Cracking Network Monitoring in DCNs with SDN

Zhiming Hu

Jun Luo

Nanyang Technological University
Singapore

by
Jean-Philippe Gauthier

Paper presentation
Fall 2015

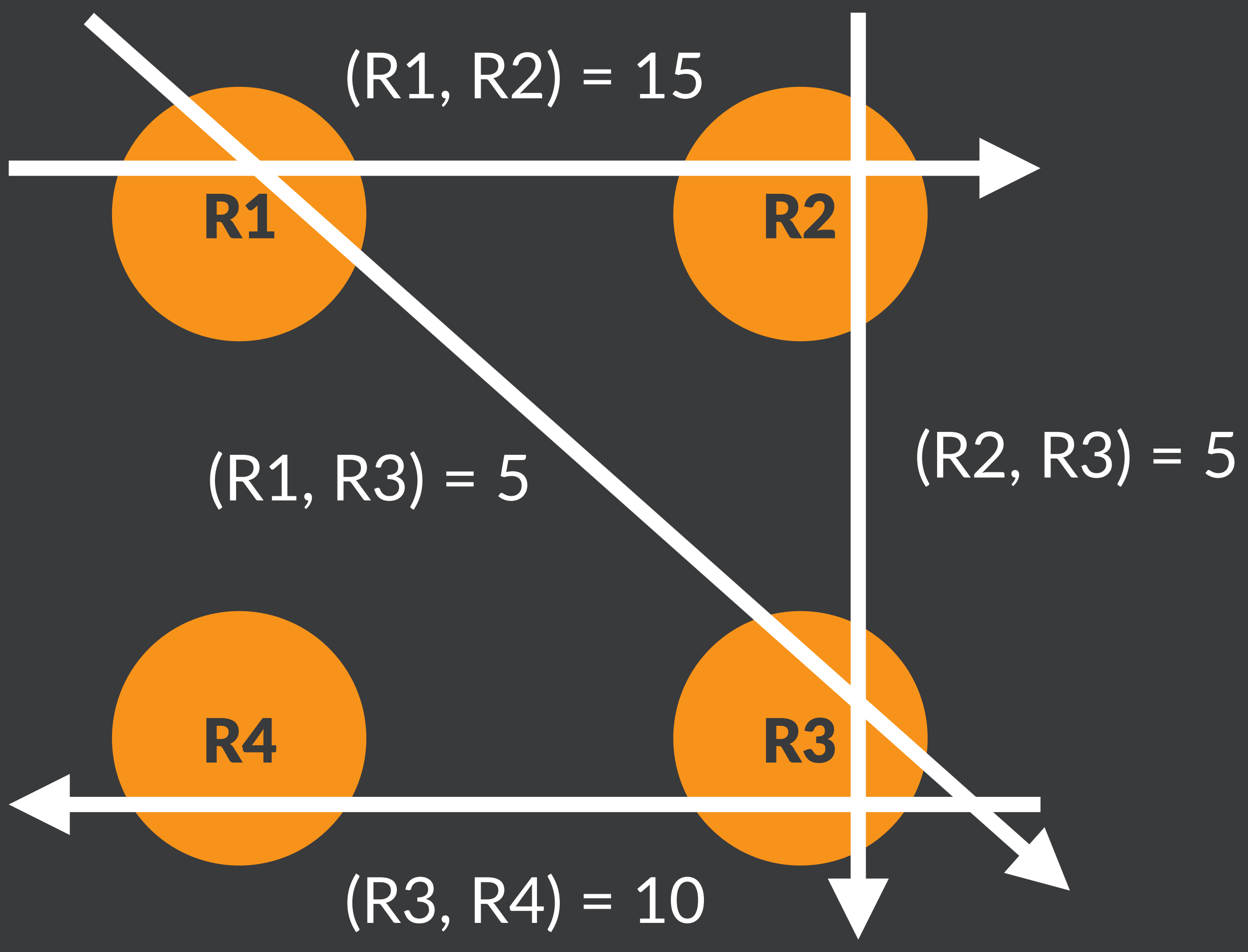
Cheriton School of Computer Science
Faculty of Mathematics

