Working around Software Defined Networks (SDN) limitations: the compression of flow table & flow management cases

Dino Lopez – dino.lopez@unice.fr

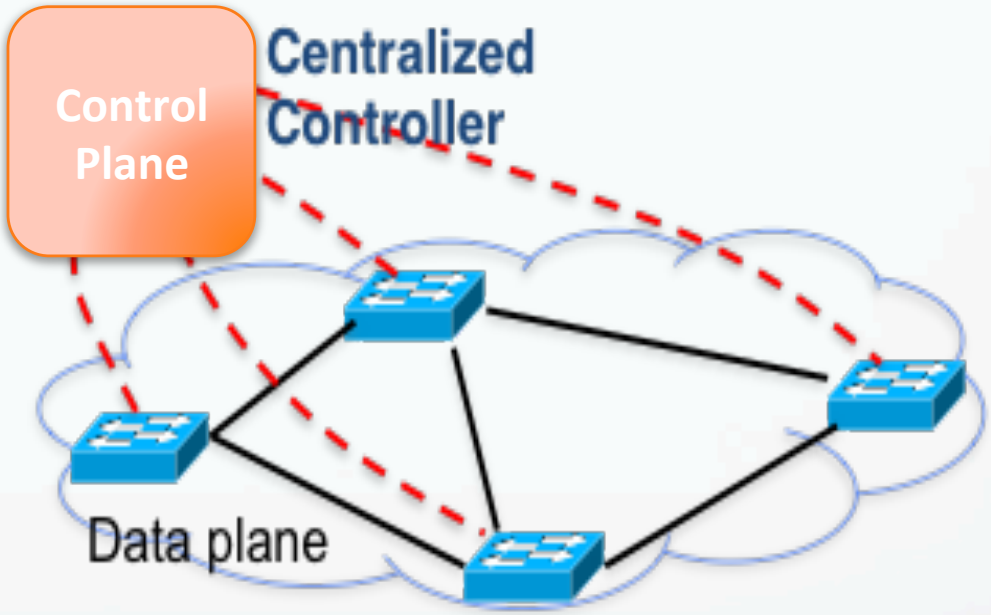
Université Côte d'Azur – Laboratoire I3S/CNRS UMR 7271

SigNet Team

Sophia Antipolis, France

Motivation

- SDN transforms network administrators into network programmers
  - The control plane and data plane are physically disjoint
  - The controller uses programs to modify the network devices' behavior
- SDN can use flow-based rules to forwards packets
- SDN devices can be hardware-based (e.g. HP switch) or software-based (e.g. OpenVSwitch)



Flow based rules:

Input port	Vlan ID	Vlan PCP	Src. MAC	Dst. MAC	Src. IP	Dst. IP	…	action

12 tuples

Some challenges

- Richer expressiveness of rules comes at a price
  - need for special hardware → TCAM (expensive and power hungry)
  - Typical hardware switches support a few thousands rules
  - Ex: our SDN platform, built with an HP 5412zl, supports 3000 hardware rules (up to 65000 software rules)

- The controller-SDN device channel may be the root of new problems
  - Bottleneck
  - Security risks
  - …

Improving Network Performance with SDN

- Several propositions based on SDN exist to improve the network performances [Popescu2015,Tu2014,Heller10,Cao2014,Han2014,Jain2013,…]
- In our case, we employed SDN to improve the performance in Data Center networks by providing
  - Fine-grain routing with per flow-based rules
  - Short flows protection

Enabling Fine-grain Routing with SDN

Work in Collaboration between the COATI team (INRIA) and SigNet (I3S)

How many forwarding rules?

