Size-based Flow Management Enabling Dynamic DMZ

Haotian Wu[1], Xin Li[1], Caterina Scoglio[1], Don Gruenbacher[1], Daniel Andresen[2],
[1] Electrical and Computer Engineering, [2] Computer Science
Kansas State University

Introduction

Campus networks require an architecture that is specifically designed for high-performance scientific applications. The DMZ model [1] is introduced to differentiate the science-data network from the general-purpose network. The current networking solutions for cybersecurity adopt static policies. Packets from and to supercomputers go through a DPI (deep packet inspection) device for security inspection before being routed toward their destinations, as shown in Fig. 1. However, a DPI always represents the system bottleneck, thus bringing down the performance of the whole network. The objective of our project is to realize a dynamic DMZ that achieves a tradeoff between network performance and network security.

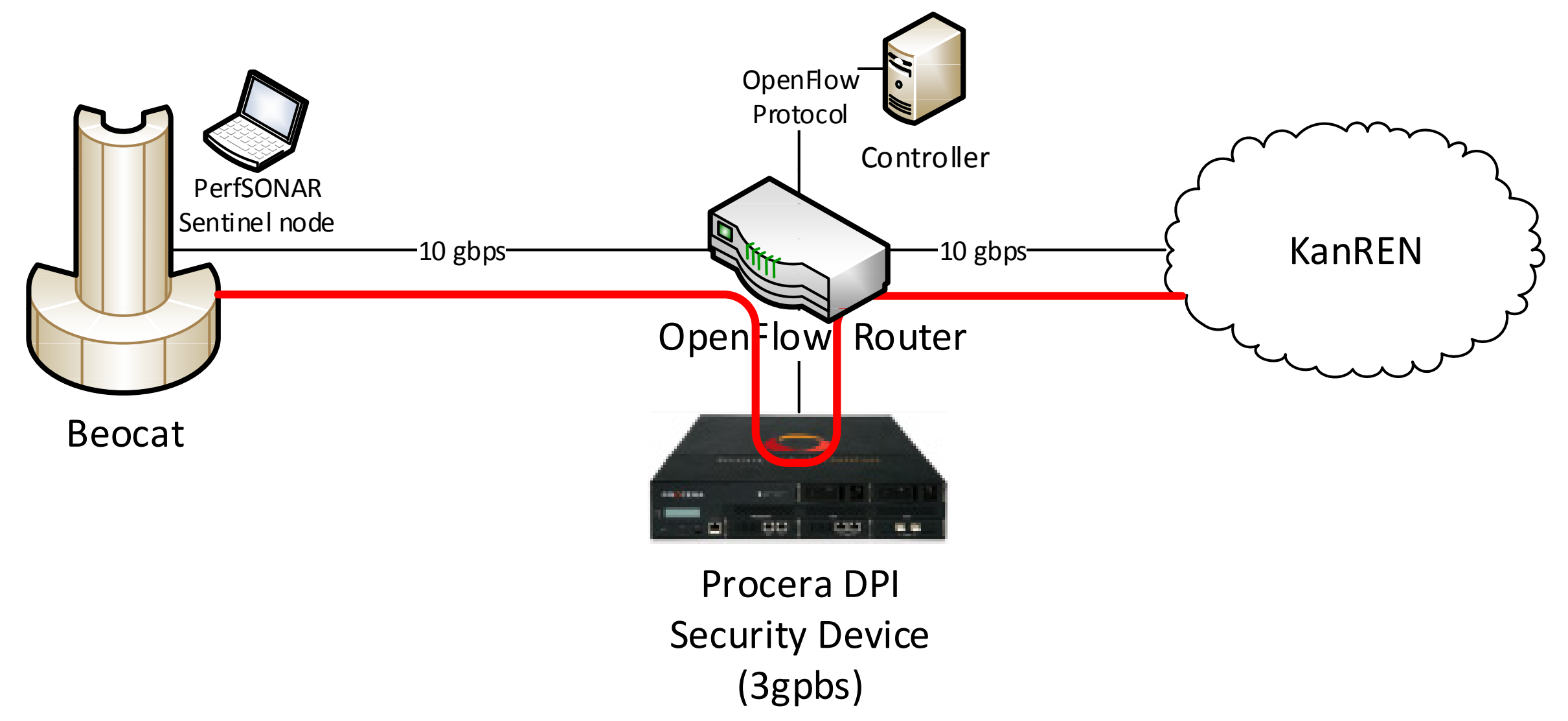


Fig. 1. DMZ configuration on K-State campus.