Network Monitoring

- A systematic effort to understand the exchanges in the network
- Traditionally used to monitor the health/reliability of a network
- The results of network monitoring is vital for
  - Operators
  - Researchers
- Two important tasks
  - Traffic Matrix estimation
  - Elephant Flow detection

Traffic Matrix (TM)

- Defines the volume of traffic for each corresponding pair of nodes



From/To	R1	R2	R3	R4
R1	0	15	5	0
R2	0	0	5	0
R3	0	0	0	10
R4	0	0	0	0

Of Mice and Elephants

- Around 2001, researchers started to use the term elephant flow
  - Observations were made that a small number of flows carry the majority of Internet traffic (elephant flows) and the remainder consists of a large number of flows that carry very little Internet traffic (mouse flow)
- An elephant flow is an extremely large continuous flow
- A mouse flow is everything else

How is this useful?

- Improve your network
  - Traffic Engineering
  - Increase link bandwidth
  - Load Balancing
  - Optimize IGP metrics
  - Change BGP exit points
  - Failover strategies
  - …

Outline

- Prior Work
- Problems formulation
- Proposed system
- Network Tomography in SDNs
- Evaluation
- Conclusion

Prior work in SDNs

- Direct measurements
    - Switch-based approaches
    - Server-based approaches
- Direct measurements are extremely expensive
    - Extra storage
    - Extra processing
    - Extra bandwidth

Prior work in ISPs (Network Tomography)

- Tomography refers to imaging by sections or sectioning, through the use of any kind of penetrating wave (radiology)
- Network Tomography is the inference of the internal behavior and topology of a network based on end-to-end network measurements and/or SNMP link counts
- Due to the rich connections between switches in DCNs, end-to-end paths (variables) are far more than the number of links (available measurements)
- Makes the problem under-constrained
- Impossible to use state-of-the-art algorithms

The proposed solution

- Network tomography alone can hardly offer accurate TM and elephant flow detection
- Direct measurements are too expensive
- Combine direct measurements AND network tomography
- Reach a balance between measurements overhead and accuracy
- Use SDN rules to provide more measurements when needed

Problems Formulation

- Let x(t) = { $x_1(t)$ … $x_n(t)$ } denotes the volume of traffic between *n* paths at time slot *t*
- Let *traffic matrix* X = { x(1) … x(τ) } represents the traffic in DCN from slot 1 to τ
- Let y(t) = { $y_1(t)$ … $y_m(t)$ } denotes the link counts over *m* links pulled from the link counters in the switch through SNMP at time slot t
- Let *link count matrix* Y = { y(1) … y(τ) } represents the link counts in DCN from slot 1 to τ
- Let routing matrix A = { a_{ij}|i = 1…m, j = 1 … n } indicates whether a path traverses a link in the topology

Problems Formulation

- The relationship between X, Y and A is

$$AX = Y$$

- However, the problem is heavily under-constrained
- Let aggregation matrix B indicates if the traffic through a certain path is measured by SDN rules
- Let SDN counters matrix Q denotes the counters defined with rules
- The relationship between X, Y, A, B, D and Q

$$\begin{pmatrix} A \\ B \end{pmatrix} X = DX = \begin{pmatrix} Y \\ Q \end{pmatrix}$$

