COATI

Compressing Two-dimensional Routing Tables with Order

Frédéric Giroire, Frédéric Havet, Joanna Moulierac

COATI – I3S (CNRS-Université de Nice)/Inria
Sophia Antipolis

COATI

Compressing Two-dimensional Routing Tables with Order

Frédéric Giroire, Frédéric Havet, Joanna Moulierac

3S (CN rsité de Nice)/Inria

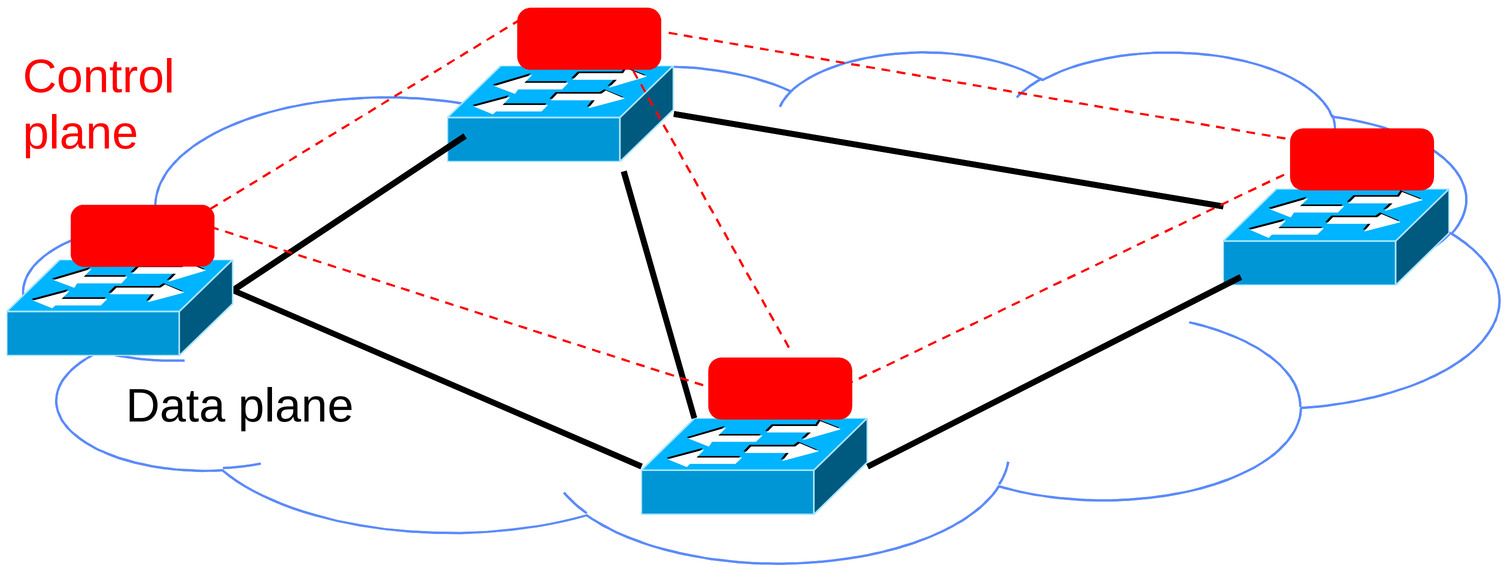
So polis

INOC 2015

7th International Network Optimization Conference

May 18-20, 2015, Warsaw, Poland

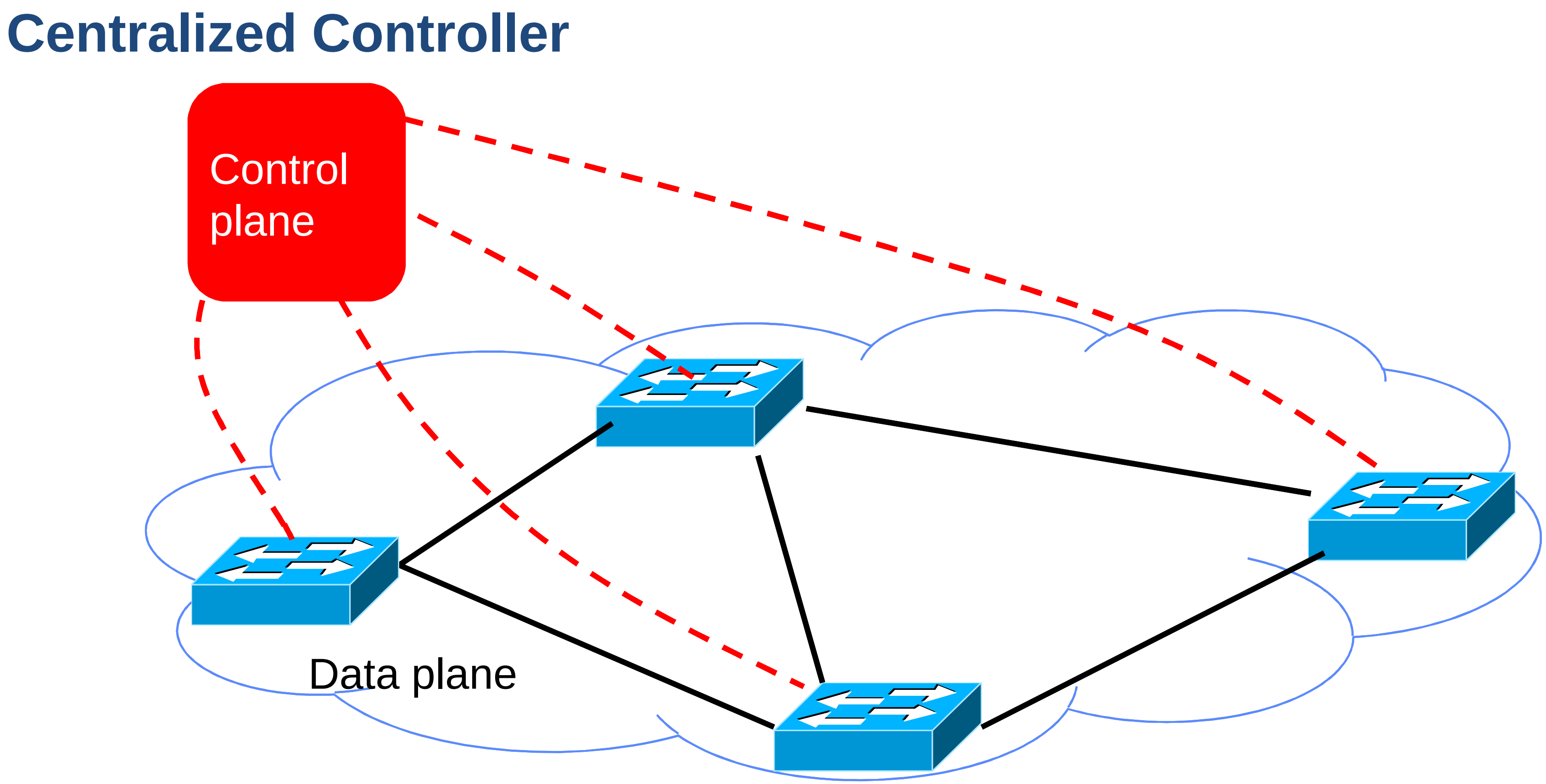
Traditional Network



Routers and switches are “closed systems” →

difficulties to deploy new network protocols.

Software Defined Networks (SDN)



Network elements are elementary switches, the intelligence is implemented by a logically centralized controller that manages the switches (i.e., install forwarding rules).

Software-Defined Networks

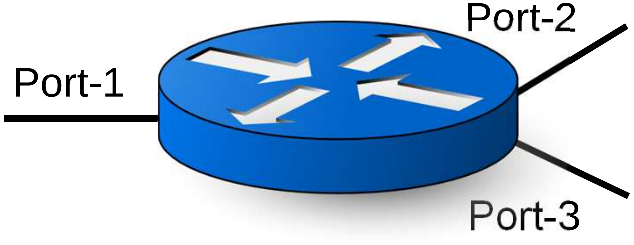
Networks are managed by configuration but

- each protocol has its own configuration set,
- each constructor has its own configuration language,
- it is hard to construct configurations that support all the possible cases.

SDN conceives the network as a **program**:

- Operators do not configure the network, they program it.
- Operators do not interact directly with devices.
- Network logic is implemented by humans but network elements are never touched by humans.

