An Interactive Monitoring Tool for OpenFlow Networks

Wassapon Watanakeesuntorn
Nara Institute of Science and Technology

Outline

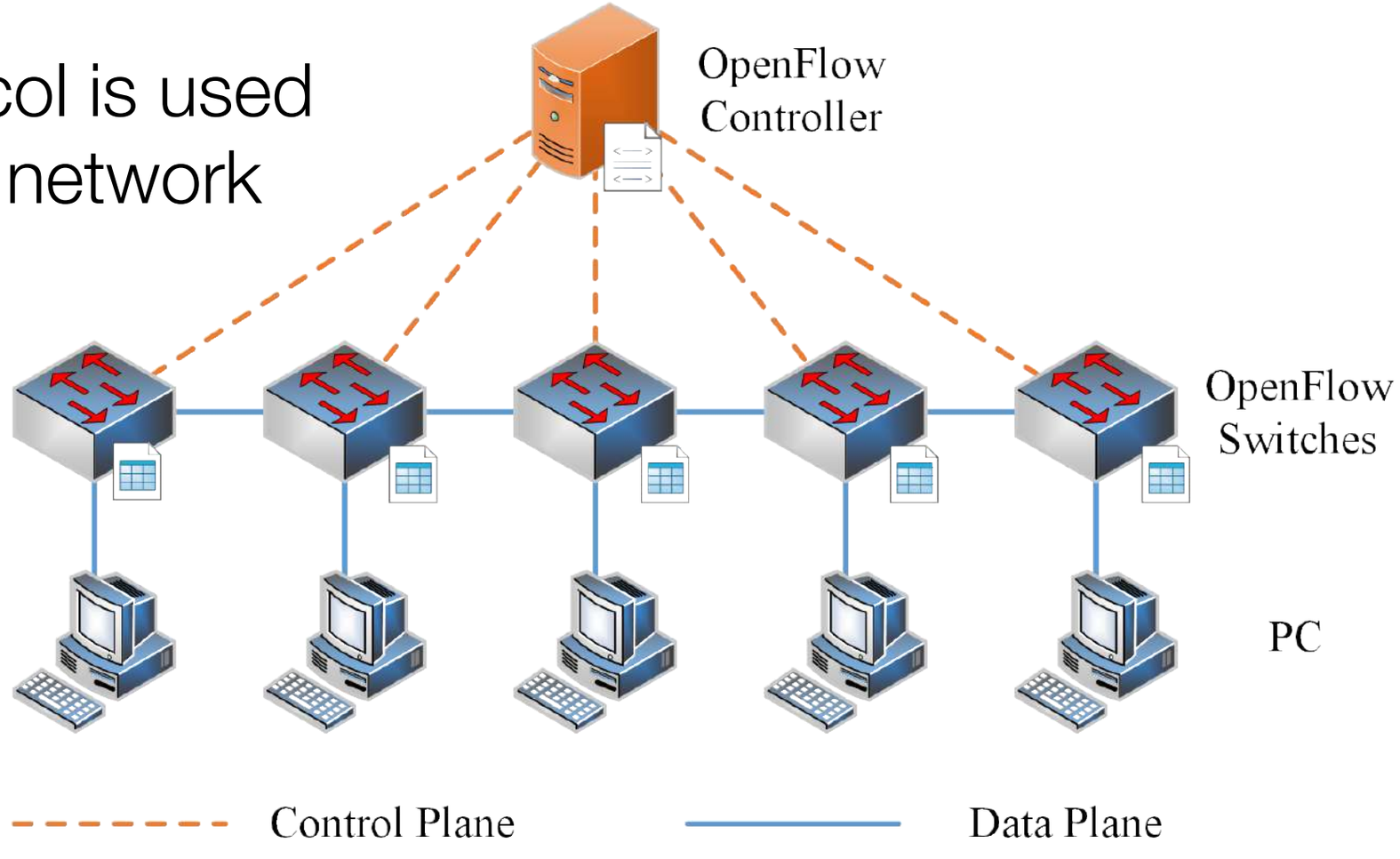
1. Introduction
2. Design & Implementation
    1. Monitoring Module
    2. Visualization Module
    3. Security Analysis Module
3. Experimental Result
4. Conclusion

Outline

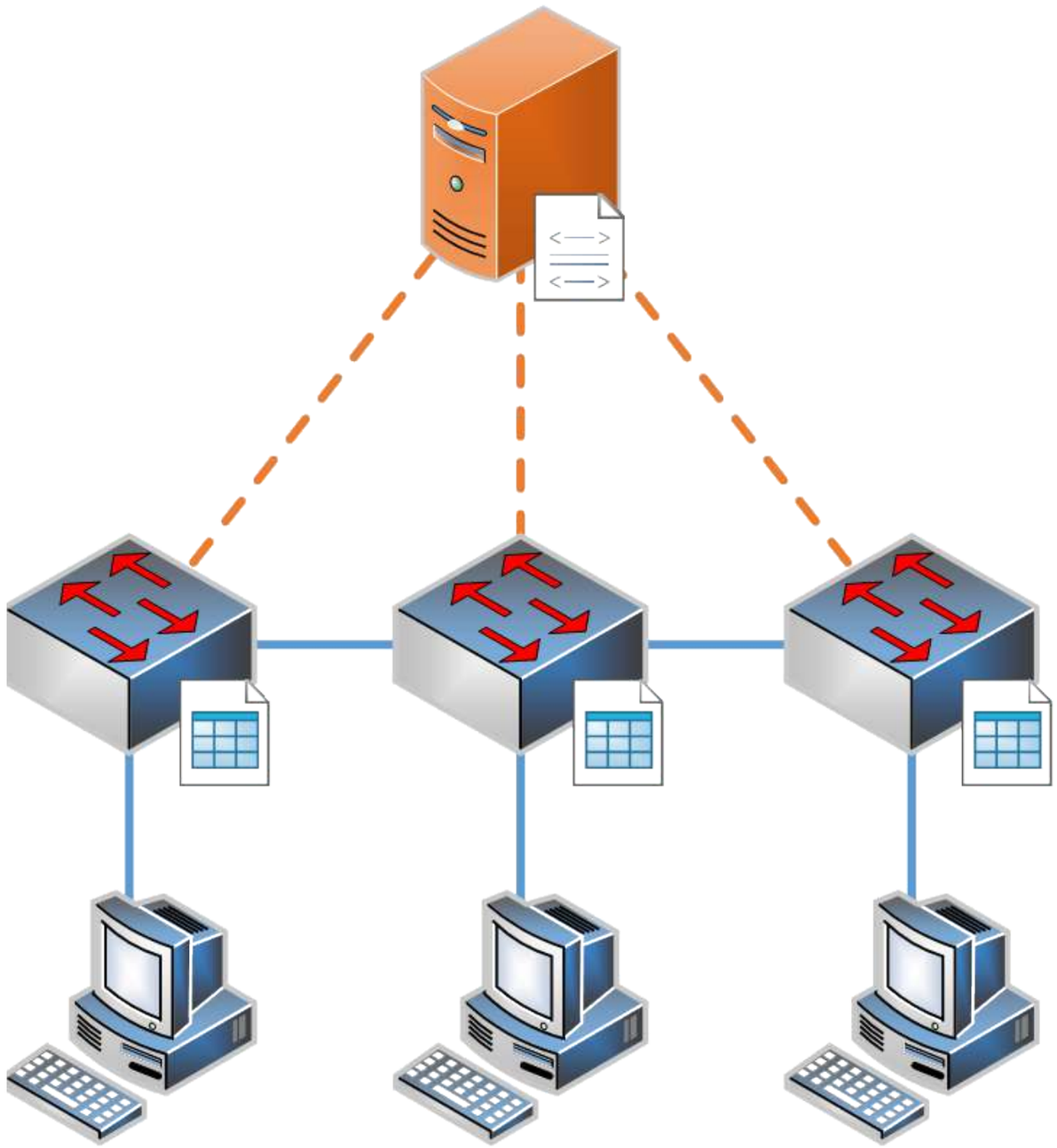
1. **Introduction**
2. Design & Implementation
    1. Monitoring Module
    2. Visualization Module
    3. Security Analysis Module
3. Experimental Result
4. Conclusion

Software-Defined Networking & OpenFlow

- SDN (Software-defined Networking) is a network technology that tries to provide centralized programmability of networks and simplify the management of a large scale network
- Decoupling data plane and control plane
- OpenFlow protocol is used to build the SDN network

