# Interactive Monitoring, Visualization, and Configuration of OpenFlow-Based SDN

***Pedro Heleno Isolani***
*Juliano Araujo Wickboldt*
*Cristiano Bonato Both*
*Lisandro Zambenedetti Granville*
*Juergen Rochol*

July 16, 2015

IM 2015 IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT | Integrated Management in the Age of Big Data
OTTAWA, CANADA • MAY 11-15 2015
IEEE IEEE COMMUNICATIONS SOCIETY
ifip
UFRGS UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
.inf INSTITUTO DE INFORMÁTICA UFRGS
Computer Networks UFRGS

# Outline



1. Introduction & Motivation
2. Control Channel Analysis
3. SDN-Management approach
4. Experimental Evaluation
5. Conclusions & Open Questions

# Introduction – SDN Principles



**SDN** is an emerging paradigm that enables network innovation and simplifies network management based on four fundamental principles:

I. *Network control and forwarding planes are clearly decoupled*

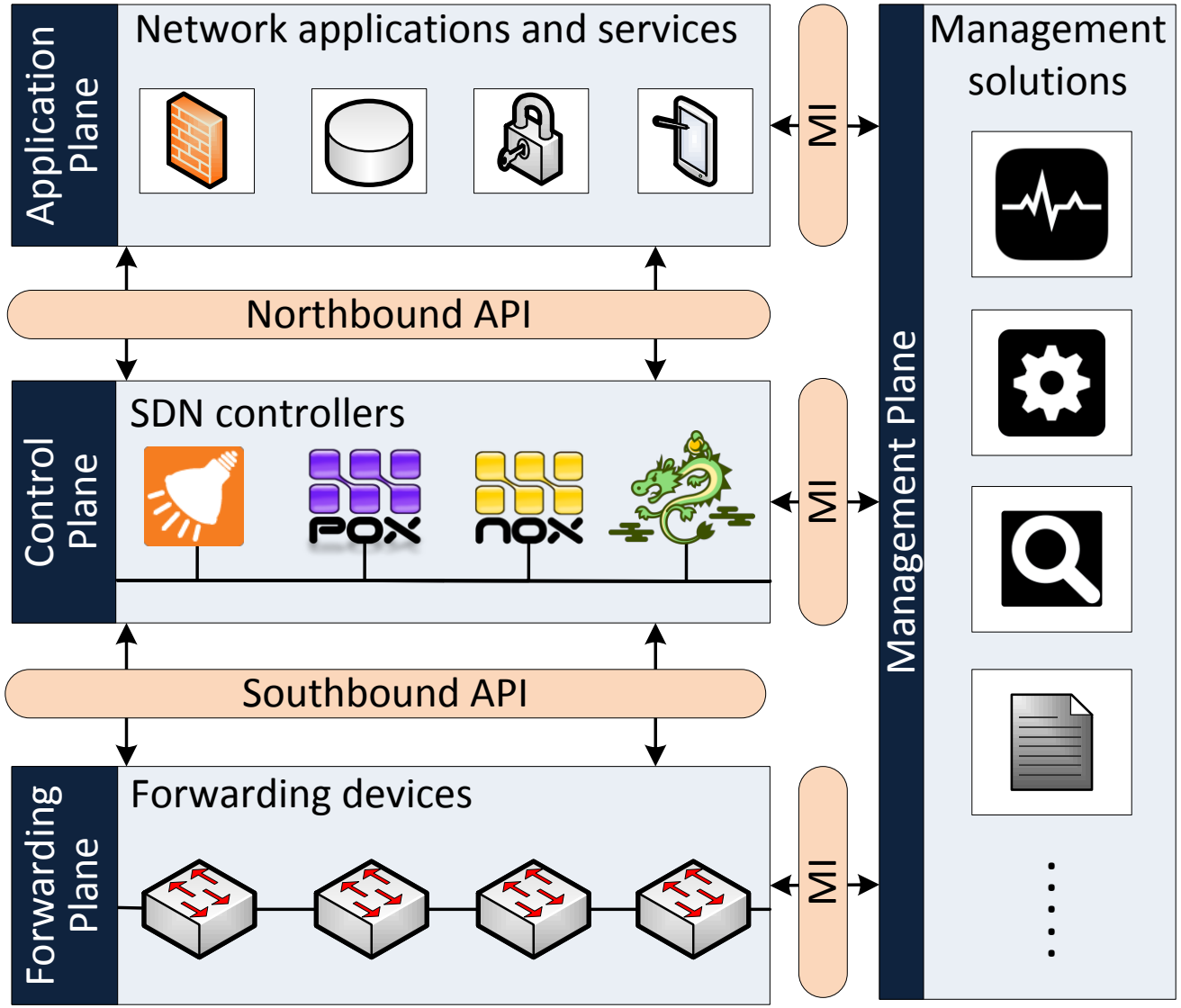
II. *Forwarding decisions are flow-based instead of destination-based*

III. *The network forwarding logic is abstracted from hardware to a programmable software layer*

IV. *An element, called controller, is introduced to coordinate network-wide forwarding decisions*

# Introduction - SDN architecture



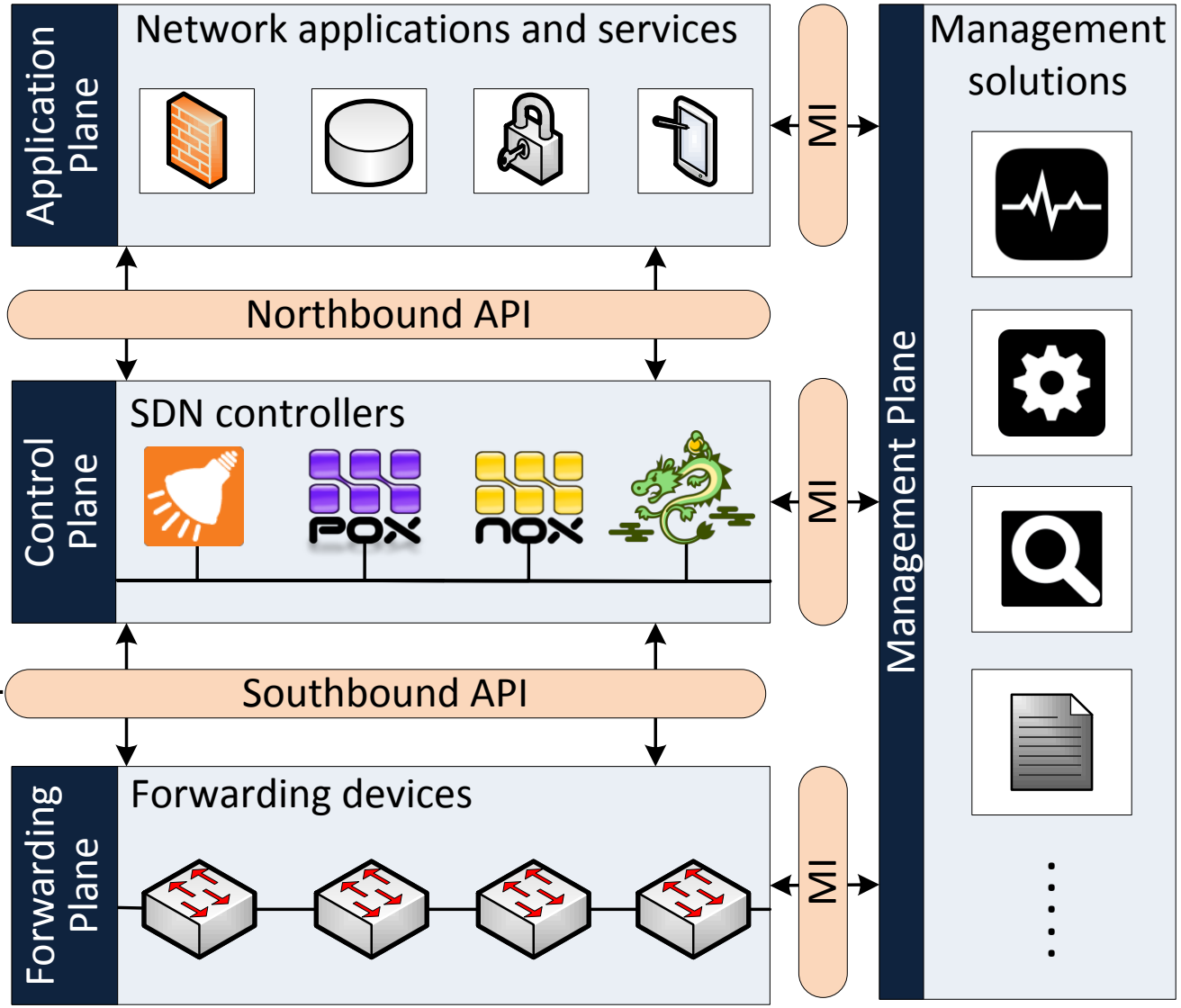


# Introduction - SDN architecture



## OpenFlow

*Currently the most relevant SDN implementation*




# Introduction - Motivation



SDN reduces or even eliminates some traditional management problems [1]

- *E.g.*, enabling network configuration in a high level language
- *E.g.*, providing support for enhanced network diagnosis and troubleshooting

# Introduction - Motivation



SDN reduces or even eliminates some traditional management problems [1]

- *E.g.*, enabling network configuration in a high level language
- *E.g.*, providing support for enhanced network diagnosis and troubleshooting

Monitoring, Visualization, and Configuration can be considerably different from traditional networks

- *E.g.*, SDN controller behavior impacts on resource consumption and forwarding performance

# Introduction - Motivation



SDN reduces or even eliminates some traditional management problems [1]

- *E.g.*, enabling network configuration in a high level language
- *E.g.*, providing support for enhanced network diagnosis and troubleshooting

Monitoring, Visualization, and Configuration can be considerably different from traditional networks

- *E.g.*, SDN controller behavior impacts on resource consumption and forwarding performance

Centralized controller creates bottlenecks at the control channel [7]

# Introduction - Motivation



SDN reduces or even eliminates some traditional management problems [1]

- *E.g.*, enabling network configuration in a high level language
- *E.g.*, providing support for enhanced network diagnosis and troubleshooting

Monitoring, Visualization, and Configuration can be considerably different from traditional networks

- *E.g.*, SDN controller behavior impacts on resource consumption and forwarding performance

Centralized controller creates bottlenecks at the control channel [7]

**Devoflow [8]** and **DIFANE [9]** attempted to alleviate these bottlenecks by distributing the control logic of OpenFlow

# Introduction - Motivation



SDN reduces or even eliminates some traditional management problems [1]

- *E.g.*, enabling network configuration in a high level language
- *E.g.*, providing support for enhanced network diagnosis and troubleshooting

Monitoring, Visualization, and Configuration can be considerably different from traditional networks

- *E.g.*, SDN controller behavior impacts on resource consumption and forwarding performance

Centralized controller creates bottlenecks at the control channel [7]

**Devoflow [8]** and **DIFANE [9]** attempted to alleviate these bottlenecks by distributing the control logic of OpenFlow

*How OpenFlow control messages can be used without affect network performance?*

# Introduction - Motivation



Moreover…

SDN proposals use monitoring information to automate tasks

- *E.g.*, **reduce control traffic overhead** [2] [3] [4]
- *E.g.*, **protecting the network** [5] [6]

# Introduction - Motivation



Moreover…

SDN proposals use monitoring information to automate tasks

- *E.g.*, **reduce control traffic overhead** [2] [3] [4]
- *E.g.*, **protecting the network** [5] [6]

In what proportion the SDN controller behavior can affect both resource consumption and traffic forwarding performance?

# Introduction - Motivation



Moreover…

SDN proposals use monitoring information to automate tasks

- *E.g.*, **reduce control traffic overhead** [2] [3] [4]
- *E.g.*, **protecting the network** [5] [6]

In what proportion the SDN controller behavior can affect both resource consumption and traffic forwarding performance?

No solution is available to integrate monitoring information with interactive visualization and configuration tools for SDN

# Contributions



I. Quantify overheads imposed by OpenFlow messages on the control channel

**Control Channel Analysis**

II. Propose an interactive approach to SDN management through monitoring, visualization, and configuration including the administrator in the management loop

**Interactive approach through monitoring, visualization, and configuration**

# Contributions



I. Quantify overheads imposed by OpenFlow messages on the control channel

**Control Channel Analysis**

II. Propose an interactive approach to SDN management through monitoring, visualization, and configuration including the administrator in the management loop

**Interactive approach through monitoring, visualization, and configuration**

# Control Channel Analysis



Quantify overheads imposed by OpenFlow messages on the control channel **using**:

- OpenFlow version 1.0
- Controller's *Forwarding* behavior implementation
- Campus network scenario

Analyzing

- **Control Channel Load** (installation and monitoring forwarding rules)
- **Resource Usage** (active and idle rules)

# Control Channel Analysis



Quantify overheads imposed by OpenFlow messages on the control channel **using**:

- OpenFlow version 1.0
- Controller's *Forwarding* behavior implementation
- Campus network scenario

Analyzing

- **Control Channel Load** (installation and monitoring forwarding rules)
- **Resource Usage** (active and idle rules)

A rule is considered **idle** when its counters do not change between two monitoring polls

# Control Channel Analysis - Control Messages



**OpenFlow 1.0 Control Messages**

| Message Type | Sub-type | Description |
|---|---|---|
| Controller-to-switch | *Features* | Obtain features and capabilities about the switches |
| | *Configuration* | Set query configuration parameters in switches |
| | ***Modify-State*** | **Manage the state of the switches** |
| | ***Read-State*** | **Retrieve statistics about switch tables, ports, flows, and queues** |
| | ***Send-Packet*** | **Send packets to a specific switch port** |
| | *Barrier* | Ensure message dependencies and receive notifications |
| Asynchronous | ***Packet-In*** | **When a packet do not match with a flow entry or an matched flow entry action is "send to the controller"** |
| | *Flow-Removed* | When a flow entry expires in the switch flow table |
| | *Port-Status* | Send port configuration state changes |
| | *Error* | Notify problems to the controller |
| Symmetric | *Hello* | Exchanged between switch and controller upon connection startup |
| | *Echo* | Sent by both controller and switch to establish connectivity |
| | *Vendor* | Functionality to store a staging area for other OpenFlow revisions |