- Up to one billion for the task management in the cloud

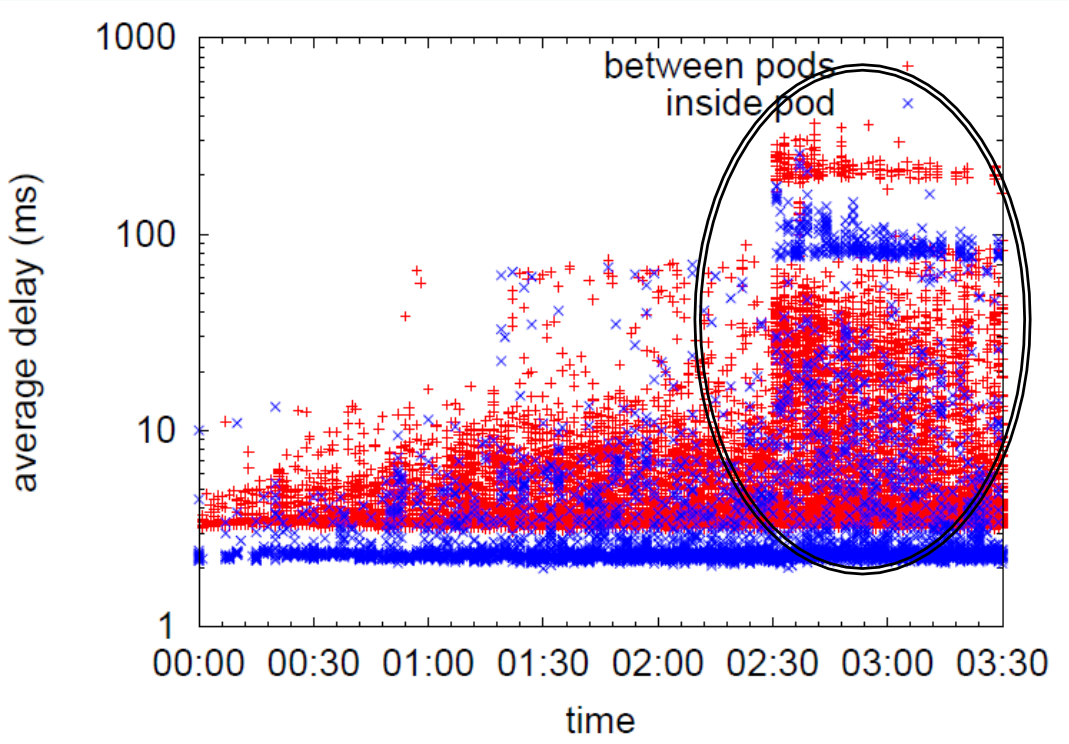
Masoud Moshref, Minlan Yu, Abhishek Sharma, and Ramesh Govin- dan, *vCRIB: Virtualized Rule Management in the Cloud*, USENIX HotCloud (Berkeley, CA, USA), 2012

- Top of the rack up to 78000 rules

Andrew R. Curtis, Jerey C. Mogul, Jean Tourrilhes, Praveen Yala- gandula, Puneet Sharma, and Sujata Banerjee, *DevoFlow: scaling flow management for high-performance networks*, SIGCOMM Comput. Commun. Rev. **41** (2011), no. 4, 254–265

- So what is the impact of limited TCAM ?

Impact of full TCAM in SDN-based networks



When the routing table is full, the switch needs to contact the controller for every incoming packet from an unknown flow (high delay)

Number of rules but also rate at which to install them….

- Problem: rule installation processed by slow path of SDN routers
  - Can handle a few hundreds events per second

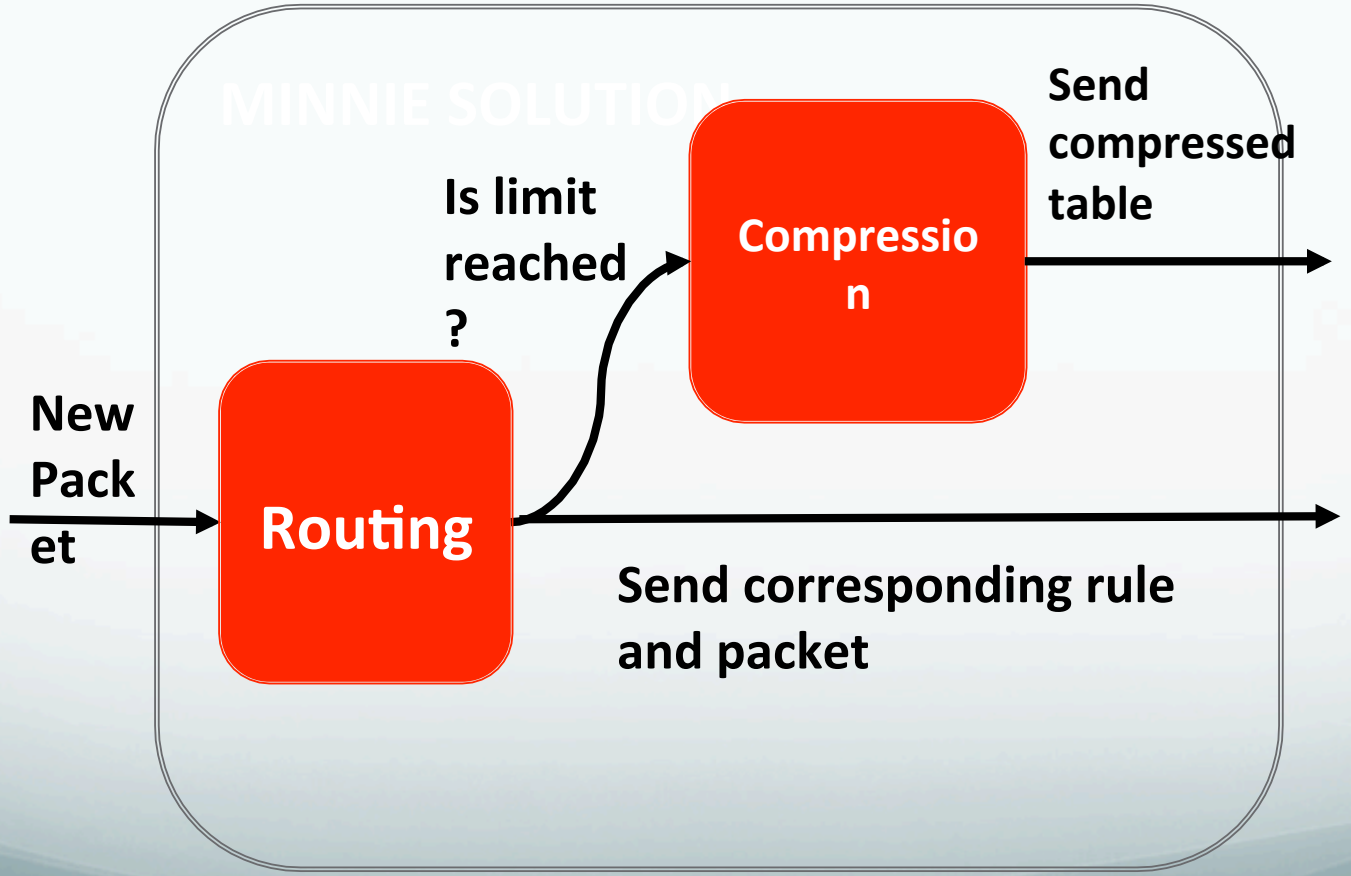
Maciej Kuzniar, Peter Peresíni, Dejan Kostic:
What You Need to Know About SDN Flow Tables. PAM 2015