Network Topology Description

In the network configuration shown in Fig. 2, the bandwidth of the links is equal to 10 gbps while the speed of the DPI is equal to 3 gbps. The DPI throughput represents the system bottleneck, bringing the bandwidth of the path down to 3 gbps.

We assume that flows generated by the supercomputer can be divided into two groups based on flow sizes: one group consists of mice flows (the average flow size is much smaller than the DPI bandwidth) and the other group contains elephant flows (the average flow size is close to or larger than the DPI bandwidth). To detect elephant flows we use a threshold criterion. As a consequence, we develop a routing strategy based on flow sizes. Since elephant flows can create congestion, we reroute them on a direct path, bypassing the DPI, to achieve higher throughput, see Fig. 3.

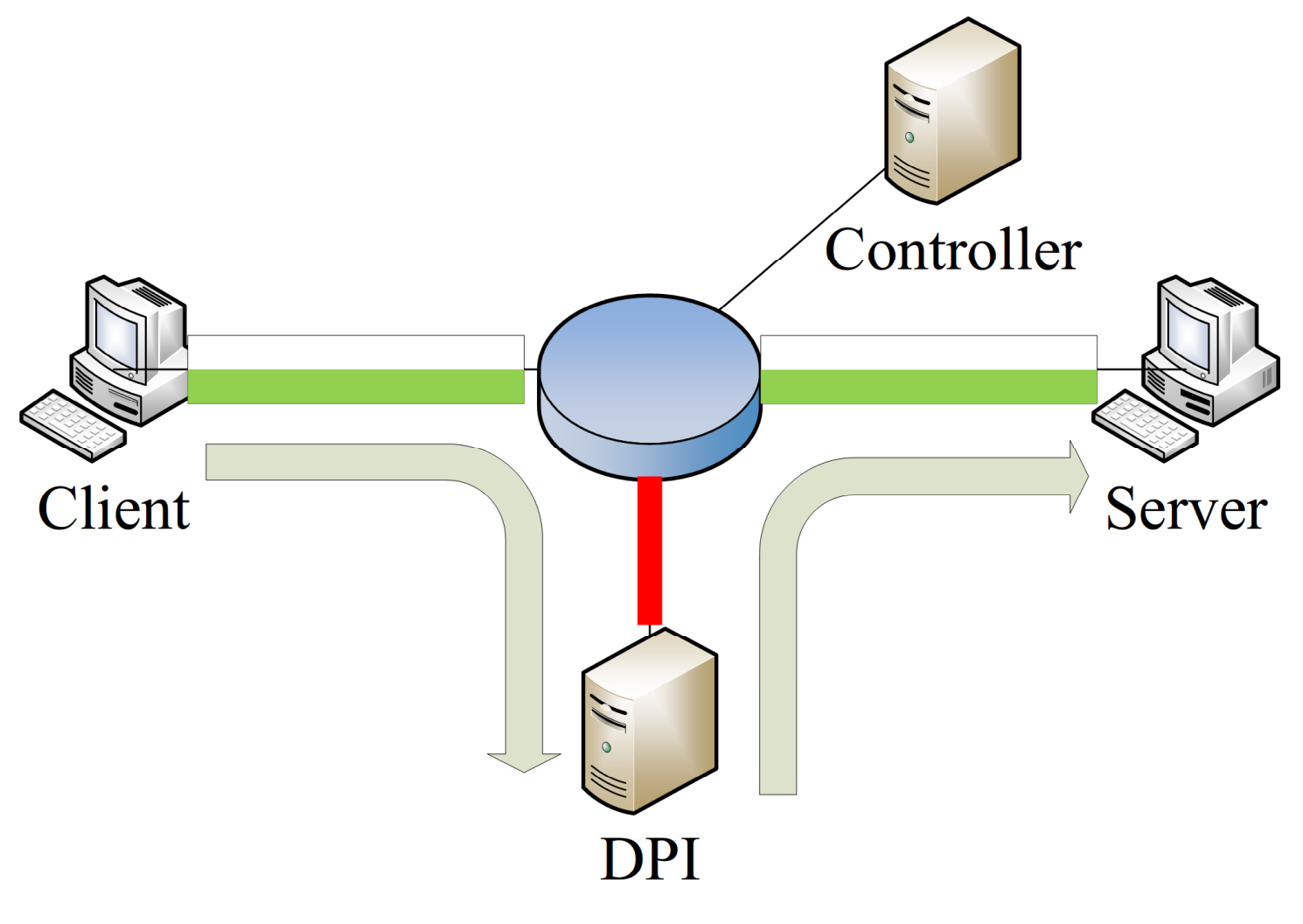


Fig. 2. DPI throughput represents the system bottleneck.