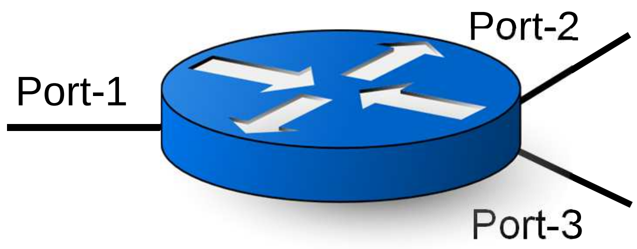
Forwarding/Routing Table



(Src, Dst)	Output port
(0, 2)	**2**
(0, 3)	**2**
(0, 4)	**1**
(1, 4)	**1**
(0, 5)	**2**
(2, 4)	**3**

Routing table

Forwarding/Routing Table



(Src, Dst)	Output port
(0, 2)	**2**
(0, 3)	**2**
(0, 4)	**1**
(1, 4)	**1**
(0, 5)	**2**
(2, 4)	**3**

Routing table

Problem: SDN routing tables are stored with TCAM memory which are expensive, power-hungry and with a limited size.

How do we deal with small rule tables?

- Eviction (e.g., LRU) or remove the least interesting rule when a new rule must be added.
- Split and distribute the rules in network.
- Use a minimum number of paths
- Compressing [1]

[1] Compressing Two-Dimensional Routing Tables (2003), Subhash Suri , Tuomas Sandholm , Priyank Warkhede

Compressing a single routing table using wildcards on src and dest

(Src, Dst)	Output port
(0, 2)	**2**
(0, 3)	**2**
(0, 4)	**1**
(1, 4)	**1**
(0, 5)	**2**
(2, 4)	**2**
(2, 5)	**1**
(1, 5)	**1**

with wildcards

(Src, Dst)	Output port
(2, 4)	**2**
(*, 4)	**1**
(0, *)	**2**
(*, 5)	**1**

The order is important

The **first matching rule** is applied

Example: If $(*,4)\rightarrow 4$ is before $(1,*)\rightarrow 6$, then $(1,4)$ will be routed through 4, and not 6.

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

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

Cycles and inconsistencies

destination

source	3	4
1	0	1
2	1	0

The number of rules cannot be reduced in this case:

(Src, Dst)	Output port
(1, 3)	**0**
(1, 4)	**1**
(2, 3)	**1**
(2, 4)	**0**

(Src, Dst)	Output port
(1, 3)	**0**
(1, *)	**1**
(*, 4)	**0**
(*, 3)	**1**

Routing-List Problem

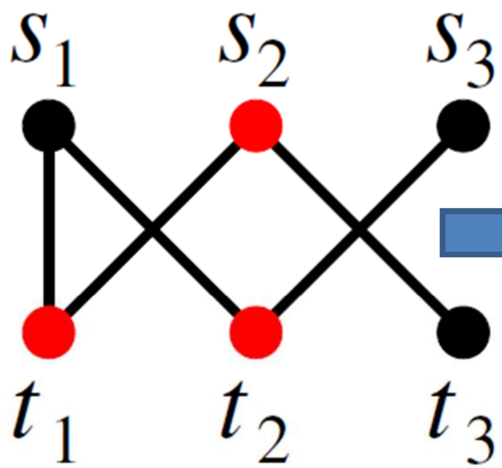
- **Input:** A routing table of a node with a set of triples (s,t,k)

- **Output:** an ordered compressed routing table with (s,t,k), (s,*,k), (*,t,k) and eventually the default (*,*,k) rules.

- **Objective:** Minimize the size of the routing table

Routing Table with 1 port is Polynomial

Chemin	Port
(s_1, t_1)	p_1
(s_1, t_2)	p_1
(s_2, t_1)	p_1
(s_2, t_3)	p_1
(s_3, t_2)	p_1



Bipartite Graph

Chemin	Port
$(*, t_1)$	p_1
$(s_2, *)$	p_1
$(*, t_2)$	p_1

Find a set S in the associated bipartite graph such that each edge has one of its endpoints in S

With more than 2 ports

- For $k >= 2$, the problem is NP-Complete
- Reduction to the Feedback Arc Set Problem

*A feedback arc set (FAS) is a set of edges which, when removed from the graph, leave a DAG.