Related Work

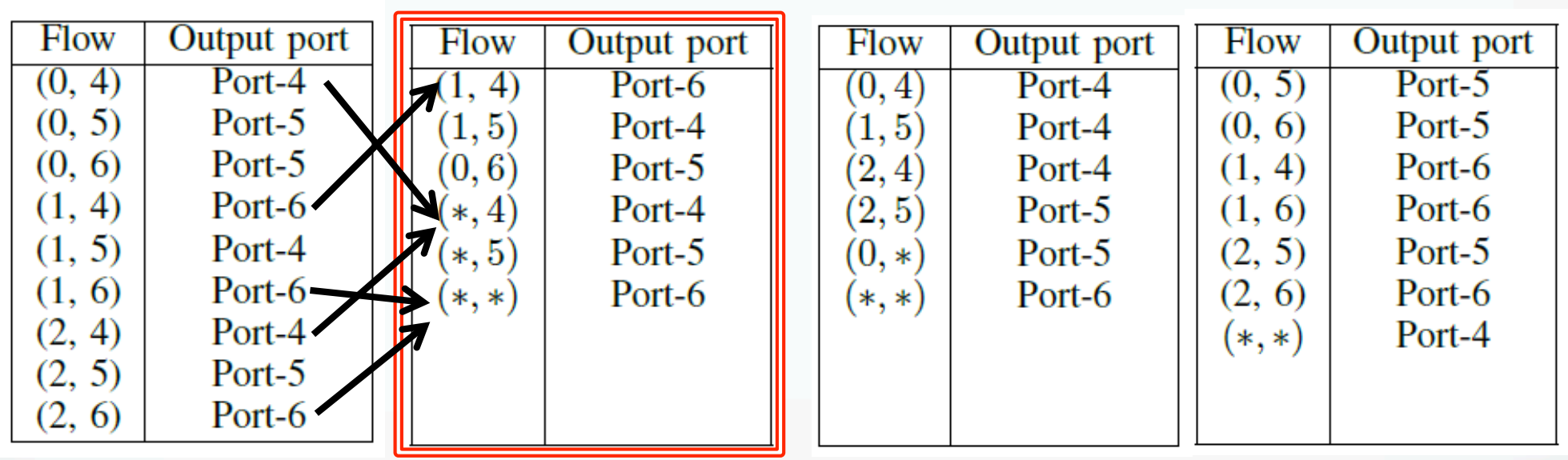
- **Eviction** to remove the least interesting rule (e.g. LRU, [Lee2013])
  - Which is the least interesting rule?
  - Frequent contact with controller
- **Split and distribute the rules** in network (e.g. DIFANE [Yu2010])
  - Needs an accurate view of the traffic matrix
- Use **default paths and deviate** from it when the target is close (e.g**. OFFICER** [DSaucez2015])
- **Compressing using wildcard rules.**
  - *Our proposition resides in this category*

The *MINNIE* Solution

Myriana Rifai, Nicolas Huin, Christelle Caillouet, Frédéric Giroire, Dino Lopez Pacheco, Joanna Moulierac, Guillaume Urvoy-Keller. “Too Many SDN Rules? Compress Them with MINNIE”. GLOBECOM 2015.



