Turning Mininet/Open vSwitch into A Detailed OpenFlow Emulator

YI-JUN CHENG

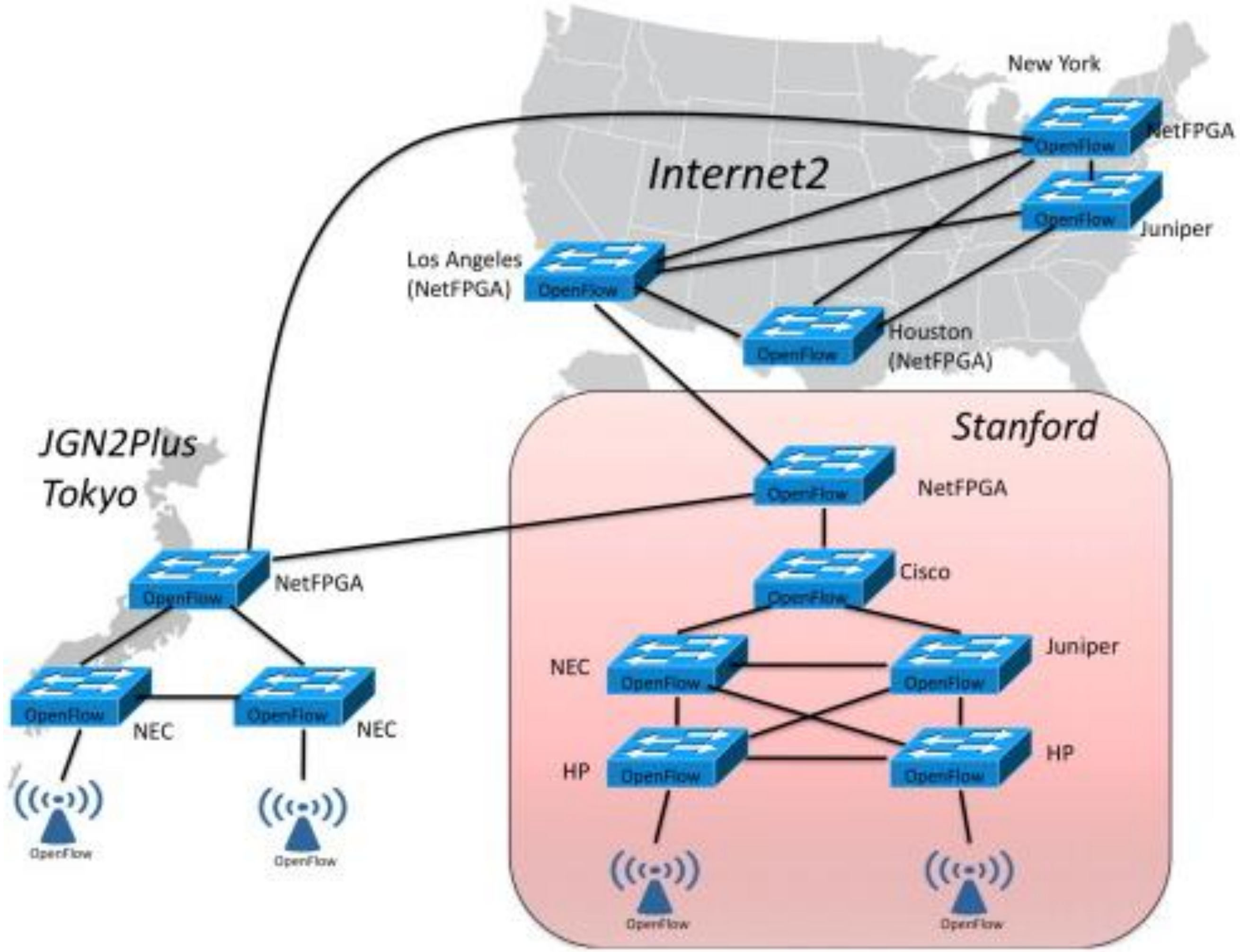
OCTOBER, 2015

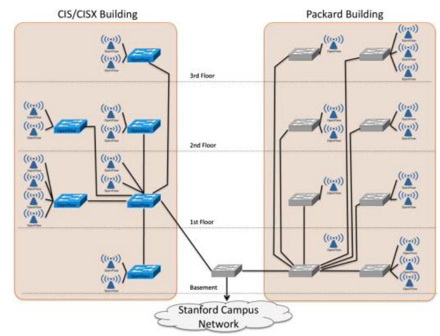
Emerging Network Architecture

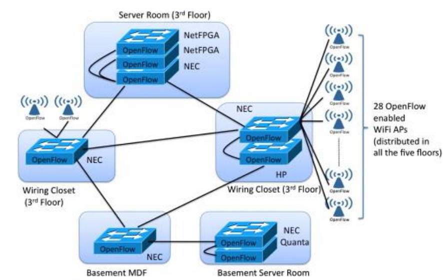
- Software-Defined Networking provides network programmability and efficient network management
- Researchers from academia and industry worked on developing innovative network services on SDN and OpenFlow
- Behavior verifications and performance evaluations are necessary to examine the possibilities of the novel ideas

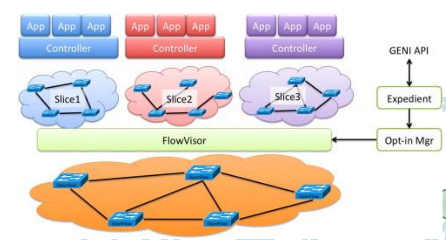
OPEN NETWORKING FOUNDATION

OpenFlow





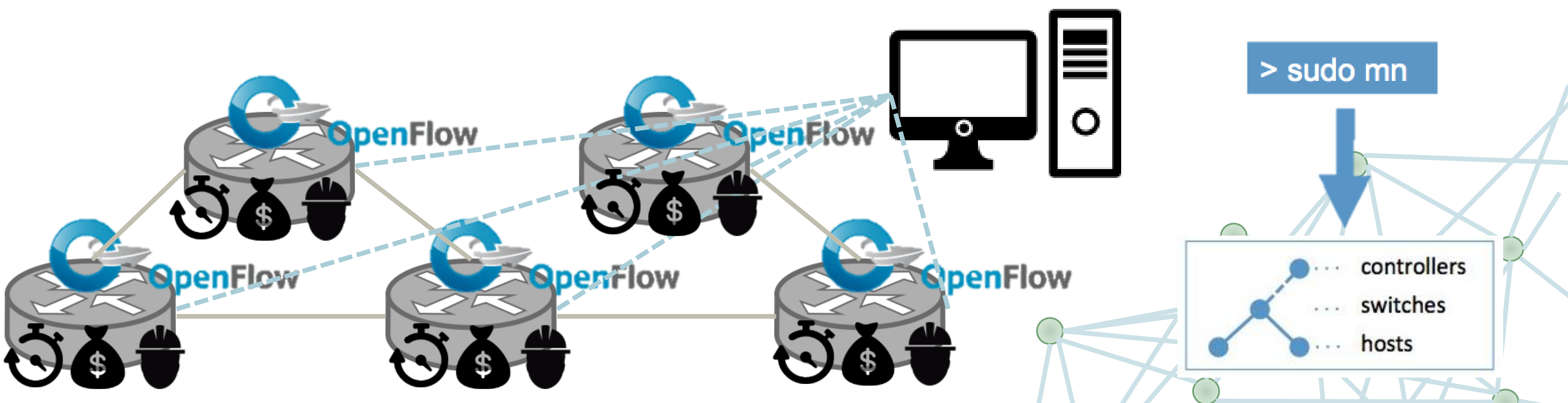




References:
[1] M. Kobayashi and S. Seetharaman and G. Parulkar and G. Appenzeller and J. Little and J. Reijendam and P. Weissmann and N. McKeown. Maturing of OpenFlow and Software-defined Networking through Deployments. *Computer Networks*, 61:151–175, November 2013.

OpenFlow Simulators/Emulators

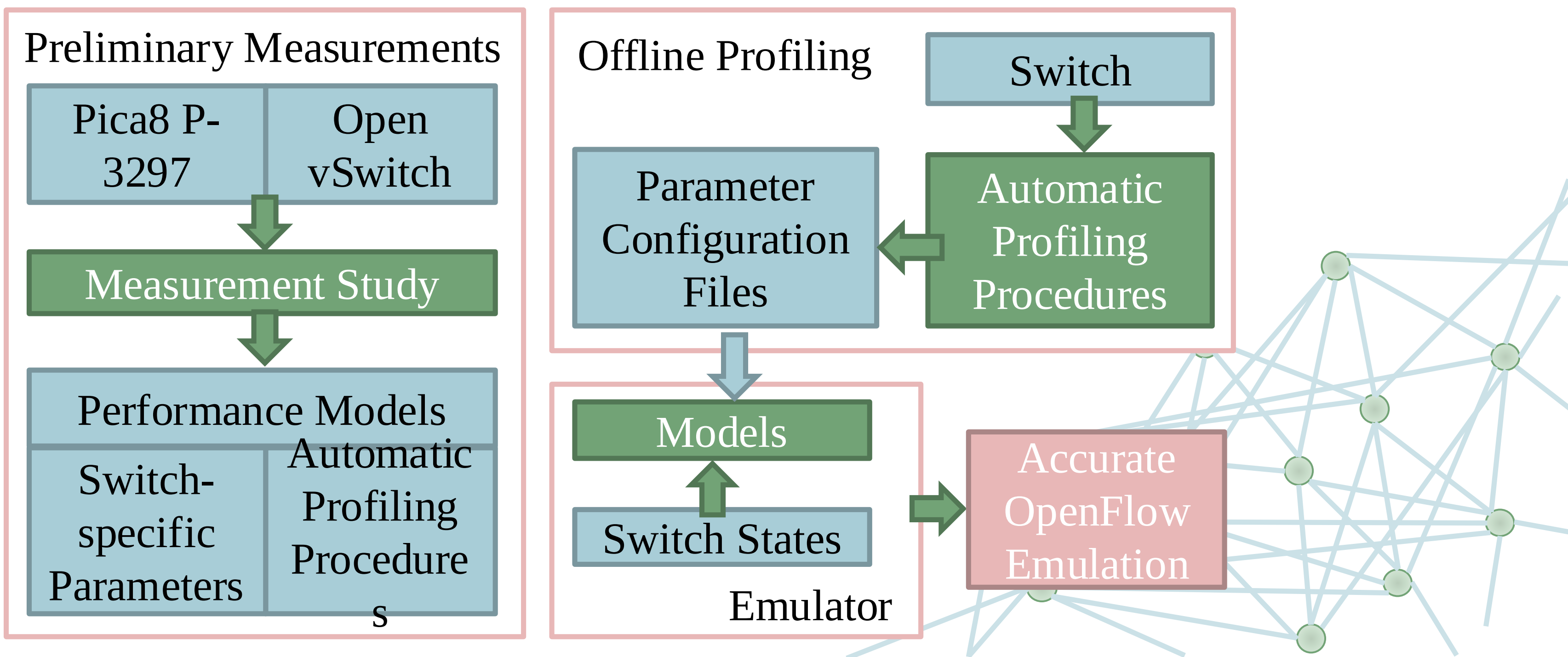
- Testbeds are necessary, but deploying ones is costly, time-consuming, and labor-intensive
- Run emulations/simulations beforehand
- Several available emulators/simulators, but fail to consider control plane performances and different switch implementations



References:
[1] Mininet: an instant virtual network on your laptop (or other PC). http://mininet.org

Goal

- Accurately emulate both behaviors and performances of SDN-based networks and support different switch implementations
- Design measurement studies for switch performances and propose performance benchmarks
- Propose performance models and switch-dependent parameters,
- Integrate with an OpenFlow emulator, Mininet/OVS