# Control Channel Analysis - Control Messages



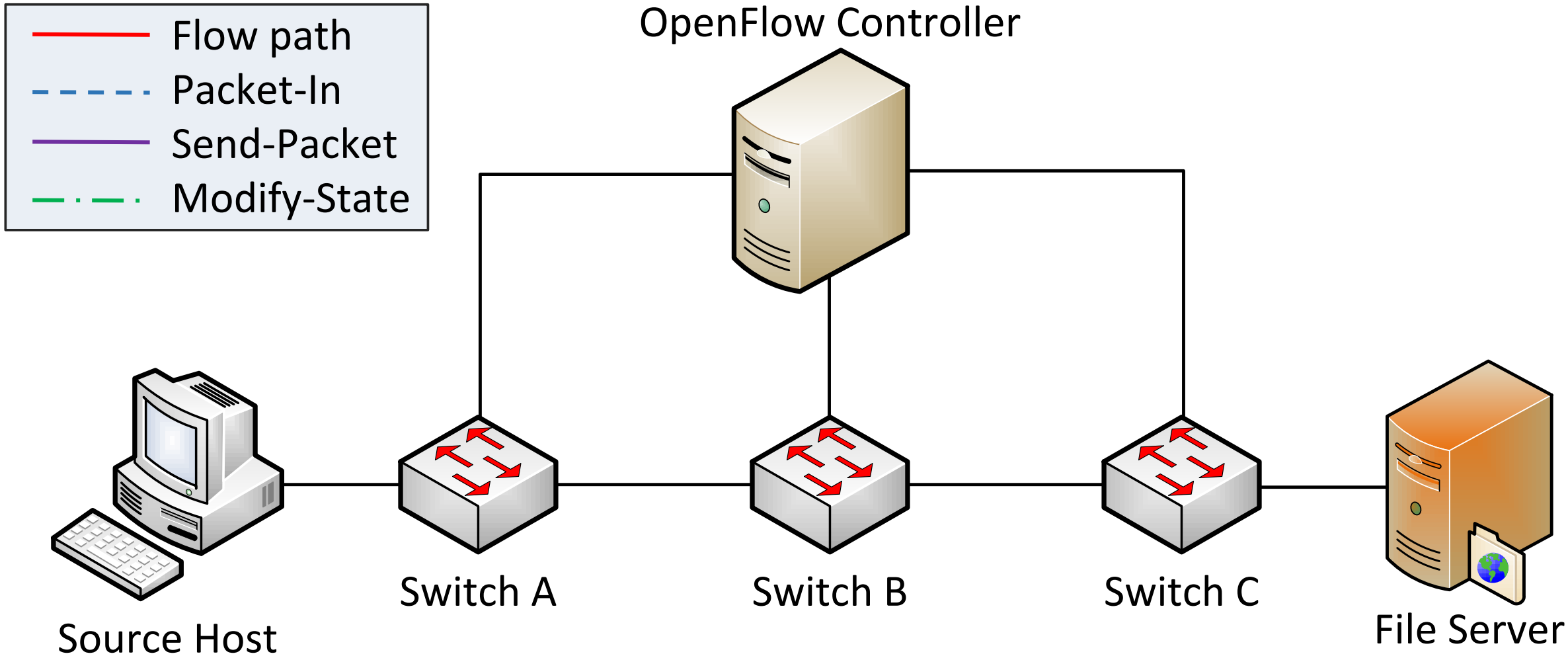
**OpenFlow 1.0 Control Messages**

| Message Type | Sub-type | Description |
|---|---|---|
| Controller-to-switch | *Features* | Obtain features and capabilities about the switches |
| | *Configuration* | Set query configuration parameters in switches |
| | ***Modify-State*** | **Manage the state of the switches** |
| | ***Read-State*** | **Retrieve statistics about switch tables, ports, flows, and queues** |
| | ***Send-Packet*** | **Send packets to a specific switch port** |
| | *Barrier* | Ensure message dependencies and receive notifications |
| Asynchronous | ***Packet-In*** | **When a packet do not match with a flow entry or an matched flow entry action is "send to the controller"** |
| | *Flow-Removed* | When a flow entry expires in the switch flow table |
| | *Port-Status* | Send port configuration state changes |
| | *Vendor* | Functionality to store a staging area for other OpenFlow revisions |

Selected sub-types represent 97.78% of the number of messages and 99.70% of the overall control traffic

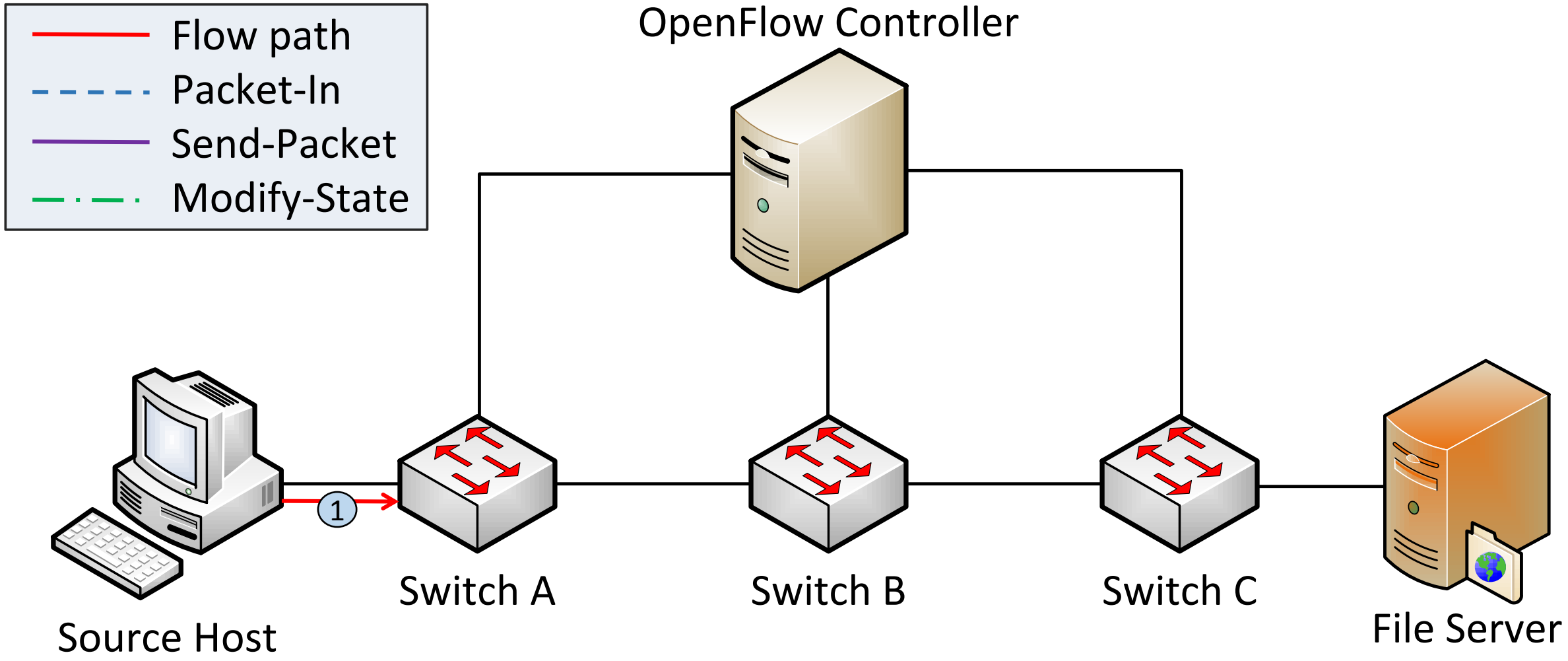
# Control Channel Analysis - *Forwarding* Behavior





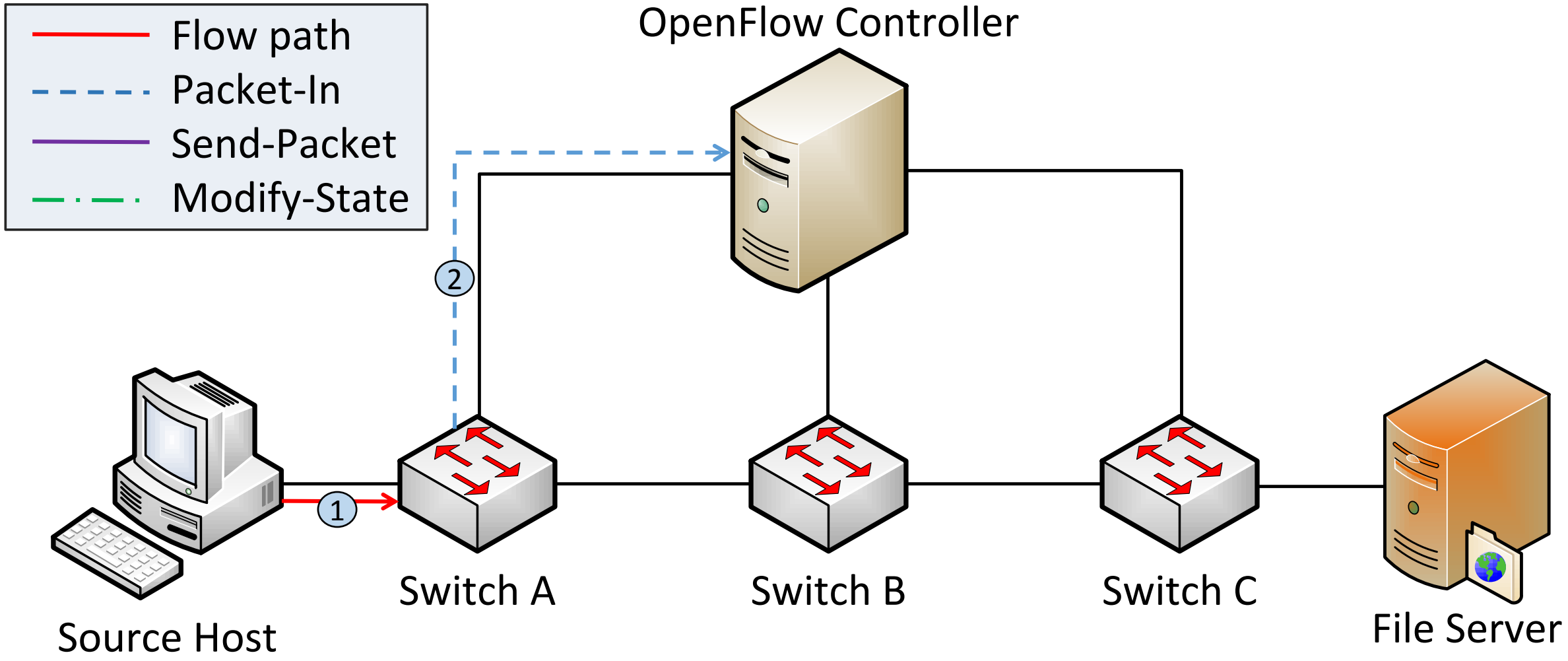
# Control Channel Analysis - *Forwarding* Behavior





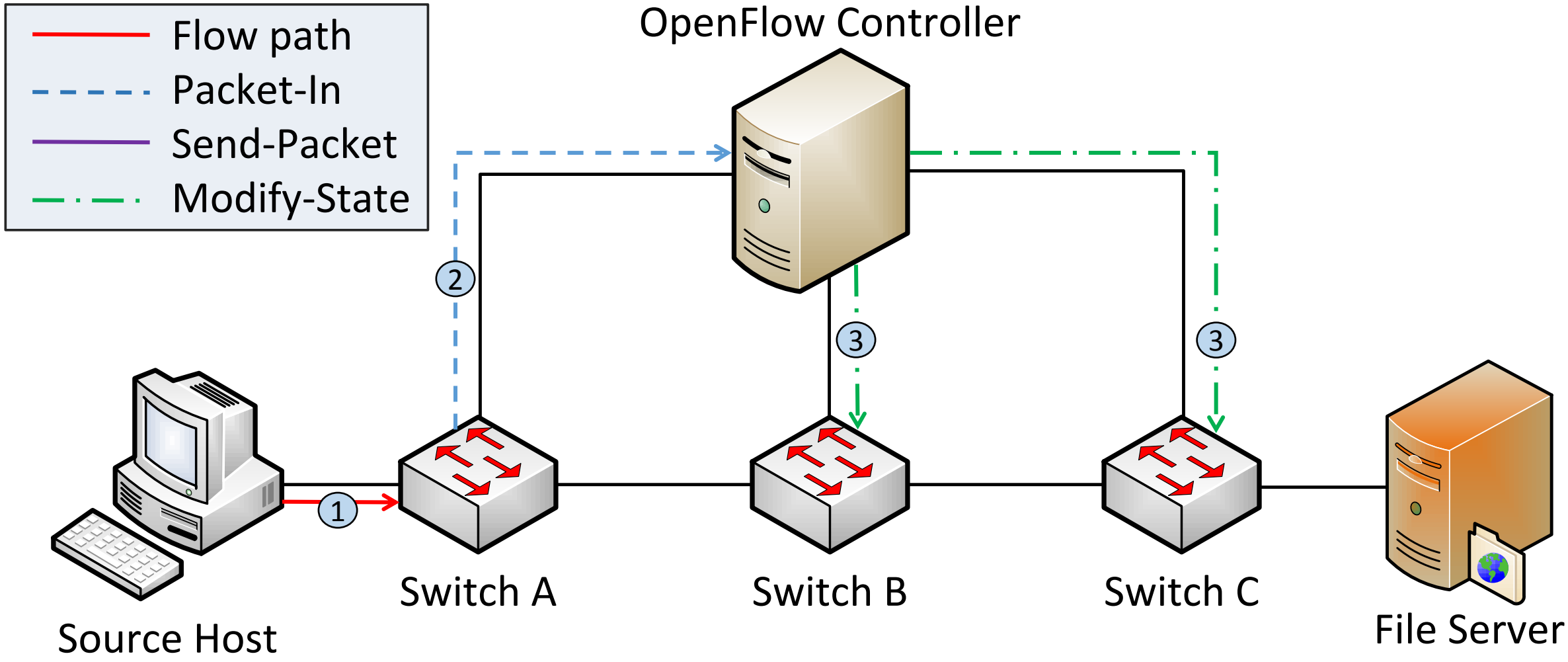
# Control Channel Analysis - *Forwarding* Behavior





