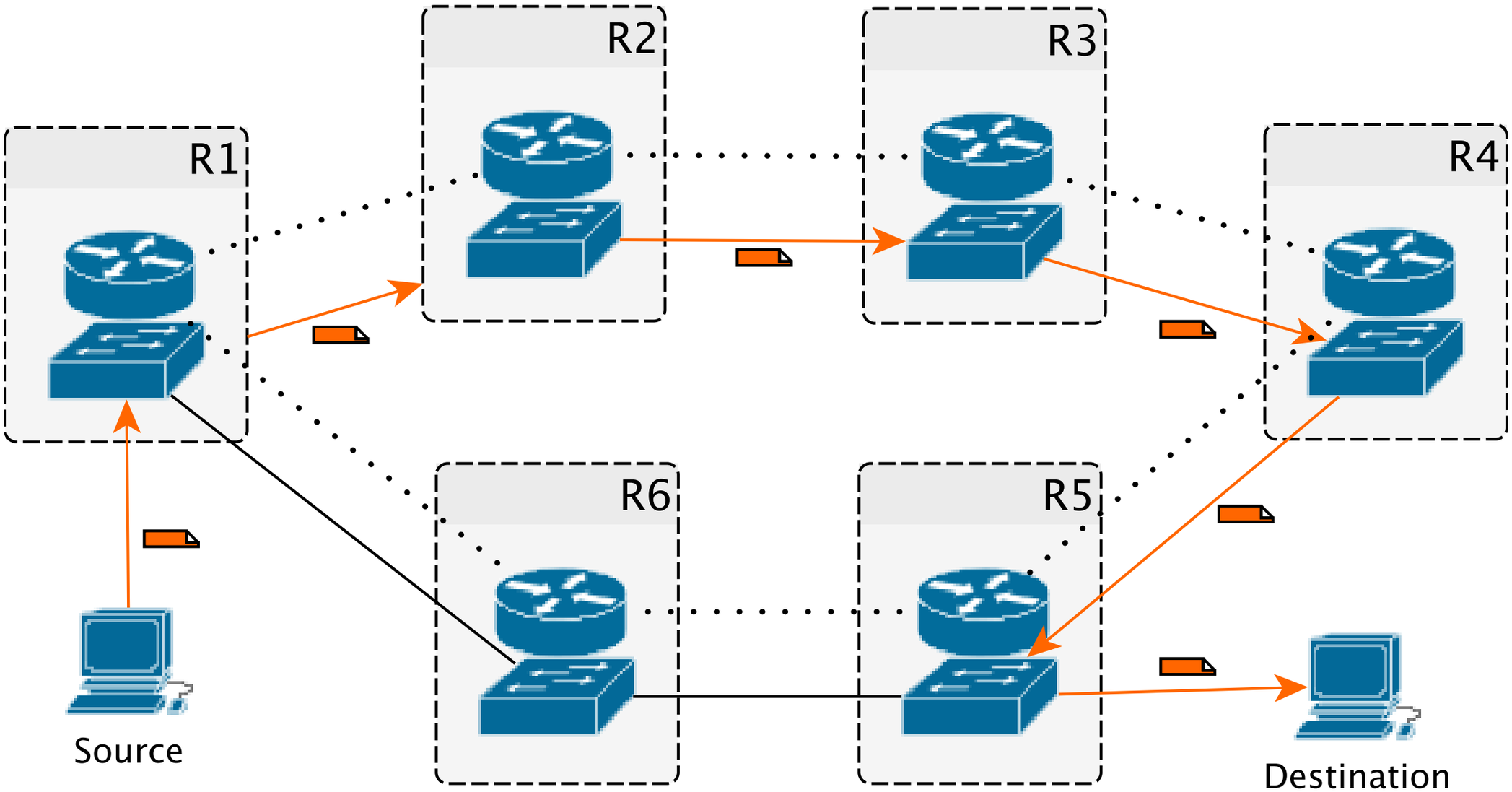


Requirements

1. Primary rules
2. Next-hop control rules
3. IGP-path control rule
4. Identifying IGP-forwarded packets

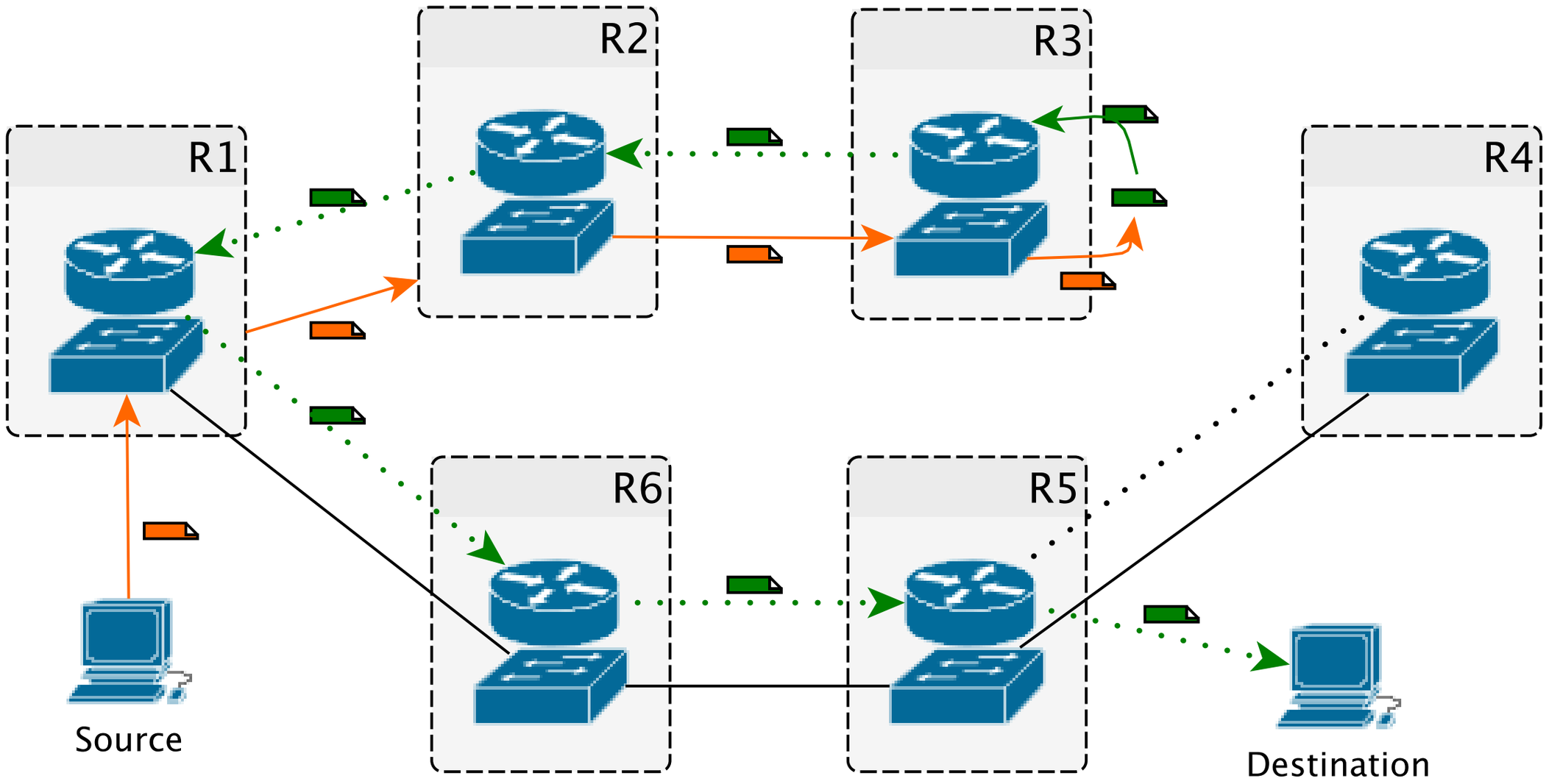
Operational model