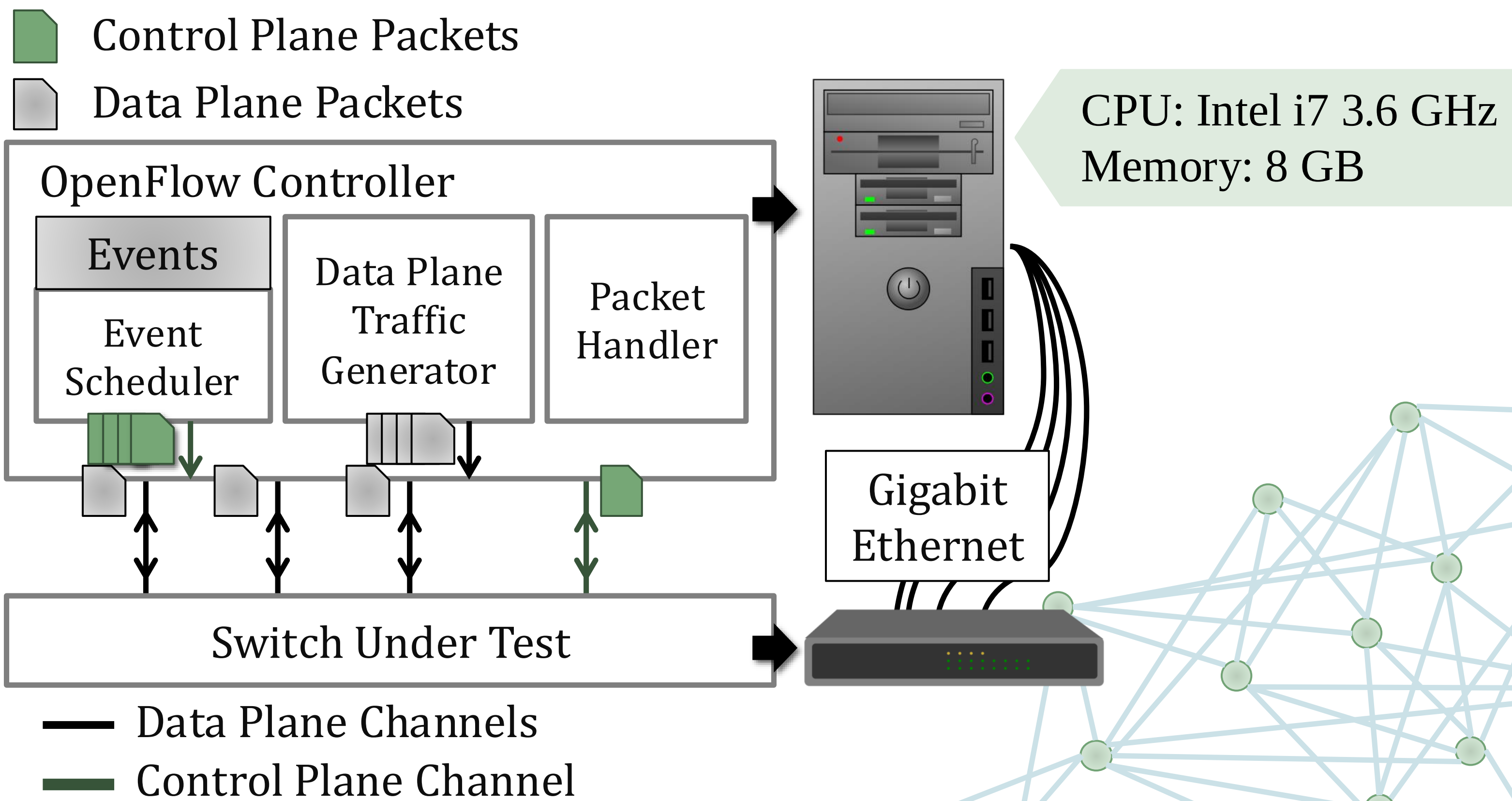


Measurement Methodology

- Conduct measurement studies on OpenFlow switches, Pica8 P-3297 and Open vSwitch
- Control plane performance measurements
  - Flow table update delay
  - Develop our testing modules based on OFLOPS
- Data plane performance measurements
  - Packet forwarding latency and throughput
  - Use OFLOPS with packet replay and capture tools

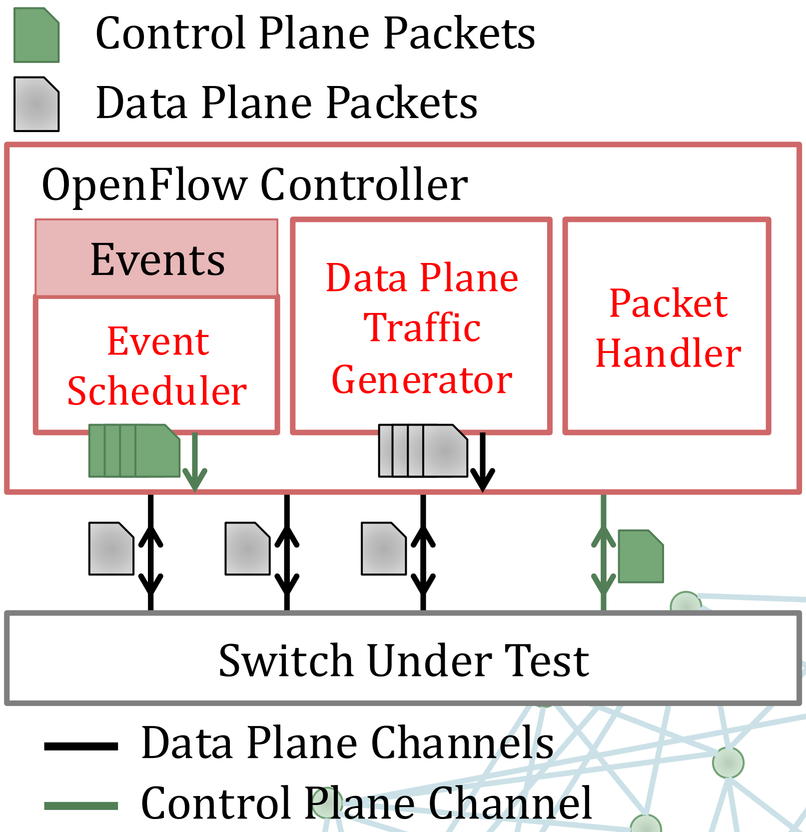
Measurement Setup

- Dedicated control plane channel
- *Event Scheduler, Data Plane Traffic Generator,* and *Packet Handler* are three main threads in OFLOPS framework



What Each Component Does

- *Event Scheduler* defines how each event works
- *Data Plane Traffic Generator* generates customized data plane packets and send them via data plane channels
- *Packet Handler* captures packets from both data plane and control plane channels



Control Plane Performance Measurement Studies and Modeling

- Test Scenarios
- Measurement Results
- Performance Models
- Model Validation

Factor Considerations for Control Plane Performance Tests

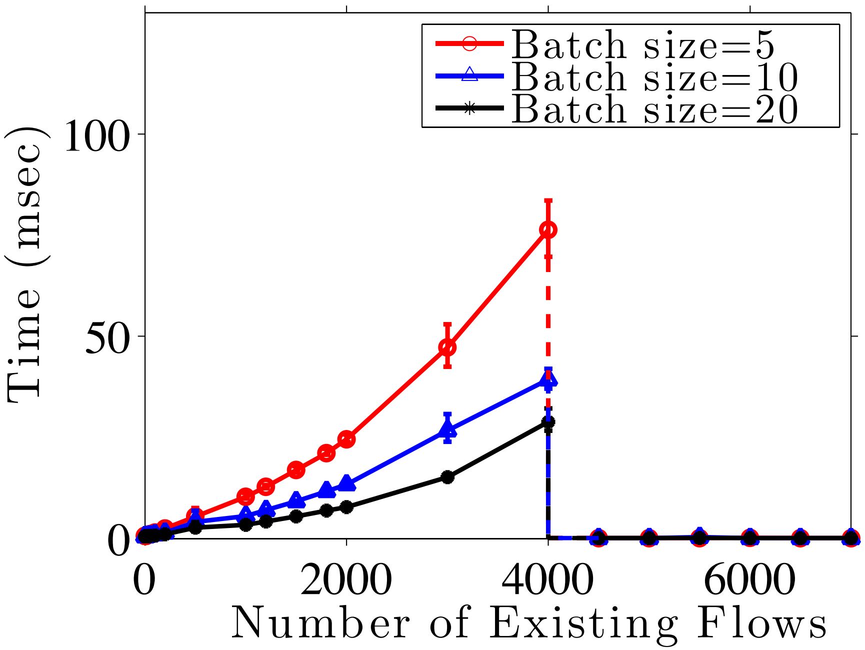
- **Performance metrics**
  - Flow table update delay
- Flow_mod command types
  - Insertion, modification, and deletion commands
- Number of existing flows
- Priority distribution of existing flows
  - Descending, ascending, and same priority distributions
- Number of batch commands
  - How many commands waiting for execution

Control Plane Tests

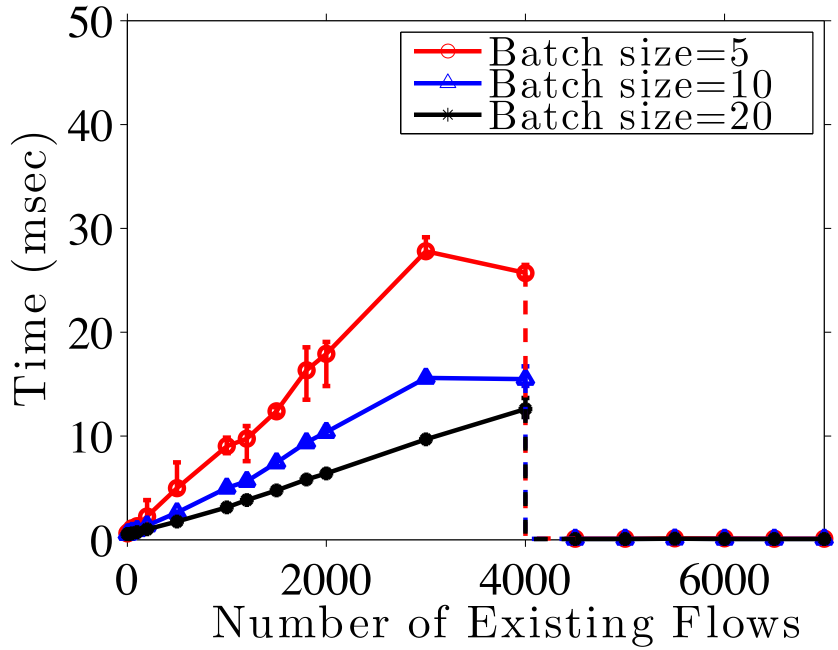
- Preinstall different number of flows with different priority distributions at first for each of the test
- Insertion test
  - Send different number of insertion commands under different number of existing flows
- Modification test
  - Send different number of modification commands under different number of existing flows
- Deletion test
  - Send a wild-carded command to delete flows in the table
- Show sample results from Pica8 in the following

Insertion Test Results on Pica8

- Proportional to existing flow size
- Different increasing rate for different batch command size
- Increasing rate decreases with more batch commands
- Similar observations in software table



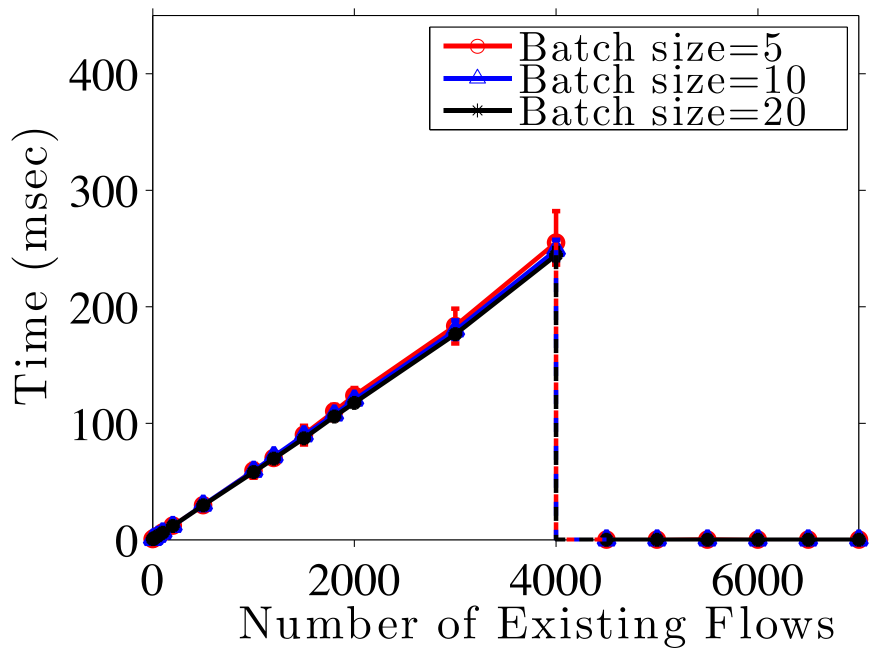
Descending priority distribution



Same priority distribution

Insertion Test Results on Pica8 (cont.)