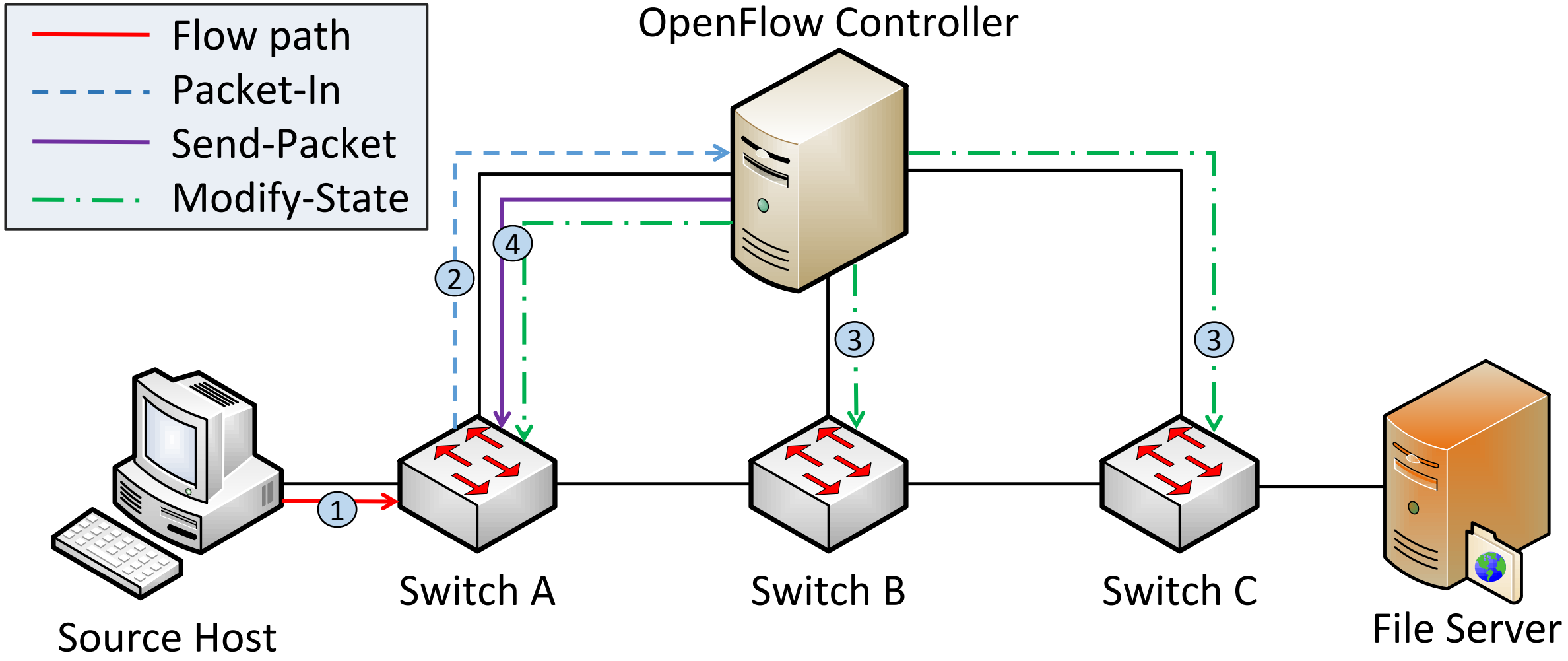
# Control Channel Analysis - *Forwarding* Behavior





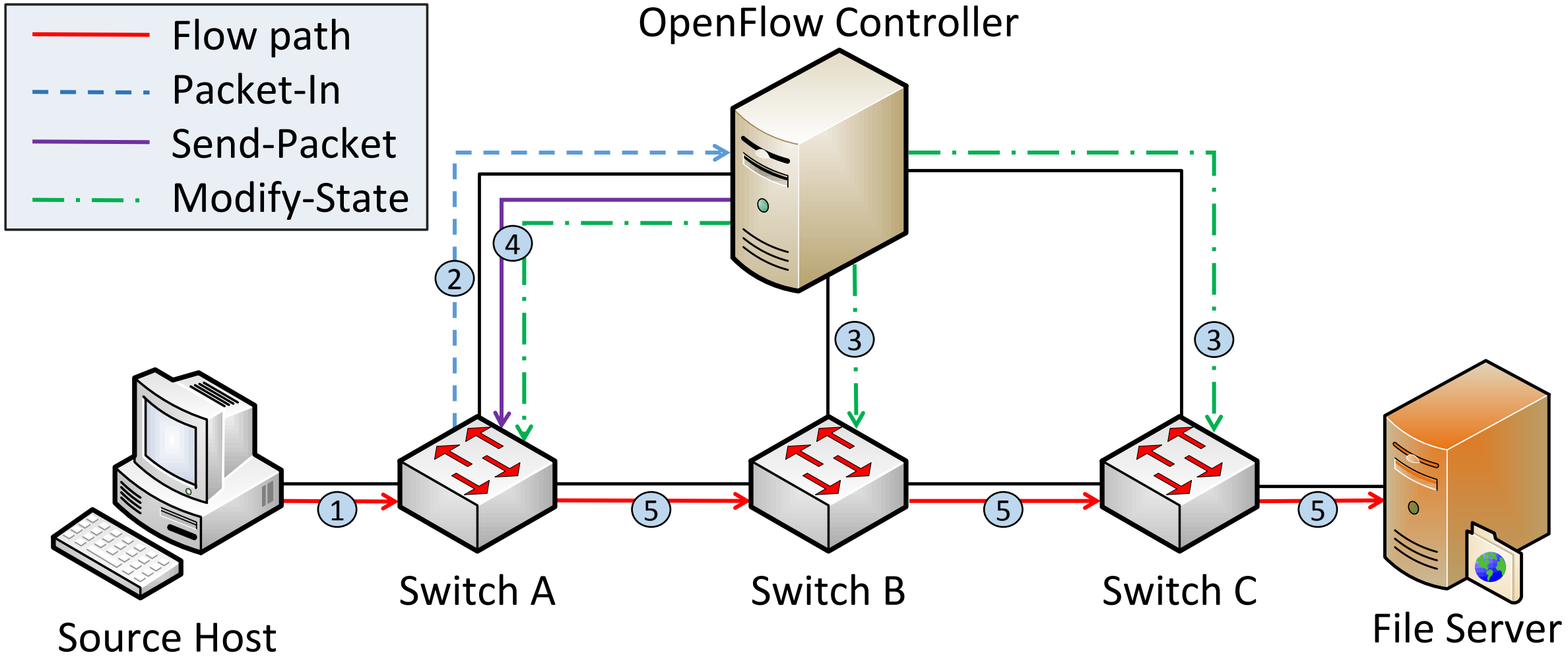
# Control Channel Analysis - *Forwarding* Behavior





# Control Channel Analysis - *Forwarding* Behavior



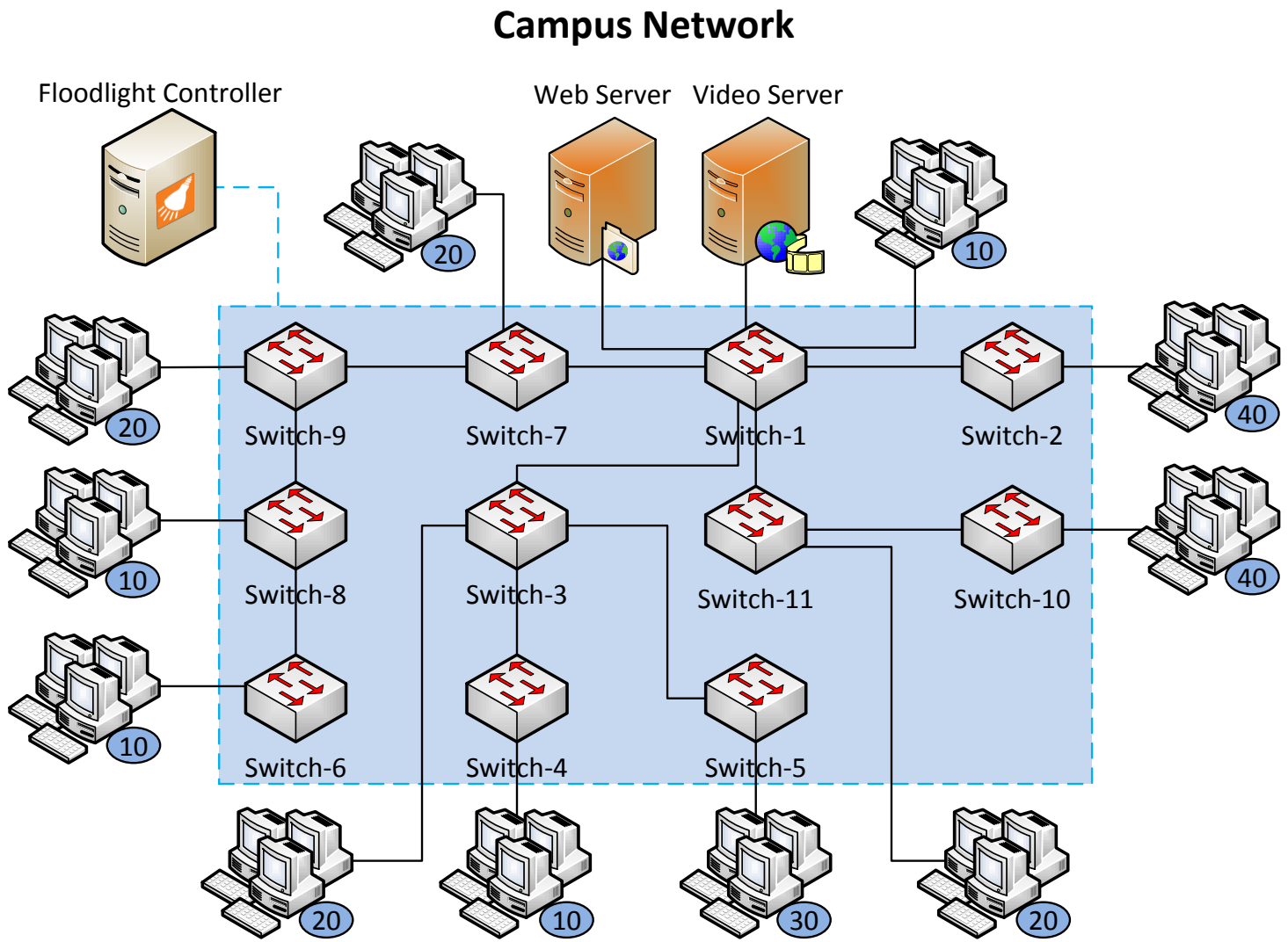


# Control Channel Analysis - Scenario



## Scenario

- Hosts: 230
- Switches: 11
- Web Servers: 1
- Video Servers: 1
- Controllers: 1
- Controller: Floodlight v.90
- Emulated over **Mininet**



# Control Channel Analysis - Workload



## Workload

- User Traffic Profile
  - Emulated Internet traffic

## Varied factor

- Idle timeout configuration

**Configuration Parameters of User Traffic Profile**

| Parameter | Value |
|---|---|
| Web request size | Lognormal Distribution ($\mu = 11.75$, $\sigma = 1.37$) Mean: 324 KBytes, Std. Dev.: 762 KBytes |
| User reading time | Exponential Distribution ($\lambda = 0.033$) Mean: 30 seconds |
| Video watch time | 180 seconds |
| Video bit rate | 300 kbps |
| Traffic Mix | Video: 75%, Web: 25% |
| User Mix | 1 video user for every 6 Web users |
| Monitoring | Polling frequency: 5 seconds |
| Controller behavior | Floodlight's default *Forwarding Behavior* implementation |
| Experiment duration | 30 min |

# Control Channel Analysis - Workload



## Workload

- User Traffic Profile
  - Emulated Internet traffic

## Varied factor

- Idle timeout configuration

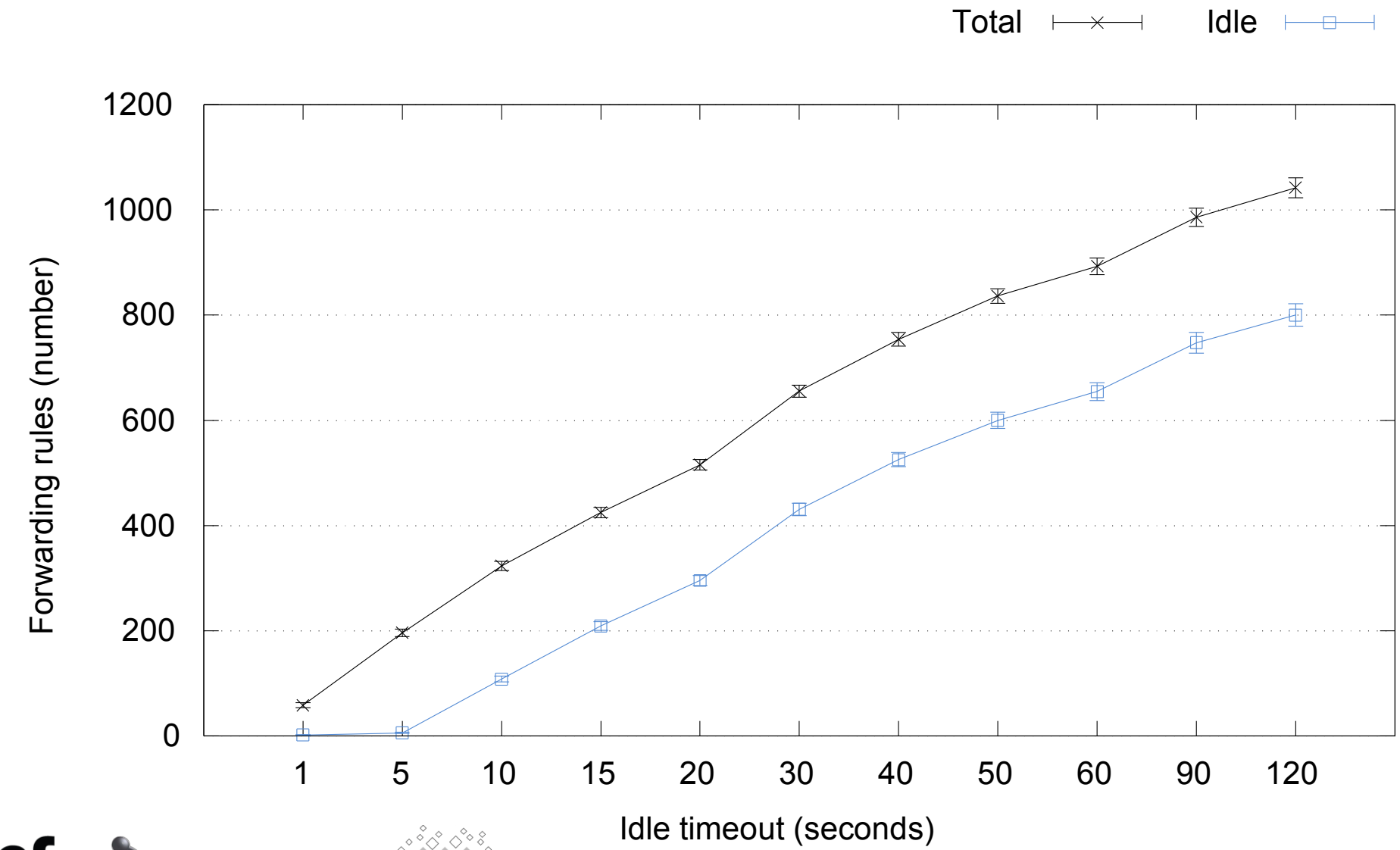
**Idle timeout** configuration indicates when a forwarding rule can be removed due to a lack of activity

**Configuration Parameters of User Traffic Profile**

| Parameter | Value |
|---|---|
| Web request size | Lognormal Distribution ($\mu = 11.75$, $\sigma = 1.37$) Mean: 324 KBytes, Std. Dev.: 762 KBytes |
| User reading time | Exponential Distribution ($\lambda = 0.033$) Mean: 30 seconds |
| Video watch time | 180 seconds |
| Video bit rate | 300 kbps |
| Traffic Mix | Video: 75%, Web: 25% |
| User Mix | 1 video user for every 6 Web users |
| Monitoring | Polling frequency: 5 seconds |
| Controller behavior | Floodlight's default *Forwarding Behavior* implementation |
| Experiment duration | 30 min |