Direction-Based Heuristic

- For each source `s`,
  - a rule `(s,*,k)` is put where k is the most frequent port for this source
  - The others rules are kept as `(s,t,k')` and put before
- Do the same for each destination `t`
- Choose the table of minimum size

Direction-Based Heuristic

- For each source `s`,
  - a rule `(s,*,k)` is put where `k` is the most frequent port for this source
  - The others rules are kept as `(s,t,k')` and put before

- Do the same for each destination `t`

- Choose the table of minimum size

This gives a **2-Approximation** if $(*,*,k)$ is not used for the List-Reduction problem:

Input: A set C of communication triples and an integer z

Output: $sav(C) \geq z$ where sav(C) is the number of saved triples

Direction-Based Heuristic

	8	9	10	11	12	13	14
1	1	1	1	2	1	0	1
2	1	1	1	1	1	1	1
3	1	0	1	0	1	0	1
4	1	1	2	1	0	1	0
5	1	0	3	0	2	1	1
6	0	1	2	1	0	1	0
7	1	1	1	0	1	1	1

Direction-Based Heuristic

	8	9	10	11	12	13	14
1	1	1	1	2	1	0	1
2	1	1	1	1	1	1	1
3	1	0	1	0	1	0	1
4	1	1	2	1	0	1	0
5	1	0	3	0	2	1	1
6	0	1	2	1	0	1	0
7	1	1	1	0	1	1	1

(1,11)→2
(1,13)→0
(1,*)→1

(Src, Dst)	Output
(1, 11)	**2**
(1, 13)	**0**
(1, *)	**1**

Direction-Based Heuristic

	8	9	10	11	12	13	14
1	1	1	1	2	1	0	1
2	1	1	1	1	1	1	1
3	1	0	1	0	1	0	1
4	1	1	2	1	0	1	0
5	1	0	3	0	2	1	1
6	0	1	2	1	0	1	0
7	1	1	1	0	1	1	1

(2,*)→1

(Src, Dst)	Output
(1, 11)	**2**
(1, 13)	**0**
(1, *)	**1**
(2, *)	**1**

Direction-Based Heuristic

	8	9	10	11	12	13	14
1	1	1	1	2	1	0	1
2	1	1	1	1	1	1	1
3	1	0	1	0	1	0	1
4	1	1	2	1	0	1	0
5	1	0	3	0	2	1	1
6	0	1	2	1	0	1	0
7	1	1	1	0	1	1	1

(3,9)→0
(3,11)→0
(3→13)→0
(3,*)→1

(Src, Dst)	Output
(1, 11)	**2**
(1, 13)	**0**
(1, *)	**1**
(2, *)	**1**
(3,9)	**0**
(3,11)	**0**
(3,13)	**0**
(3,*)	**1**

Direction-Based Heuristic

	8	9	10	11	12	13	14
1	1	1	1	2	1	0	1
2	1	1	1	1	1	1	1
3	1	0	1	0	1	0	1
4	1	1	2	1	0	1	0
5	1	0	3	0	2	1	1
6	0	1	2	1	0	1	0
7	1	1	1	0	1	1	1

The heuristic does the same for the destinations, and and chooses the smallest of the two tables

(6,8)→0
(*,8)→1

3-Approximation with (*,*) rule

- If (*,*) rule is used, a third table is considered, the minimum of the three tables is chosen

	8	9	10	11	12	13	14
1	1	1	1	2	1	0	1
2	1	1	1	1	1	1	1
3	1	0	1	0	1	0	1
4	1	1	2	1	0	1	0
5	1	0	3	0	2	1	1
6	0	1	2	1	0	1	0
7	1	1	1	0	1	1	1

Table :
Specific triples and the default rule
(*,*)→1

Example - checkerboard

	7	8	9	10	11	12
1	1	0	1	0	1	0
2	0	1	0	1	0	1
3	1	0	1	0	1	0
4	0	1	0	1	0	1
5	1	0	1	0	1	0
6	0	1	0	1	0	1

Source-based → entries
Destination-based → entries
With Global rule → entries
Optimal (with default rule) → entries

Example - checkerboard

	7	8	9	10	11	12
1	1	0	1	0	1	0
2	0	1	0	1	0	1
3	1	0	1	0	1	0
4	0	1	0	1	0	1
5	1	0	1	0	1	0
6	0	1	0	1	0	1

Source-based → 24 entries
Destination-based → 24 entries
With Global rule → 19 entries
Optimal (with default rule) → 16 entries

Example – checkerboard - OPT

	7	8	9	10	11	12
1	1	0	1	0	1	0
2	0	1	0	1	0	1
3	1	0	1	0	1	0
4	0	1	0	1	0	1
5	1	0	1	0	1	0
6	0	1	0	1	0	1