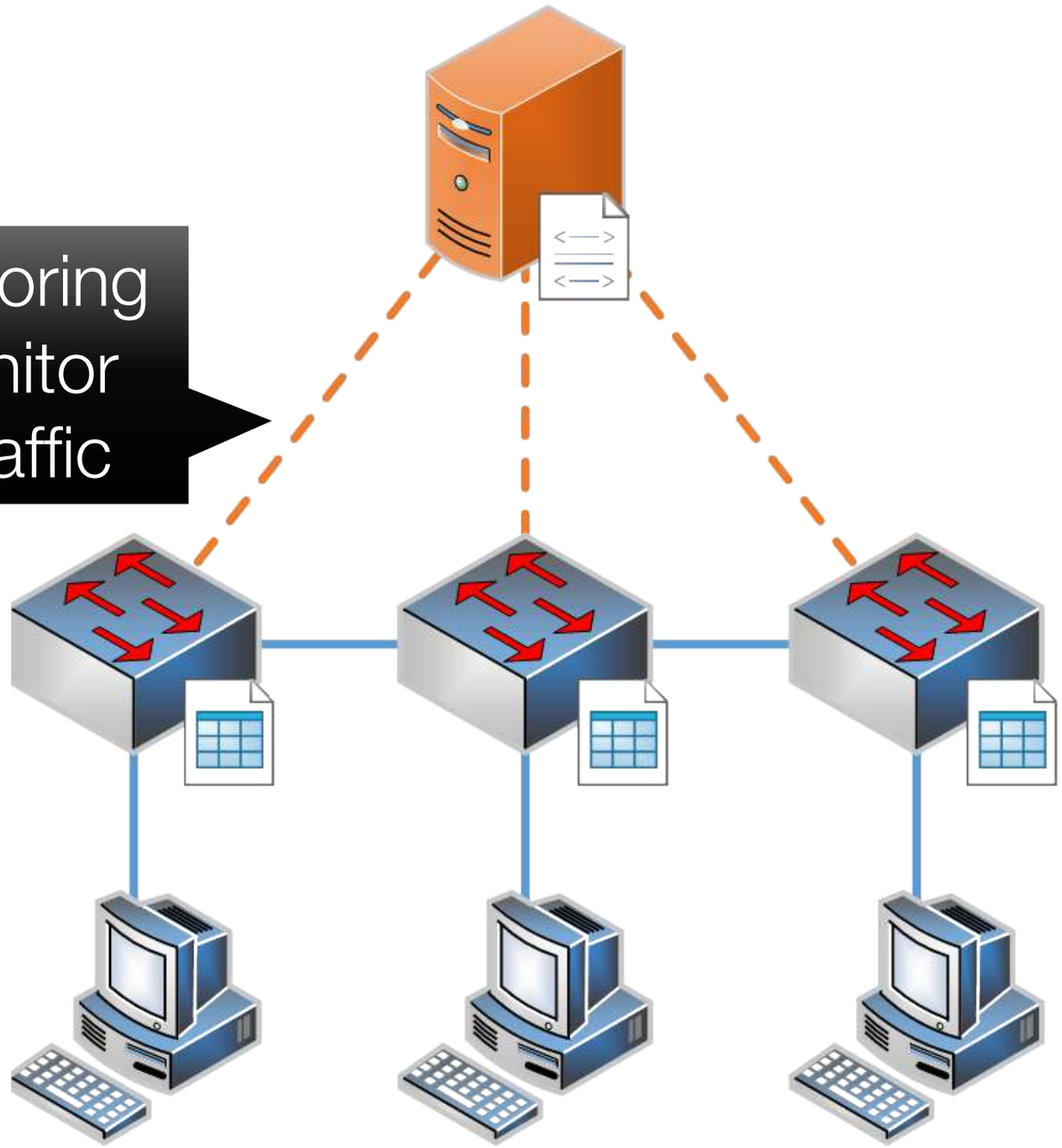


Issues of OpenFlow network



Issues of OpenFlow network

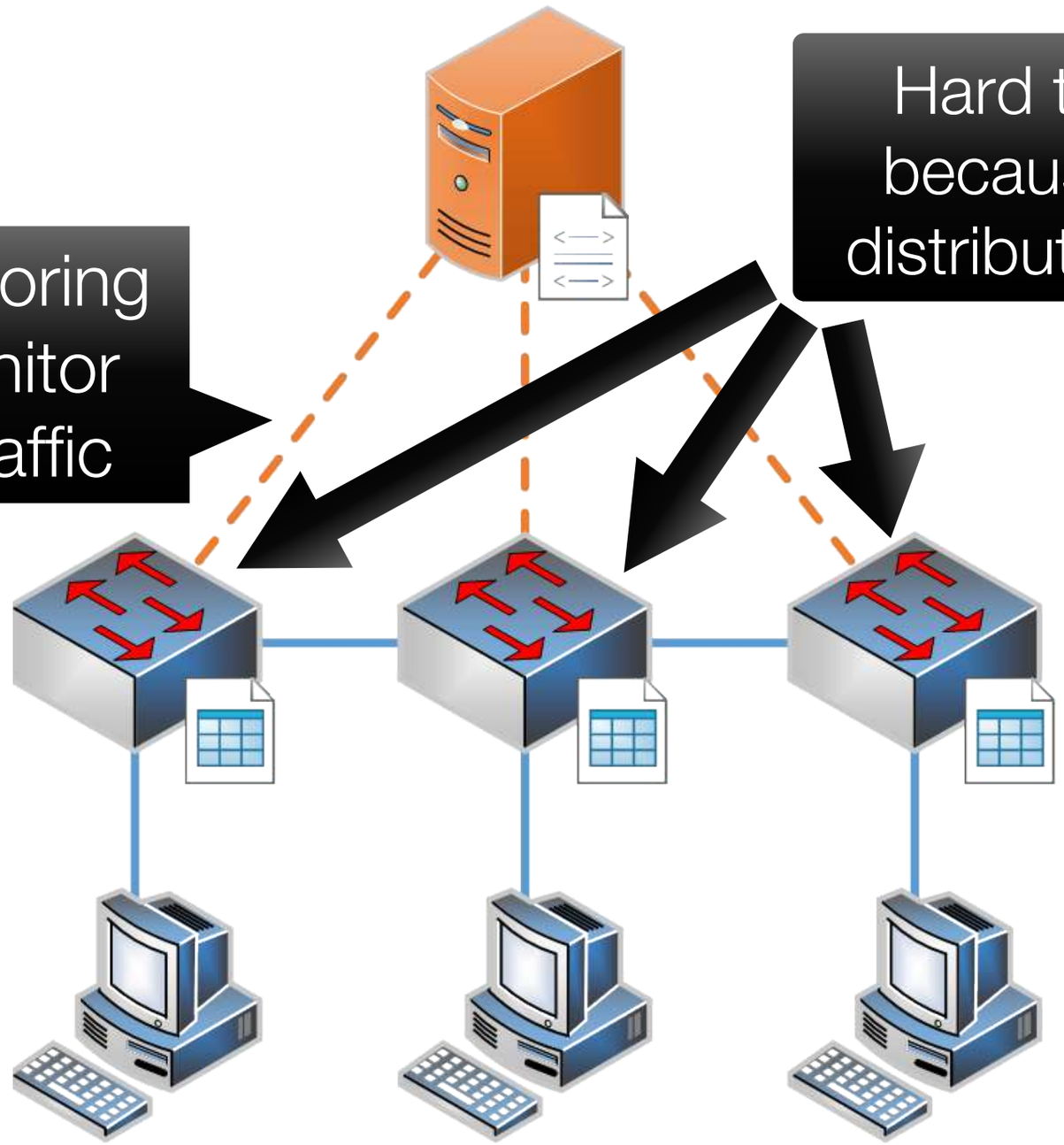
Traditional monitoring tool cannot monitor control plane traffic



Issues of OpenFlow network

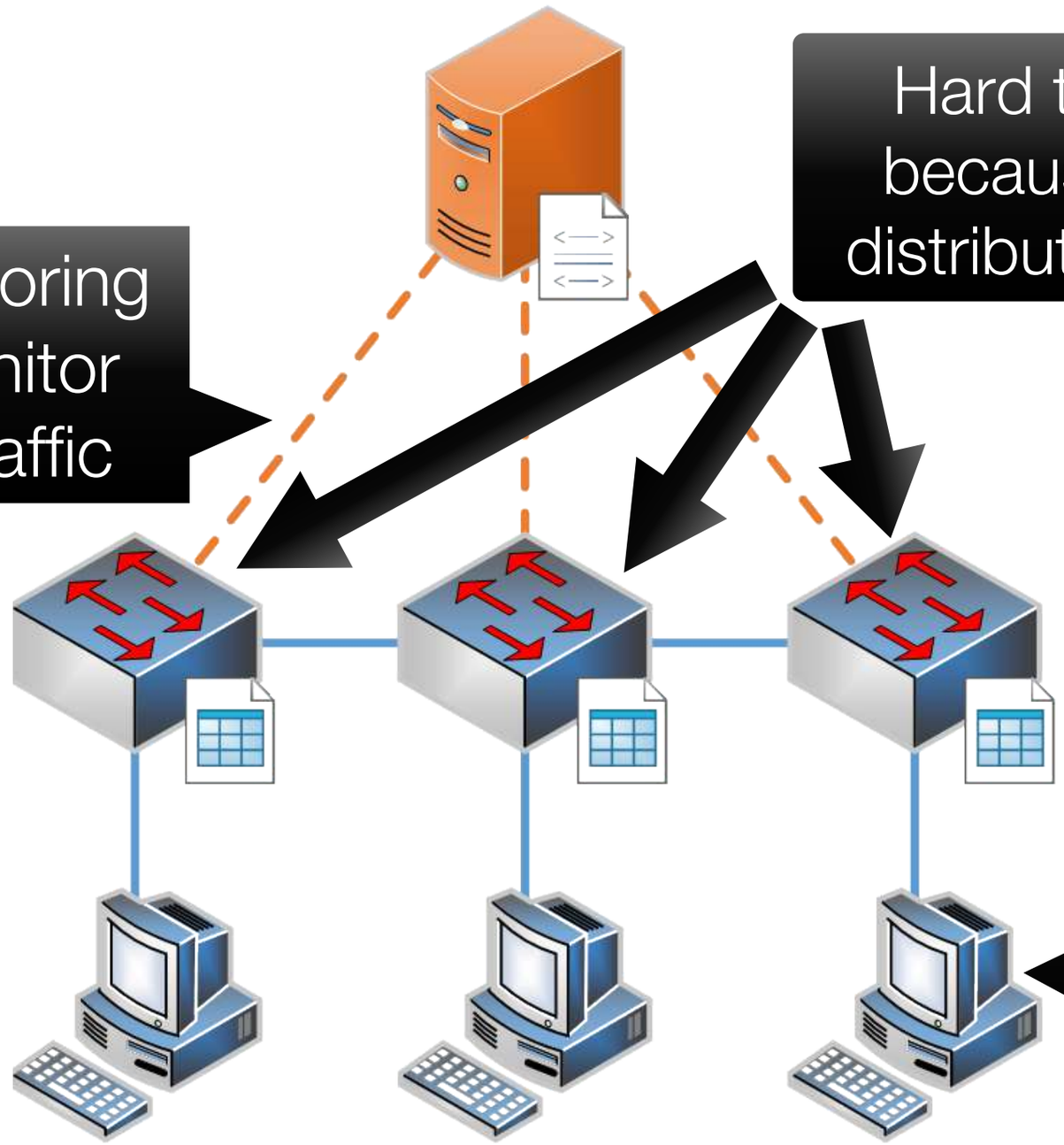
Traditional monitoring tool cannot monitor control plane traffic