Minnie: An any-field Compression Solution



Flow	Output port
(0, 4)	Port-4
(0, 5)	Port-5
(0, 6)	Port-5
(1, 4)	Port-6
(1, 5)	Port-4
(1, 6)	Port-6
(2, 4)	Port-4
(2, 5)	Port-5
(2, 6)	Port-6

(a) Original routing table

Flow	Output port
(1, 4)	Port-6
(1, 5)	Port-4
(0, 6)	Port-5
(∗, 4)	Port-4
(∗, 5)	Port-5
(∗, ∗)	Port-6

(b) Comp by source

Flow	Output port
(0, 4)	Port-4
(1, 5)	Port-4
(2, 4)	Port-4
(2, 5)	Port-5
(0, ∗)	Port-5
(∗, ∗)	Port-6

(c) Comp by destination

Flow	Output port
(0, 5)	Port-5
(0, 6)	Port-5
(1, 4)	Port-6
(1, 6)	Port-6
(2, 5)	Port-5
(2, 6)	Port-6
(∗, ∗)	Port-4

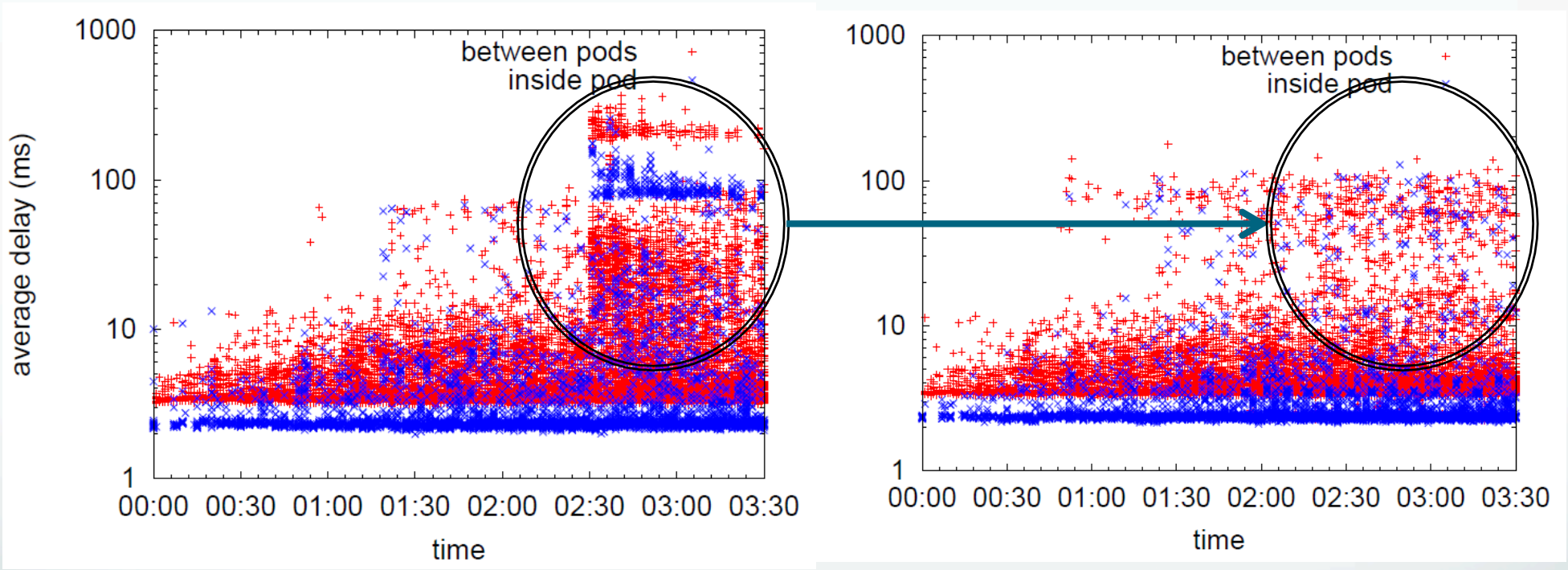
(d) Best default route

The smallest forwarding table is chosen

Evaluating MINNIE

- MINNIE has been evaluated by both simulations and experiments in a testbed
- Same code used during the simulation phase was used for the experimental part
  - We deployed MINNIE on top of a Beacon-based controller
- We could evaluate several parameters
  - Compression rate for large and most famous DC topologies : simulation
  - Impact of compression over the network traffic (delay & jitter): experiments on a testbed
- More information available at [Rifai2015]