Normal operation



Operational model

Failure of the link R3-R4



Guarantees

- Connectivity is preserved for any combination of failures

Guarantees

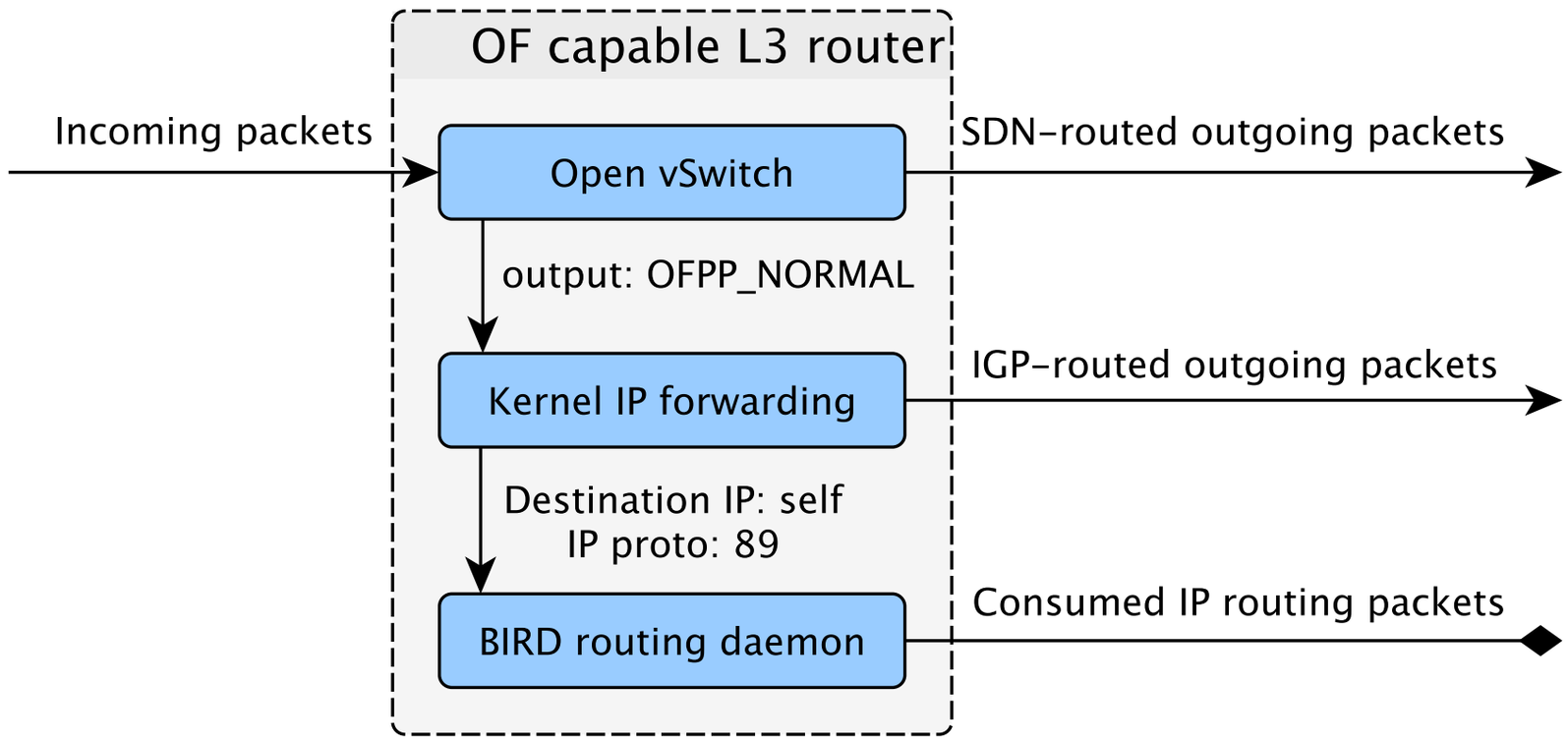
- Connectivity is preserved for any combination of failures
- Restoration of connectivity does not involve the controller

Guarantees

- Connectivity is preserved for any combination of failures
- Restoration of connectivity does not involve the controller
- Safety