Hard to understand because information distribute over network



Issues of OpenFlow network

Traditional monitoring tool cannot monitor control plane traffic

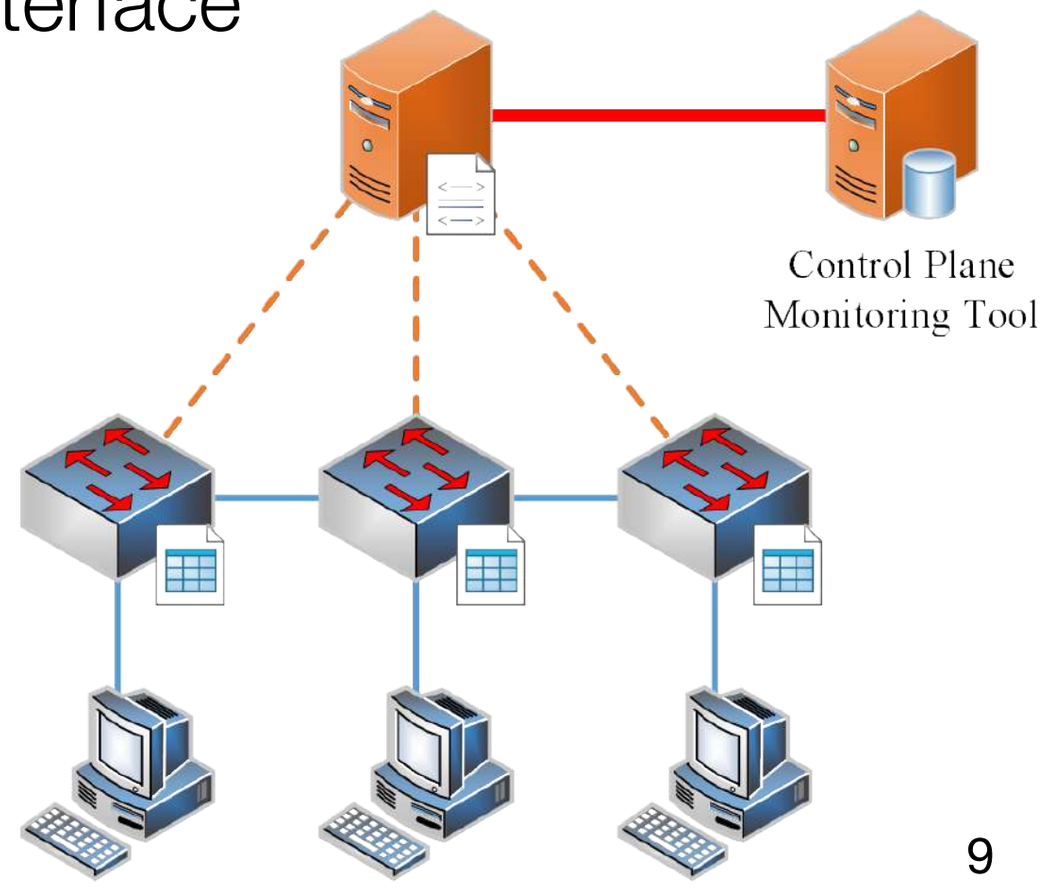


Hard to understand because information distribute over network

Difficult to detect security issues in the network

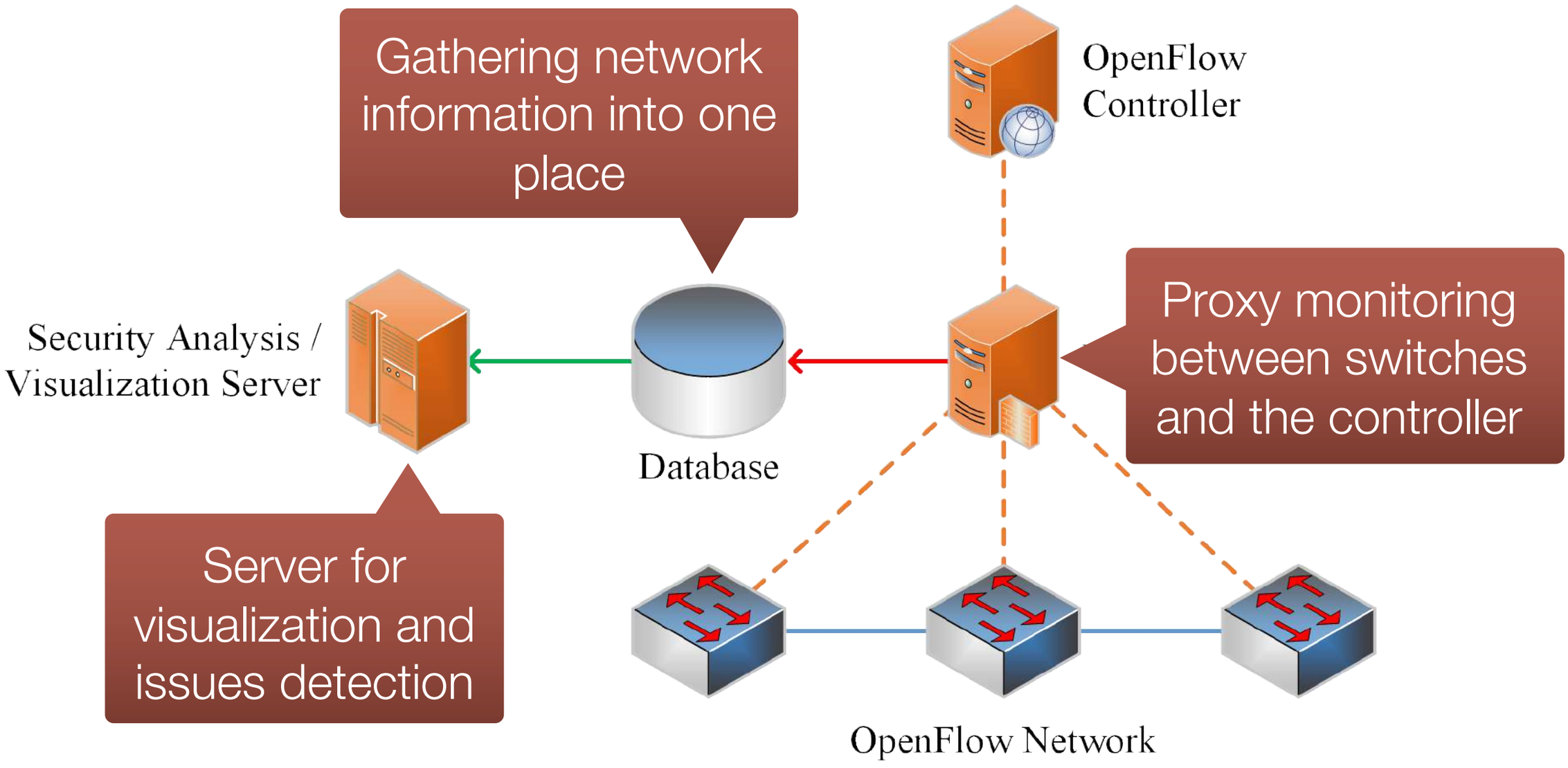
Research Objectives

- Develop a monitoring tool for OpenFlow network
  - Monitor communication on control plane
  - Gather information into one place
  - Visualize information on web interface
    - Network topology
    - Flow tables
- Analyze security issues in network
  - DDoS detection



Approach

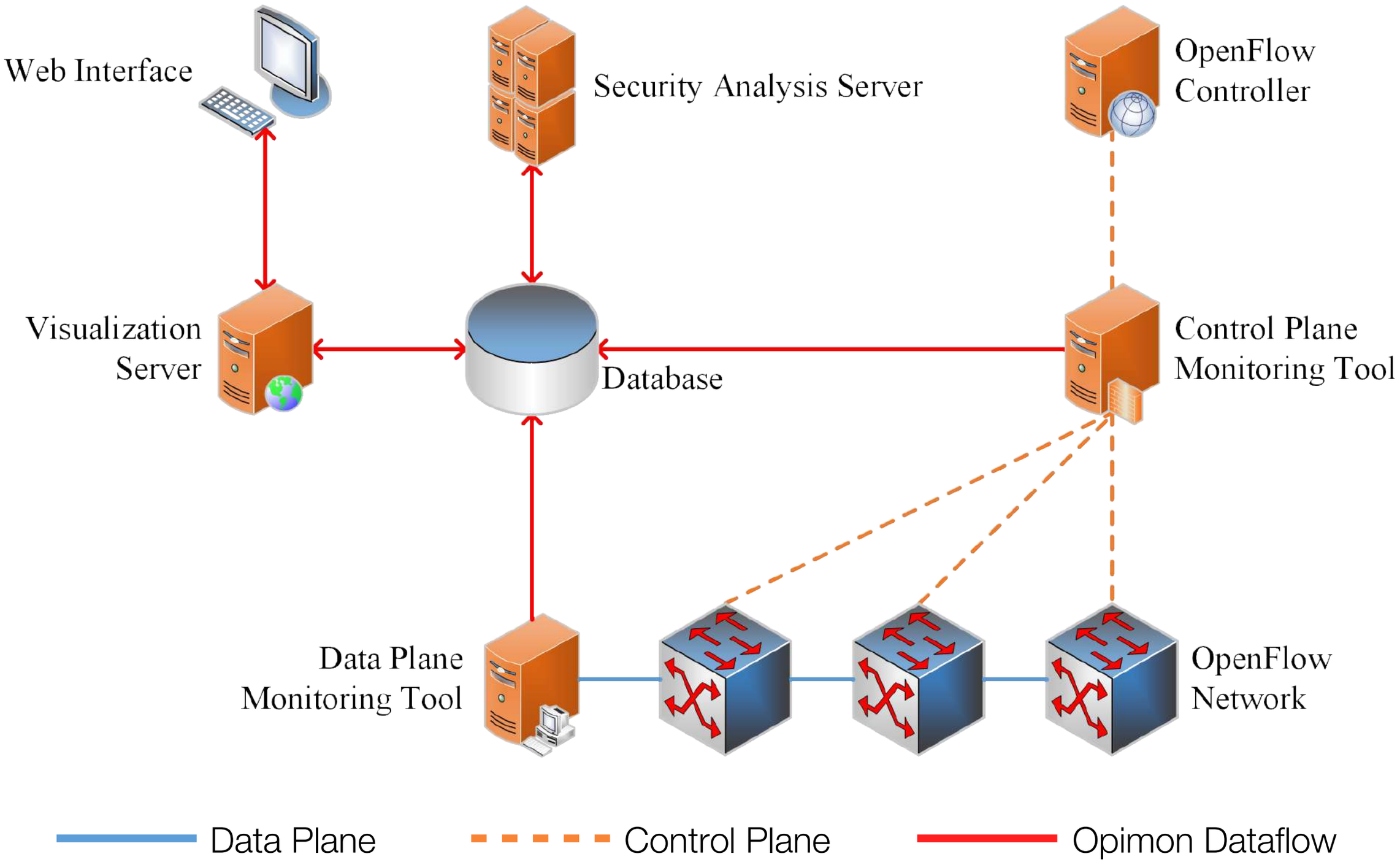
- Proxy monitoring between switch and controller
  - Without modify the implementation of the controller and switches