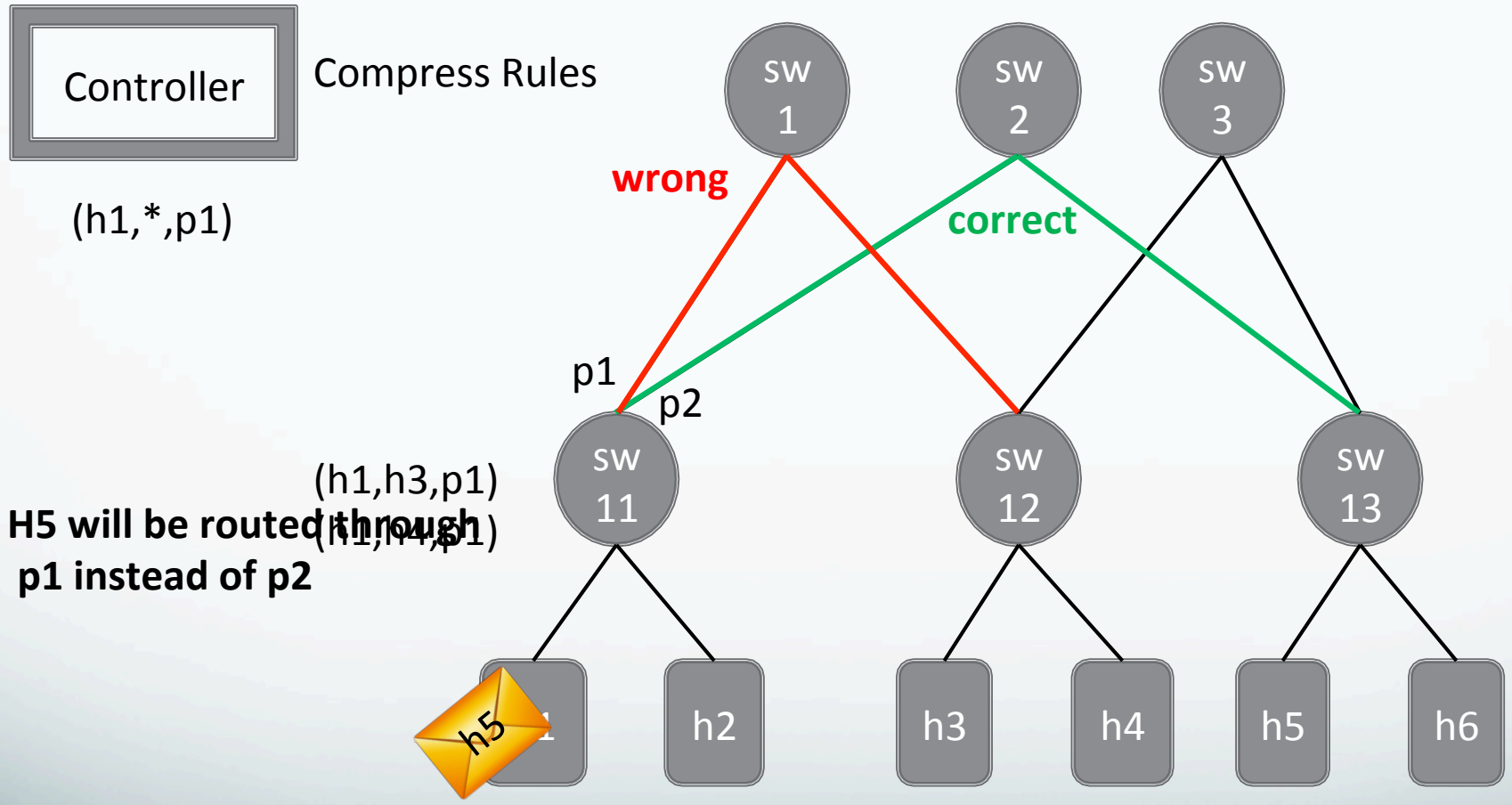
MINNIE in action



No compression

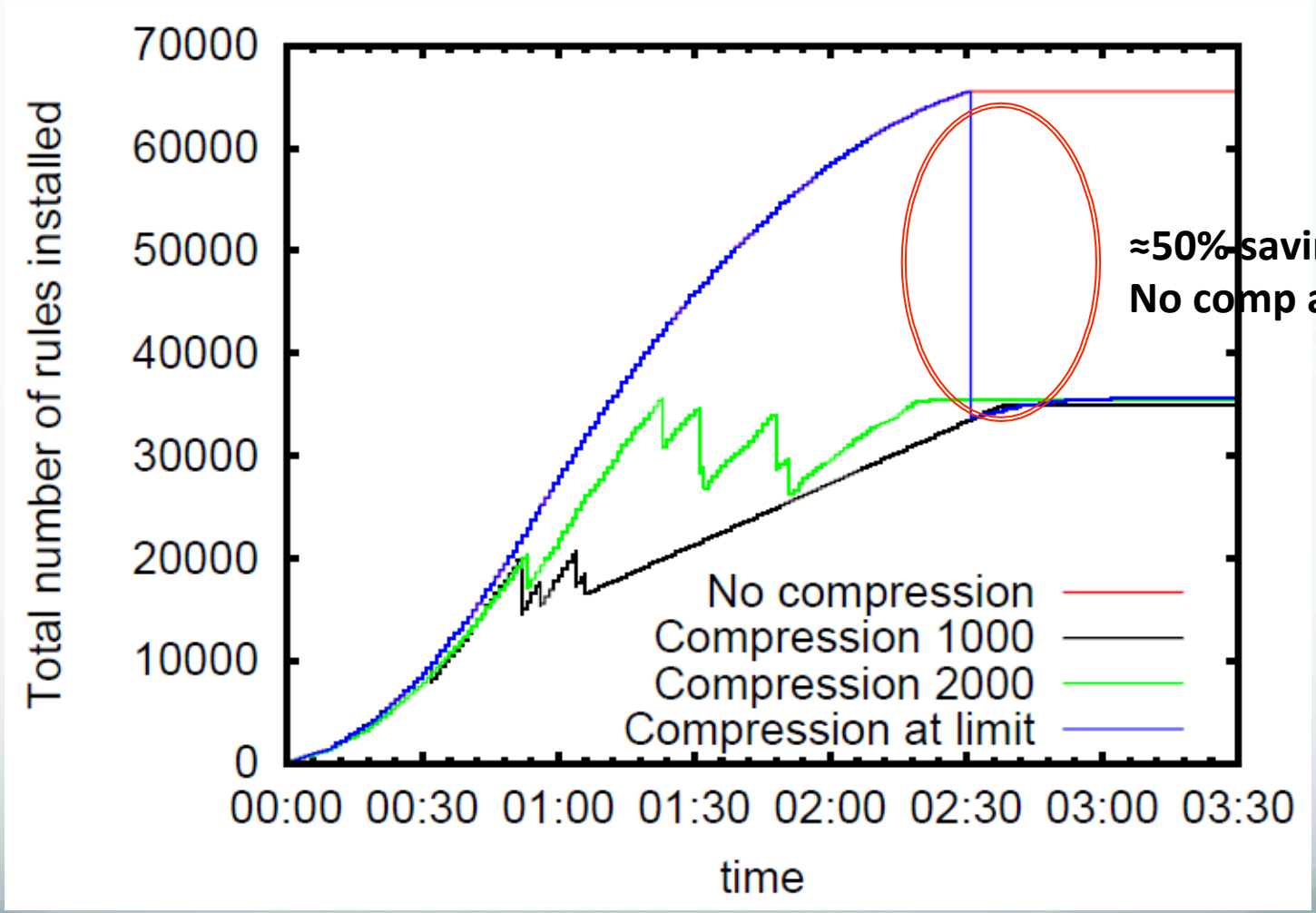
MINNIE

But… can we safely compress the routing table at the access switches?



Results
k-4 pods Fat Tree topology

Total number of rules installed on the physical switch



Short-Lived flows protection

Motivation

- Large flows represent a least percentage of network connections [Benson2010]
- These large flows tend to penalize short flows by consuming all network resources:
  - Majority of network connections are penalized (latency, losses, low QoE)