OPT = 3*4+

4 entries : 3 triples + (*,7)→1

Example – checkerboard - OPT

	7	8	9	10	11	12
1	1	0	1	0	1	0
2	0	1	0	1	0	1
3	1	0	1	0	1	0
4	0	1	0	1	0	1
5	1	0	1	0	1	0
6	0	1	0	1	0	1

(1,*)→0

(3,*)→0

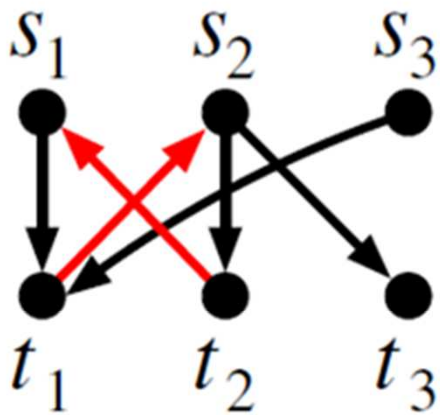
(5,*)→0

And Default =
(*,*)→1

OPT = 12+3+1 =16

With two ports

Chemin	Port
(s_1, t_1)	p_1
(s_1, t_2)	p_2
(s_2, t_1)	p_2
(s_2, t_2)	p_1
(s_2, t_3)	p_1
(s_3, t_1)	p_1



Associate Bipartite graph

Chemin	Port
(s_1, t_2)	p_2
(s_2, t_1)	p_2
$(*, t_1)$	p_1
$(s_2, *)$	p_1

There exists a 4-approximation for Feedback Arc Set for Bipartite Tournament [Van Zuylen, 2009]

ILP for acyclic

Minimize

$$\sum_{\forall s,t,k} r(s,t,k) + g(*,t,k) + g(s,*,k)$$

Subject to:

$$\textstyle\sum_{\forall k} g(s,*,k) \leq 1$$

$$\textstyle\sum_{\forall k} g(*,t,k) \leq 1$$

$\forall (s,t,k),$

$$r(s,t,k) + g(*,t,k) + g(s,*,k) \geq 1$$

ILP with order

Minimize

$$\sum_{\forall s,t,k} r(s,t,k) + g(*,t,k) + g(s,*,k)$$

Subject to:

$$\textstyle\sum_{\forall k} g(s,*,k) \leq 1\,;\, \sum_{\forall k} g(*,t,k) \leq 1$$

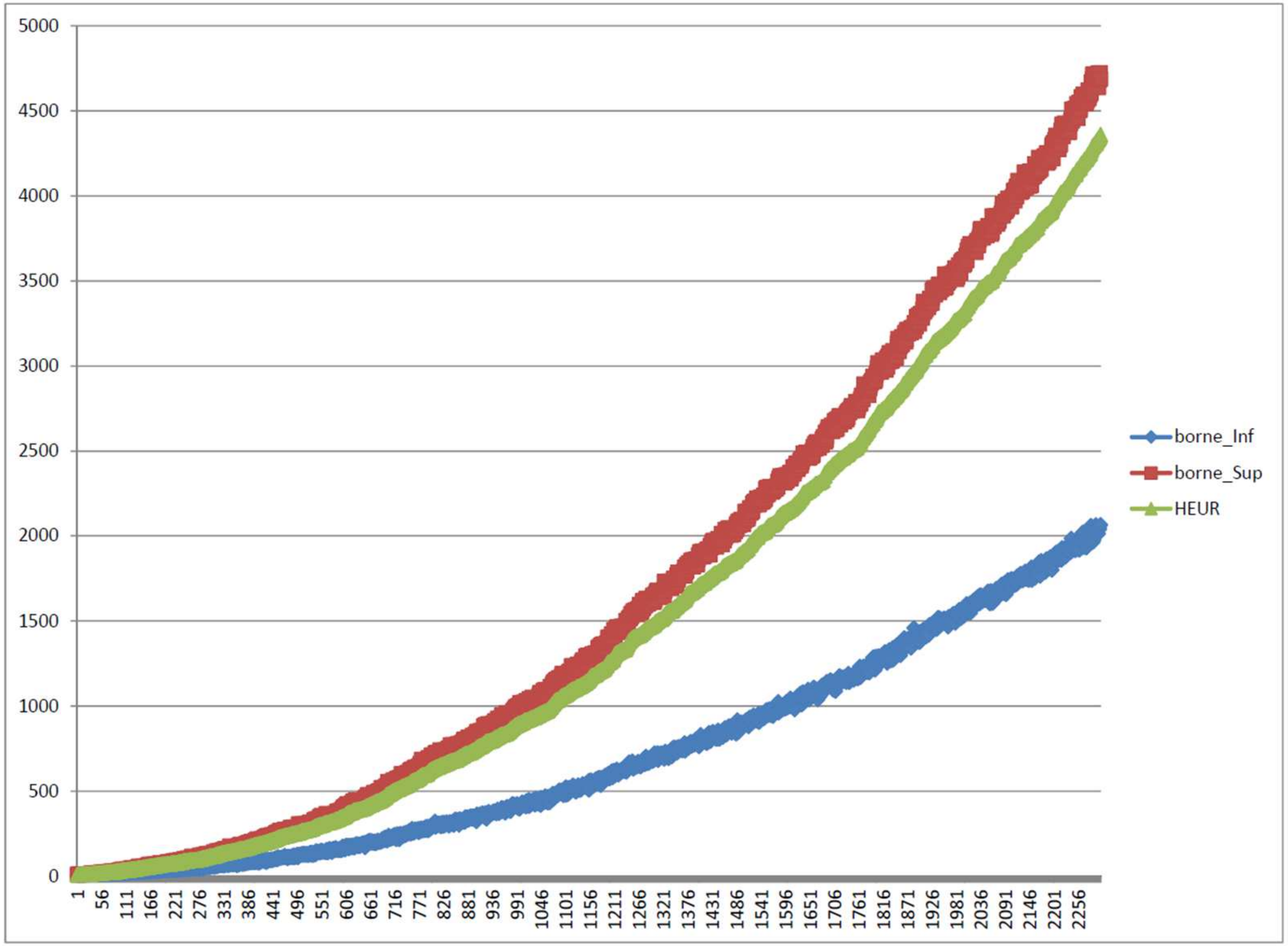
$$\forall (s,t,k), \qquad r(s,t,k) + g(*,t,k) + g(s,*,k) \geq 1$$