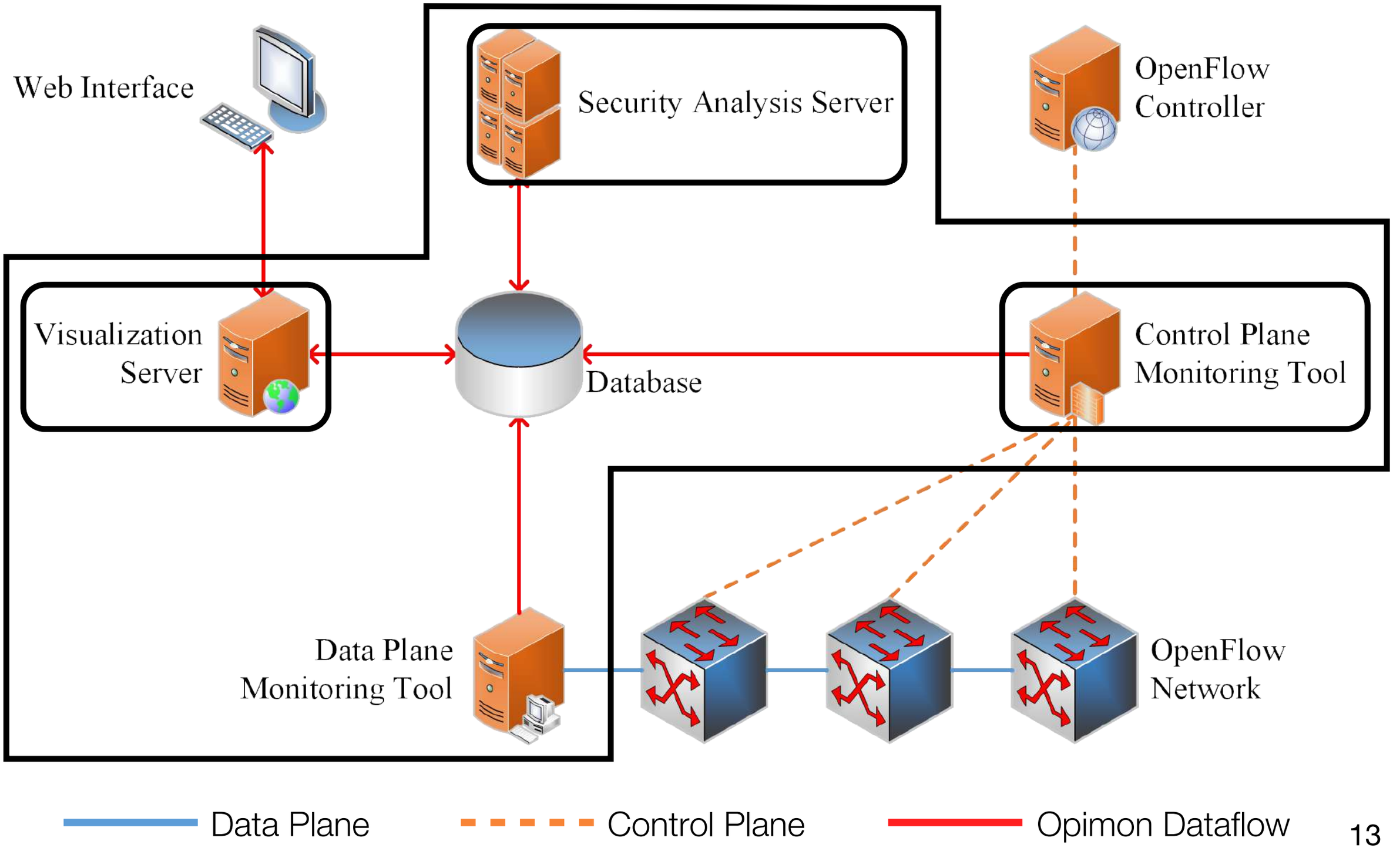
Outline

1. Introduction
2. **Design & Implementation**
    1. Monitoring Module
    2. Visualization Module
    3. Security Analysis Module
3. Experimental Result
4. Conclusion

OpenFlow Interactive Monitoring Tool (Opimon)



OpenFlow Interactive Monitoring Tool (Opimon)

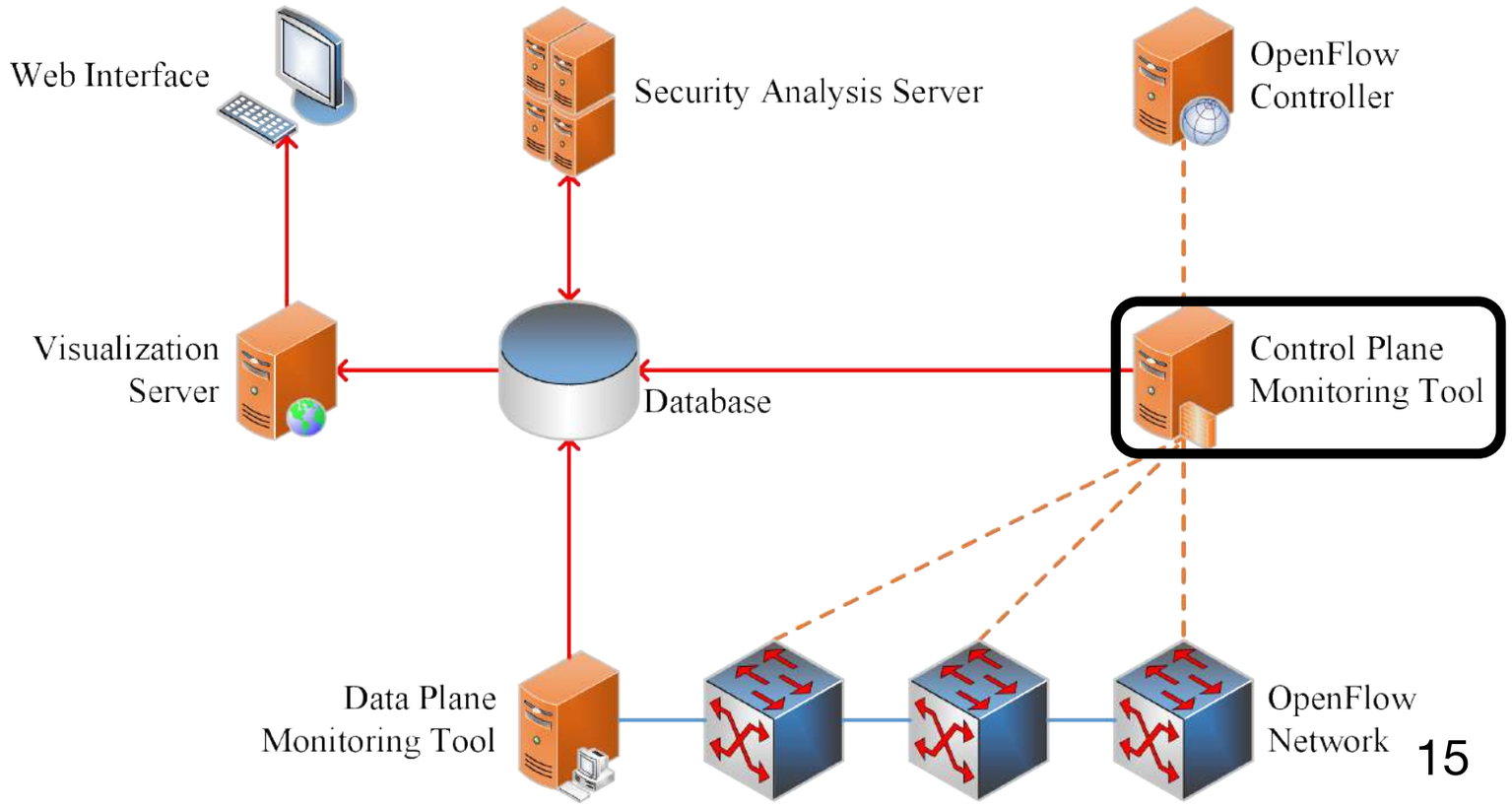


Outline

1. Introduction
2. Design & Implementation
    1. **Monitoring Module**
    2. Visualization Module
    3. Security Analysis Module
3. Experimental Result
4. Conclusion

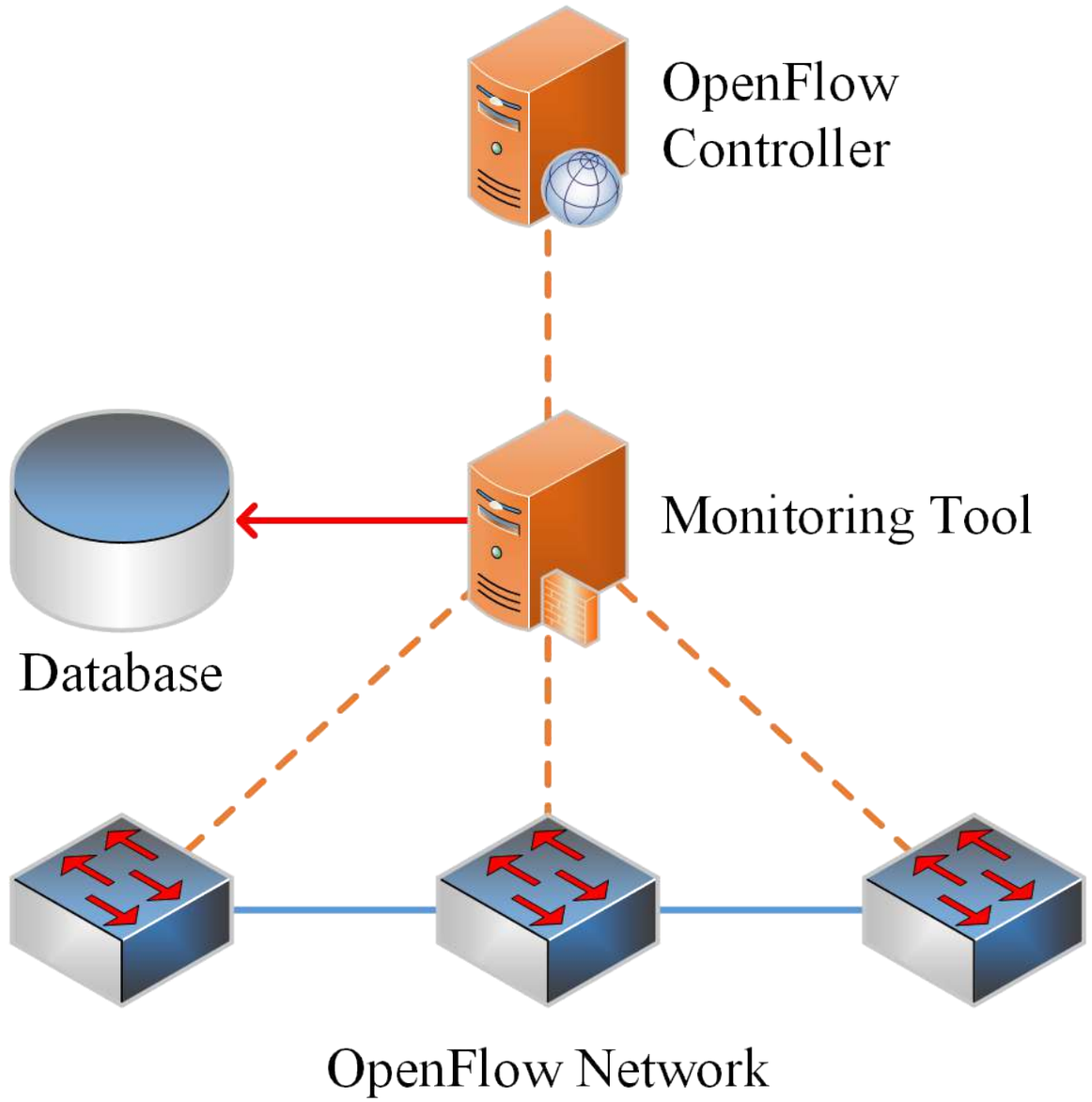
Monitoring Module

- Lie between OpenFlow controller and OpenFlow switches
- Monitor the communication on control plane
- Collect the messages into database