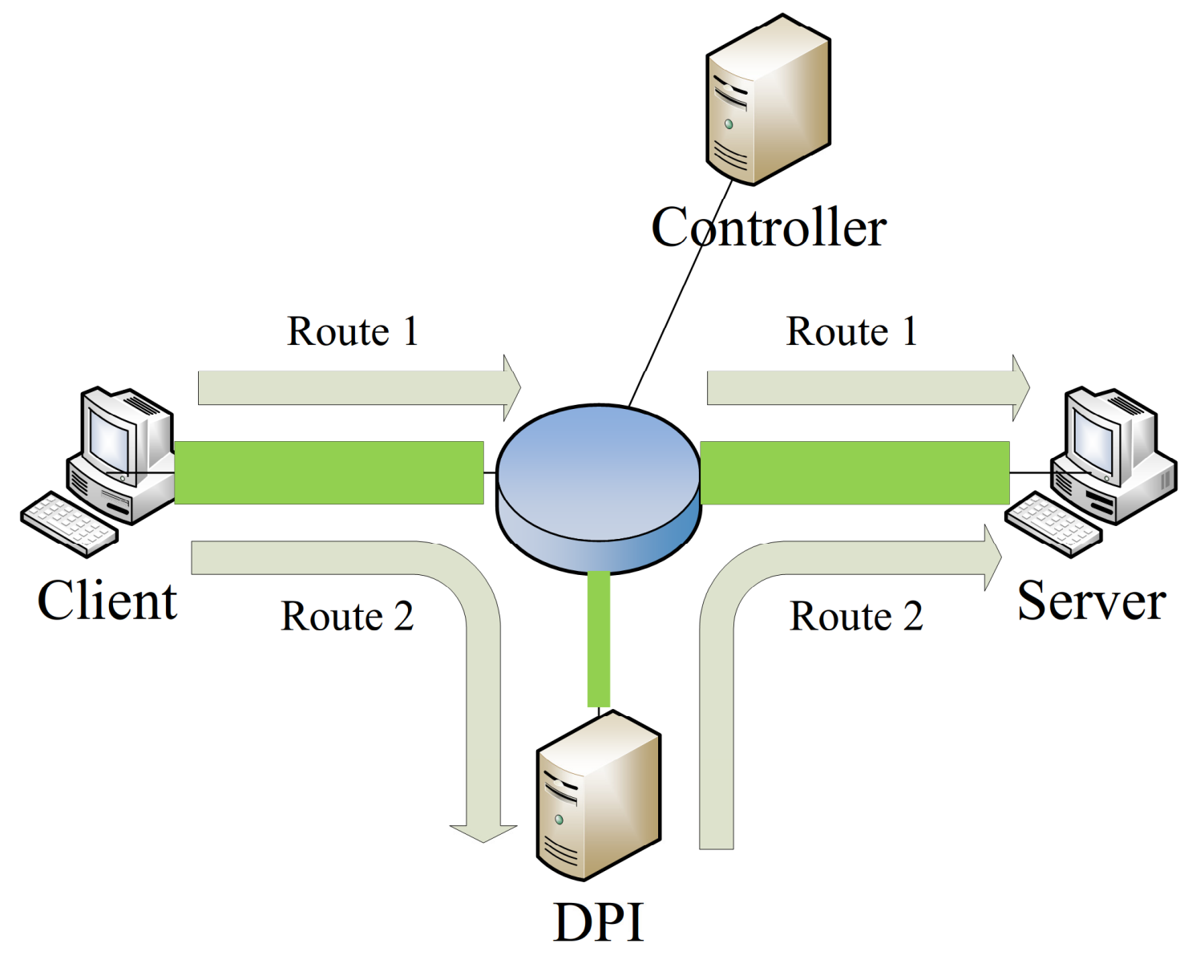