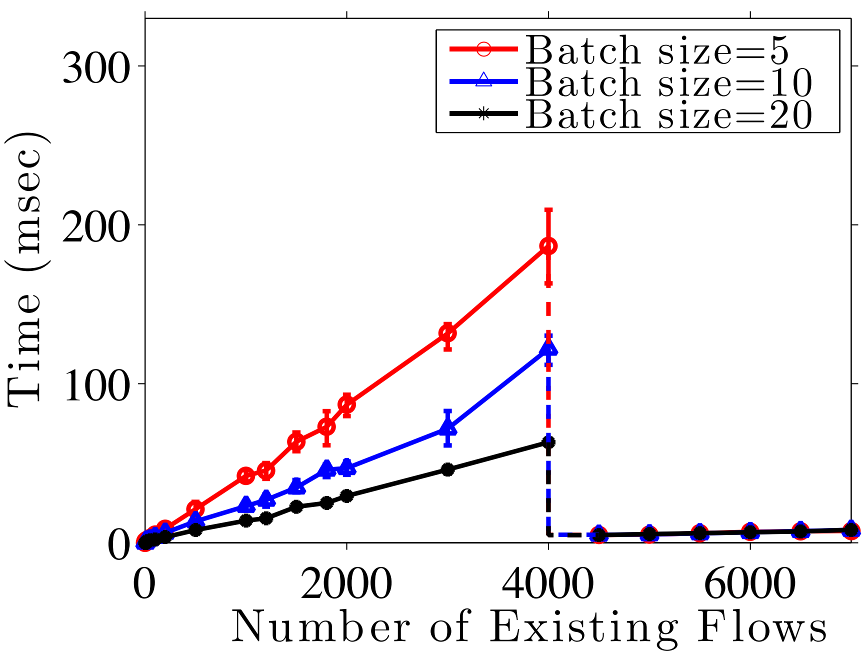
- Flows should be in priority order in TCAM
- Flow shifting time dominates the insertion delays, so little differences observed among different batch command sizes



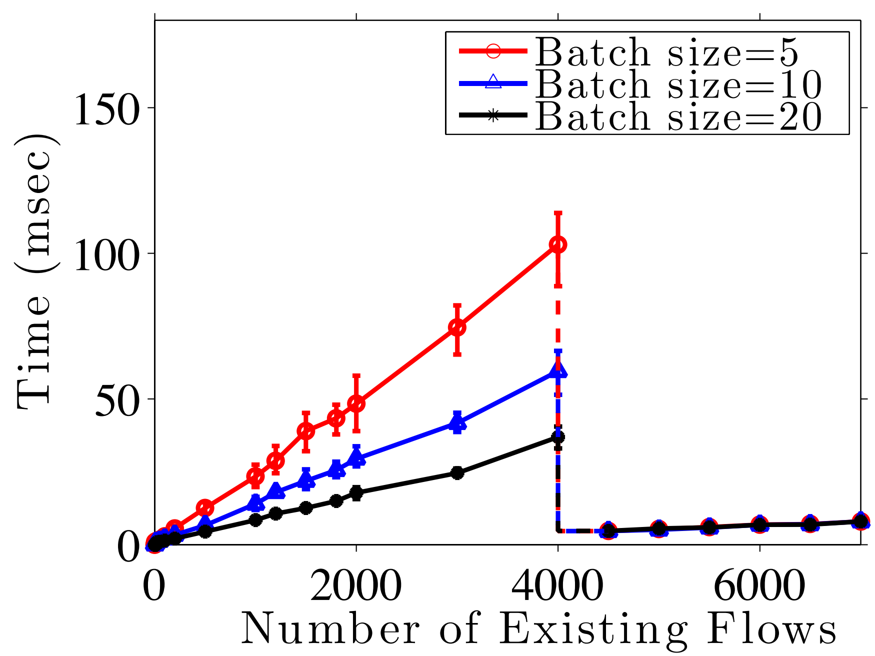
Ascending priority distribution

Modification Test Results on Pica8

- Proportional to existing flow size
- Increasing rate decreases with more batch commands
- Different increasing rates for different priority distributions



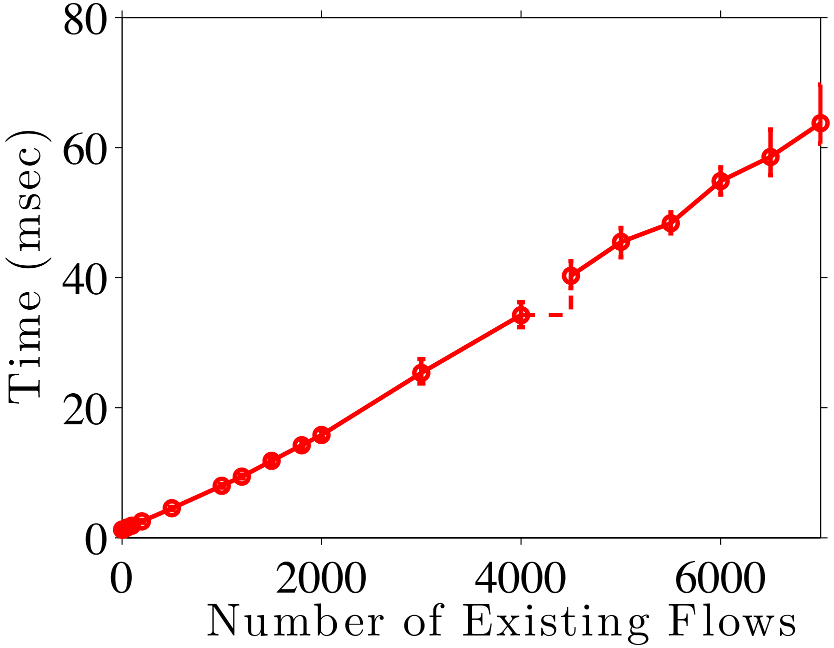
Ascending priority distribution



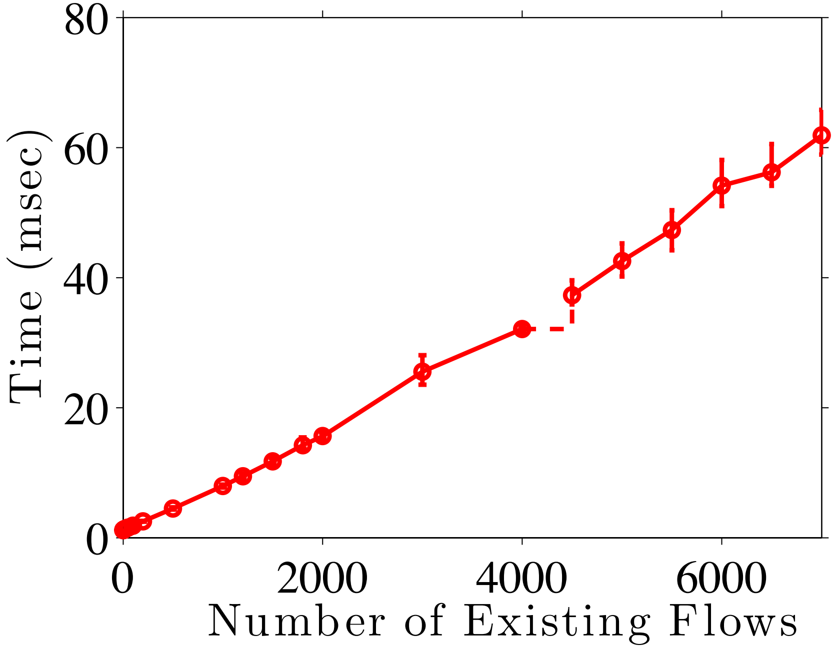
Same priority distribution

Deletion Test Results on Pica8

- Proportional to number of deleted flows
- Existing flows with different priority distributions share same results



Ascending priority distribution

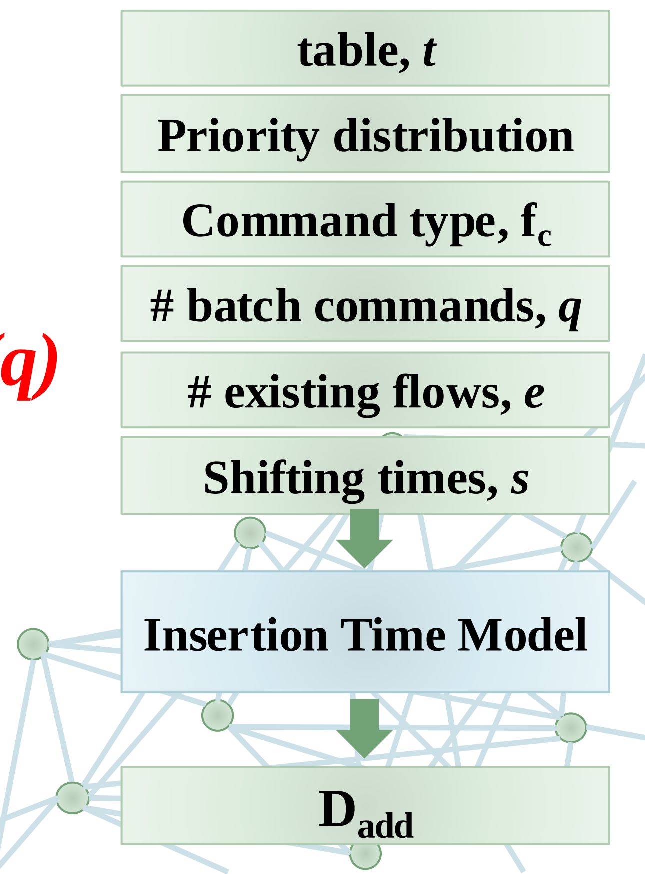


Same priority distribution

Insertion Time Model

$$D_{add} = \frac{P^t_{f_c,\omega^t_c}}{R^t_{f_c,\omega^t_c}(q_c)} \times e^t_c + S_t \times s^t_c + W_t$$

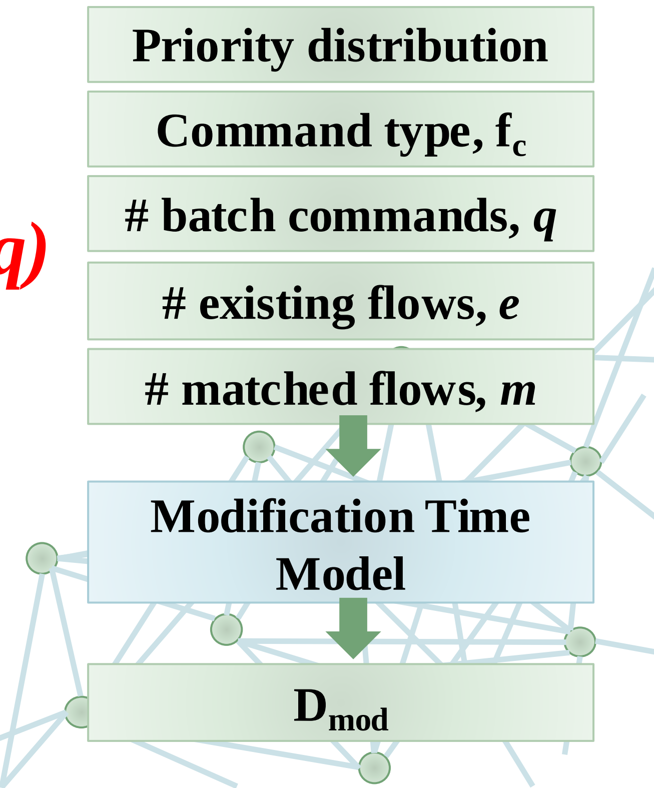
- **c** denotes the index of flow_mod command
- ***t*** is the index of the table
- Proportional to existing flow size → ***P***
- Decrease with more batch commands → ***R(q)***
- ***S*** is flow shifting time
- ***W*** is the time to update a flow table entry



Modification Time Model

$$D_{mod} = \sum_{t=1}^{T}\left(\frac{P^{t}_{f_c,\omega^{t}_{c}}}{R^{t}_{f_c,\omega^{t}_{c}}(q_c)} \times e^{t}_{c} + M_t + W_t \times m^{t}_{c}\right)$$

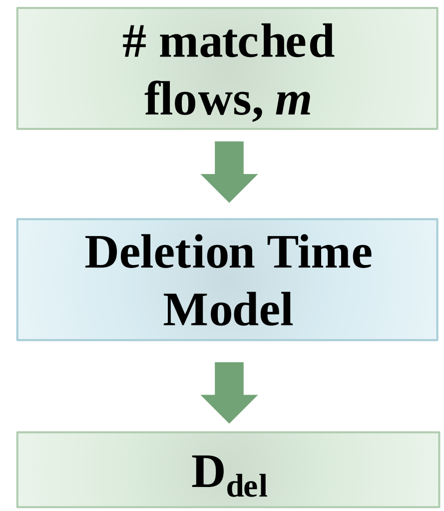
- ***T*** denotes number of tables
- Proportional to existing flow size → ***P***
- Decrease with more batch commands → ***R(q)***
- ***M*** is the time for searching matching flows
- ***m*** denotes the number of matched flows