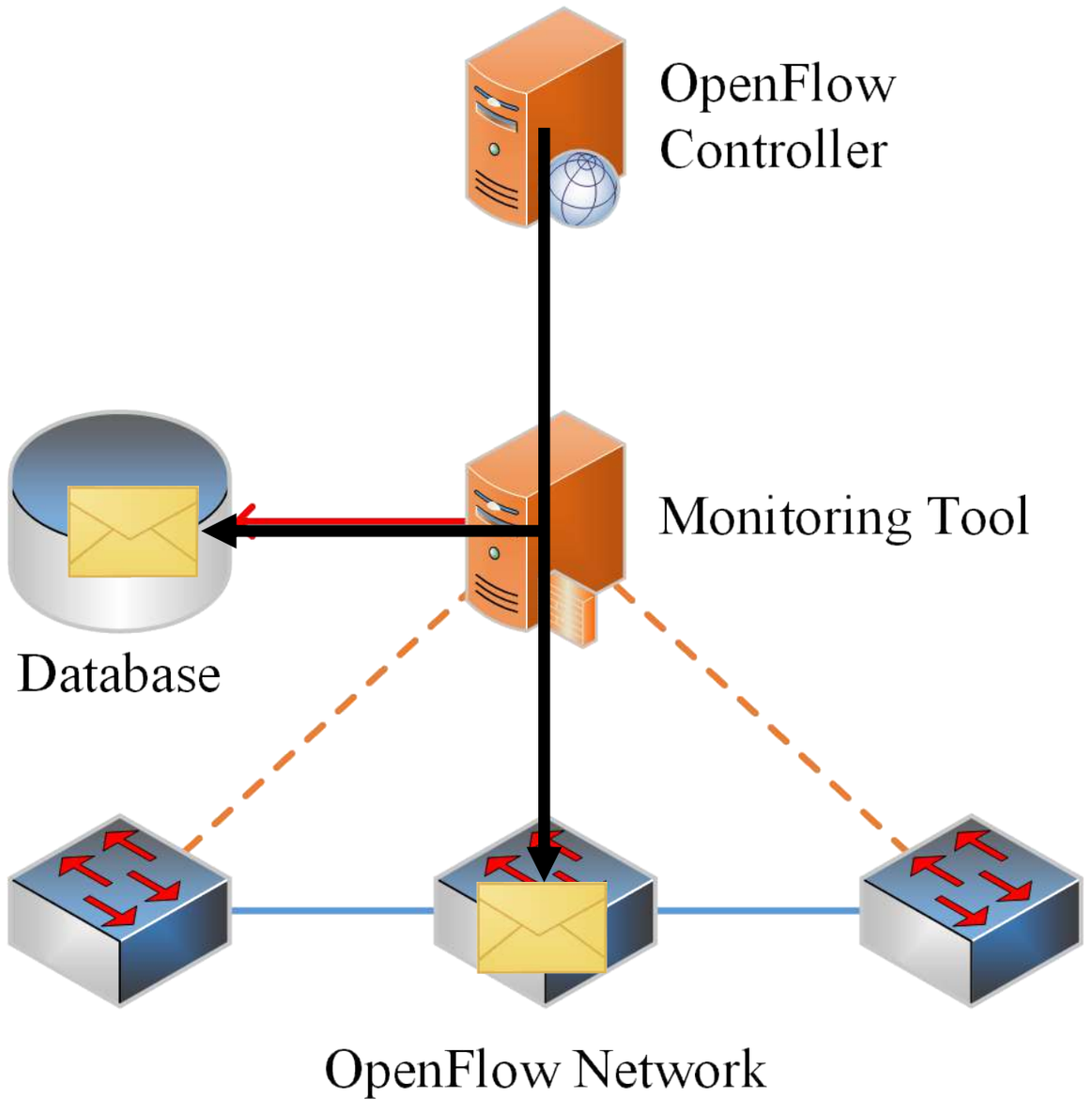
Monitoring Module

- Two approaches to ensure the transparency of the monitoring process
  - Collecting all messages that are forwarded through the monitoring module
  - Injecting messages or packets explicitly from the monitoring module to query the information of the switches



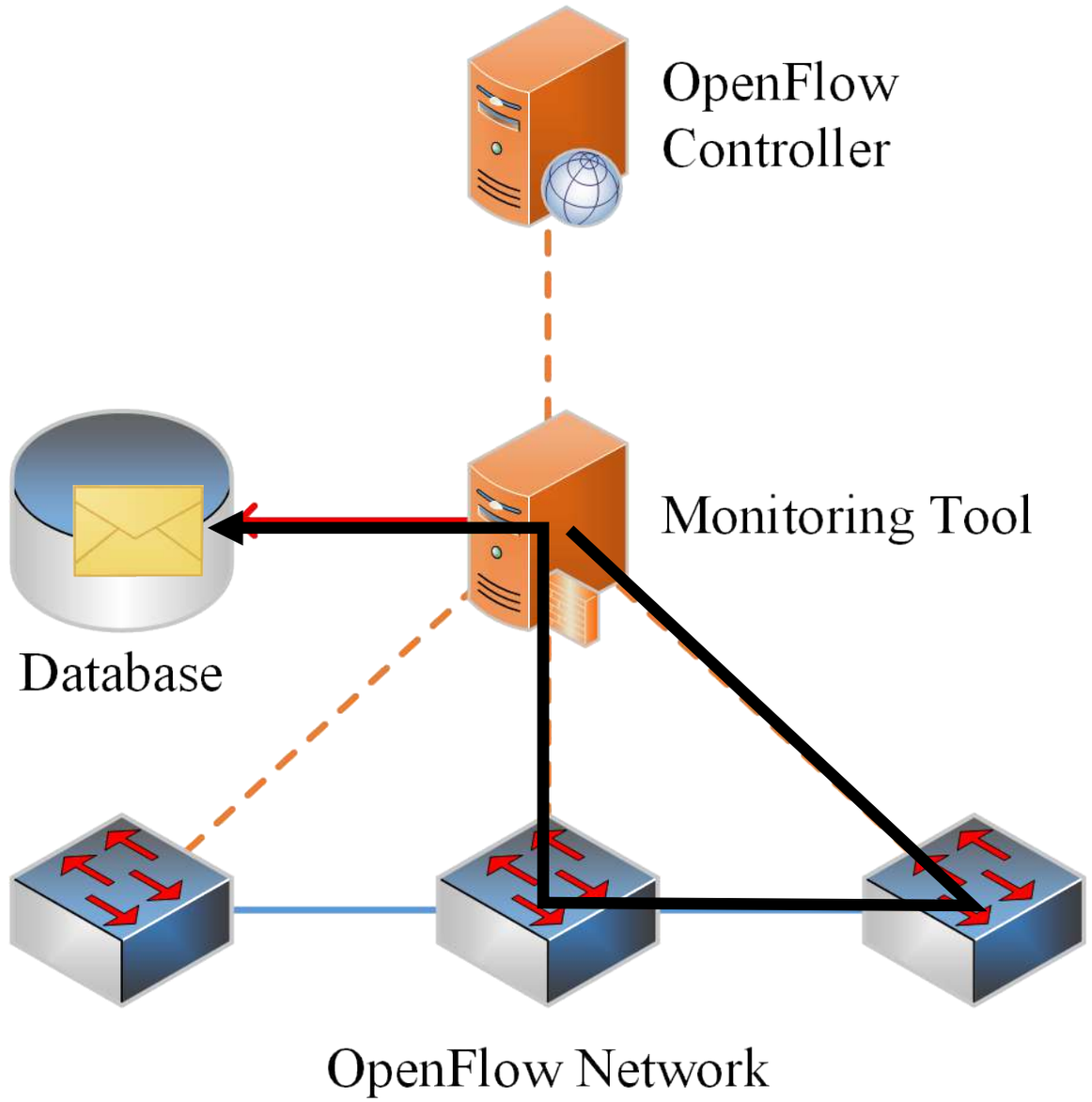
Monitoring Module

- Two approaches to ensure the transparency of the monitoring process
  - Collecting all messages that are forwarded through the monitoring module
  - Injecting messages or packets explicitly from the monitoring module to query the information of the switches



Monitoring Module

- Two approaches to ensure the transparency of the monitoring process
  - Collecting all messages that are forwarded through the monitoring module
  - Injecting messages or packets explicitly from the monitoring module to query the information of the switches

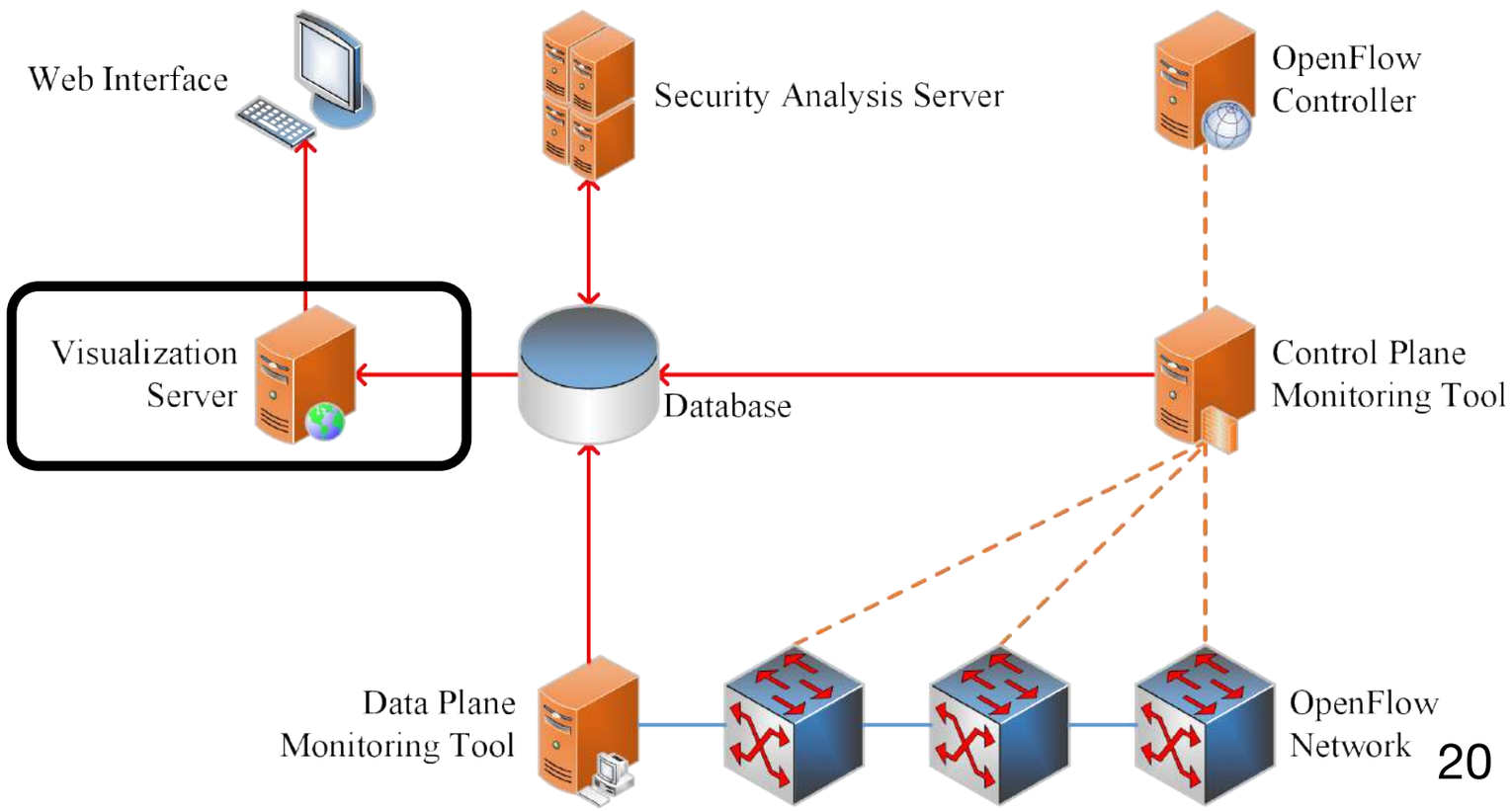


Outline

1. Introduction
2. Design & Implementation
    1. Monitoring Module
    2. **Visualization Module**
    3. Security Analysis Module
3. Experimental Result
4. Conclusion