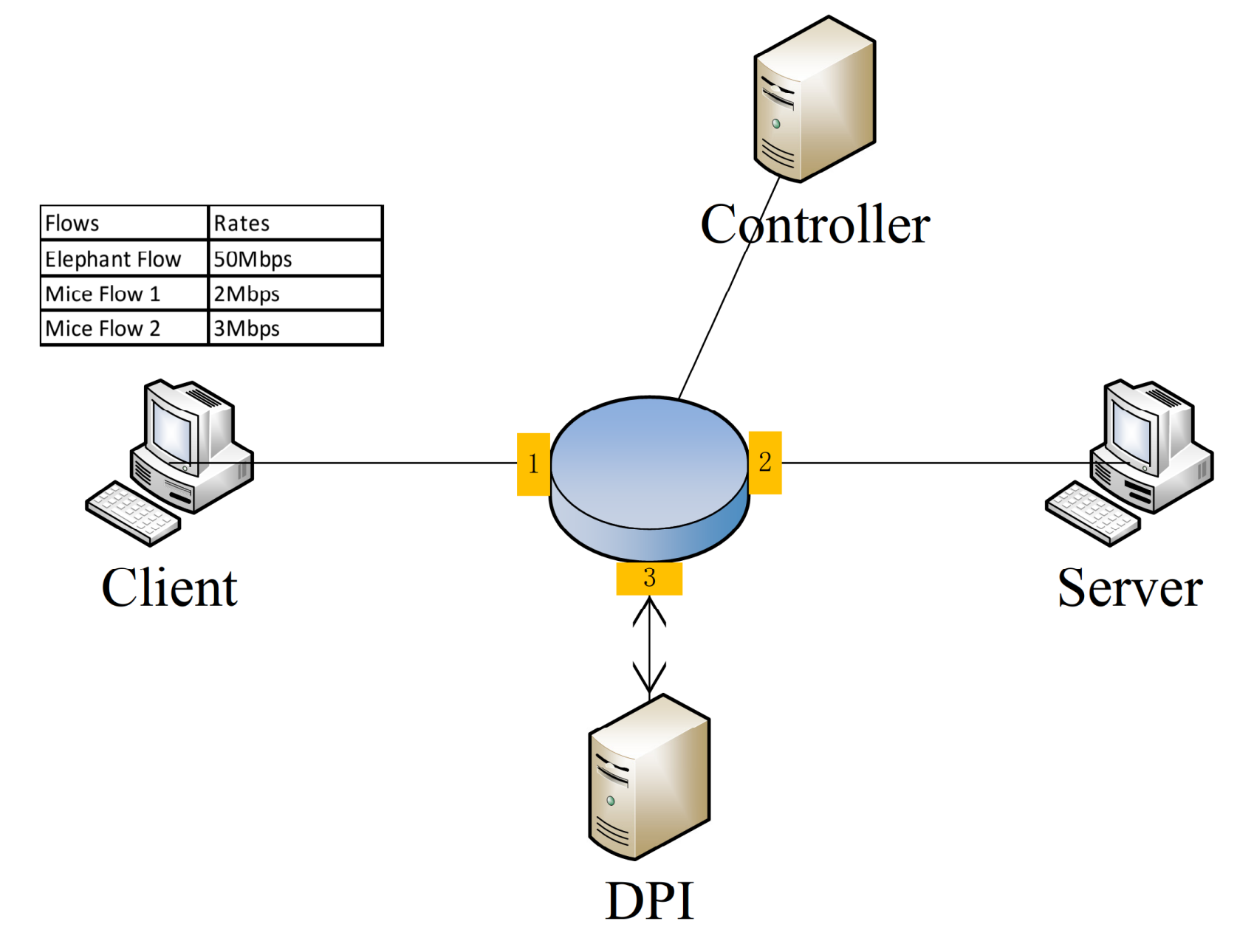