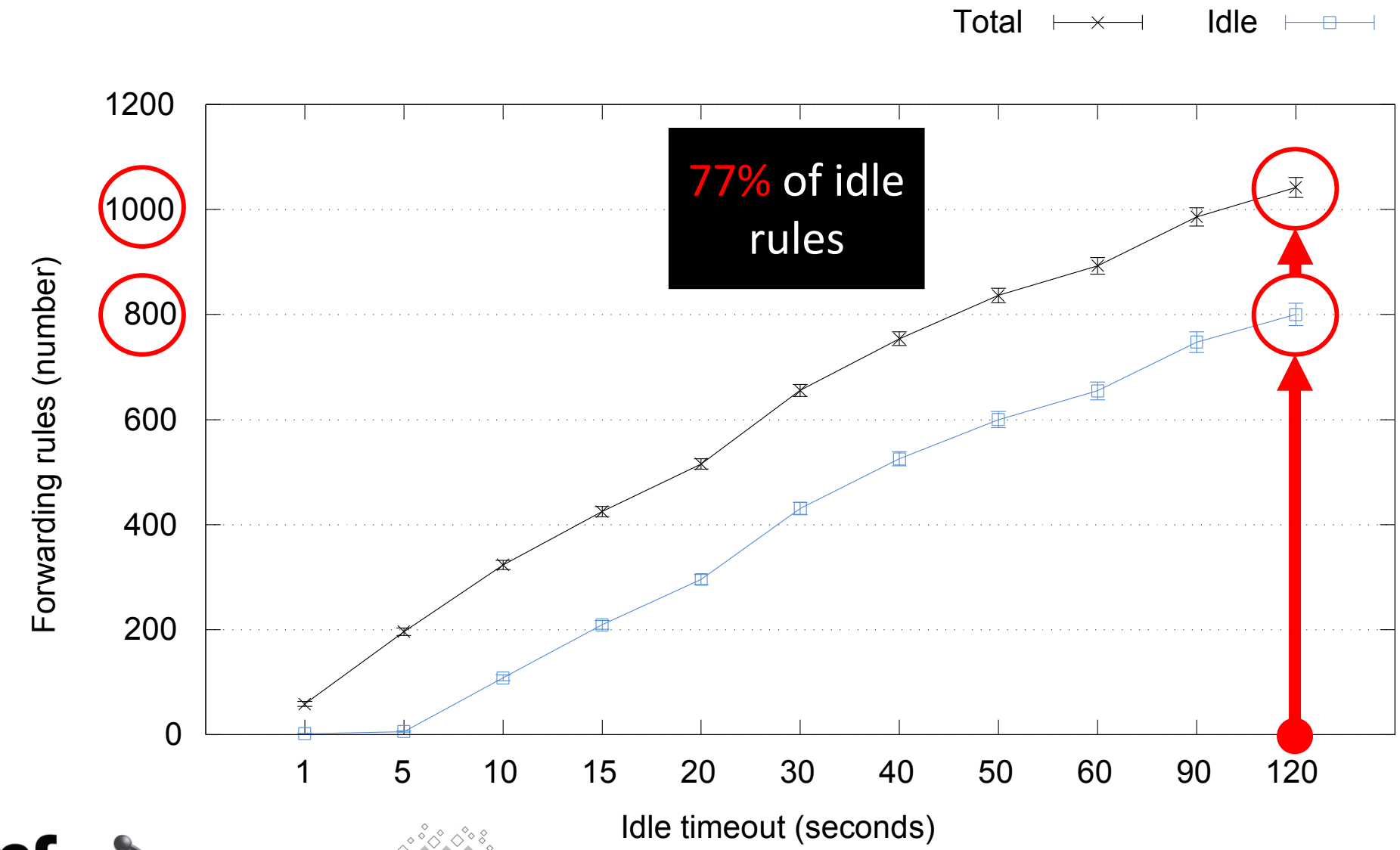
# Control Channel Analysis – Forwarding rules





# Control Channel Analysis – Forwarding rules

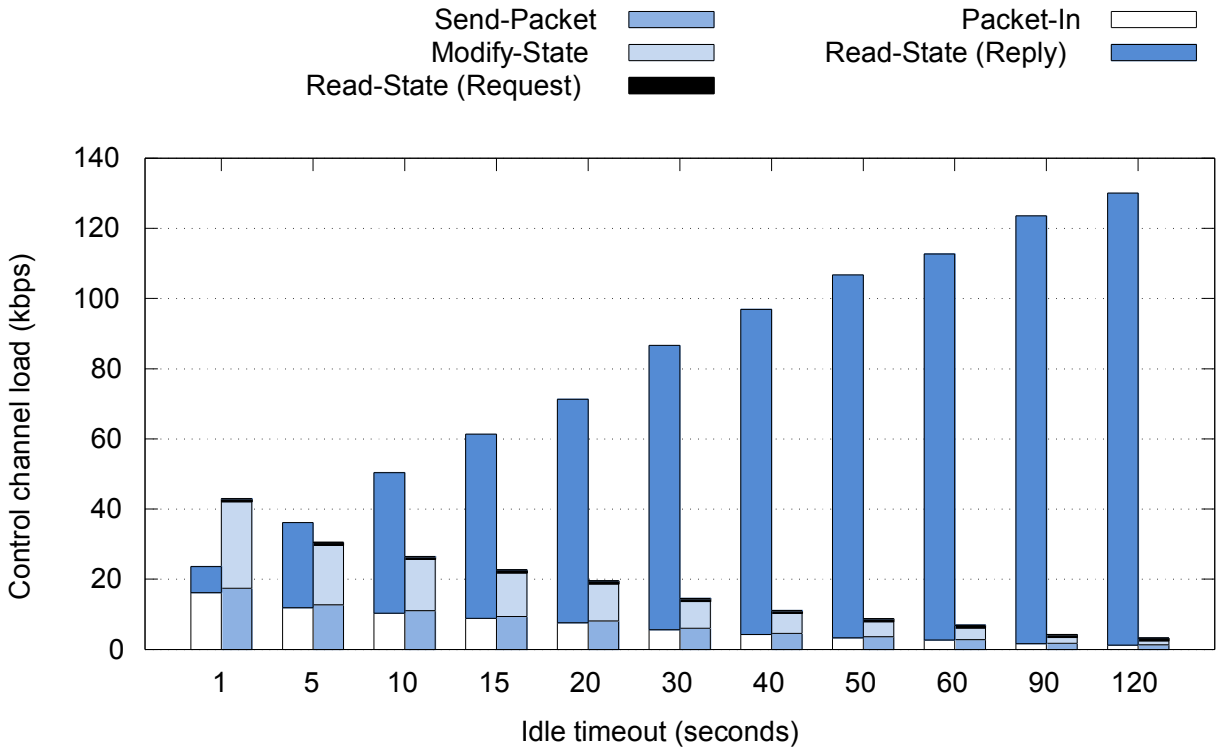




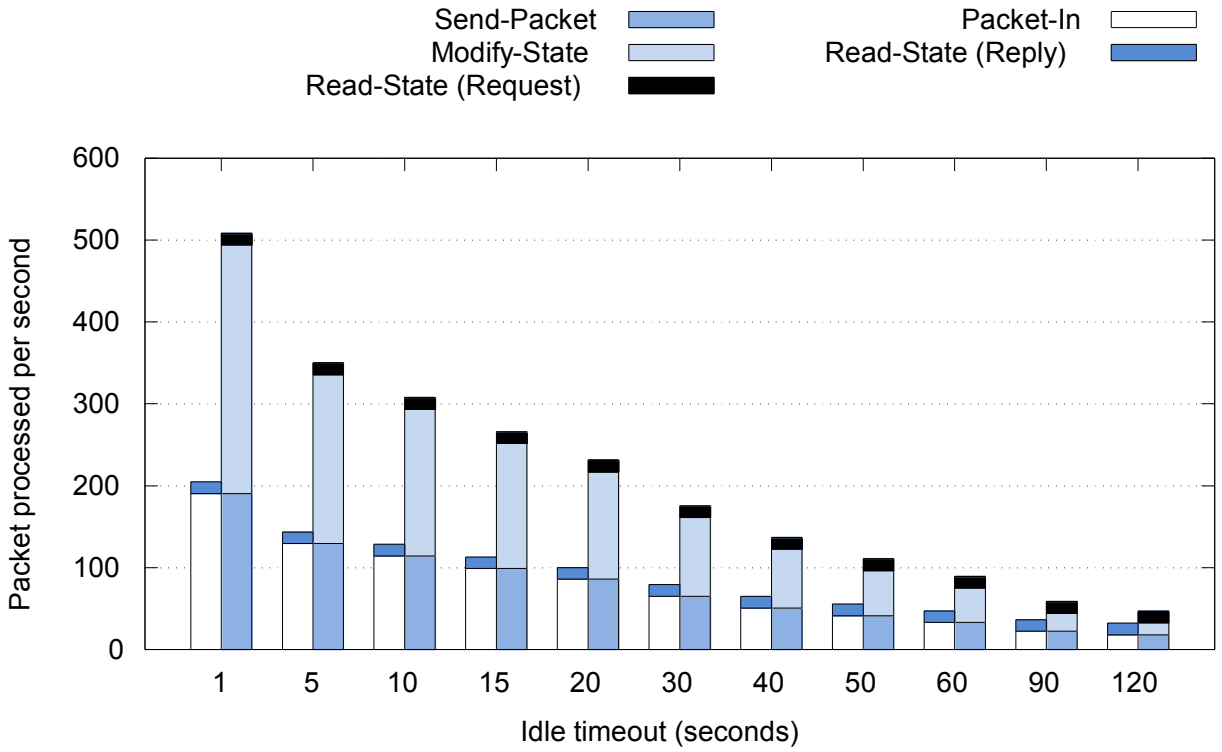
# Control Channel Analysis – Channel load



## Control channel load x Idle timeout



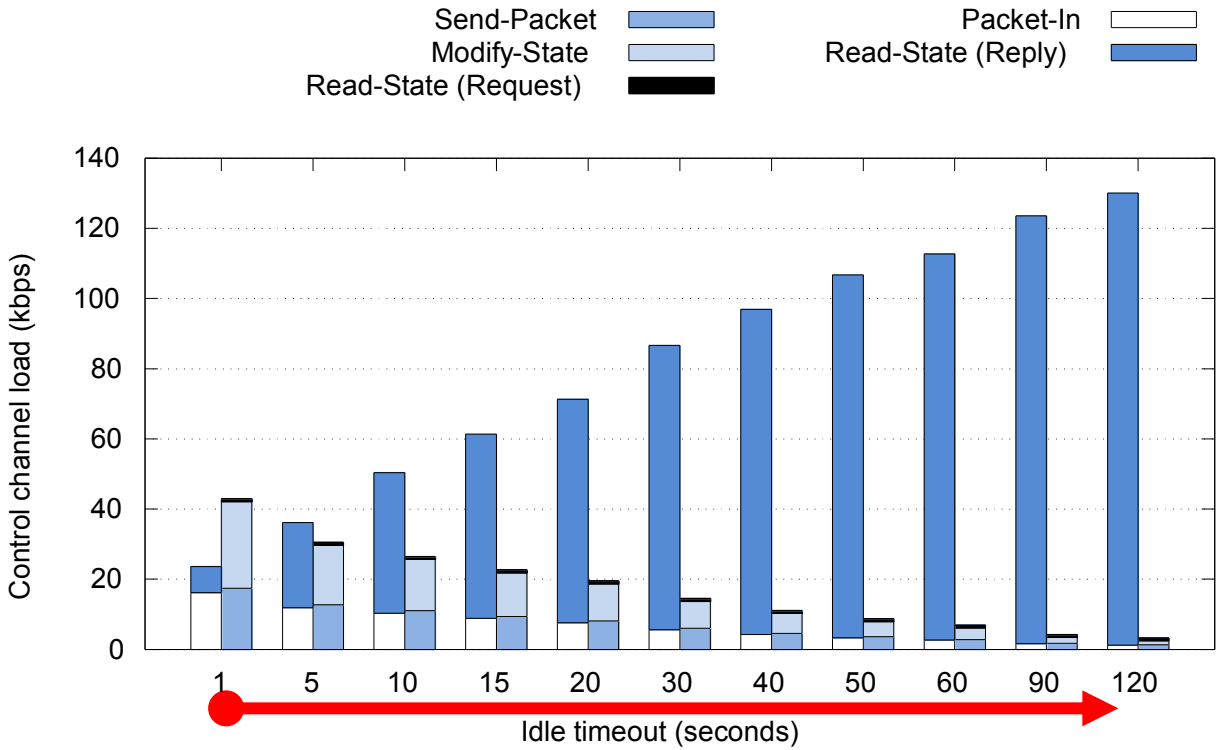

## Packets processed per second x idle timeout



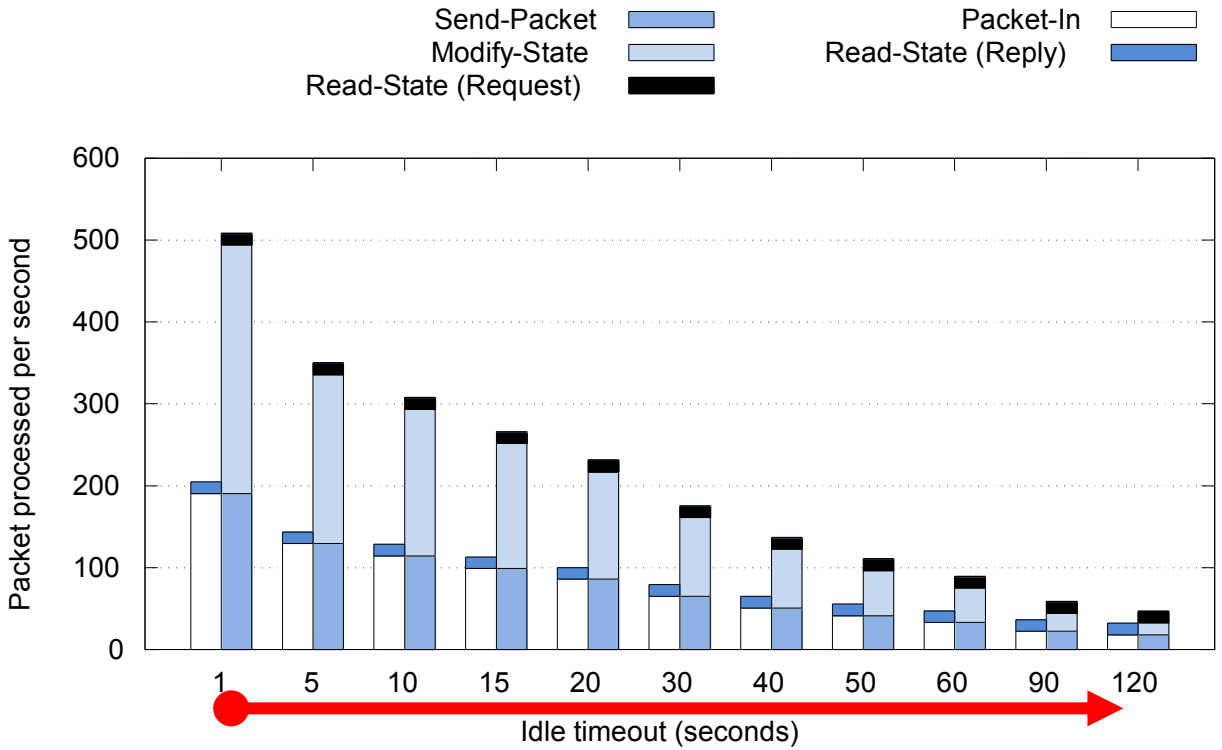
# Control Channel Analysis – Channel load