Visualization Module

- Visualization Server is used for processing raw collected data into JSON format
- Show information on web interface
  - Network topology
  - Switch information
  - Flow tables

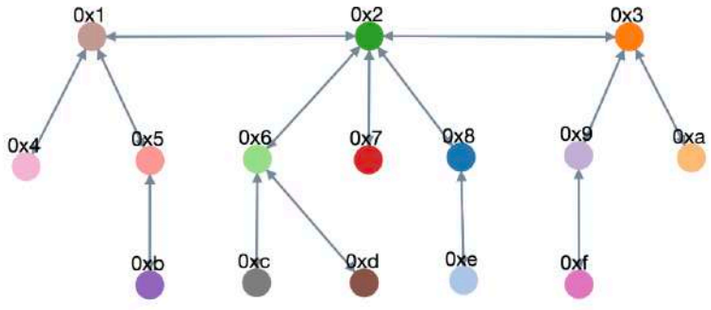


Network Topology

Thu Aug 01 2019 02:36:51 GMT+0900 (Japan Standard Time)

Refresh

Submit



Switch ID : 0x1

Switch Detail

Port	MAC Address
1	36:95:e6:b8:dd:73
• Received packets : 185 • Transmitted packets : 79 • Received bytes : 14154 • Transmitted bytes : 6078 • Packets dropped by RX : 0 • Packets dropped by TX : 0	• Receive errors : 0 • Transmit errors : 0 • Frame alignment errors : 0 • Packet with RX overrun : 0 • CRC errors : 0 • Collisions : 0
2	ce:df:9f:6c:4e:51
• Received packets : 38 • Transmitted packets : 223 • Received bytes : 2972 • Transmitted bytes : 17034 • Packets dropped by RX : 0 • Packets dropped by TX : 0	• Receive errors : 0 • Transmit errors : 0 • Frame alignment errors : 0 • Packet with RX overrun : 0 • CRC errors : 0 • Collisions : 0
3	ea:81:ff:7b:52:1c
65534	2e:9e:ec:57:97:4c

Flow Table (Switch ID: 0x1)

Match	Actions
• Wildcard : 3678454 • Switch Input Port : 1 • Destination MAC Address : ca:bc:ea:68:53:f1 • Idle Timeout : 0 • Hard Timeout : 0	• Type : 0 (OFPActionOutput) • Switch Output Port : 2 • Max Length : 65509
• Wildcard : 3678454 • Switch Input Port : 2 • Destination MAC Address : 46:1c:2a:5d:92:79 • Idle Timeout : 0 • Hard Timeout : 0	• Type : 0 (OFPActionOutput) • Switch Output Port : 1 • Max Length : 65509

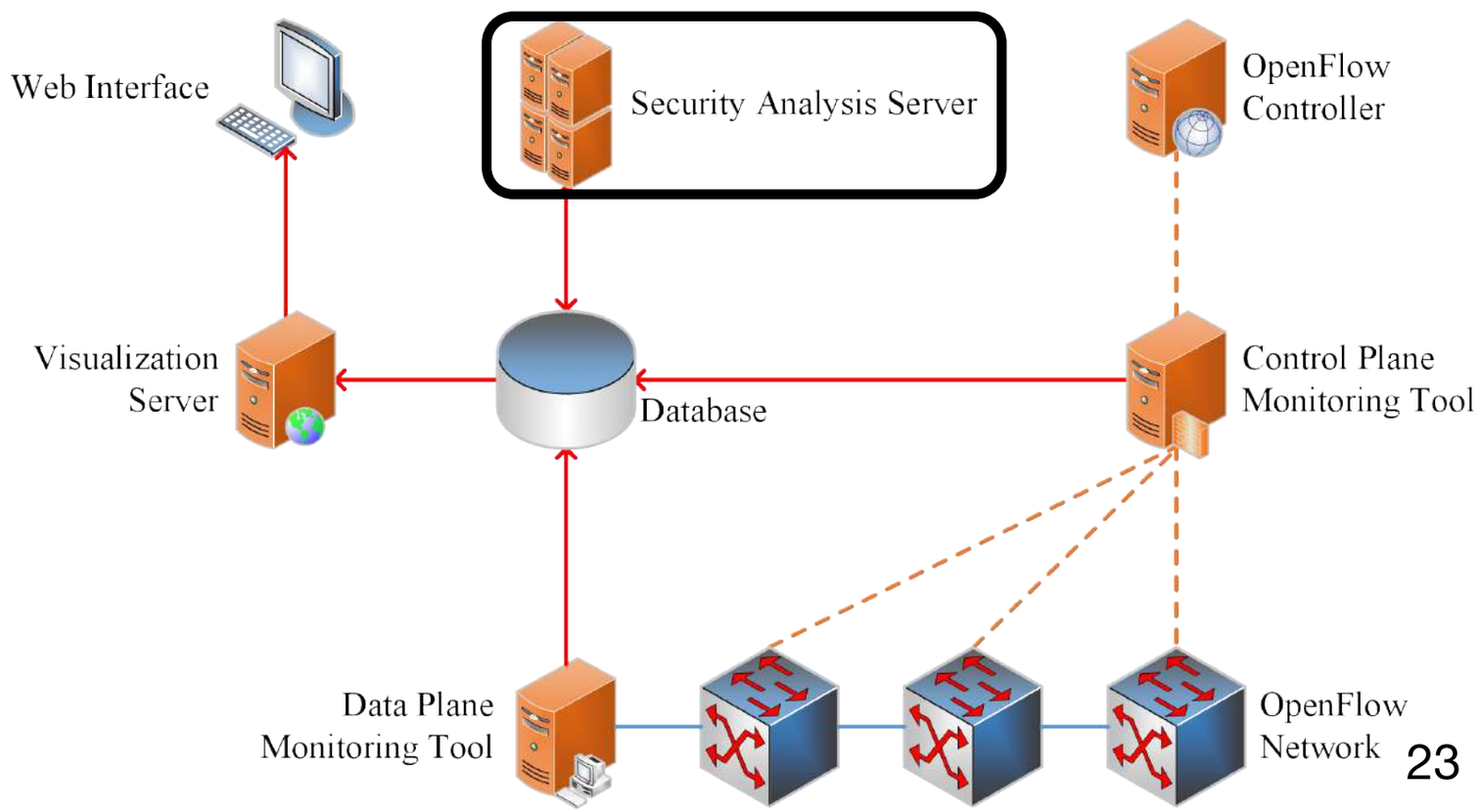
Visualization Tool | Web Interface

Outline

1. Introduction
2. Design & Implementation
    1. Monitoring Module
    2. Visualization Module
    3. **Security Analysis Module**
3. Experimental Result
4. Conclusion

Security Analysis Module

- Use machine learning techniques to detect DDoS attack
  - Support Vector Machine (SVM) and Deep Feed Forward (DFF) are used to compare the performance
  - Use DARPA dataset for training a machine learning model



Dataset

Dataset

- 2009 DARPA Intrusion Detection: Background traffic
- DARPA-2009 DDoS Attack-20091105: SYN flood DDoS attack

Number of samples

Dataset	DDoS attack	Normal	Total
Time window aggregated	335	365	700
Packet specific (S)	331	369	700
Packet specific (L)	481,903	518,097	1,000,000

Security Analysis Module

Selected Features

Time Windows Aggregated Features

- 16 features: # of src/dst IPs, # of src/dst ports, etc.
- Aggregate packet information for a certain time window

Packet Specific Features

- 26 features: MAC, IP, port, Protocol, etc.
- Extract value of each packet

Outline

1. Introduction
2. Design & Implementation
    1. Monitoring Module
    2. Visualization Module
    3. Security Analysis Module
3. **Experimental Result**
4. Conclusion

Evaluation Methodology

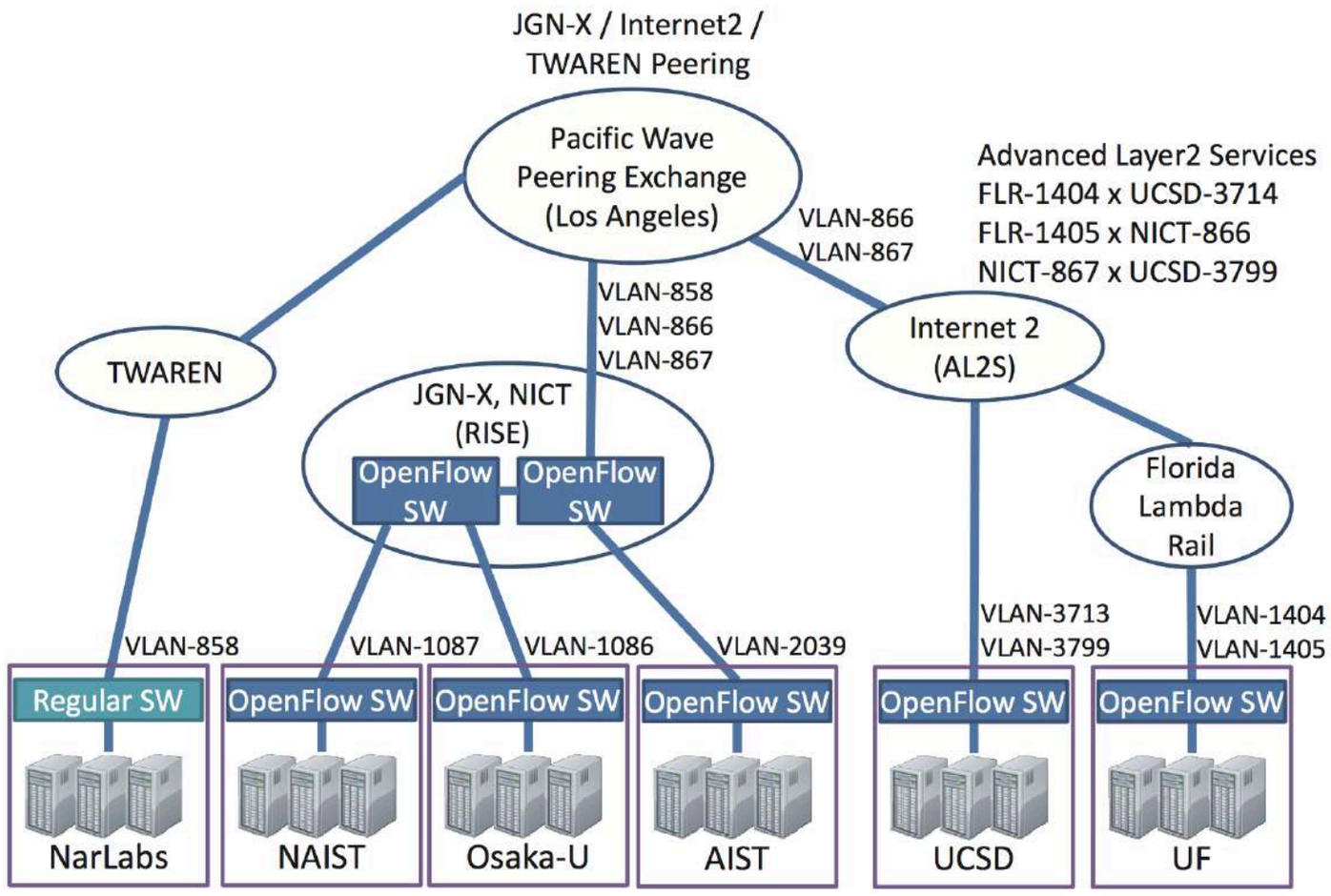
- Evaluation of the monitoring module
  - Evaluation on international network testbed
  - Benchmarking of the monitoring tool
- Evaluation of the security analysis module
  - Accuracy of DDoS detection