Implementation overview

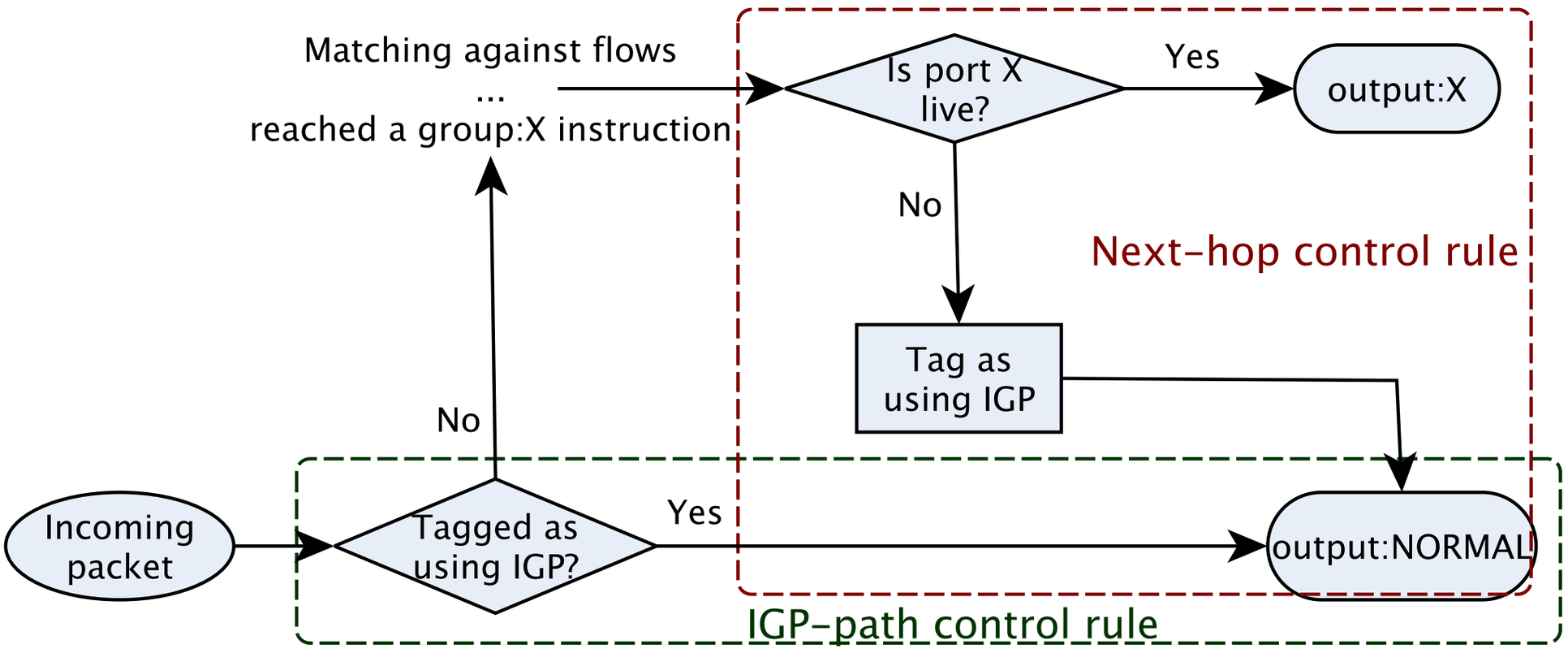
- Controller built against the Ryu framework
- Nodes are Linux hosts



Implementation overview (cont'd)