Deletion Time Model

$$D_{del} = \sum_{t=1}^{T} (M_t + W_t \times m_c^t)$$

- ***M*** → time for searching all matched ***m*** flows
- ***W*** → time for updating a flow entry

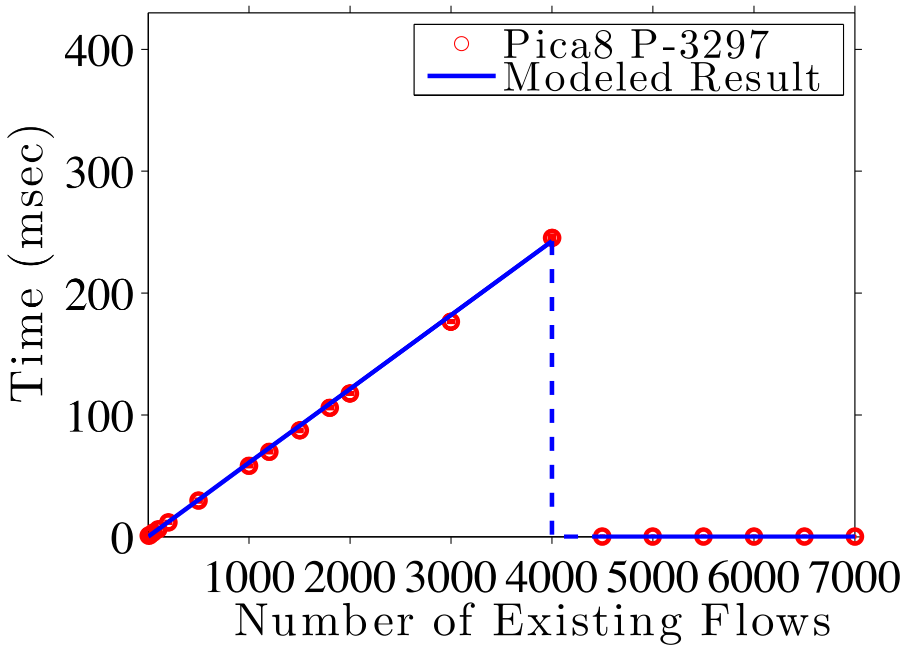


Validation Experiments

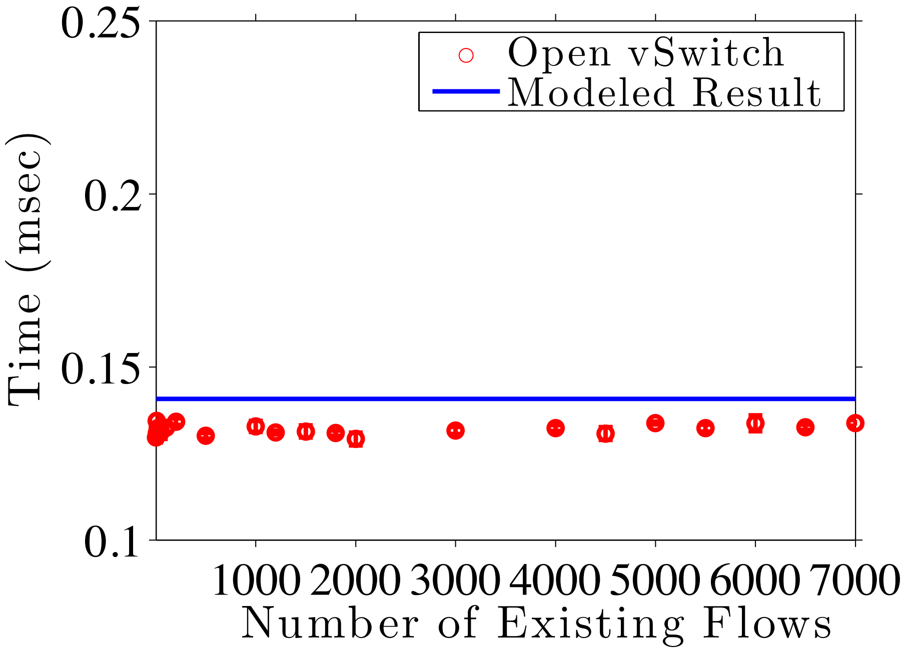
- Test scenarios
  - Insertion tests
  - Modification tests
  - Deletion tests
  - Random tests
    - Random priorities, IP addresses, arrival time, and command types
- Validation results from Pica8 and OvS

Insertion Test Validation

- Validation results using ascending priority distribution
- Modeled results follow the results of OpenFlow switches



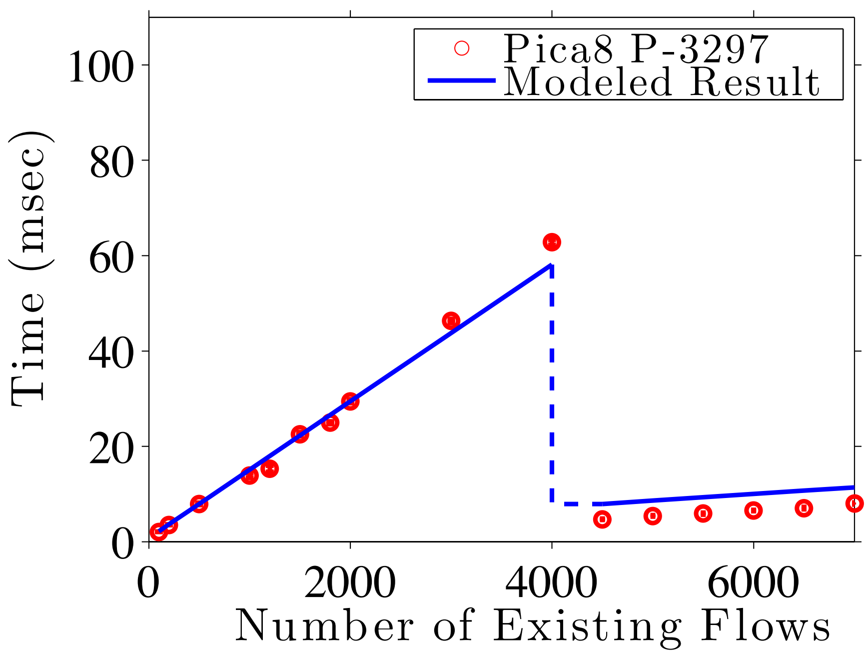
Pica8 P-3297



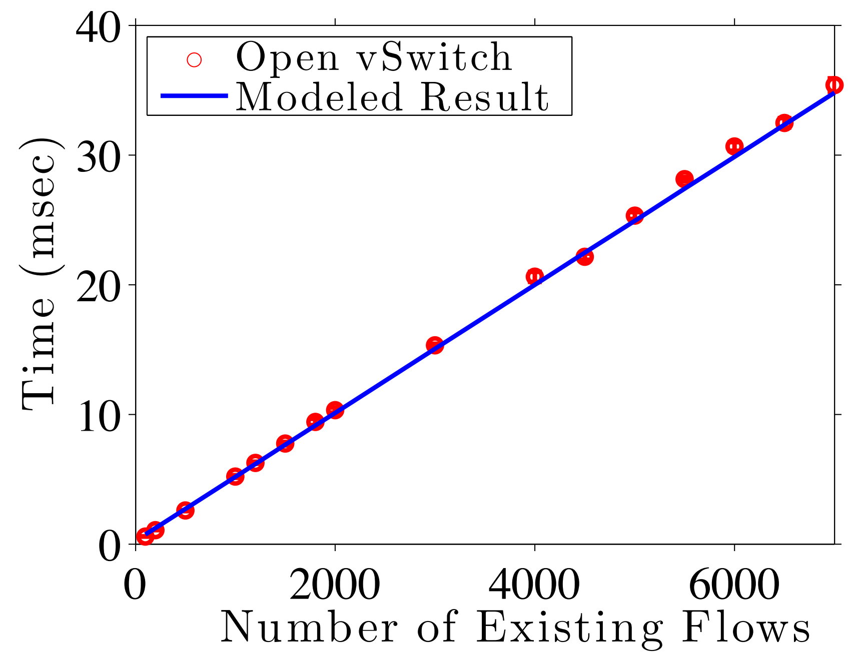
Open vSwitch

Modification Test Validation

- Validation results using ascending priority distribution
- Modeled results follow the results of OpenFlow switches



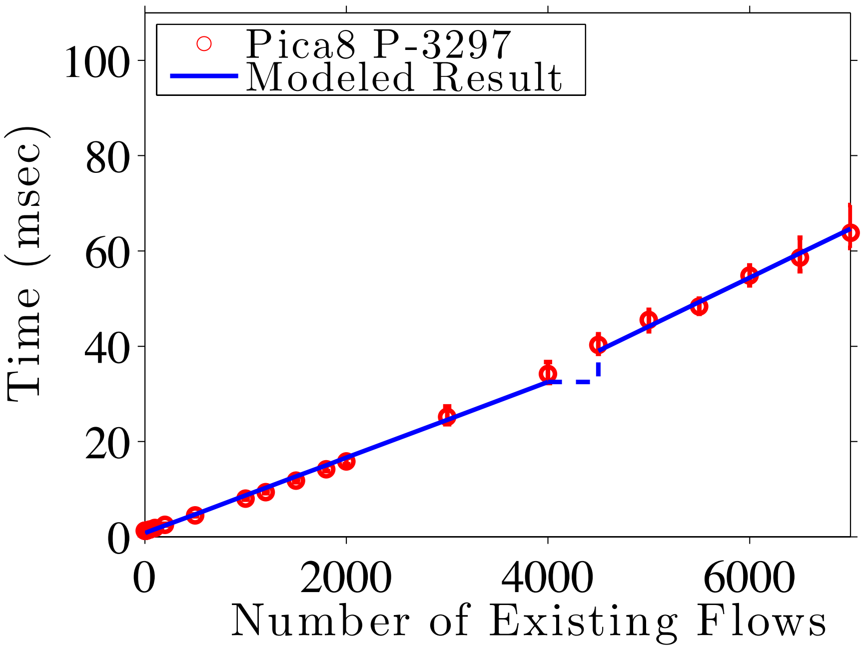
Pica8 P-3297



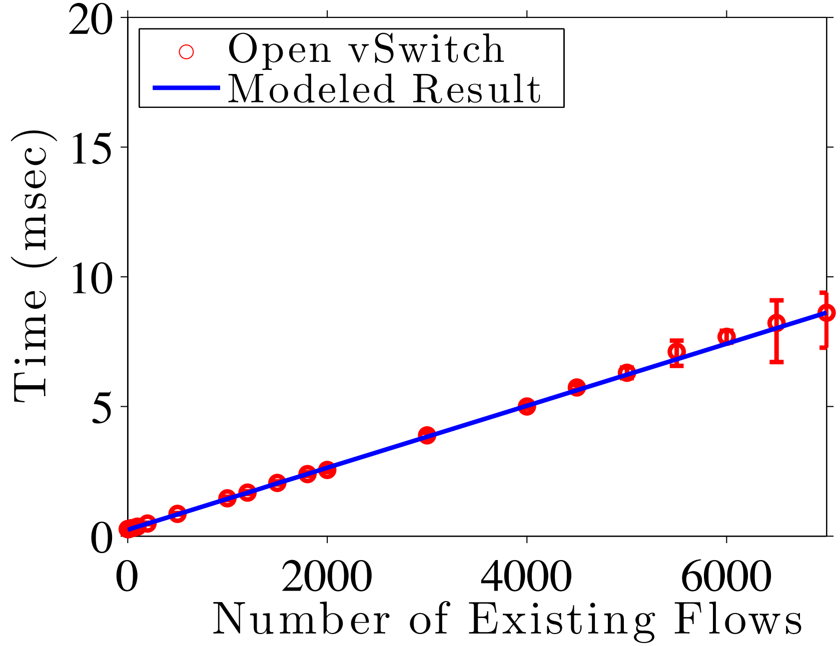
Open vSwitch

Deletion Test Validation

- Validation results using ascending priority distribution
- Modeled results follow the results of OpenFlow switches