## Control channel load x Idle timeout




## Packets processed per second x idle timeout

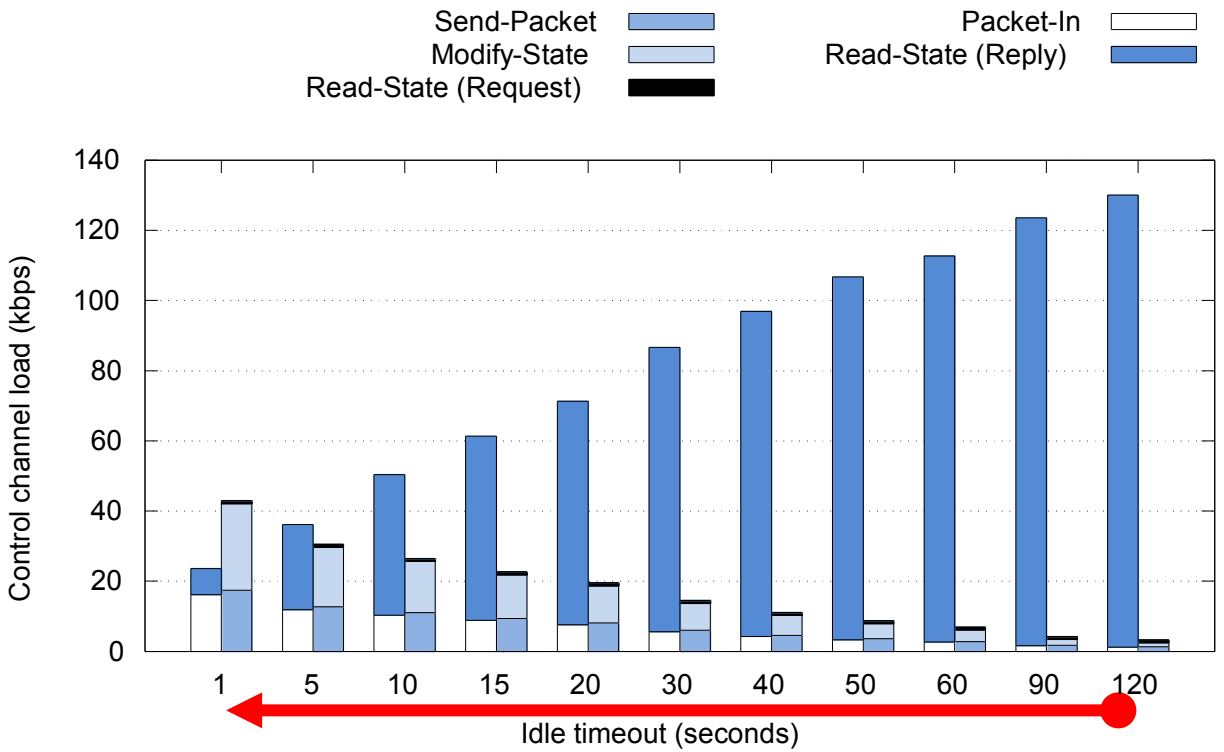



**Increasing de idle timeout value**

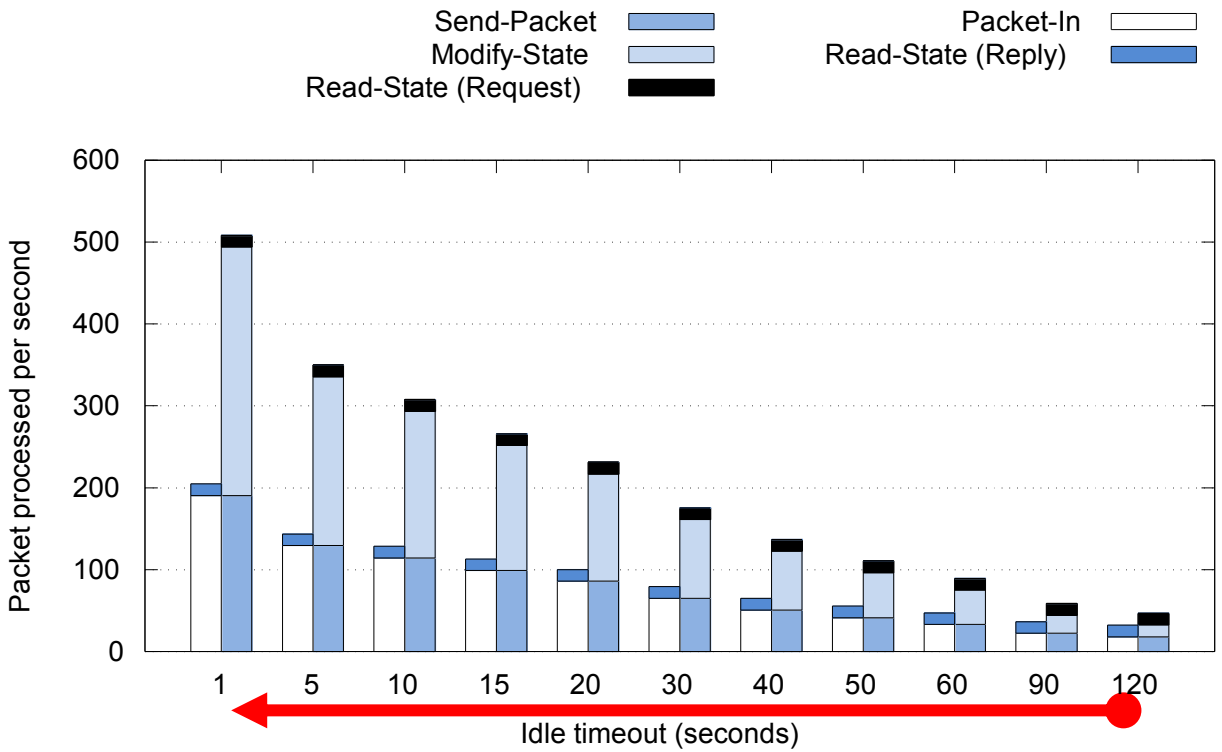
# Control Channel Analysis – Channel load



## Control channel load x Idle timeout




## Packets processed per second x idle timeout




**Decreasing de idle timeout value**

# Contributions



I. Quantify overheads imposed by OpenFlow messages on the control channel

**Control Channel Analysis**

II. Propose an interactive approach to SDN management through monitoring, visualization, and configuration including the administrator in the management loop

**Interactive approach through monitoring, visualization, and configuration**

# Contributions



I. Quantify overheads imposed by OpenFlow messages on the control channel

**Control Channel Analysis**

II. Propose an interactive approach to SDN management through monitoring, visualization, and configuration including the administrator in the management loop