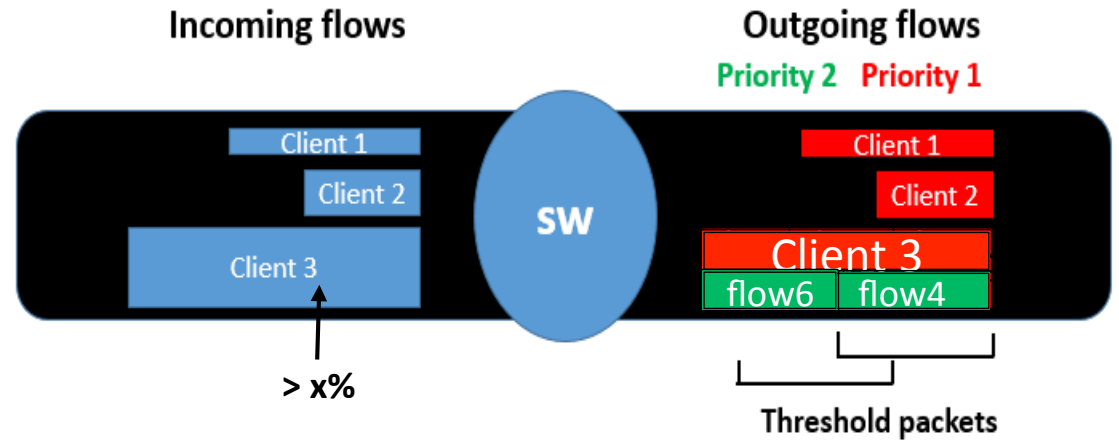
Related Work

- Use longer rarely used paths for large flows to decrease completion time [Tso2013]
- Detect long flows at the servers and find the most adapted path for such flows [Curtis2011]
- Active queue management strategies: RED, CoDel, etc.
- TCP modifications + multi-bit network signaling: DCTCP

Coarse-grained Scheduling with Software-Defined Networking Switches

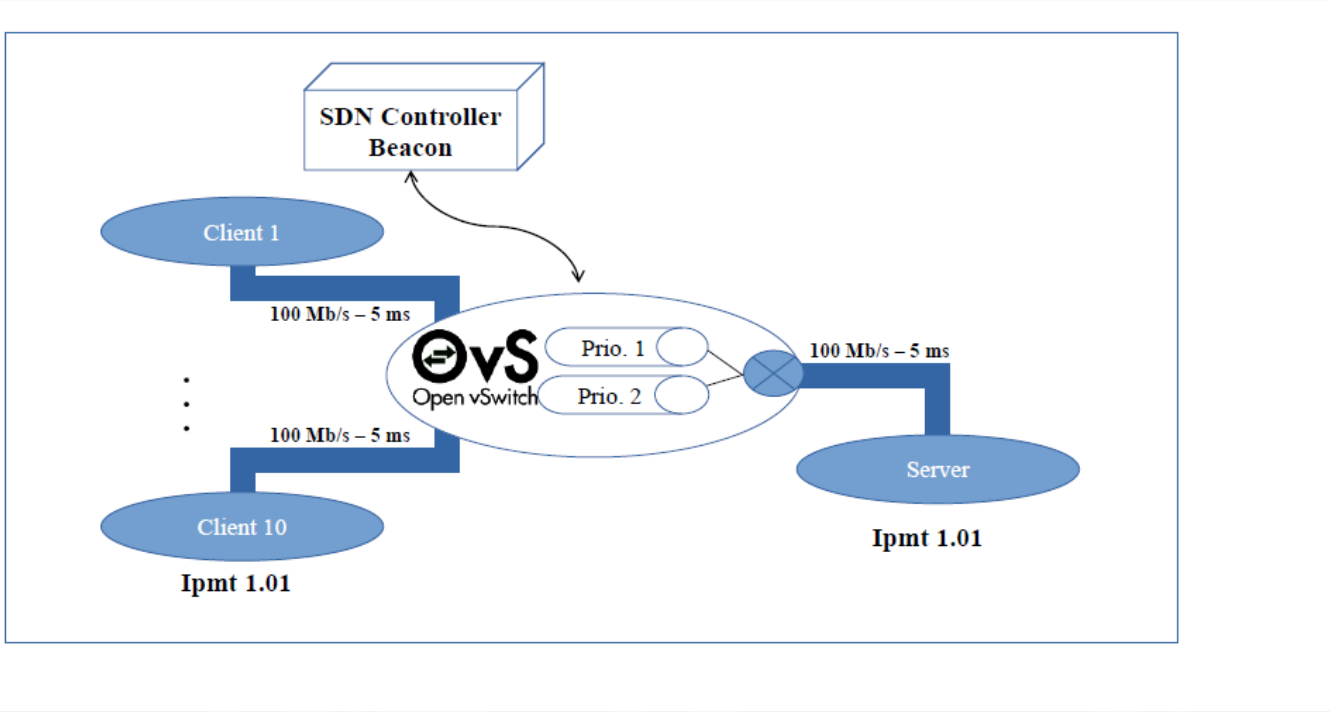
- The scheduler monitors and *retrieves* per rules statistics every *T* ms

1. State-full scheduler:
   - Install one rule per flow
   - Use the lower priority queue after threshold_pkts sent by the flow



2. Scalable scheduler:
   - One rule → several flows
   - When such a rule uses more than x% of link capacity, install rules per flow
   - Monitor the new per flow rules
   - Use the lower prioirity queue after threshold_pkts sent by the flow
   - After few Tmonitor cycles, large flows are isolated and the grouped forwarding rule is reinstalled