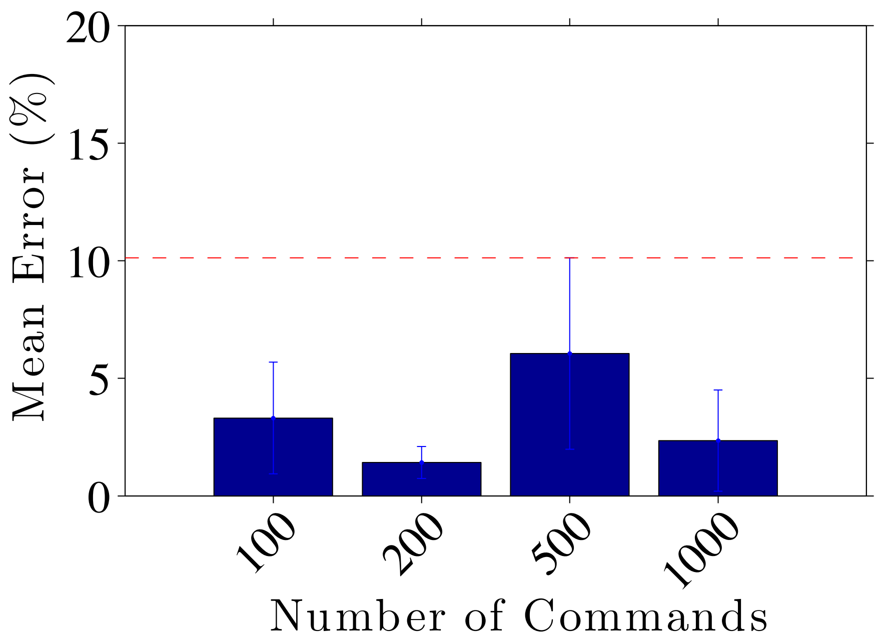
Pica8 P-3297



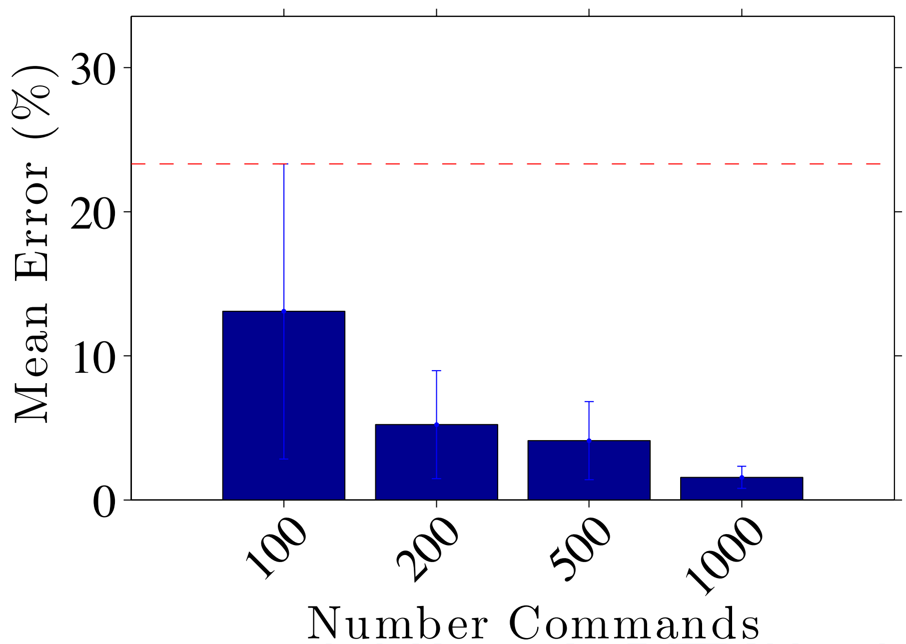
Open vSwitch

Random Test Validation

- Random commands, priorities, IP addresses, and arrival time
- Arrival time follows Poisson process with 100 flows/sec
- 16 random configurations for each command size
- Error rates are mostly under 20% on Pica8 and OvS



Pica8 P-3297



Open vSwitch

Data Plane Performance Measurement Studies and Modeling

- Test Scenarios
- Measurement Results
- Performance Models
- Model Validation

Factor Considerations for Data Plane Performance Tests

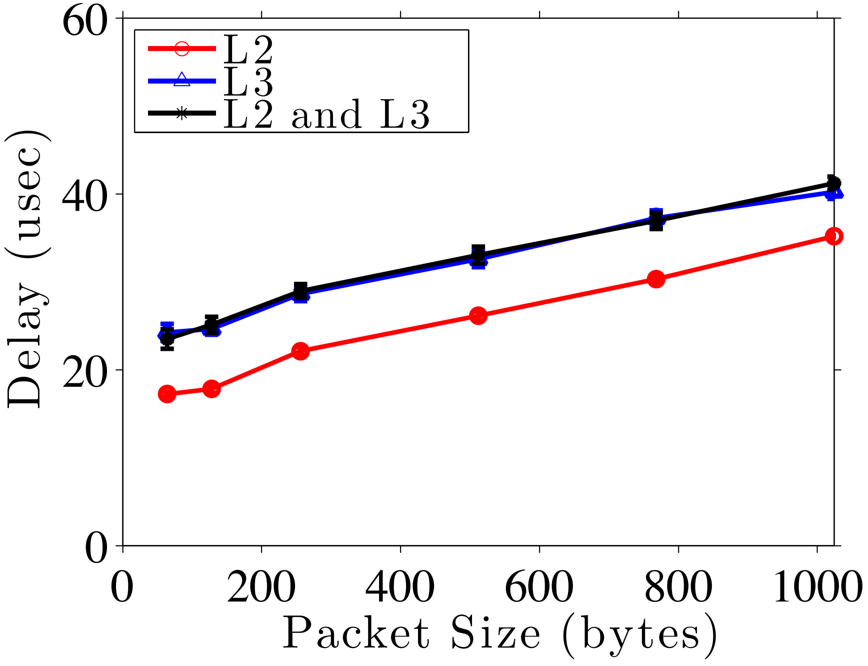
- Different matching fields used of tables flows
  - L2, L3, and both L2 and L3
- Number of existing flows in the flow table
- Inter-packet time
  - Time difference between last packet and current packet arrival time
- Packet size

Data Plane Tests

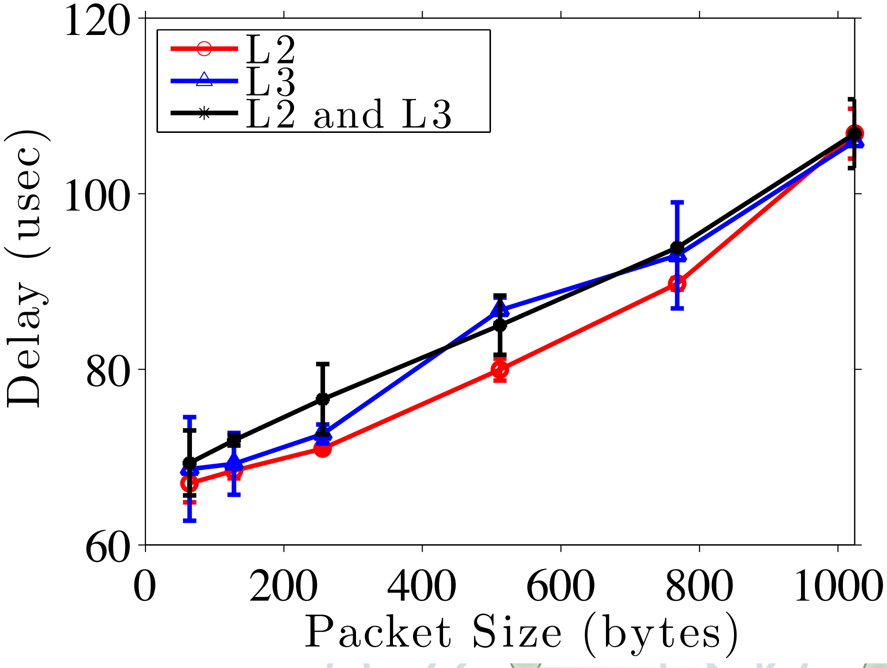
- **Performance metrics**
  - Forwarding delay
  - Throughput
- Preinstall corresponding flows for data plane traffic, with:
  - Different existing flow size
  - Different matching fields used
- Send data plane traffic, with:
  - Different packet size
  - Different inter-packet time

Packet Sizes and Matching Fields

- Larger packet sizes result in higher forwarding delays
- Delay time varies with different matching fields used