**Interactive approach through monitoring, visualization, and configuration**

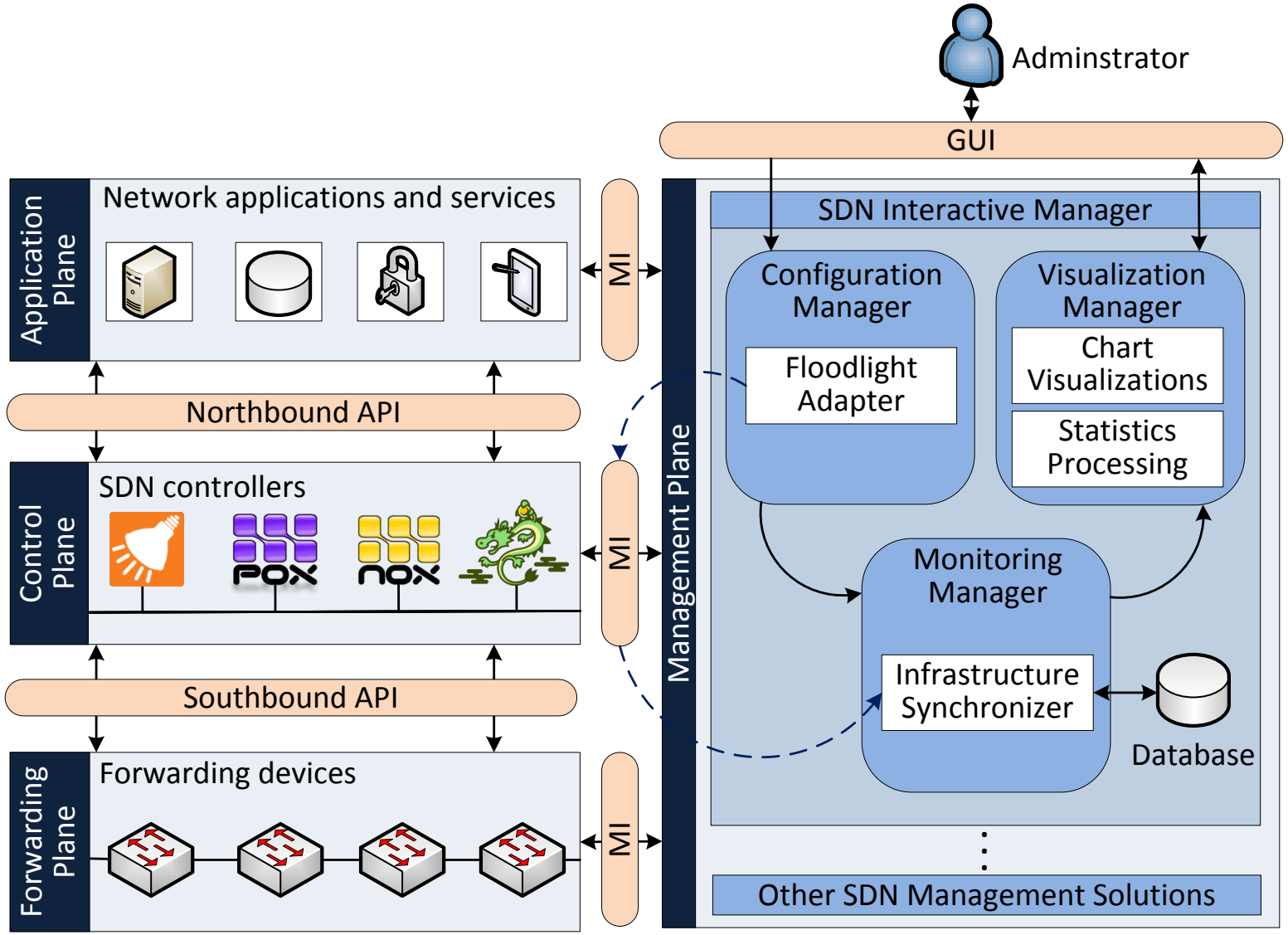
# Proposal - Conceptual Architecture



## SDN Interactive Manager

- Monitoring Manager
- Visualization Manager
- Configuration Manager

A management loop with the Administrator interactions



# Proposal - Conceptual Architecture

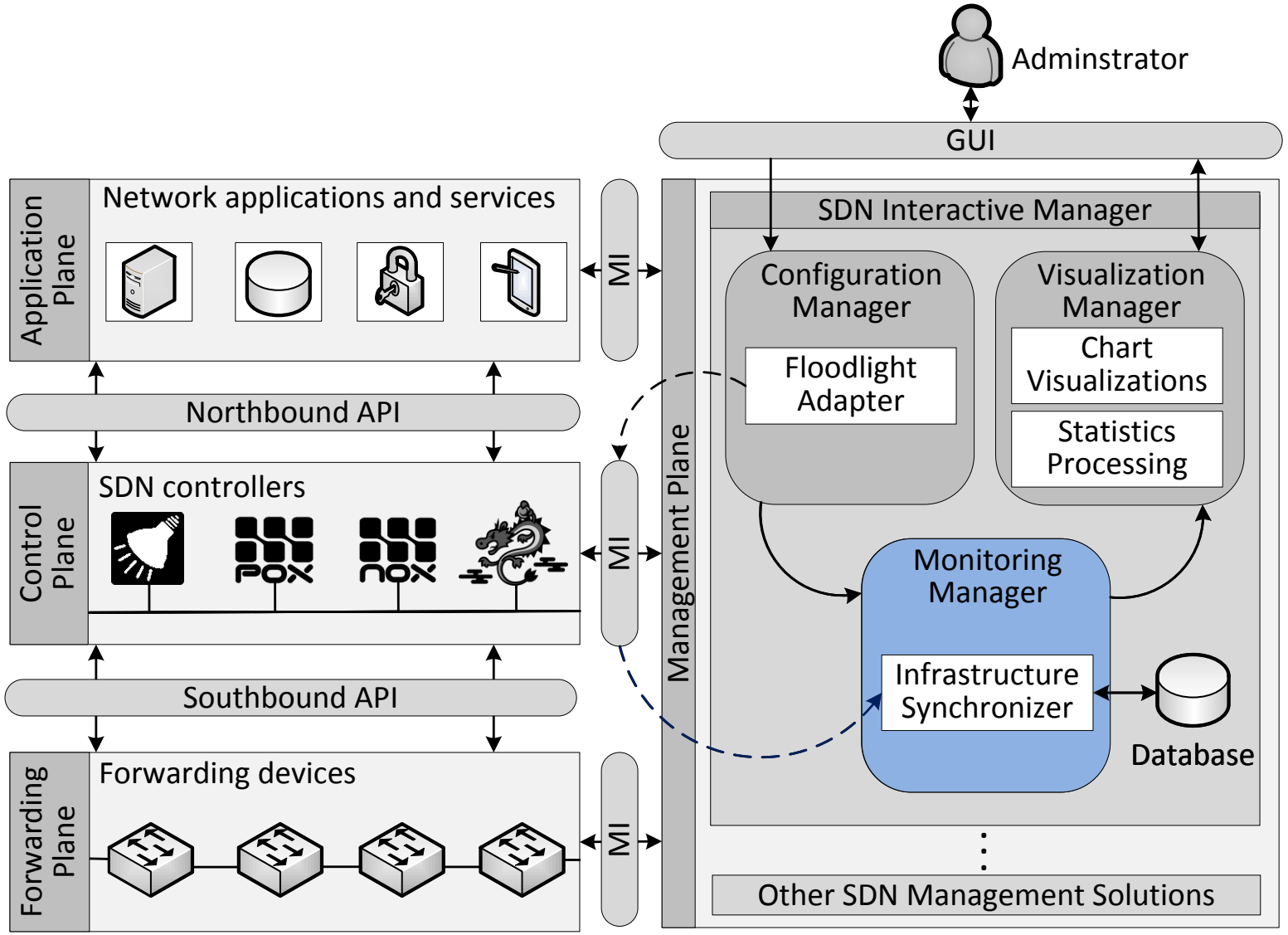


## Monitoring Manager

Retrieve information about the network and storing it in a local *Database*

## ***Modules***

*Infrastructure Synchronizer*



# Proposal - Conceptual Architecture



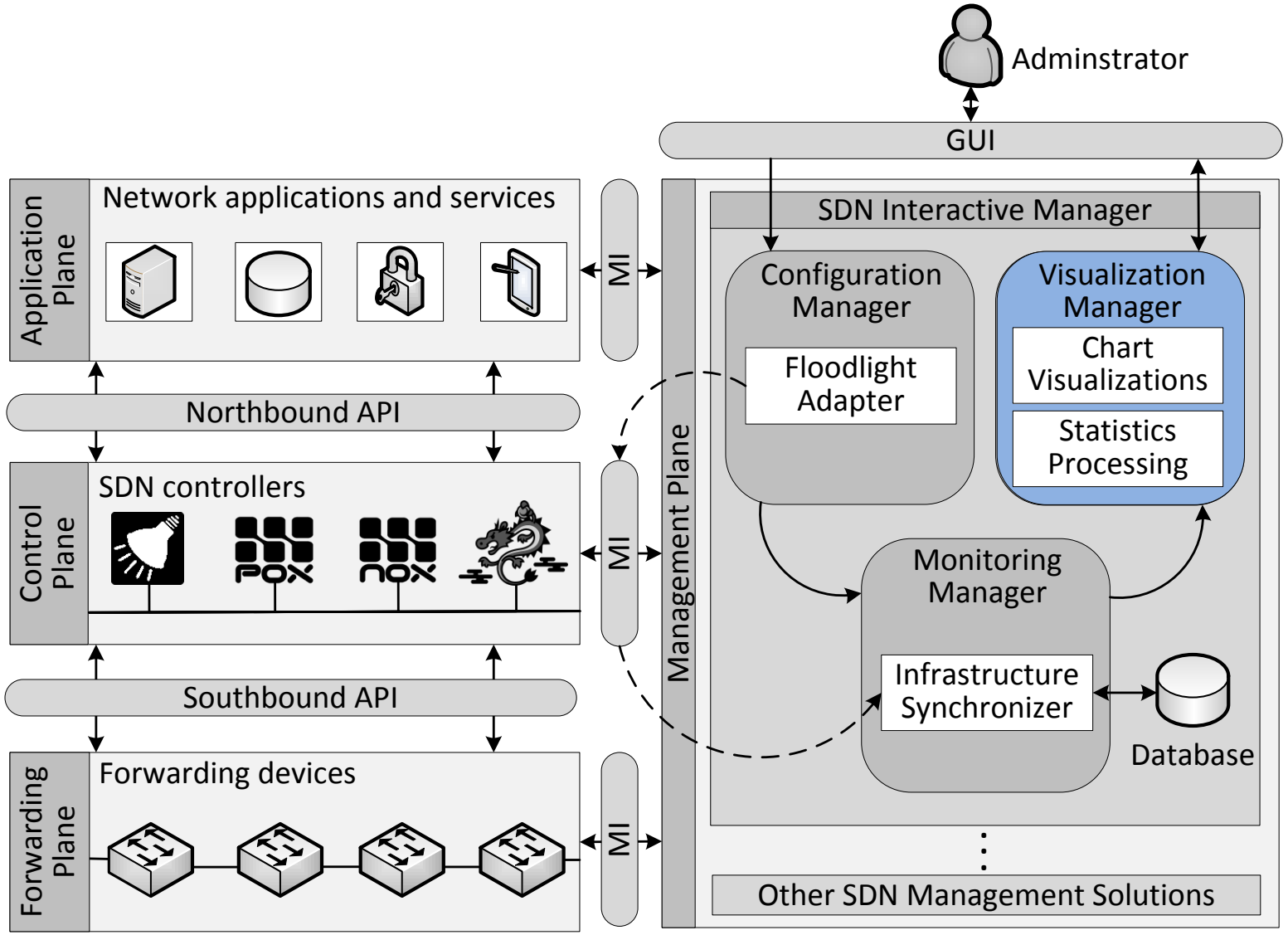
## Visualization Manager

Aggregates data providing interactive visualizations to the administrator

## Modules

*Chart Visualizations*

*Statistics Processing*



# Proposal - Conceptual Architecture

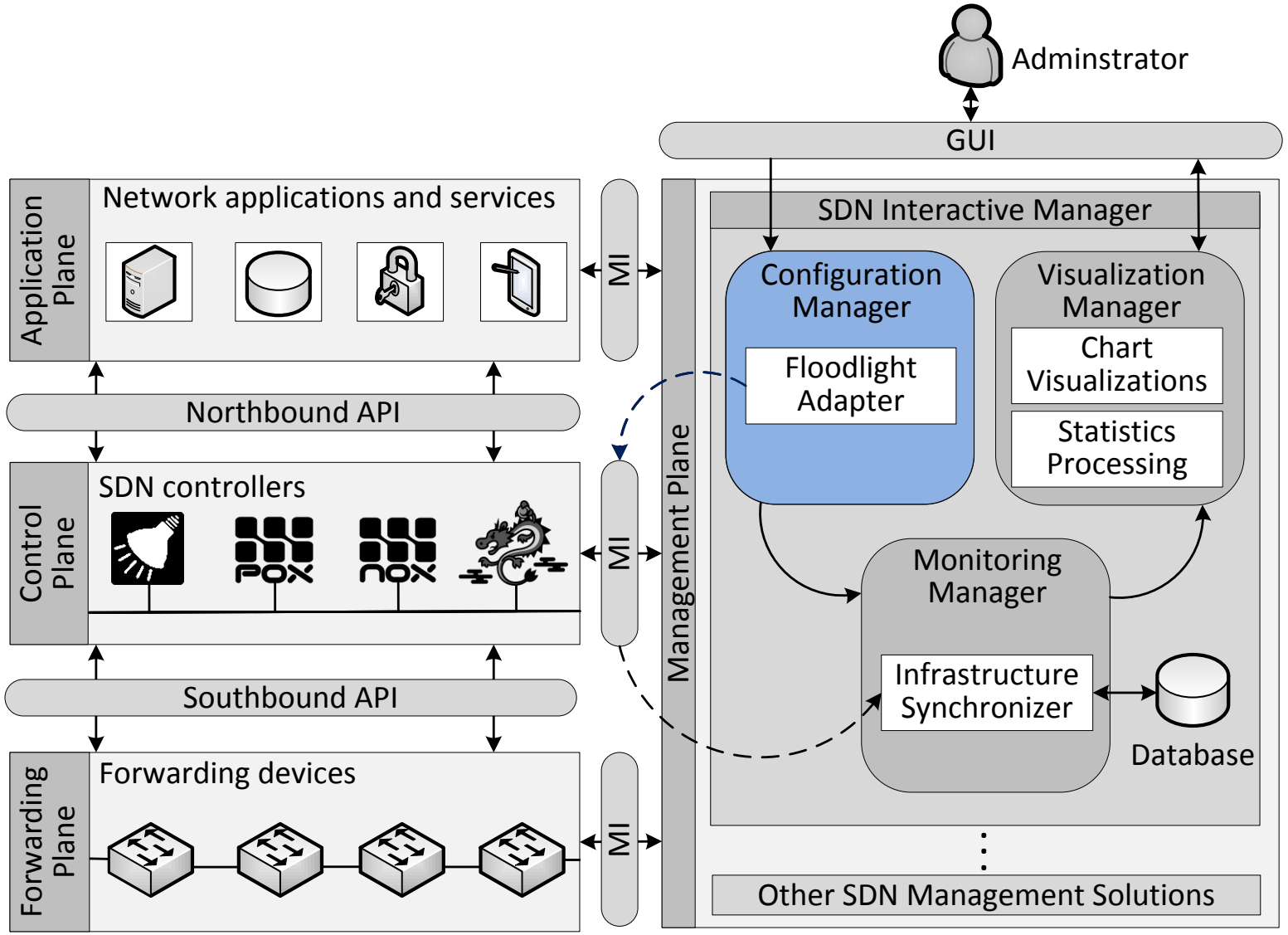


## Configuration Manager

Check and configure SDN-related parameters on network controllers

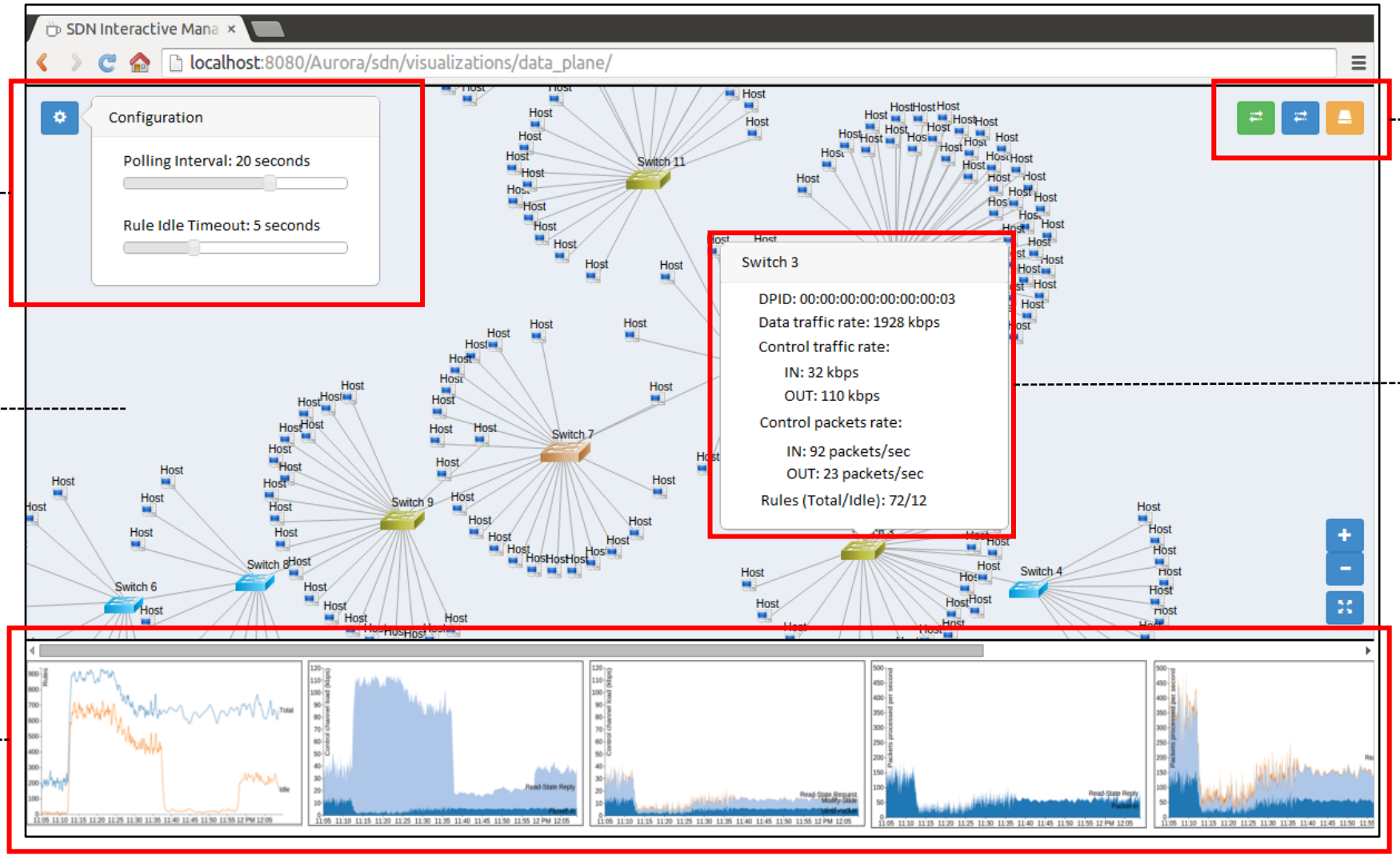
## Modules

*Floodlight Adapter*



# Experimental Evaluation - Prototype GUI






Configurations

Topology View

Interactive Charts

Perspectives

Nodes Information

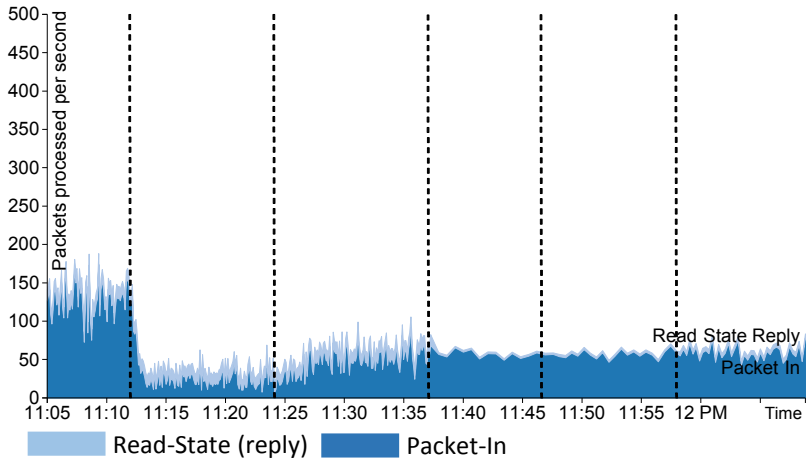
# Experimental Evaluation - Charts

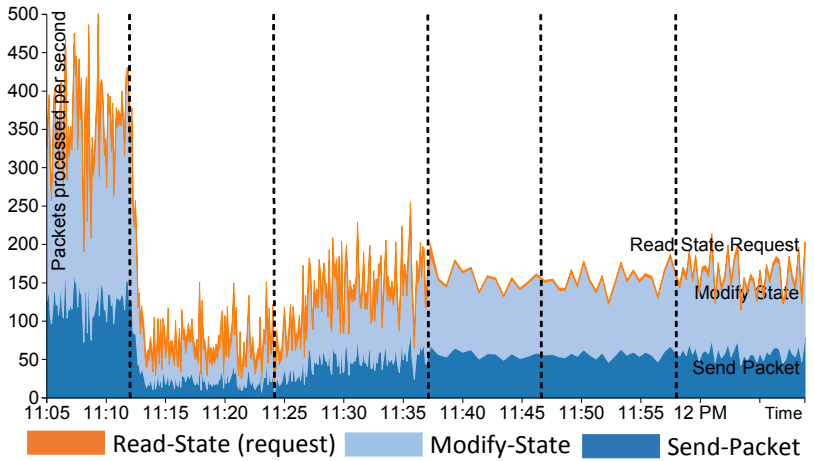


## Interactive Chart Visualizations

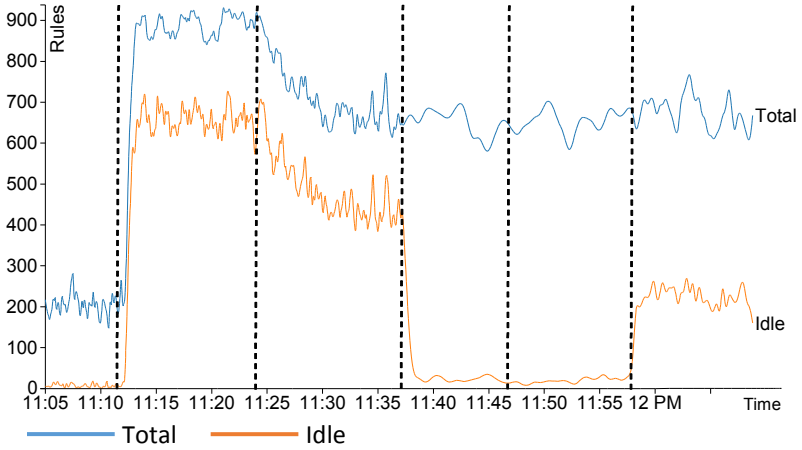
| Parameter | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| ***Idle timeout*** | 5 | 60 | 30 | 30 | 30 | 30 |
| ***Polling*** | 5 | 5 | 5 | 40 | 30 | 15 |