- IGP-forwarded packets are tagged in their TOS byte

- Uses OpenFlow fast failover groups



Agenda

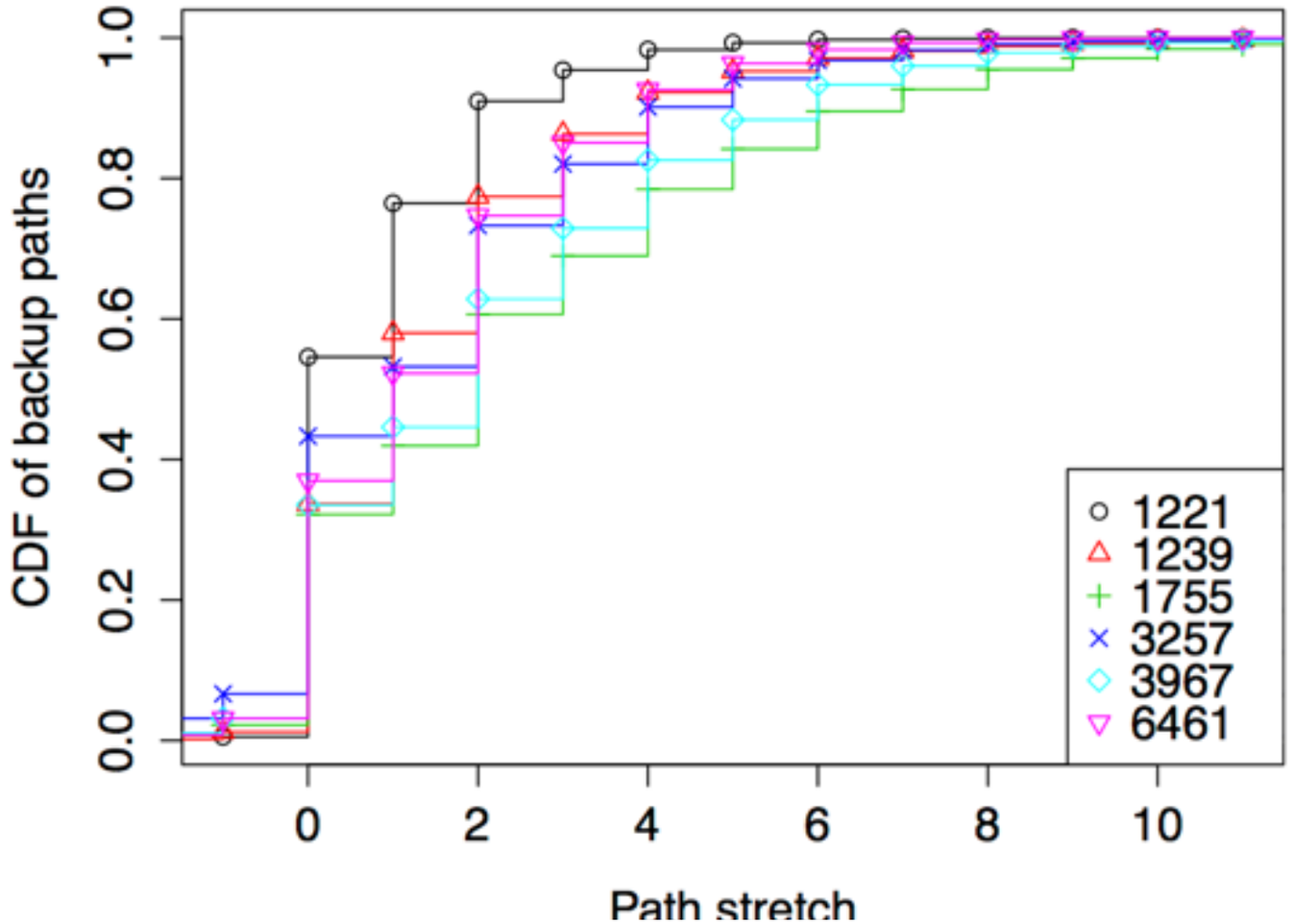
1. Software-Defined Networking
2. IBSDN
3. Evaluation
4. Discussion
5. Summary