Fig. 3. Rerouting the elephant flow to achieve higher link utilization.

Experiment Testbed

Flow generation: we generated flows using *iperf* at different rates, as shown in table in Fig. 4.
Flow monitoring: we monitored flow statistics at the three interfaces of the software switch , marked orange in Fig. 4, using *Wireshark*.



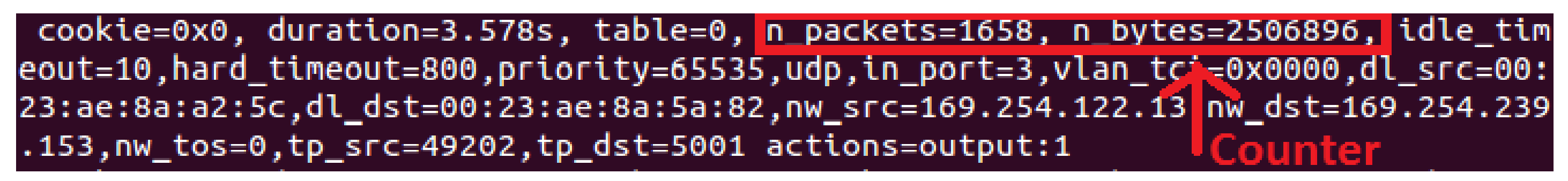
Flows	Rates
Elephant Flow	50Mbps
Mice Flow 1	2Mbps
Mice Flow 2	3Mbps

Fig. 4. Experiment testbed.

- **Software switch:** Open vSwitch 2.3.0, running on Ubuntu 12.04 LTS with Intel i7 Processor, 8GB RAM
- **Server**: 10/100Mb Ethernet card, iperf on Windows 7 Enterprise
- **Client**: 10/100Mb Ethernet card, iperf on Windows 7 Enterprise
- **DPI**: Open vSwitch on Ubuntu
- **OpenFlow controller:** POX carp branch, running on Ubuntu 12.04 LTS with Intel i7 Processor and 8GB RAM.

Size-based Flow Management (Part One)

Our controller can detect elephant flows by obtaining the flow statistics periodically from the switch. In the OpenFlow-enabled switch, a flow entry in the routing table consists of a counter which can count the flow size by bytes and packets [2]. The statistics of flow sizes are collected by the counter in our project, as shown in Fig. 5.



```
 cookie=0x0, duration=3.578s, table=0, n_packets=1658, n_bytes=2506896, idle_tim
eout=10,hard_timeout=800,priority=65535,udp,in_port=3,vlan_tci=0x0000,dl_src=00:
23:ae:8a:a2:5c,dl_dst=00:23:ae:8a:5a:82,nw_src=169.254.122.13 nw_dst=169.254.239
.153,nw_tos=0,tp_src=49202,tp_dst=5001 actions=output:1
```

Fig. 5. A specific flow entry, created by *POX* controller and displayed using *Open vSwitch*. We can see counters, headers to match, and actions in the flow entries.

The client sends multiple flows, see Fig. 6, to the server. The flows are forwarded by a software switch (OpenFlow switch) to the DPI first. Flows sent to the DPI are inspected for network security purposes. When the software switch detects a legitimate elephant flow, the elephant flow will be forwarded directly to the server bypassing the DPI.