Evaluation Methodology

- Evaluation of the monitoring module
  - **Evaluation on international network testbed**
  - Benchmarking of the monitoring tool
- Evaluation of the security analysis module
  - Accuracy of DDoS detection

Evaluation on International Network Testbed

- Evaluated the practical usage of the Opimon
  - Deployed into the real OpenFlow network
- PRAGMA-ENT was used to evaluate the tool



Evaluation on International Network Testbed

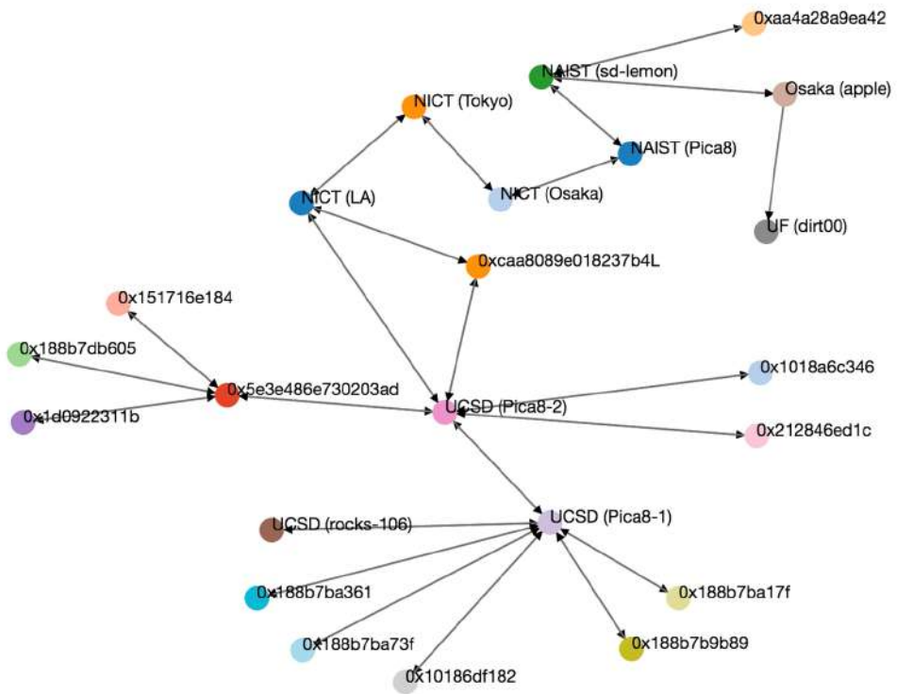
Opimon

Network Topology

Refresh

Submit

Sun Apr 23 2017 01:07:23
GMT+0700 (+07)



Switch ID : 0x1018c1900e (NAIST (sd-lemon))

Switch Detail

Port	MAC Address
1	56:20:df:69:5f:11
• Received packets : 17103 • Transmitted packets : 22982 • Received bytes : 2053667 • Transmitted bytes : 2592857 • Packets dropped by RX : 0 • Packets dropped by TX : 0	• Receive errors : 0 • Transmit errors : 0 • Frame alignment errors : 0 • Packet with RX overrun : 0 • CRC errors : 0 • Collisions : 0
2	0e:06:bd:7d:df:f0
3	76:06:82:37:aa:8c
7	00:10:18:c1:90:0e
10	52:ca:50:8a:1e:98
11	f6:79:06:fb:8b:fe
27	32:1b:24:ca:b1:cd
29	96:b1:b8:d9:74:e7
33	fe:5b:d7:c0:e6:63
37	fa:58:92:ab:7e:3f
65534	00:10:18:c1:90:0e

Flow Table

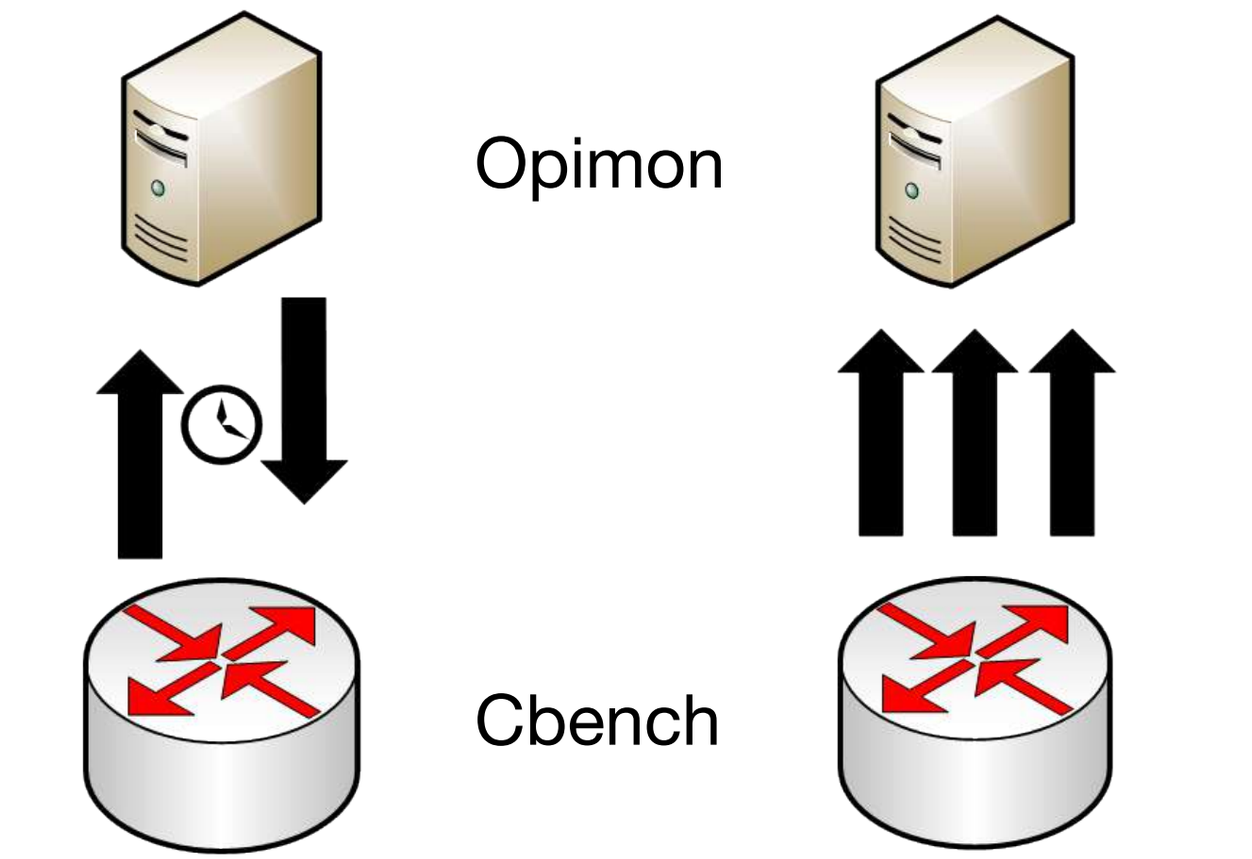
Match	Actions
• Wildcard : 3145954 • Switch Input Port : 2 • Source MAC Address : 8a:4a:58:e1:2b:4f • Destination MAC Address : 0e:83:bb:a8:b8:a7 • Ethernet Frame Type : 2054 • Source IP Address : 10.112.0.222 • Destination IP Address : 10.112.0.3 • Idle Timeout : 61 • Hard Timeout : 0	• Type : 0 (OFPActionOutput) • Switch Output Port : 3 • Max Length : 65535

Evaluation Methodology

- Evaluation of the monitoring module
  - Evaluation on international network testbed
  - **Benchmarking of the monitoring tool**
- Evaluation of the security analysis module
  - Accuracy of DDoS detection