Benchmarks

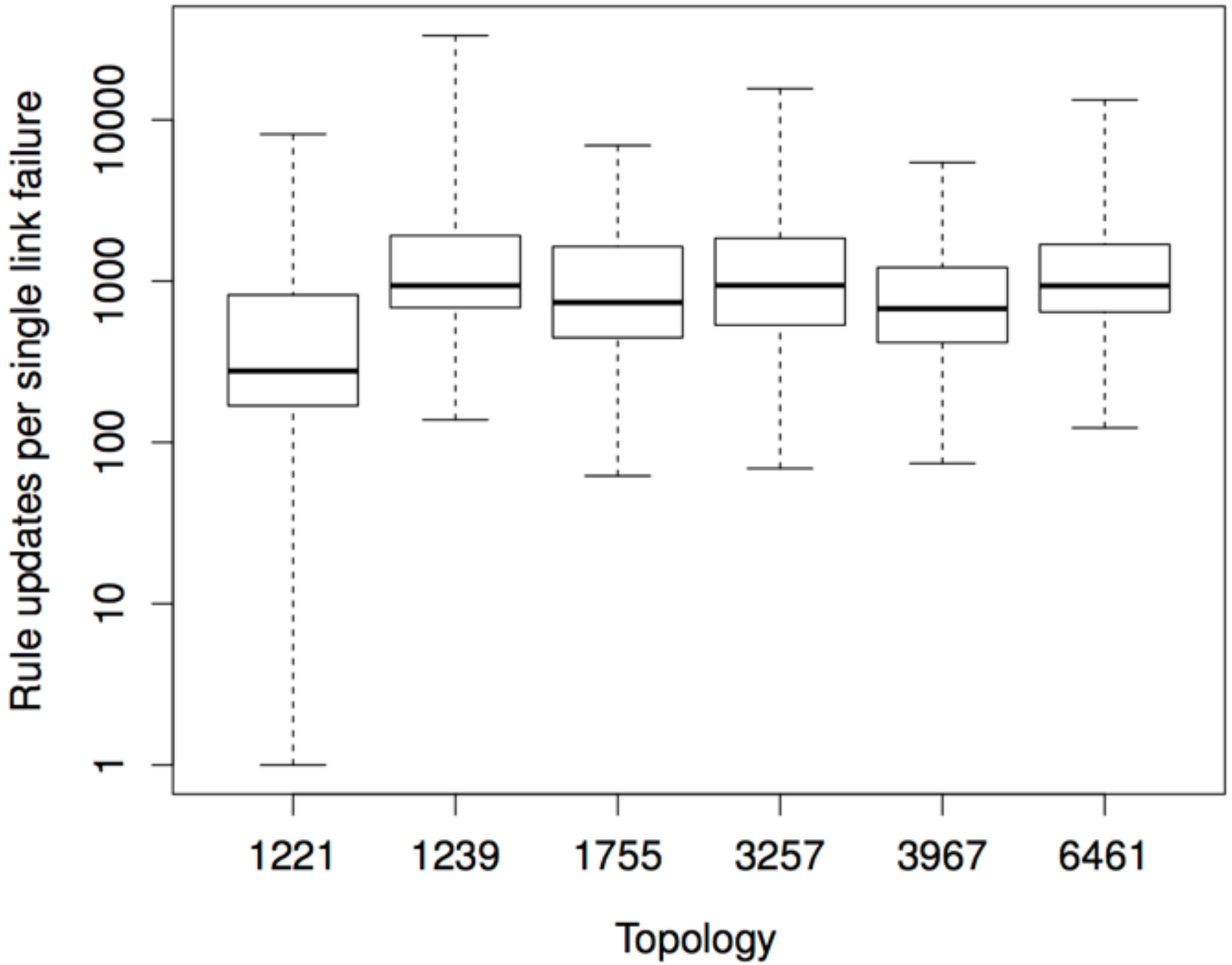
- Micro benchmark

Benchmarks

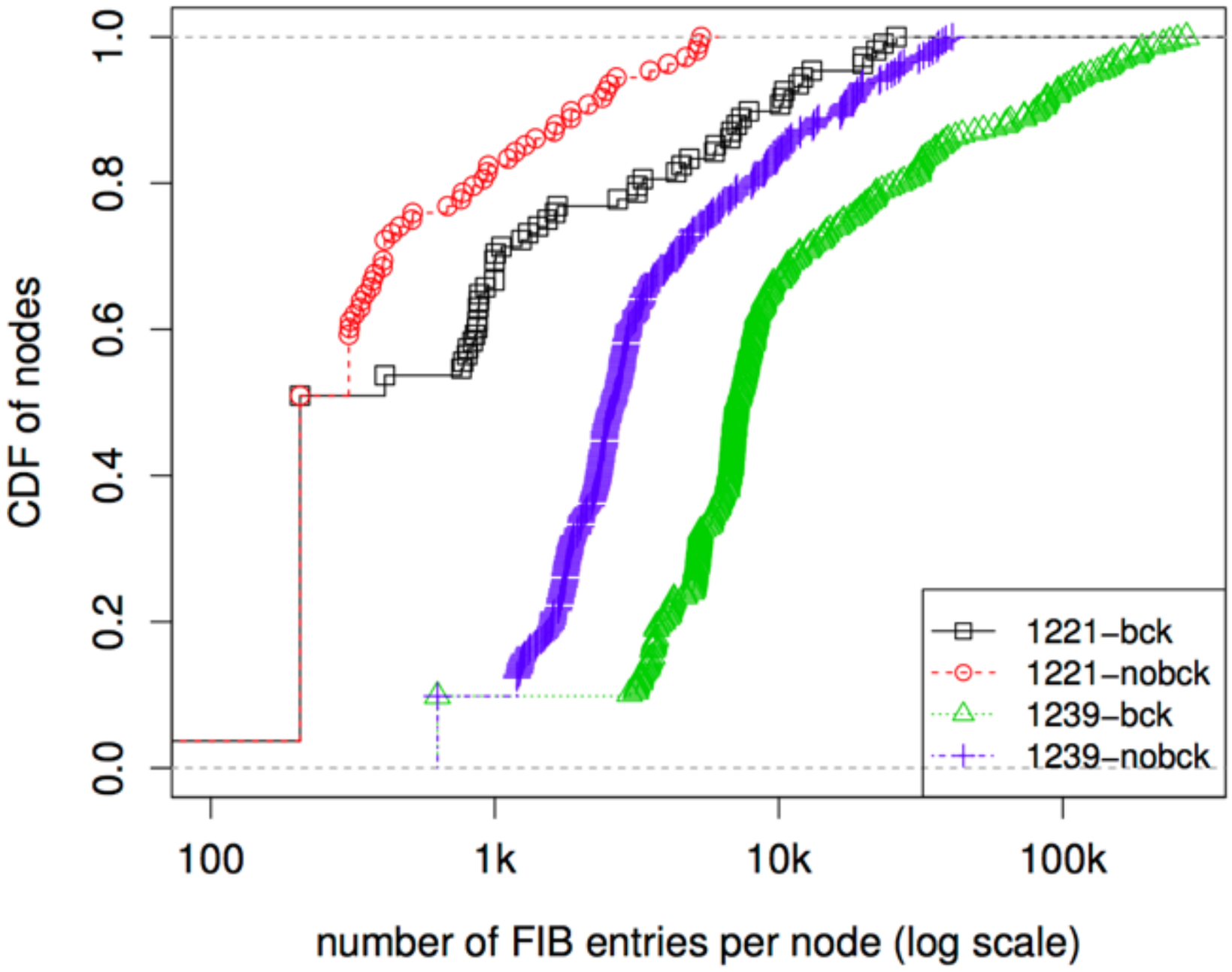
- Micro benchmark
- Macro benchmark



Comparison with purely reactive SDN technique



Comparison with purely proactive SDN technique



Agenda

1. Software-Defined Networking
2. IBSDN
3. Evaluation
4. Discussion
5. Summary

Benefits

- Robust against arbitrary set of failures
- Offers the expressiveness of SDN under normal operation
- Simplifies network design

Limitations

- IGP convergence
- Cannot enforce arbitrary policies with IGP
- Path stretch

Future work

- Reduce path-stretch
  - Remove U-turns
- Enforce some policies during the recovery process
  - Strict policies (do or drop)

Agenda

1. Software-Defined Networking
2. IBSDN
3. Evaluation
4. Discussion
5. Summary

Summary

- Failure management is hard in pure SDN
- IBSDN adds an IGP beneath the SDN control-plane to deal with failures
- IBSDN ensures:
  - Maximal robustness
  - Scalability
  - Upper bound on restoration time