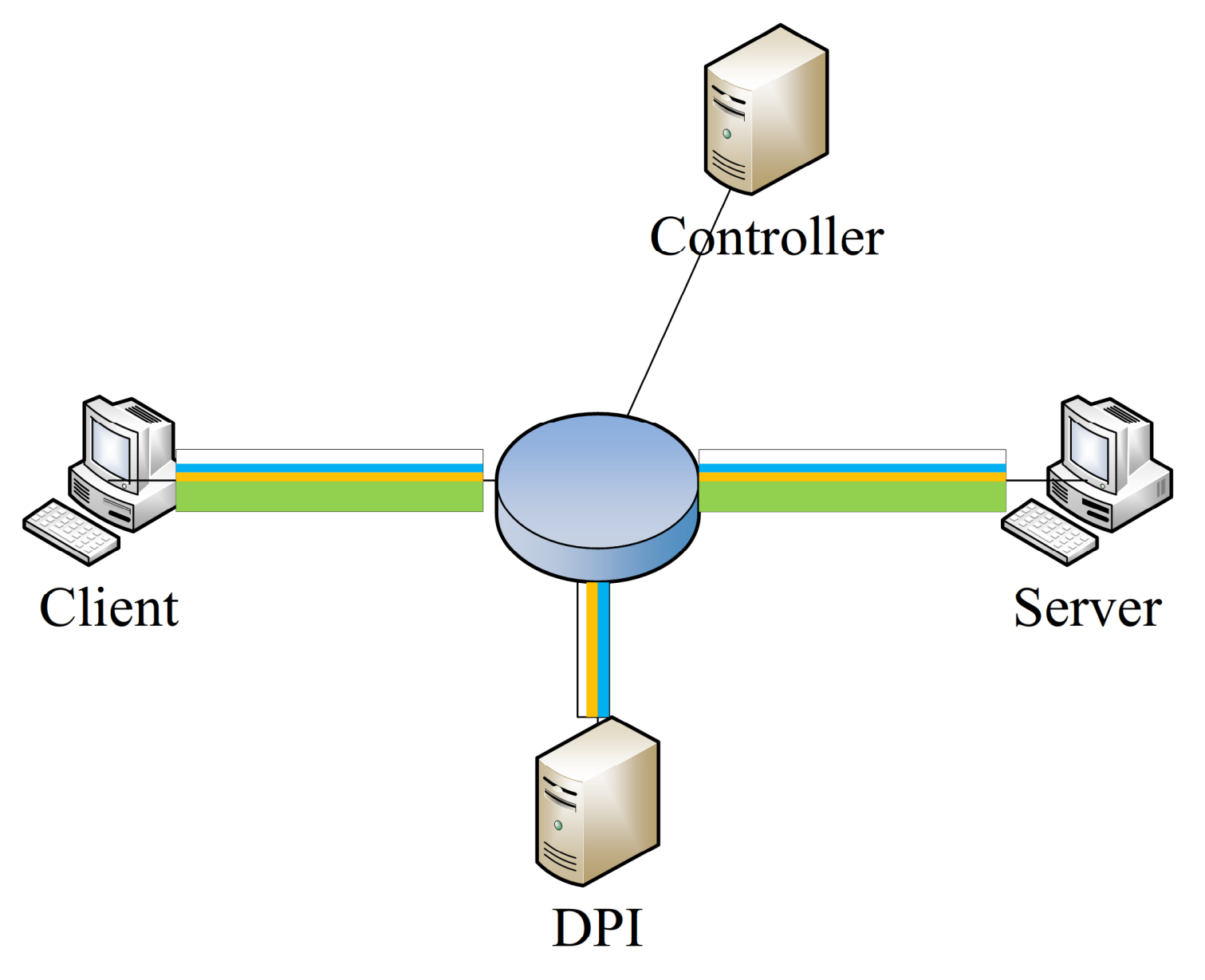
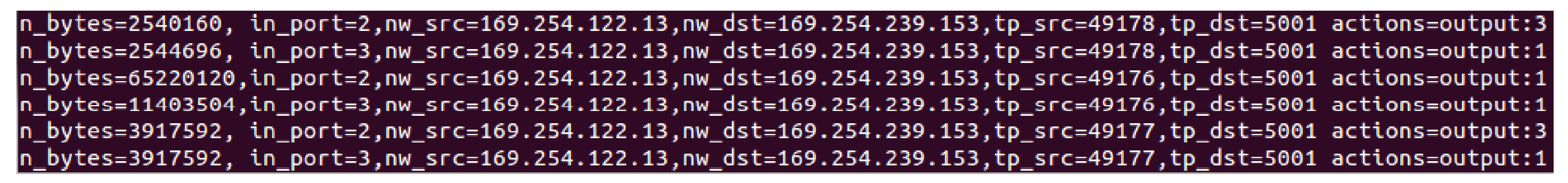


Fig. 6. Routing paths.

Size-based Flow Management (Part Two)

We monitored flow statistics and the process of routing table installation. We can see several flow tables in Fig. 7. We created two flow entries for each flow: one flow entry redirects the flow to the DPI, and second flow entry takes the flow from the DPI to its original destination. When an elephant flow is detected, the controller sends a command to the switch modifying the first flow entry to route the flow directly to the destination.



```
n_bytes=2540160, in_port=2,nw_src=169.254.122.13,nw_dst=169.254.239.153,tp_src=49178,tp_dst=5001 actions=output:3
n_bytes=2544696, in_port=3,nw_src=169.254.122.13,nw_dst=169.254.239.153,tp_src=49178,tp_dst=5001 actions=output:1
n_bytes=65220120,in_port=2,nw_src=169.254.122.13,nw_dst=169.254.239.153,tp_src=49176,tp_dst=5001 actions=output:1
n_bytes=11403504,in_port=3,nw_src=169.254.122.13,nw_dst=169.254.239.153,tp_src=49176,tp_dst=5001 actions=output:1
n_bytes=3917592, in_port=2,nw_src=169.254.122.13,nw_dst=169.254.239.153,tp_src=49177,tp_dst=5001 actions=output:3
n_bytes=3917592, in_port=3,nw_src=169.254.122.13,nw_dst=169.254.239.153,tp_src=49177,tp_dst=5001 actions=output:1
```

Fig. 7. Routing table setup.