### Packets processed per second x idle timeout

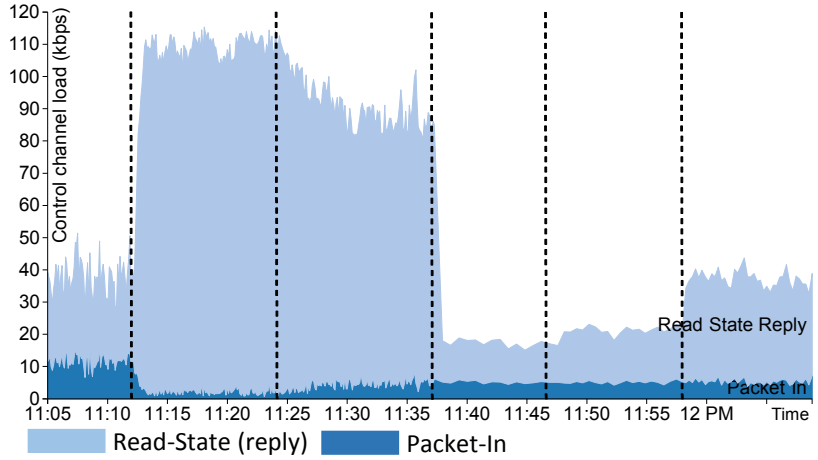


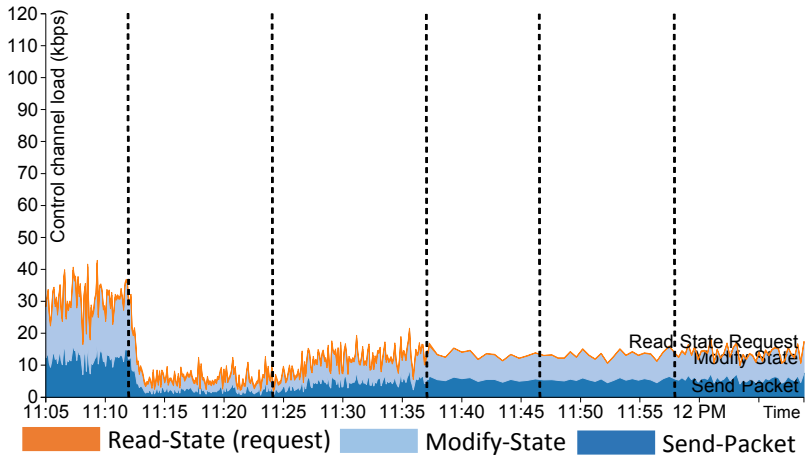




### Rules x idle timeout




### Control channel load x Idle timeout



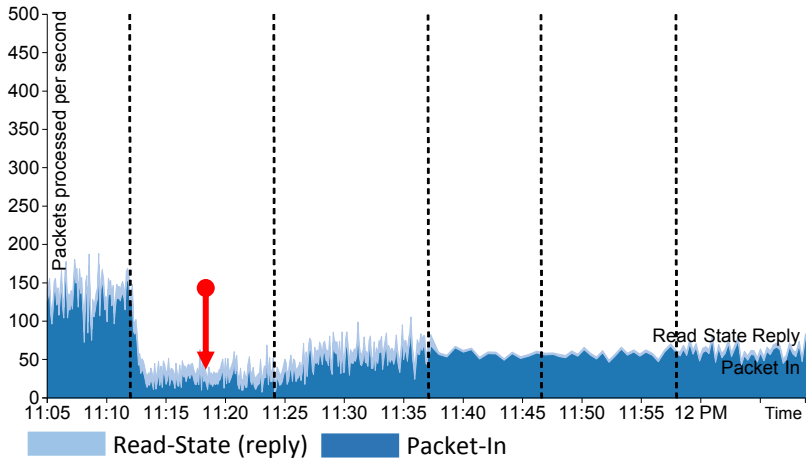
# Experimental Evaluation - Charts

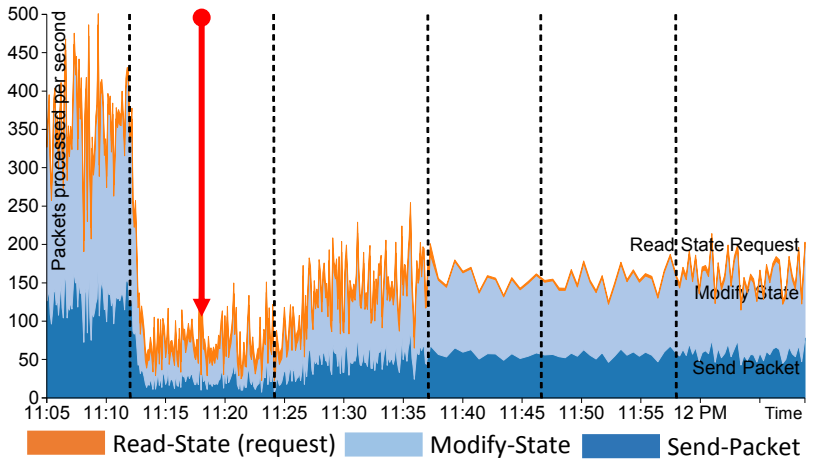


## Interactive Chart Visualizations

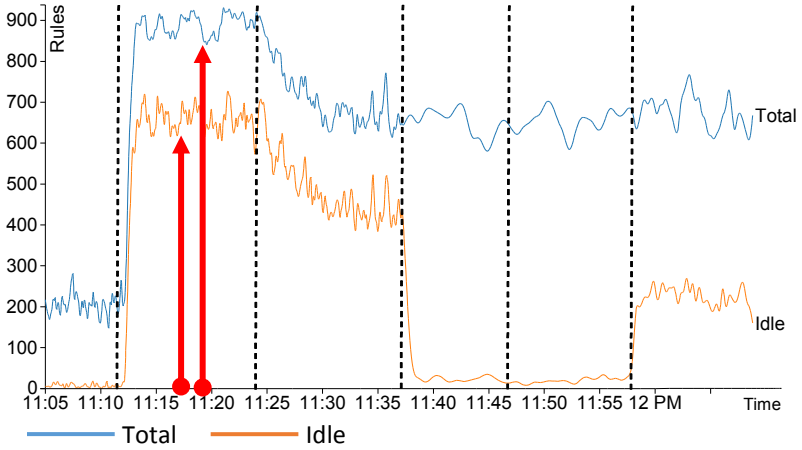
| Parameter | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| *Idle timeout* | 5 | 60 | 30 | 30 | 30 | 30 |
| *Polling* | 5 | 5 | 5 | 40 | 30 | 15 |

### Packets processed per second x idle timeout

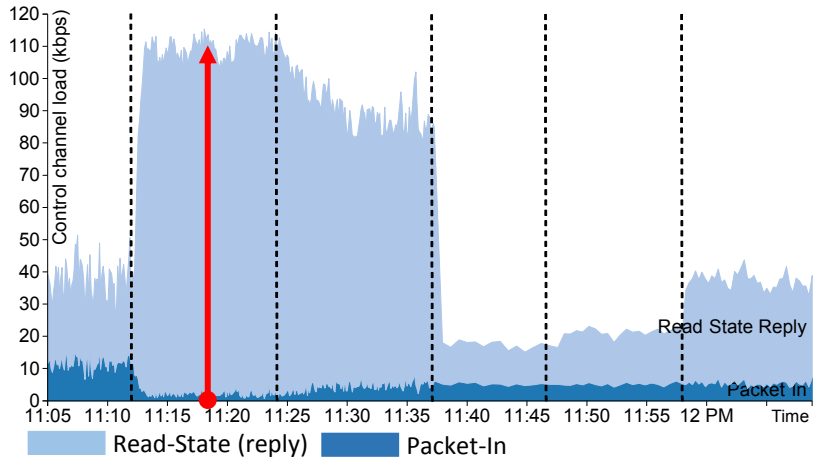


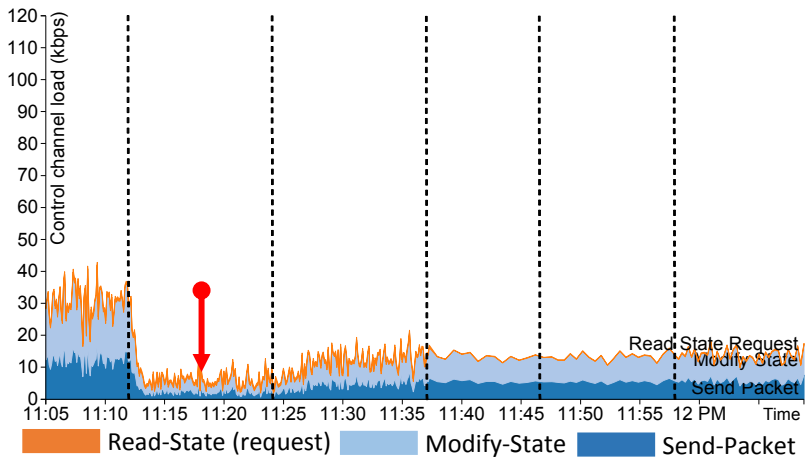




### Rules x idle timeout




### Control channel load x Idle timeout



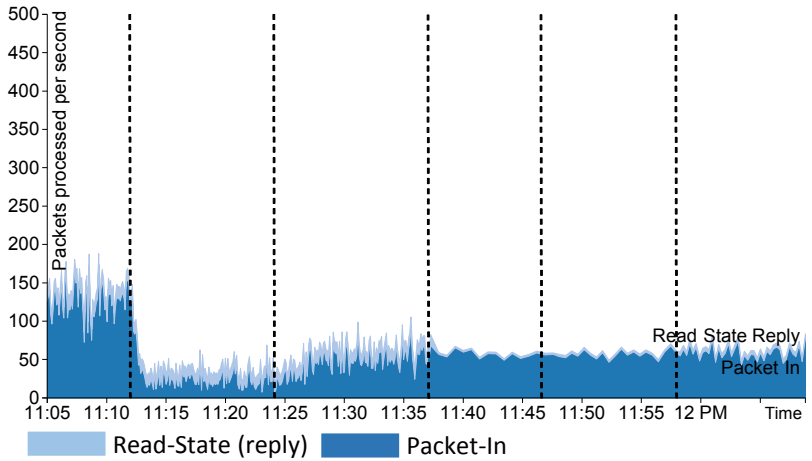
# Experimental Evaluation - Charts

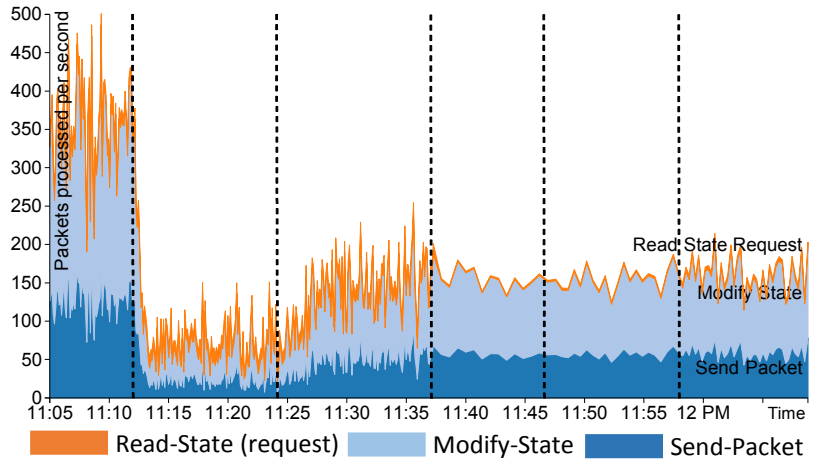


## Interactive Chart Visualizations

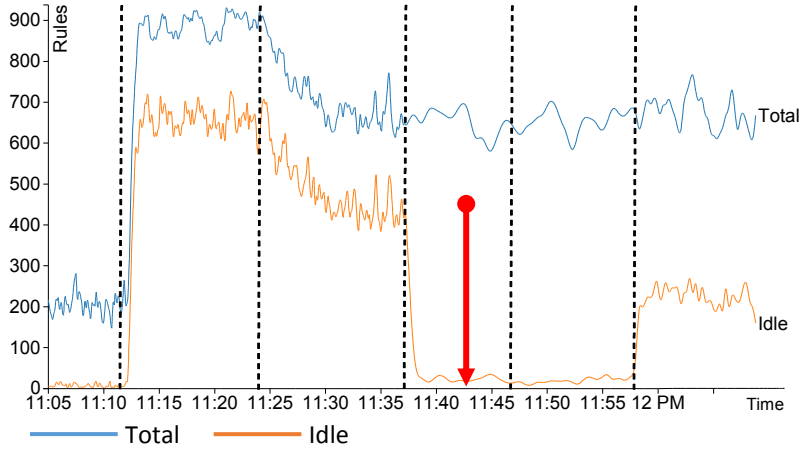
| Parameter | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| ***Idle timeout*** | 5 | 60 | 30 | 30 | 30 | 30 |
| ***Polling*** | 5 | 5 | 5 | 40 | 30 | 15 |