$$r(s,t,k) + \mathrm{order}(g(s,*,k), g(*,t,k')) \geq g(*,t,k')$$
$$r(s,t,k) + g(s,*,k) \ \geq g(*,t,k')$$

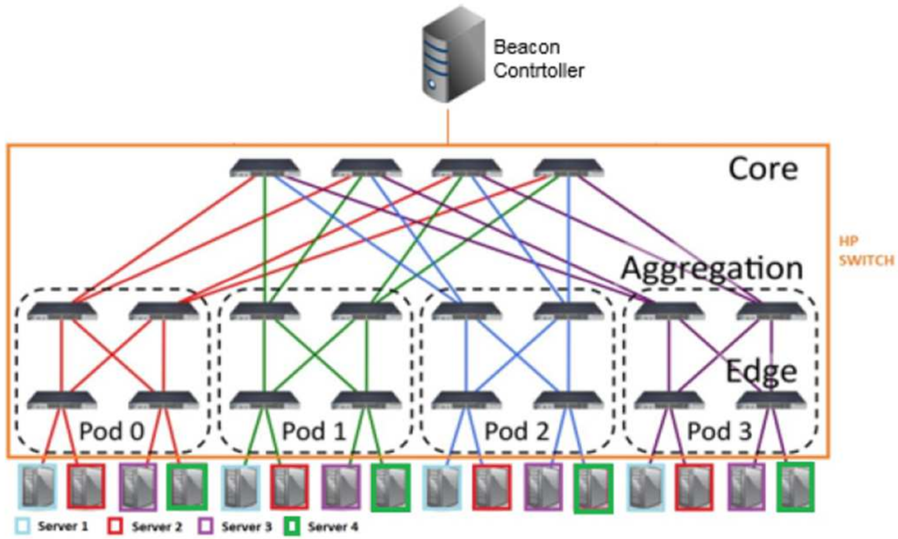
$$r(s,t,k) + \mathrm{order}(g(*,t,k), g(s,*,k')) \ \geq g(s,*,k')$$
$$r(s,t,k) + g(*,t,k) \ \geq g(s,*,k')$$