We captured the packets in the 3 interfaces on the software switch, and drew a graph in Fig. 8 showing the number of packets received or sent as a function of time. Since the controller obtains statistics every 2 seconds, we see a peak in the DPI interface (Interface 3). The peak represents the small amount of the elephant flow initially routed through the DPI. From the beginning to the end, mice flows are transmitted normally. Since the interface of DPI records bidirectional packets, its graph is twice as high as those in other interfaces.

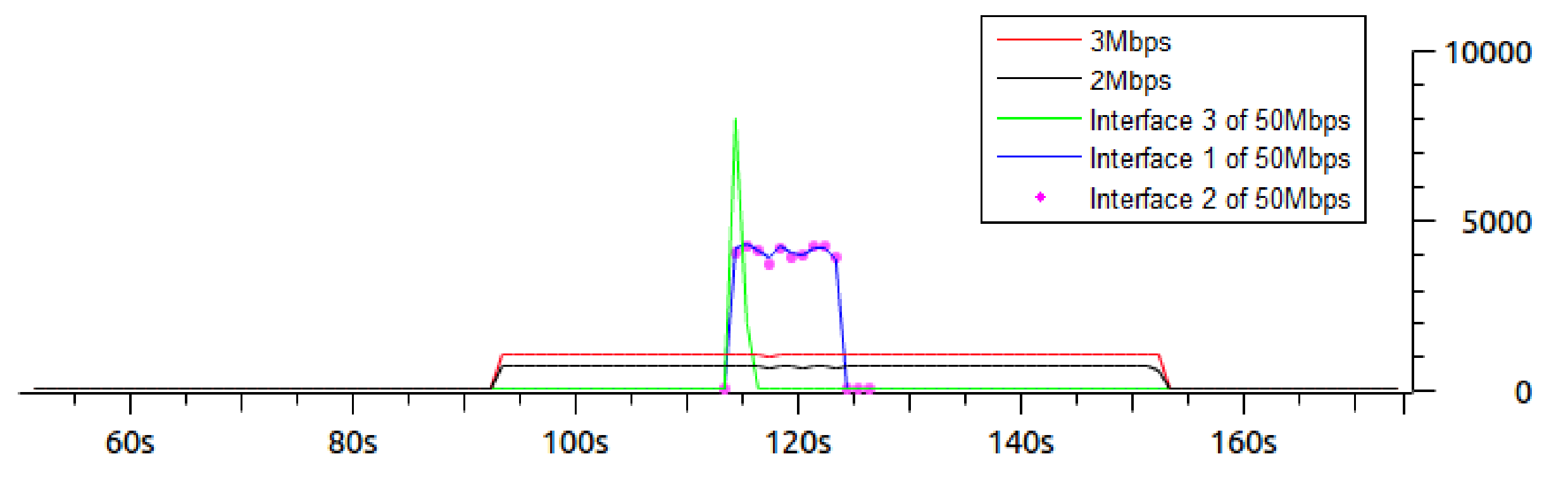


Fig. 8. Experiment results captured by *Wireshark*.

Conclusion and Future Work

We have developed a controller which accomplishes the simultaneous goals of achieving high network performance for research data flows and higher network security than traditional DMZ configurations.
In the future, by developing efficient algorithms in the controller, we will realize dynamical flow routing in large network topology. Furthermore, we plan to achieve the same results with Cisco and Brocade hardware switches in K-State Smart Grid Lab.

Reference

[1] "Science DMZ" - http://fasterdata.es.net/science-dmz
[2] OpenFlow Switch Consortium. "OpenFlow Switch Specification Version 1.0. 0." (2009).

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1341026, and by the Electrical Power Affiliates Program at Kansas State University. Experiments are conducted using the resources on GENI and equipment in the K-State Smart Grid Lab.