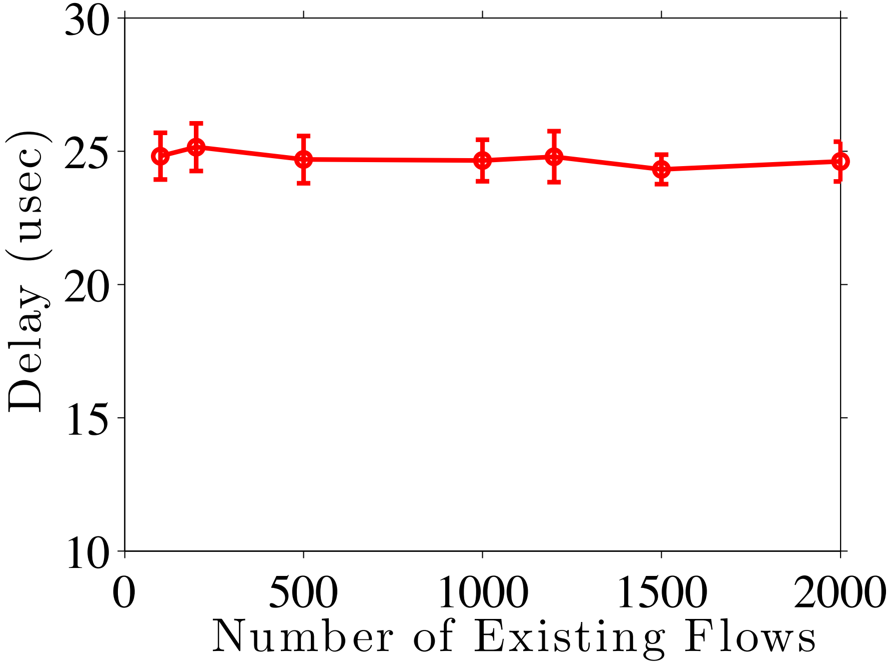
Pica8 P-3297



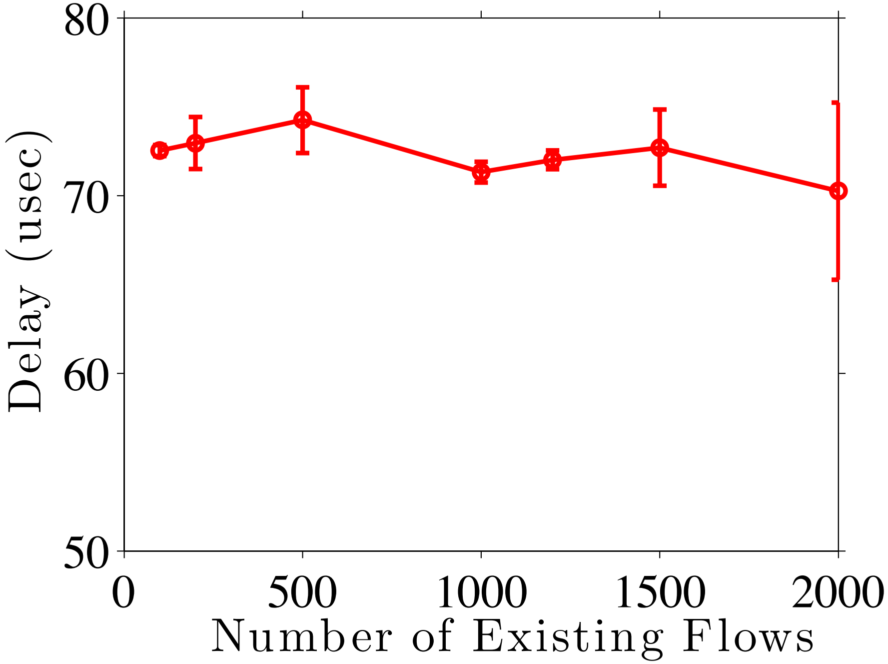
Open vSwitch

Existing Flow Sizes

- Existing flow sizes have little impact on forwarding delays



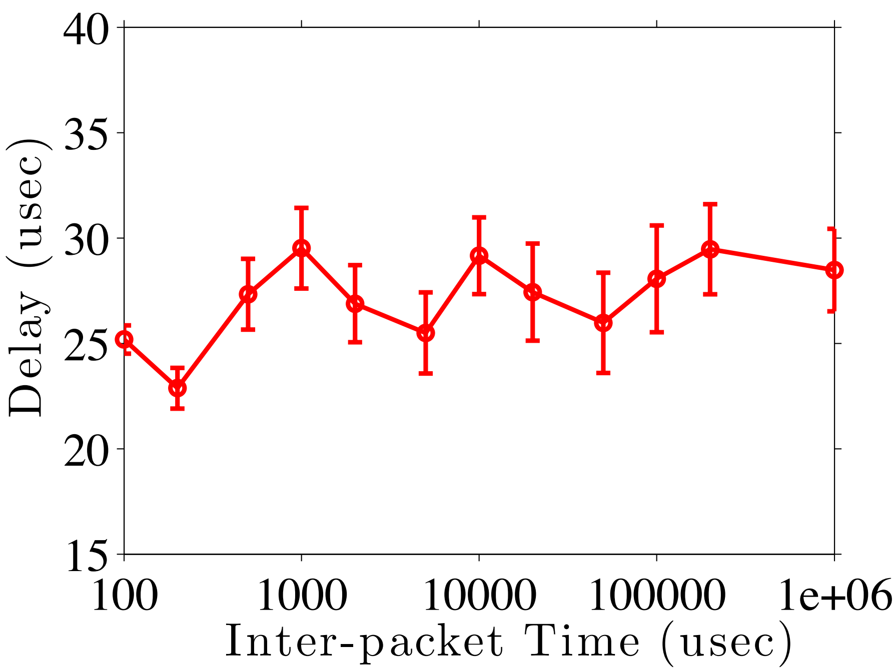
Pica8 P-3297



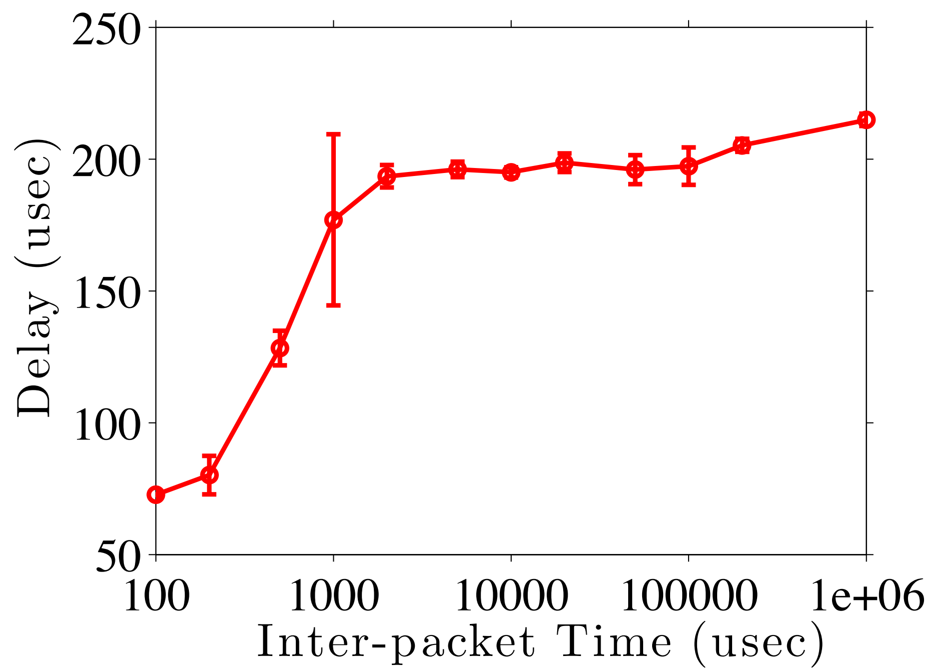
Open vSwitch

Inter-packet Time

- Multi-levels of forwarding delays with different inter-packet time



Pica8 P-3297

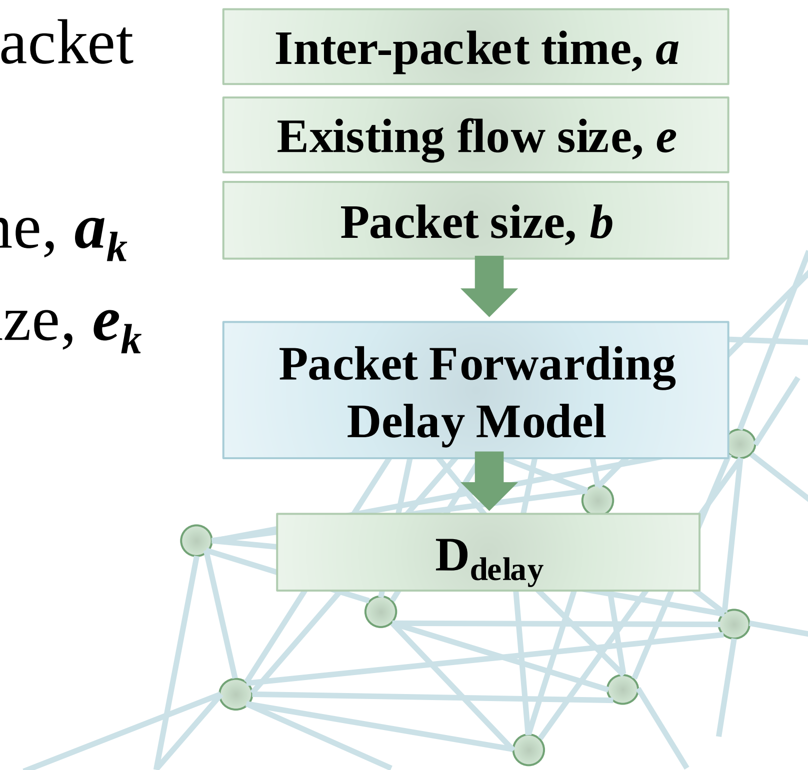


Open vSwitch

Packet Forwarding Delay Model

$$D_{delay} = \beta^t_{h_k} + \gamma^t_a \times \Delta a_k + \gamma^t_e \times \Delta e_k + \gamma^t_b \times \Delta b_k$$

- **k** denotes the index of the data plane packet
- $\beta^t_{h_k}$ denotes the base time
- γ^t_a : increasing rate for inter-packet time, $\boldsymbol{a_k}$
- γ^t_e : increasing rate for existing flow size, $\boldsymbol{e_k}$
- γ^t_b : increasing rate for packet size, $\boldsymbol{b_k}$

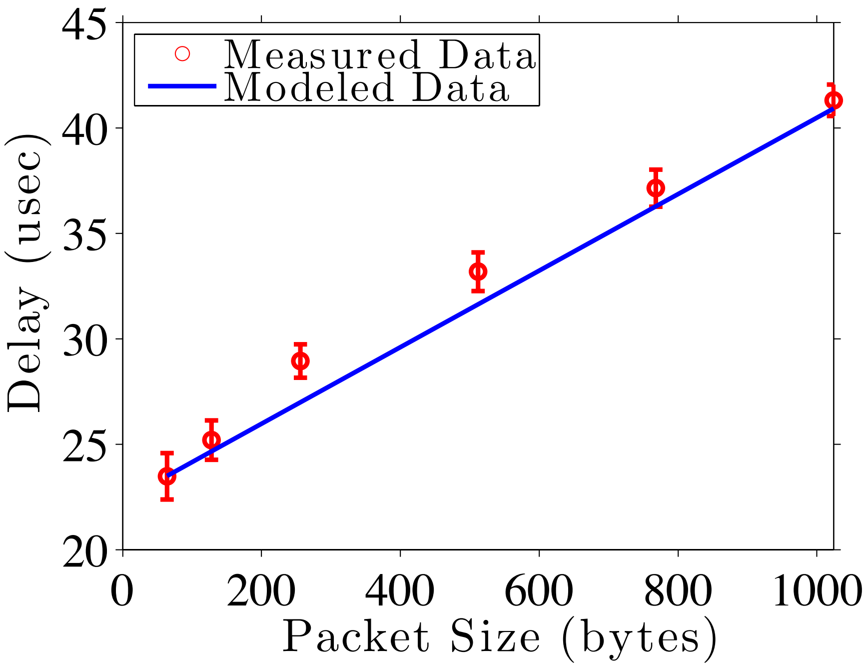


Validation Experiments

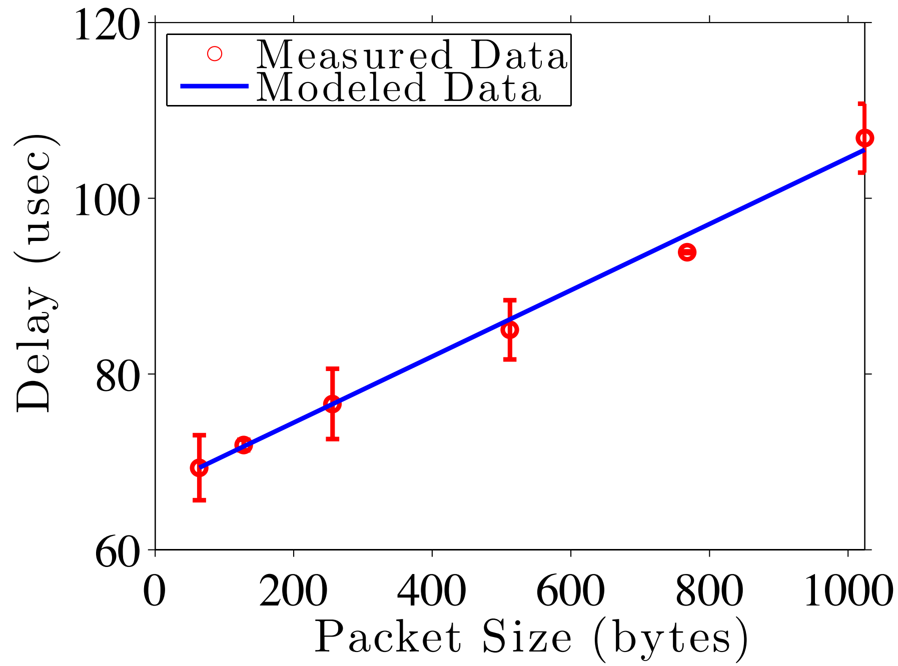
- Test scenarios
  - Different packet sizes
  - Different existing flow sizes
  - Different inter-packet time
  - Real world data plane traffic
    - Pcap trace collected from an educational site
- Validation results from Pica8 and OvS

Validation Results
- Different Packet Sizes

- 500 flows with L2/L3 matching fields used in the table
- Inter-packet time of 100 us packets sent
- Modeled results follow the result of real OpenFlow switch



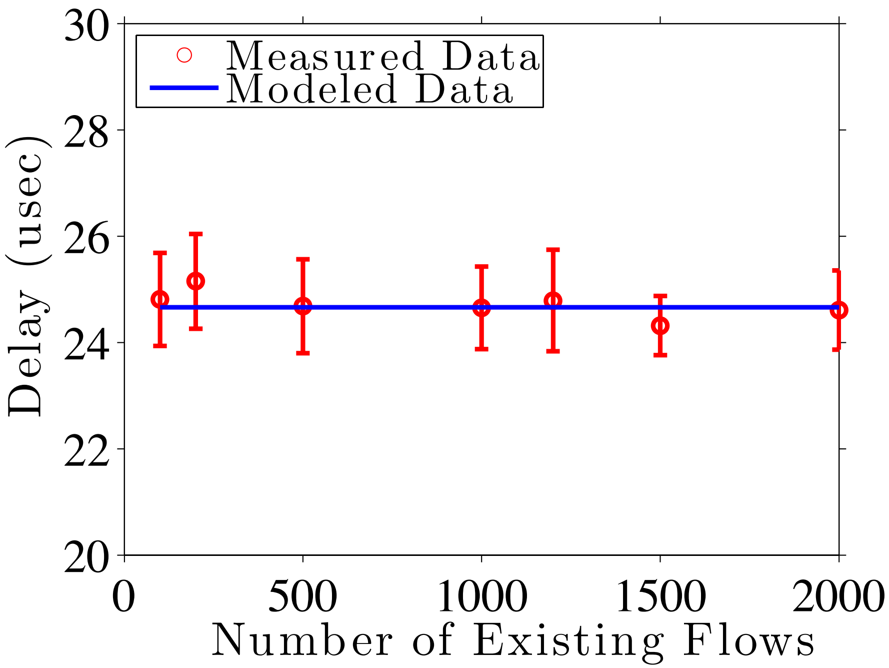
Pica8 P-3297



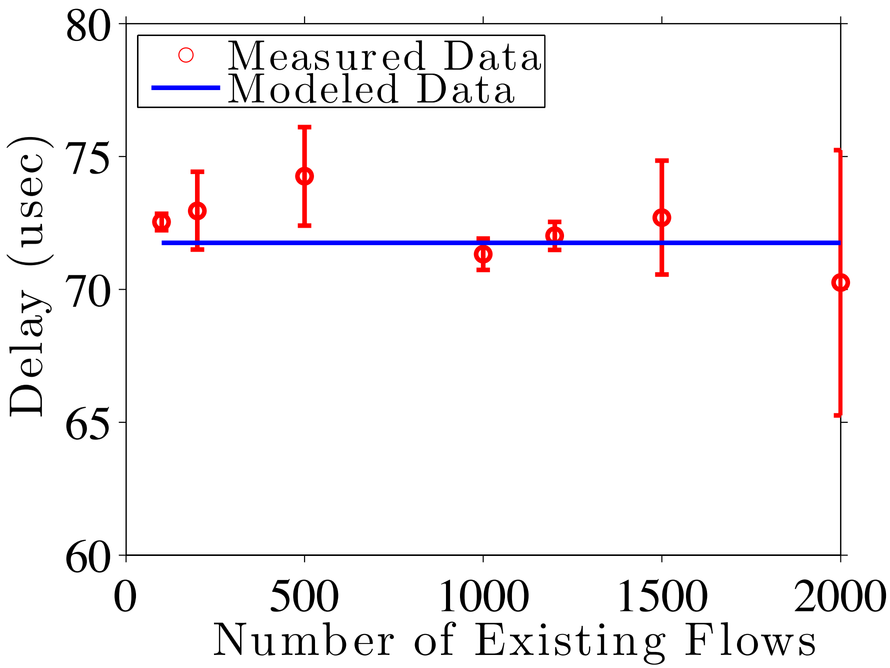
Open vSwitch

Validation Results
- Different Existing Flow Sizes

- L2/L3 matching fields used in the table
- Packet size of 128 bytes, inter-packet time of 100 us packets are sent