### Packets processed per second x idle timeout

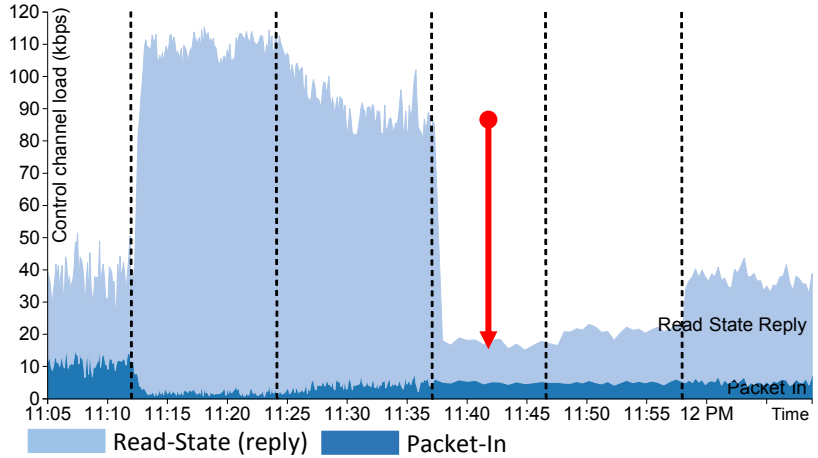


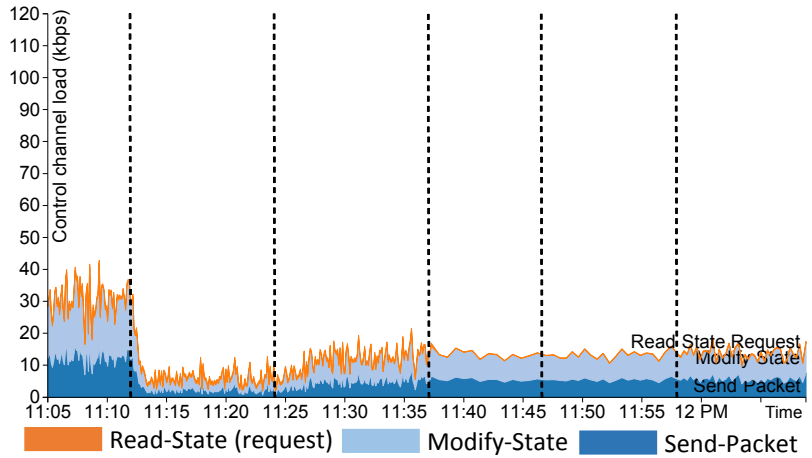




### Rules x idle timeout




### Control channel load x Idle timeout



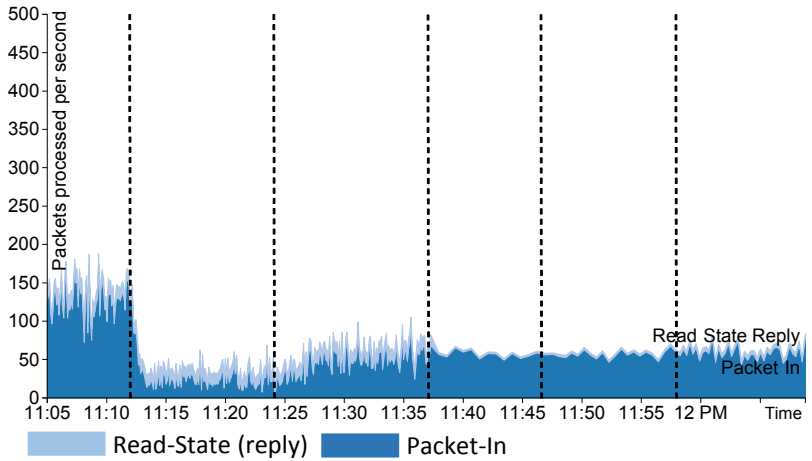
# Experimental Evaluation - Charts

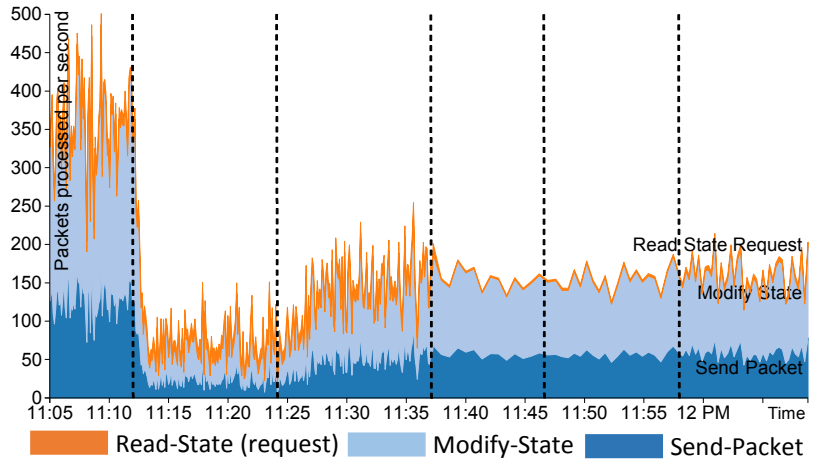


## Interactive Chart Visualizations

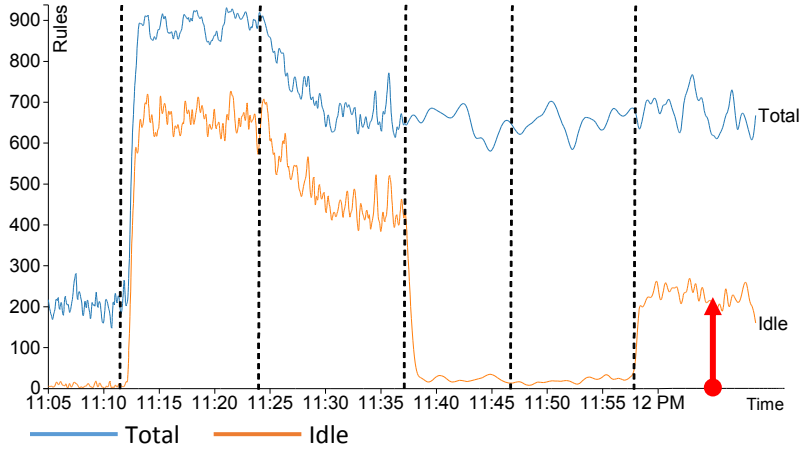
| Parameter | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| *Idle timeout* | 5 | 60 | 30 | 30 | 30 | 30 |
| *Polling* | 5 | 5 | 5 | 40 | 30 | 15 |

### Packets processed per second x idle timeout

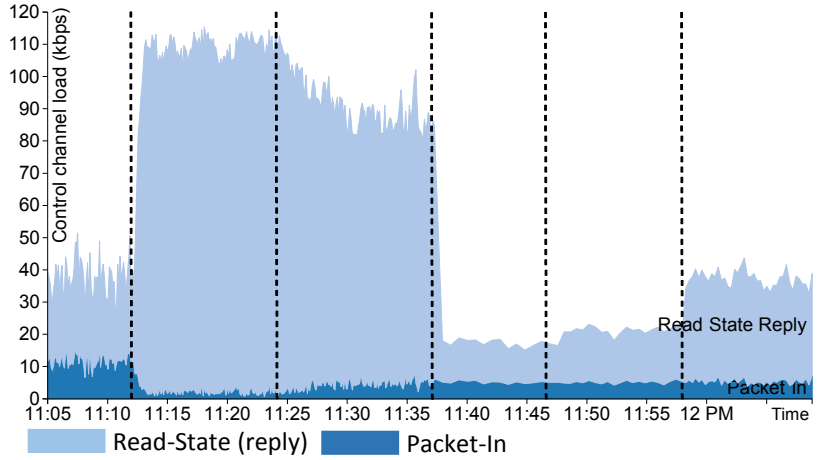


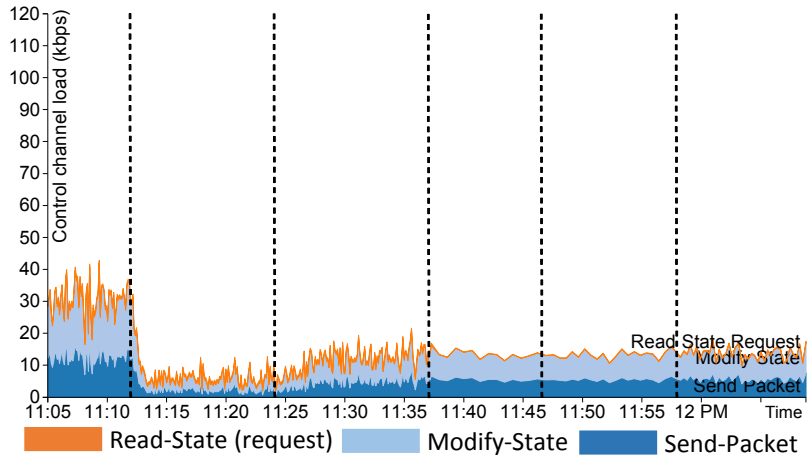




### Rules x idle timeout




### Control channel load x Idle timeout



# Conclusions & Open Questions



## Control Channel Analysis

The proportion of both resource usage and control channel load are affected by configuration of SDN-related parameters (*i.e.*, idle timeout of forwarding rules)

## Interactive Monitoring, Visualization, and Configuration

- Retrieves statistics about the control channel traffic
- Allows the administrator to interact with SDN
- Based on interactive visualizations, administrators are able identify potential issues and change configurations of SDN parameters

# Conclusions & Open Questions

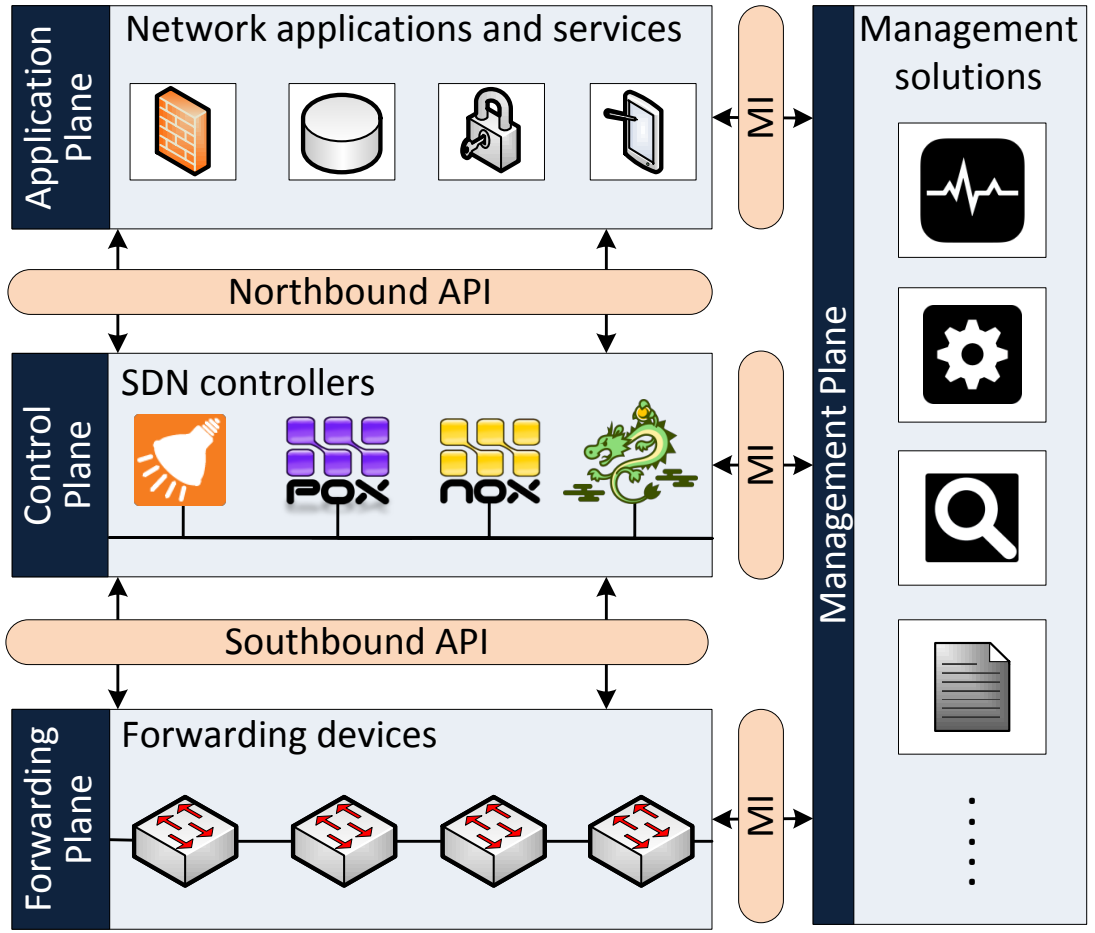


Deal with control channel statistics was not so simple.

- Different implementations of control channel handlers among controllers

- Absence of a common Management Interface (MI) to allow management solutions or even the administrator for himself to access data and control channel statistics

We argue that the standardization process of all these MIs could be an interesting may foster the development of SDN management solutions to manage any plane regardless of applications, controllers, or forwarding devices.



# References


[1] H. Kim and N. Feamster, “Improving Network Management with Software Defined Networking,” *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, February 2013.

[2] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, “FlowSense: Monitoring Network Utilization with Zero Measurement Cost,” in *Proceedings of the 14th International Conference on Passive and Active Measurement* (PAM), March 2013, pp. 31–41.

[3] S. Chowdhury, M. Bari, R. Ahmed, and R. Boutaba, “PayLess: A low cost network monitoring framework for Software Defined Networks,” in *Proceedings of the 14th IEEE/IFIP Network Operations and Management Symposium* (NOMS), May 2014, pp. 1–9.

[4] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, “OpenTM: Traffic Matrix Estimator for OpenFlow Networks,” in *Proceedings of the 11th International Conference on Passive and Active Measurement* (PAM), April 2010, pp. 201–210.

[5] Y. Zhang, “An Adaptive Flow Counting Method for Anomaly Detection in SDN,” in *Proceedings of the 9th ACM Conference on Emerging Networking Experiments and Technologies* (CoNEXT), December 2013, pp. 25–30.

[6] L. Jose, M. Yu, and J. Rexford, “Online Measurement of Large Traffic Aggregates on Commodity Switches,” in *Proceedings of the 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services* (Hot-ICE), October 2011, pp. 13–13.

[7] S. Hassas Yeganeh and Y. Ganjali, “Turning the Tortoise to the Hare: An Alternative Perspective on Event Handling in SDN,” in *Proceedings of the 2014 ACM International Workshop on Software-defined Ecosystems* (BigSystem), June 2014, pp. 29–32.

[8] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, “DevoFlow: Scaling Flow Management for Highperformance Networks,” in *Proceedings of the ACM SIGCOMM 2011 Conference*, August 2011, pp. 254–265.

[9] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, “Scalable Flow-based Networking with DIFANE,” in *Proceedings of the ACM SIGCOMM 2010 Conference*, August 2010, pp. 351–362.

# Thank you all!

# Questions?

***Pedro Heleno Isolani***
*Juliano Araujo Wickboldt*
*Cristiano Bonato Both*
*Lisandro Zambenedetti Granville*
*Juergen Rochol*

E-mail: phisolani@inf.ufrgs.br
Website: www.inf.ufrgs.br/~phisolani

July 16, 2015

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

.inf
INSTITUTO
DE INFORMÁTICA
UFRGS

Computer Networks
UFRGS