Benchmarking of the Monitoring Tool

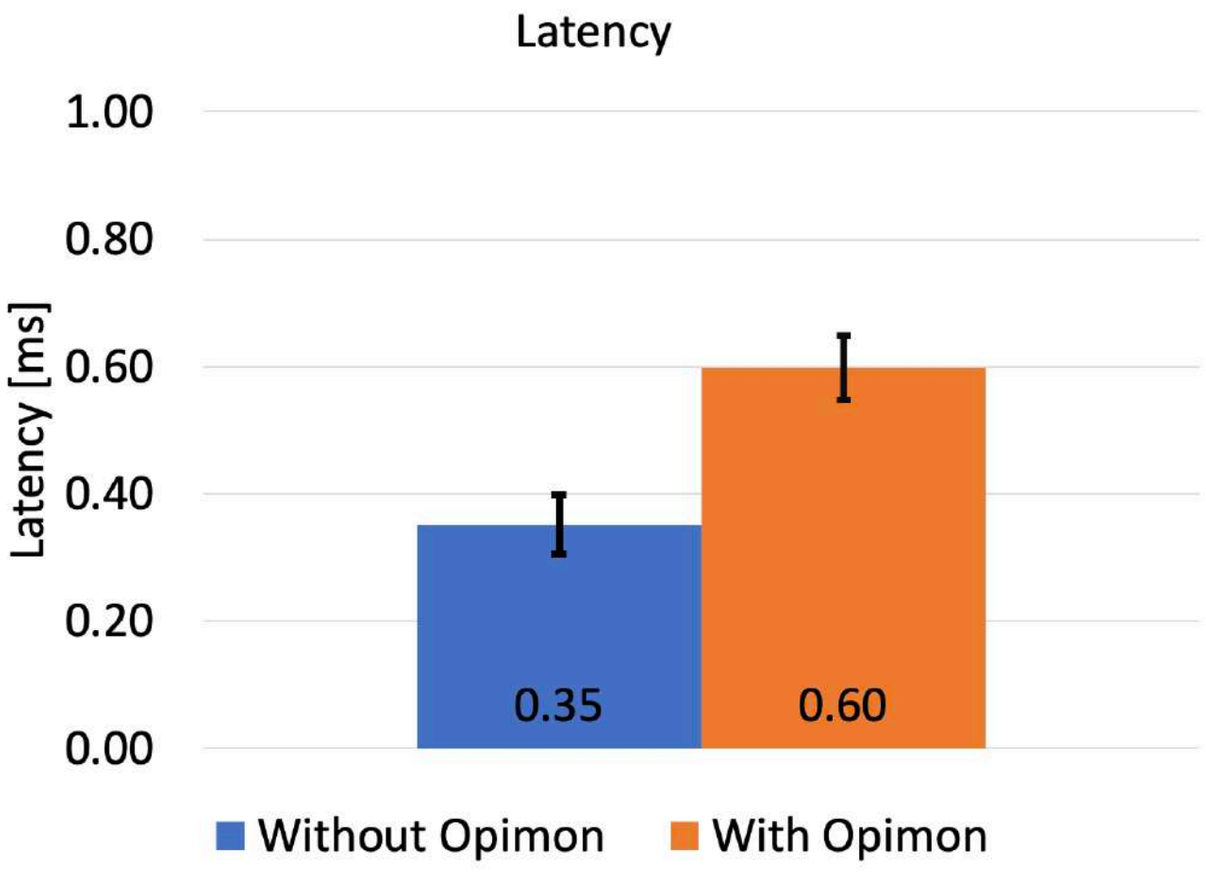
- Evaluated the practical applicability of Opimon
- Cbench, a benchmarking tool for OpenFlow controllers
  - Act as switches and send packets to the controller

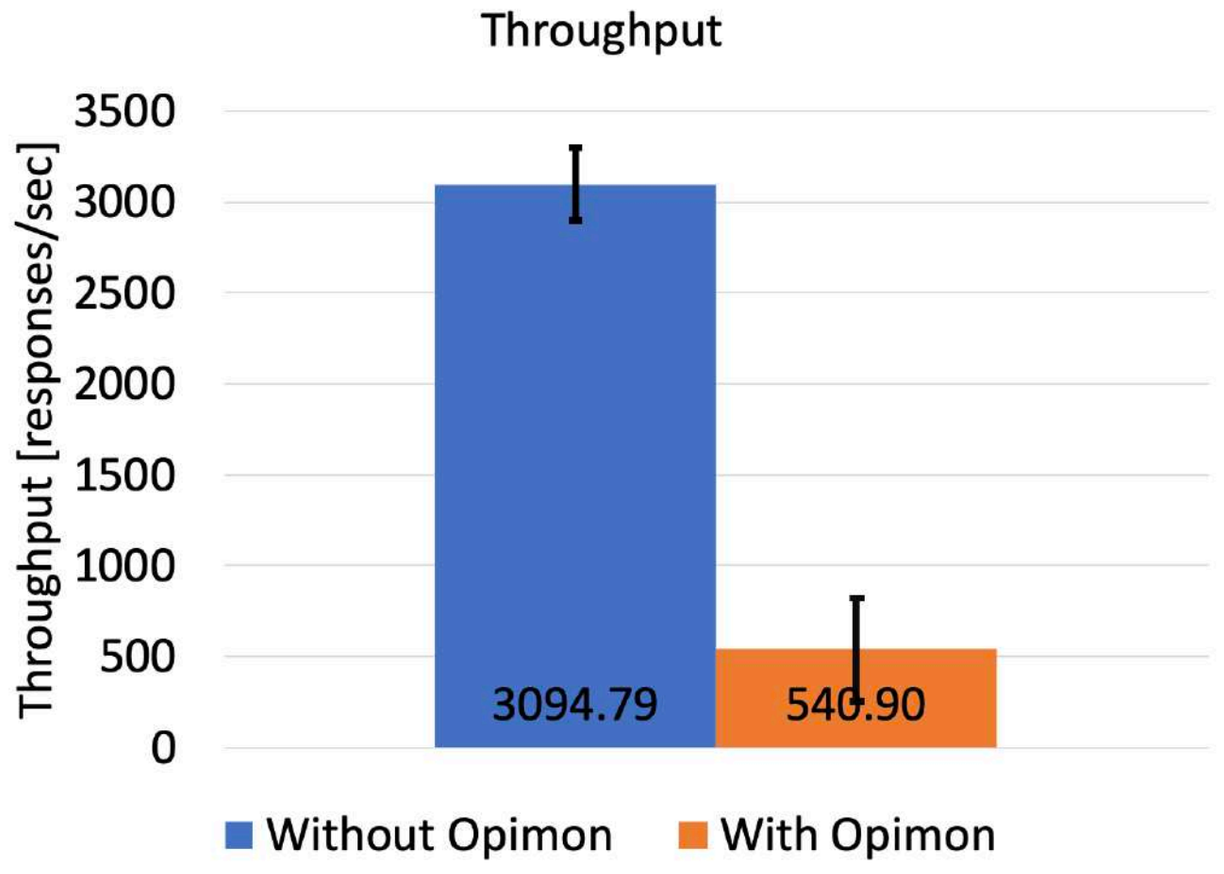


Benchmarking Result

- **Latency**: Increased for 70.09% with Opimon
- **Throughput**: Decreased for 82.52% with Opimon

Overhead comes from database connection and parsing messages





Evaluation Methodology

- Evaluation of the monitoring module
  - Evaluation on international network testbed
  - Benchmarking of the monitoring tool
- Evaluation of the security analysis module
  - **Accuracy of DDoS detection**

Accuracy of DDoS detection

- Performance comparison between Support Vector Machine (SVM) and Deep Feed Forward (DFF)
  - In terms of time and accuracy
- SVM: compare several kernels to find the best suitable
  - Polynomial kernel is used for the comparison
    - Provide the best accuracy among the compared kernels
- DFF: run with trial and error to find the best accuracy from the model

Accuracy of DDoS detection

Evaluation Result

- DFF can provide better accurate of DDoS detection
- SVM can provide faster traffic classification

SVM	Model Performance				Time Used (s)	
	Accuracy	Recall	Precision	F1-score	Train	Test
Time window aggregated	93.01%	0.922	0.933	0.927	371.118	0.003
Packet specific (S)	92.58%	0.894	0.933	0.906	379.417	0.003
Packet specific (L)	81.23%	0.927	0.756	0.826	138.260	0.500

Deep Learning	Model Performance				Time Used (s)	
	Accuracy	Recall	Precision	F1-score	Train	Test
Time window aggregated	61.30%	0.240	0.452	0.295	3.314	0.117
Packet specific (S)	68.30%	0.366	0.922	0.504	3.438	0.115
Packet specific (L)	99.63%	0.994	0.998	0.996	239.614	14.651

Outline

1. Introduction
2. Design & Implementation
    1. Monitoring Module
    2. Visualization Module
    3. Security Analysis Module
3. Experimental Result
4. **Conclusion**

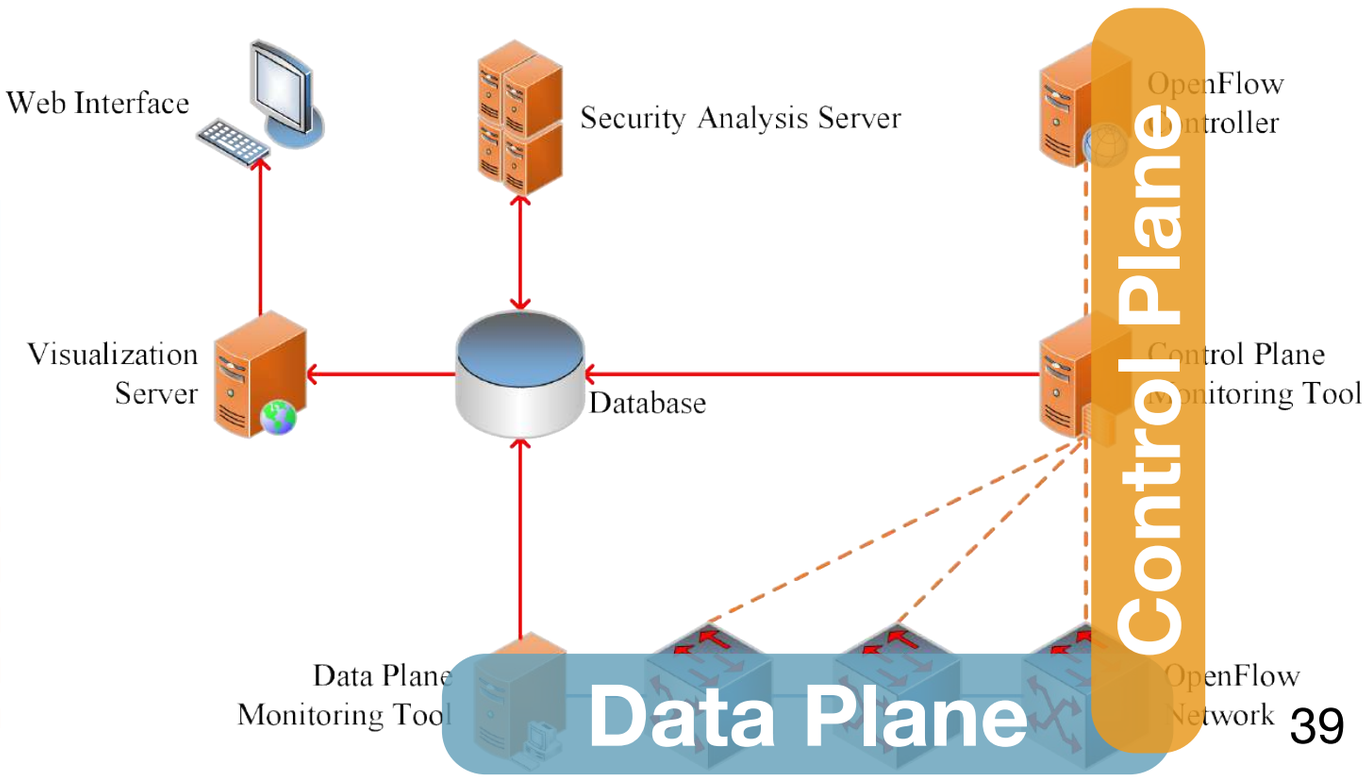
Summary

- Develop a monitoring tool for OpenFlow network
  - Can be used without modify the controller
  - Overview OpenFlow network information in real-time
  - Detect DDoS in the OpenFlow network
    - Using machine learning techniques

Future Work

- Optimize monitoring performance
- Apply machine learning into the data plane and control plane dataset
  - Simulate the DDoS traffic for getting the control plane dataset

Dataset	Packet	DDoS Detection
Data Plane	- Actual data - Big size	Detect DDoS on the host network
Control Plane	- Control message - Small size	Detect DDoS on the controller



Q&A