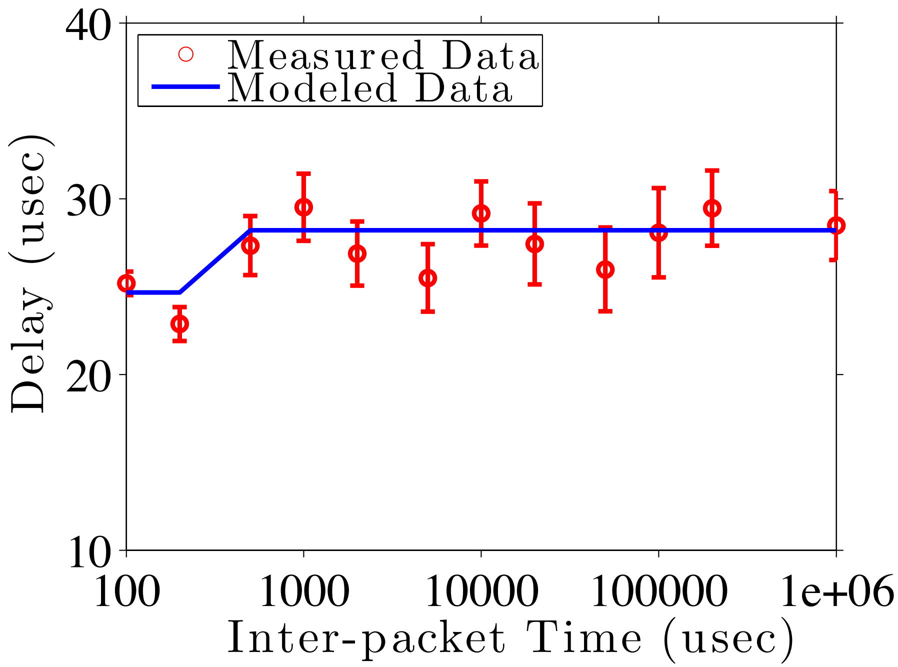
Pica8 P-3297



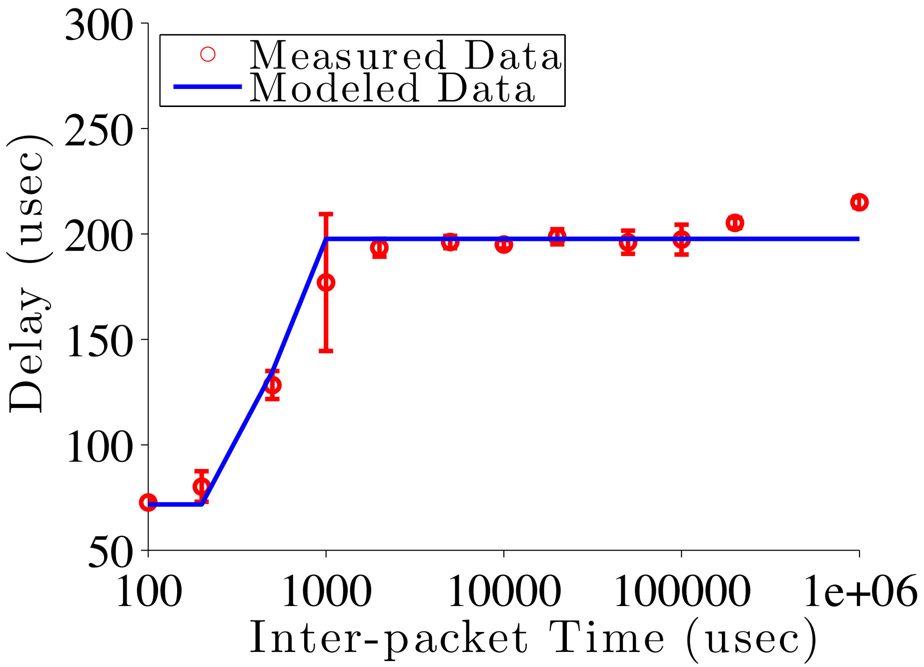
Open vSwitch

Validation Results
- Different Inter-packet Time

- 500 flows with L2/L3 matching fields used in the table
- Packet size of 128 bytes are sent



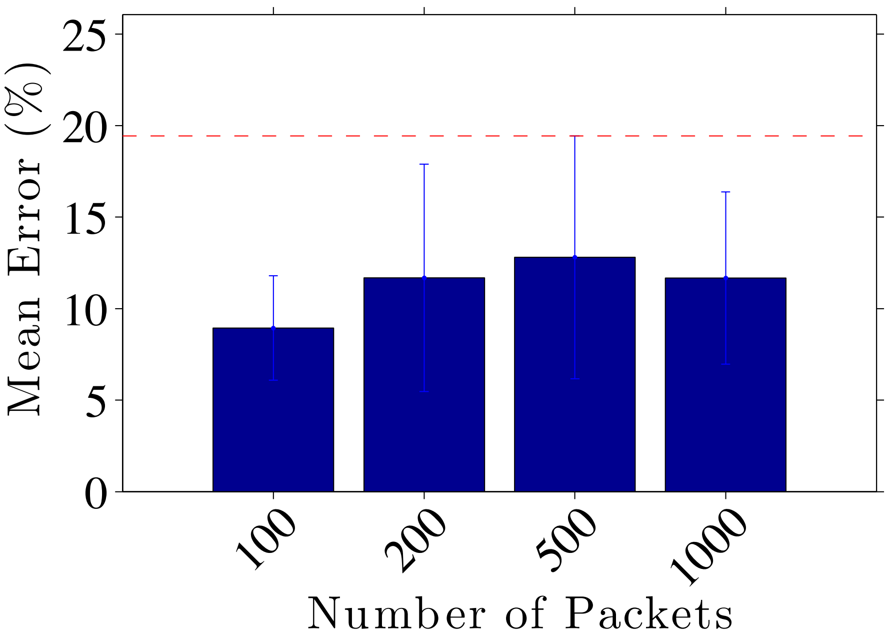
Pica8 P-3297



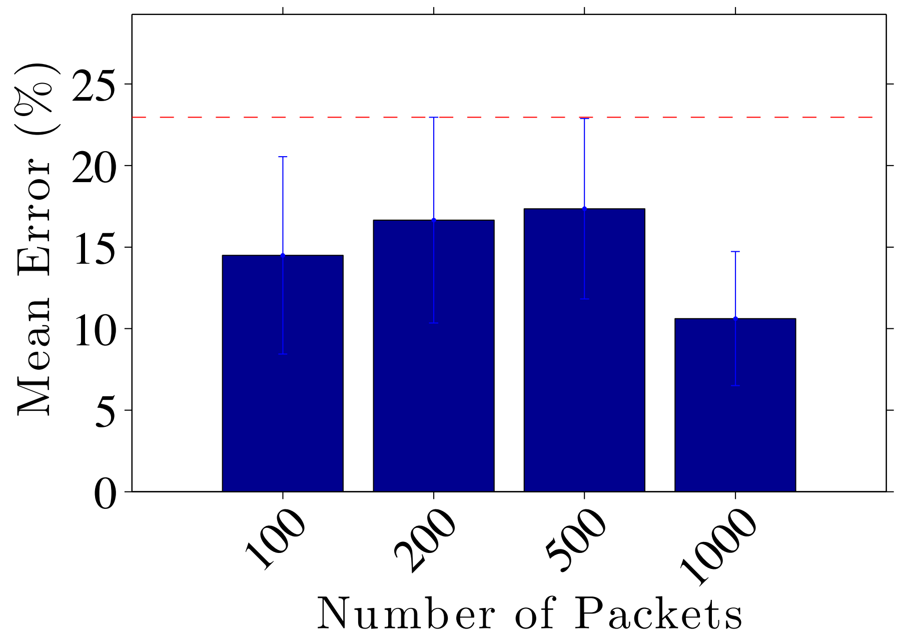
Open vSwitch

Validation Results
- Real World Data Plane Traffic

- Traces collected from a educational organization, with hundreds of students and employees in 2007, and over 200,000 packets captured
- Randomly select packets among 200,000 packets
- 16 different ranges for each number-of-packet sample



Pica8 P-3297

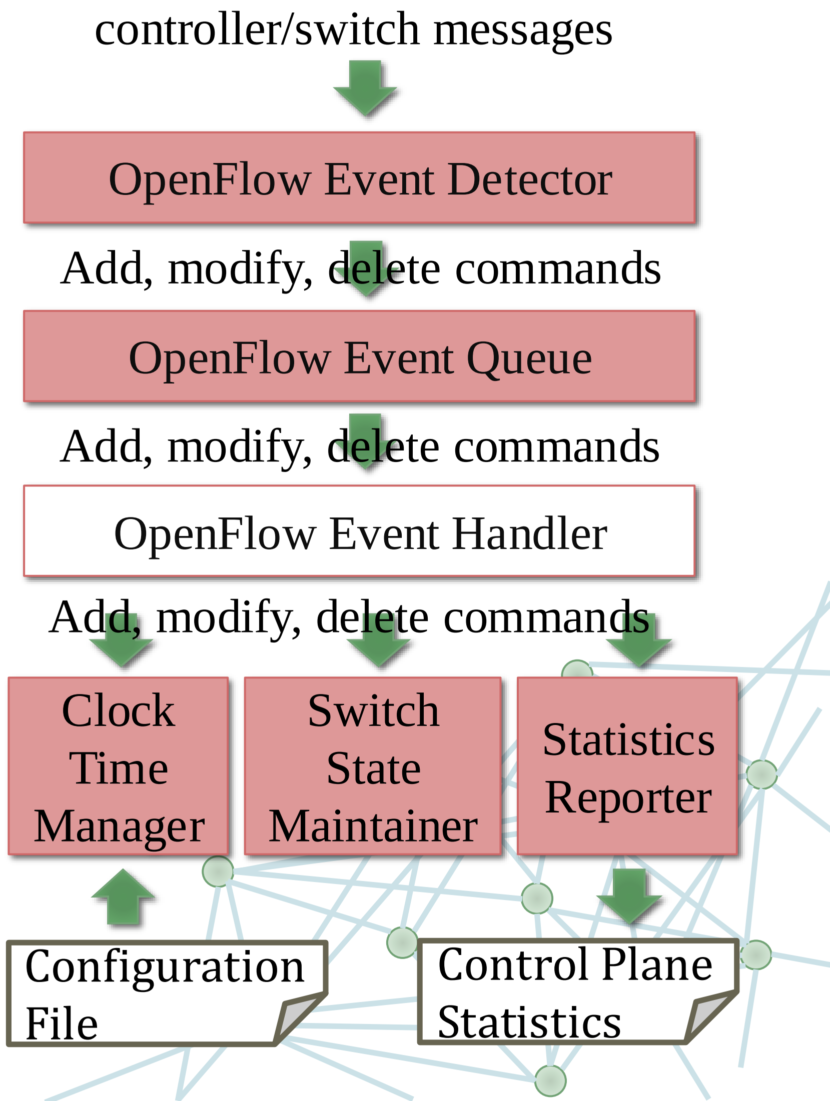


Open vSwitch

Emulator Implementations and Evaluations

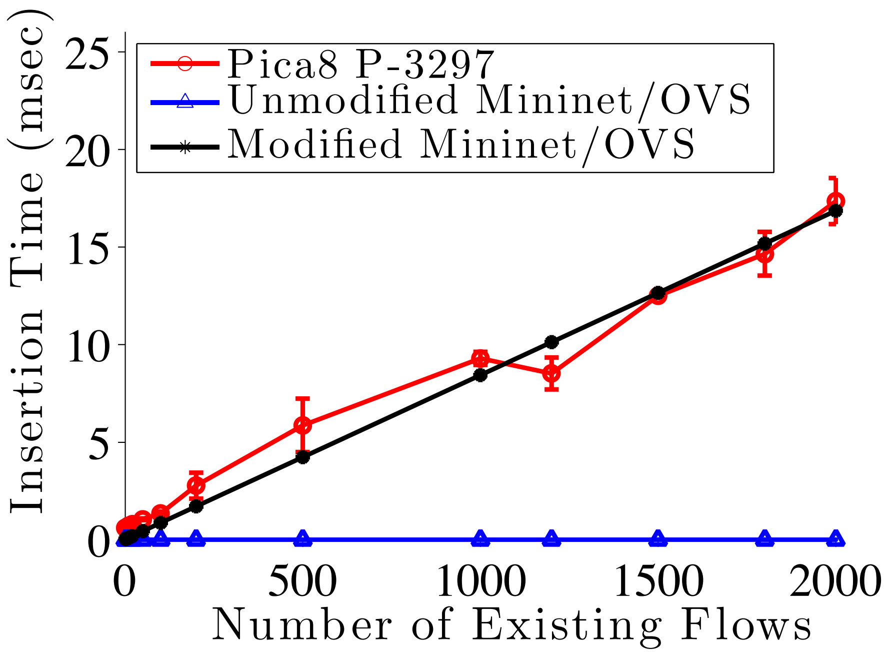
Emulator Implementation

- OpenFlow Event Detector extracts ***flow_mod*** events from controller/switch message and put them into OpenFlow Event Queue
- OpenFlow Event Handler fetches events from the queue and manipulate the events
- Clock Time Manager calculates modeled time and adjusts the time
- Switch State Maintainer updates switch states
- Statistics Reporter records each command information and performance