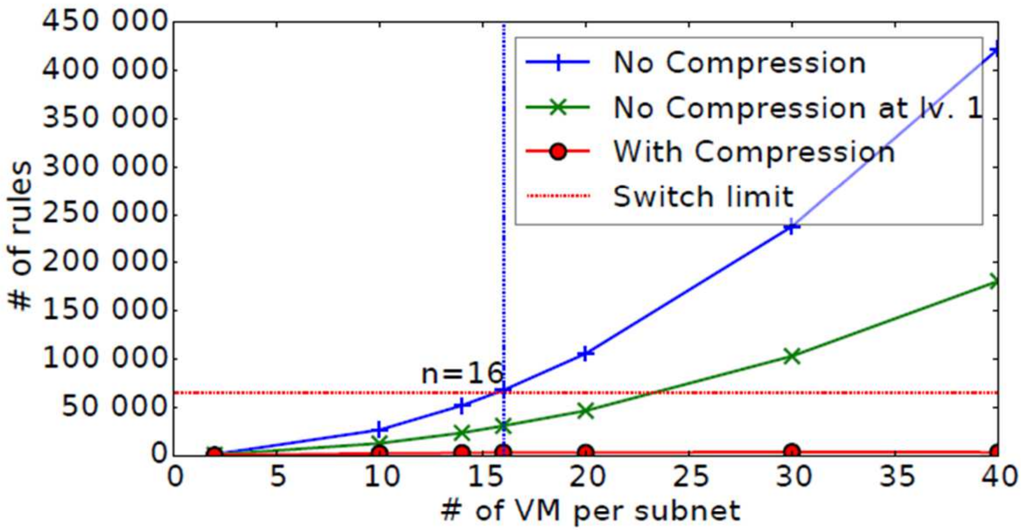
$$1 \leq order(g1,g2) + order(g2,g3) + order(g3,g1) \leq 2$$

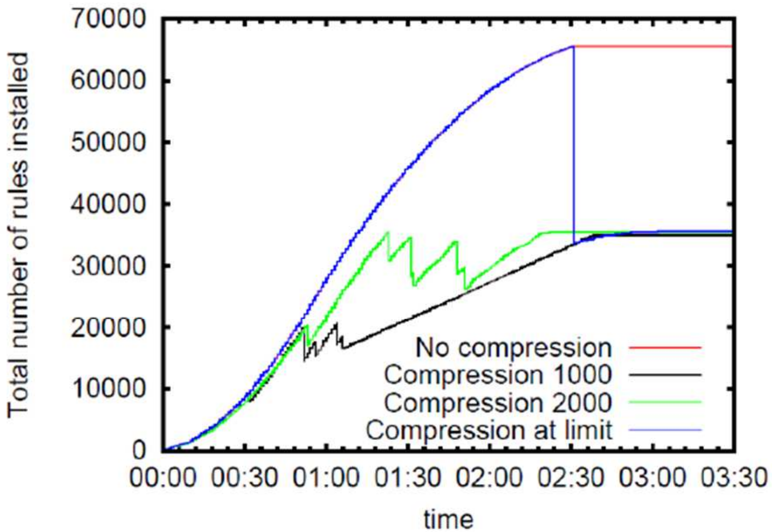
Number of rules – Upper and Lower Bounds



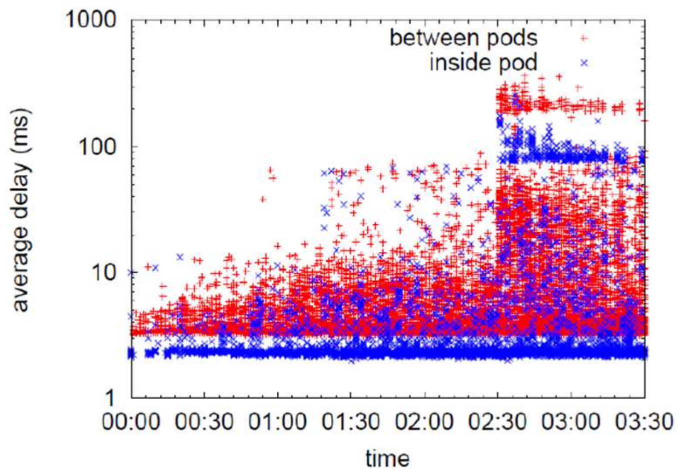
Experimentation on a fat-tree







Total number of rules installed on the physical switch



Average delay: no compression, 8 clients per VM

Conclusion

- Fixed-Parameter Tractability
- Lower and Upper Bounds for the number of rules
- Best Approximation Ratio for List-Reduction?
- Routing-List Problem is APX-Complete?

Find more: *https://hal.inria.fr/hal-01097910*