Experimental testbed with Mininet

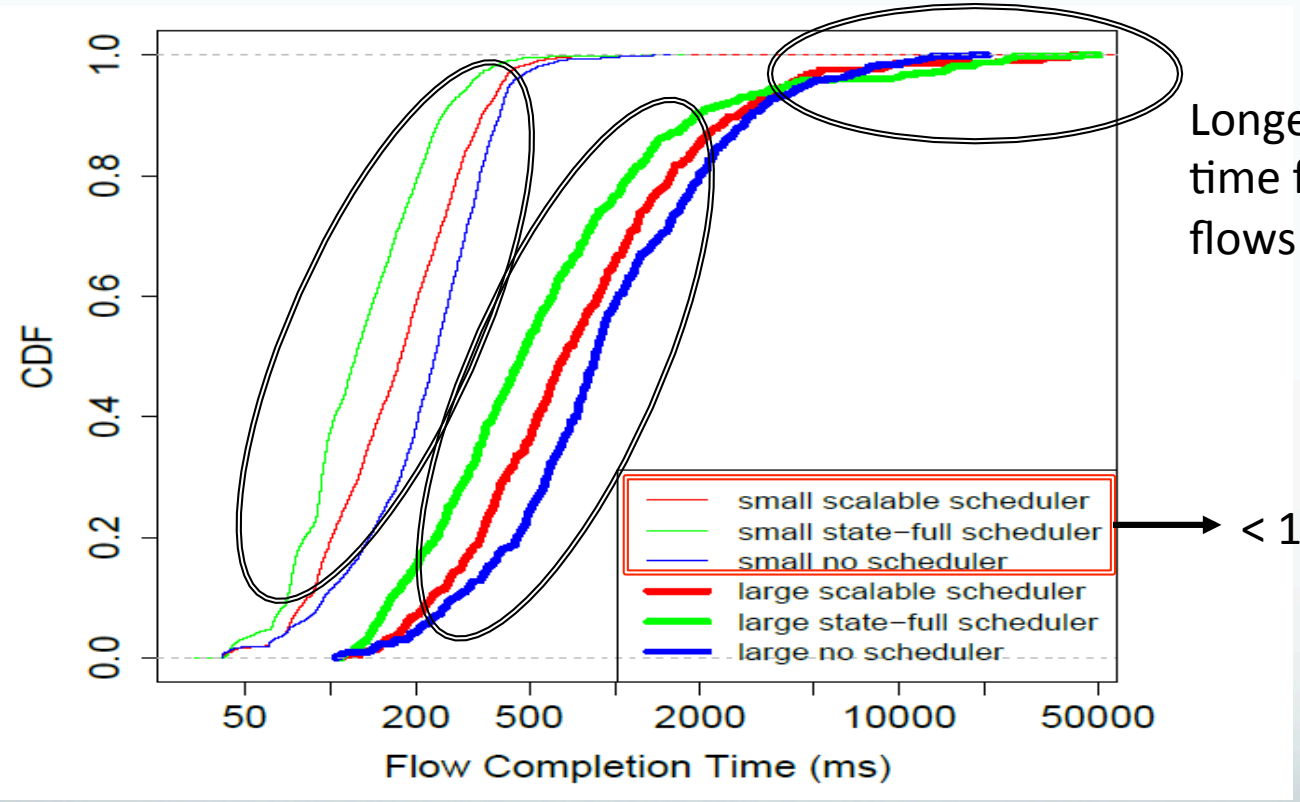


Network traffic characteristics:

- Zipf distribution with a flow size between 15KB (10 packets) and 10 MB
- The average flow size is around 100 packets
- Short flows represents, in average, 90% of the total number of flows

Results

Flow completion time is better in statefull and scalable scheduler for short and medium long flows



Longer completion time for very long flows

< 100 packets

Flow completion time CDF for long and short flows

Return on experience & conclusions

Technical Challenges

- Are SDN devices mature enough? Some actions required by the OpenFlow standard are not supported in ASICs by some SDN hardware models. Ex.
  - MAC address changes
  - Port matching (transport header dissection?)
  - Per-flow statistics retrieval
- SDN devices are completely reactive. Is it the best design choice?
  - Every new action (forward, drop, …) must be explicitly indicated. Can it be improved? Push up some “intelligence”?
  - Statistics retrieval needs request/response messages. Should it be part of the OF-Mon (Notification Framework)?
- Minor technical problems
  - Some switches model does not support queuing priority

Conclusions

- Our experiences show that SDN can improve the network performance
  - Current SDN hardware devices can indeed provide fine grain network management
  - Monitoring for management can be done by leveraging SDN features
- From the research perspective
  - SDN catalyzes the research reproducibility
  - SDN solutions can be more easily tested and improved
    - SDN hardware can be used to test your proposition in real environments
    - SDN software switches are really useful in absence of hardware devices
- It is now in your hands: deploy your solution
  - Make SDN evolve in the right way