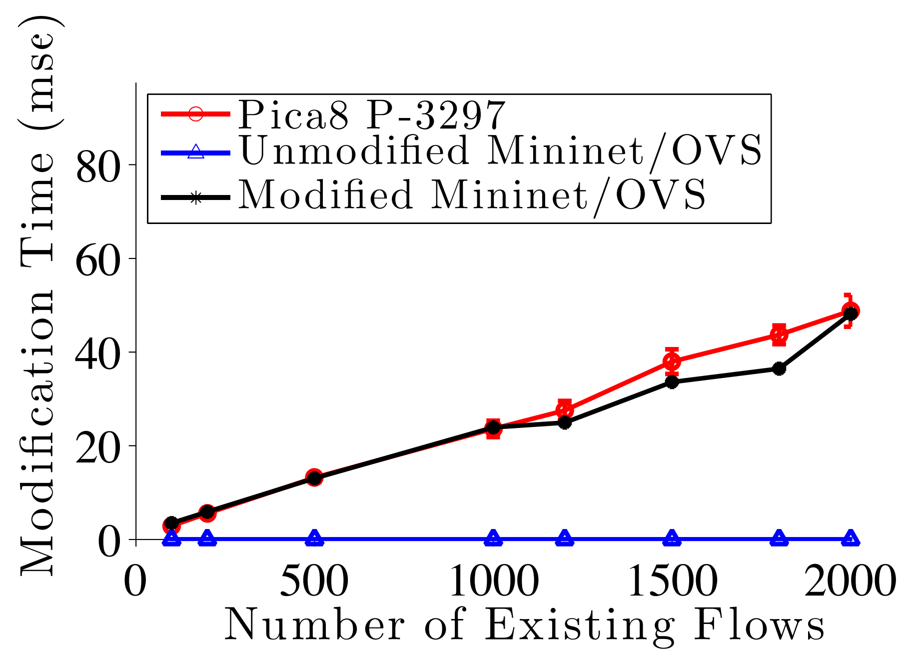


Evaluations

- Insertion/modification command tests
- Performance accuracy is much better than original Mininet/OvS



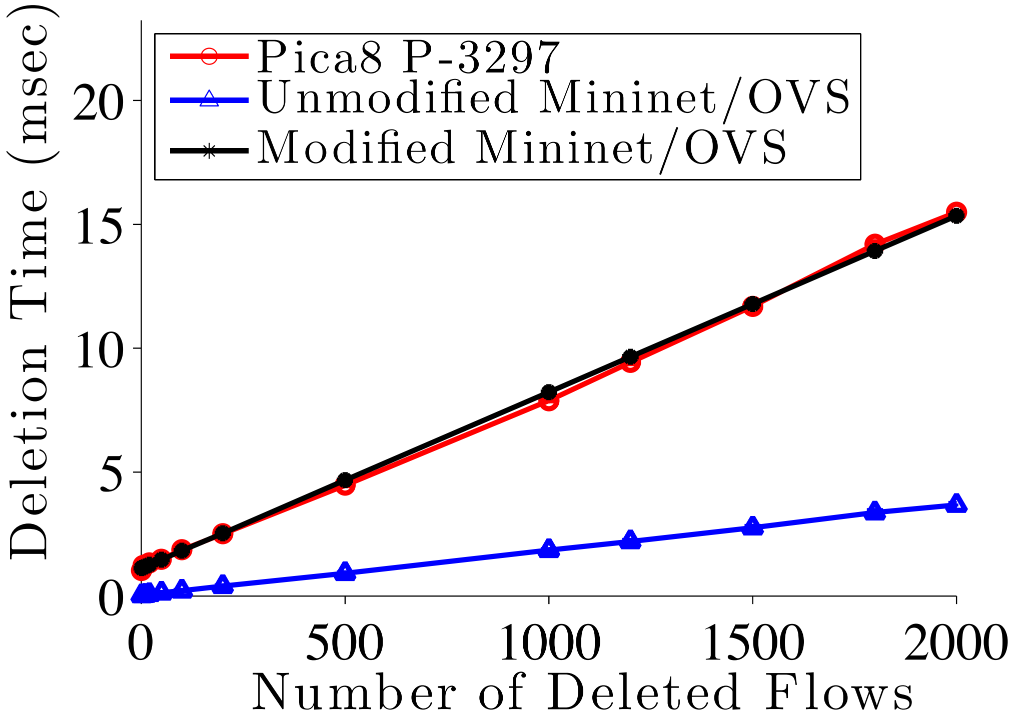
Insertion



Modification

Evaluations (cont.)

- Deletion tests
- No differences between our emulator and real Pica8 results



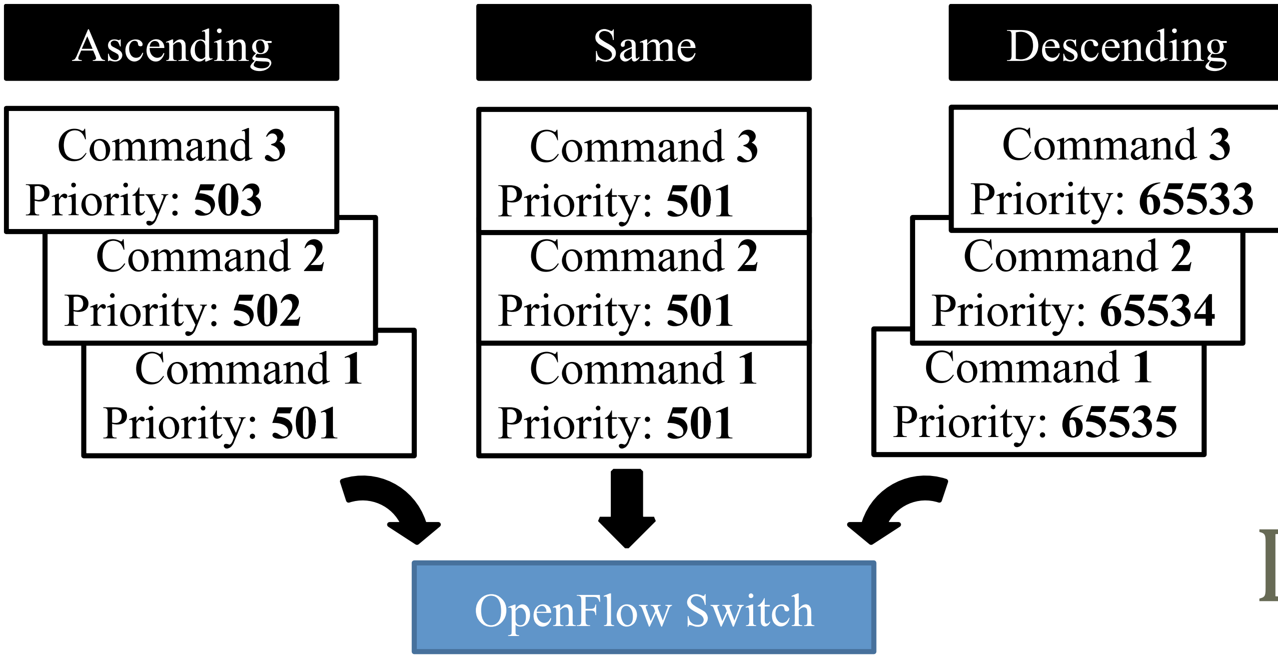
Deletion

Conclusion and Future Work

- **Switch Performance Benchmark**
  Propose automatic procedures for switch performance benchmarking
- **Performance Model and Switch-dependent Parameters**
  Propose control plane and data plane performance models for diverse OpenFlow switches
- **Emulator Implementation**
  Integrate performance models with OpenFlow emulator, Mininet/OvS
- **Future directions**
  - Adjustments on control plane performance models
  - Emulator implementation for data plane performance model, and thorough evaluations of the emulator using real-world traces
  - Update OFLOPS for OpenFlow higher versions support

Thanks much for your listening!

Different Priority Distributions



Ascending

Flow table		
…	…	Priority
…	…	503
…	…	502
…	…	501

Same

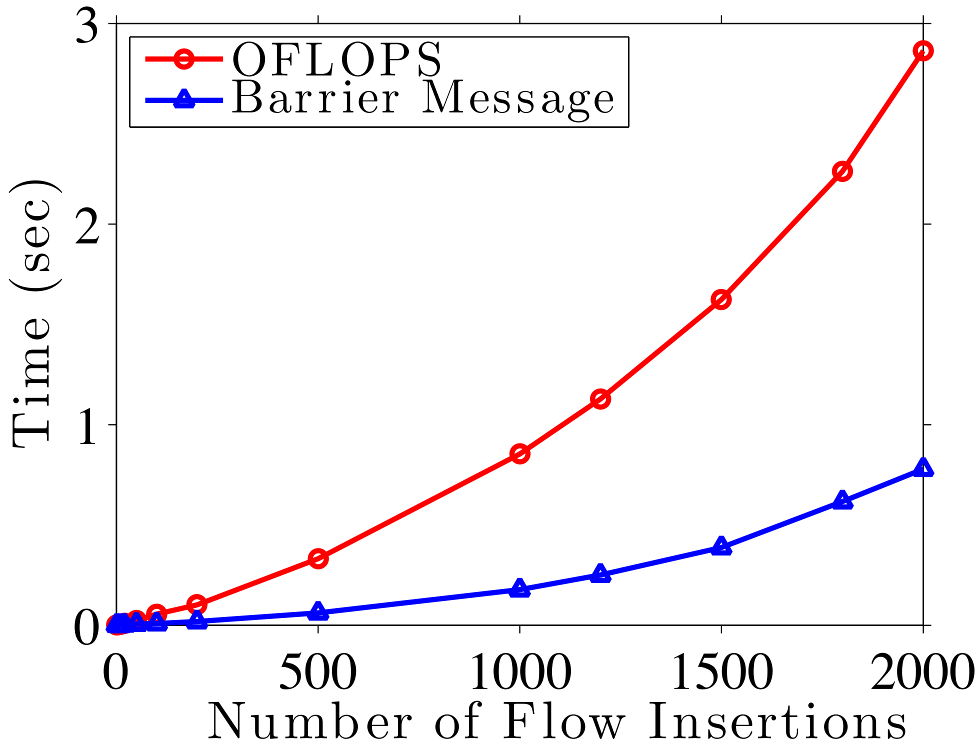
Flow table		
…	…	Priority
…	…	501
…	…	501
…	…	501

Descending

Flow table		
…	…	Priority
…	…	65535
…	…	65534
…	…	65533

Why OFLOPS?

- Barrier reply message should notify the completion of a series of commands sent before the barrier request message
- Not correctly implemented in all OpenFlow switches



Switch Benchmark Tool

Benchmark tool

Data plane profiling experiments

Control plane profiling experiments

Any OpenFlow switch

Flow insertion time

Flow modification time

Flow deletion time

Packet forwarding latency

Throughput

Sample figures