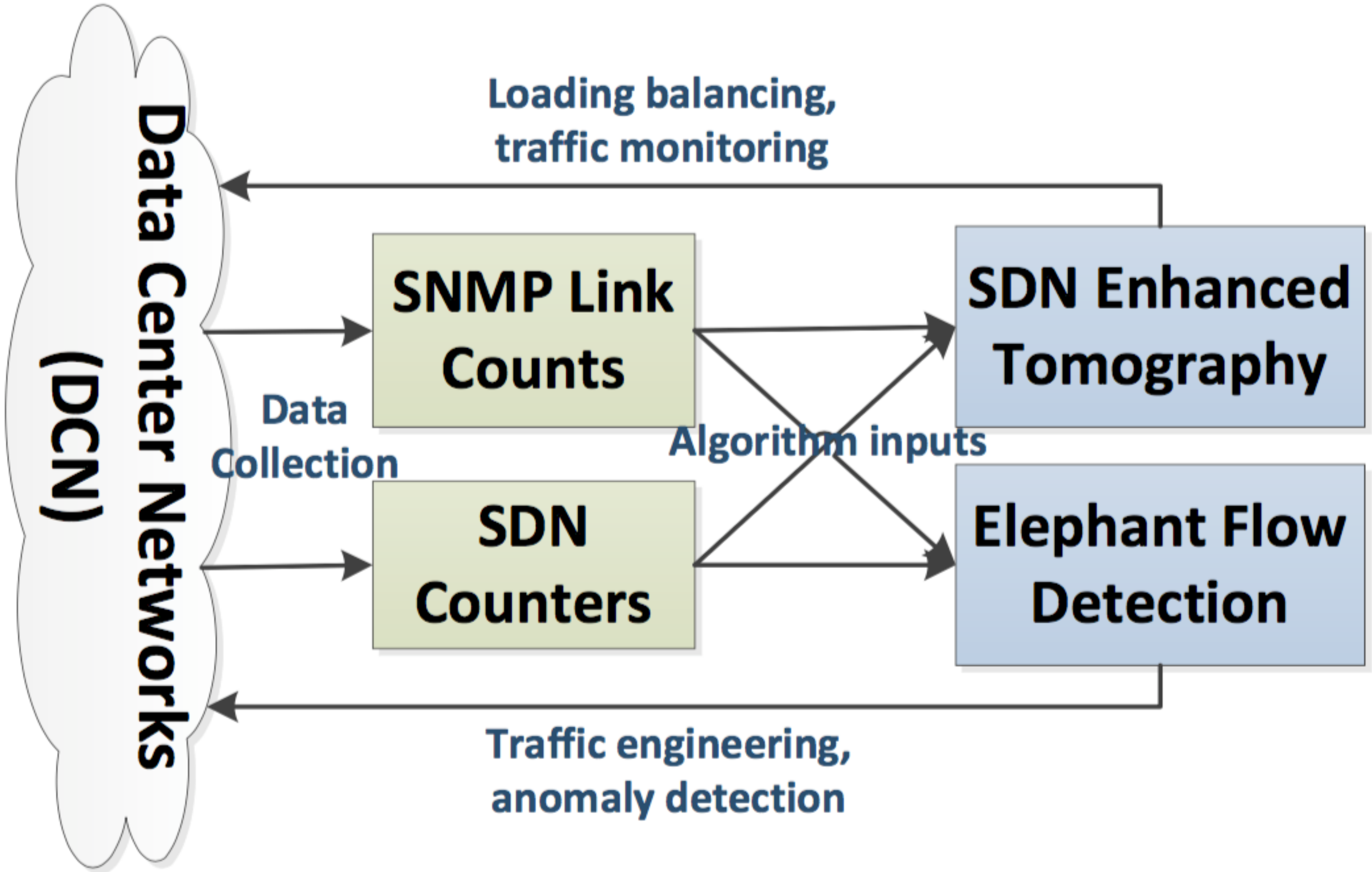
Problems Formulation

- The relationship between X, Y, A, B, D and Q

$$\begin{pmatrix} A \\ B \end{pmatrix} X = DX = \begin{pmatrix} Y \\ Q \end{pmatrix}$$

- The goal is to maximize the rank of the measurement matrix D to increase identifiability and estimation accuracy of X (TM)
- Problem 1: How to determine matrix B to maximize rank of D
- Problem 2: How to identify elephant flows with all this information

Proposed System



Network Tomography in SDNs

- Maximizing rank matrix completion is a complicated optimization problem
- One proposed solution is to adapt the norm minimization approach in compressive sensing
  - Results in a non-linear program
  - Transfer the problem into an integer linear program (ILP) by introducing new variables and constraints
  - Solve with CPLEX
  - Complexity of $O((m+r)^2 n^2)$

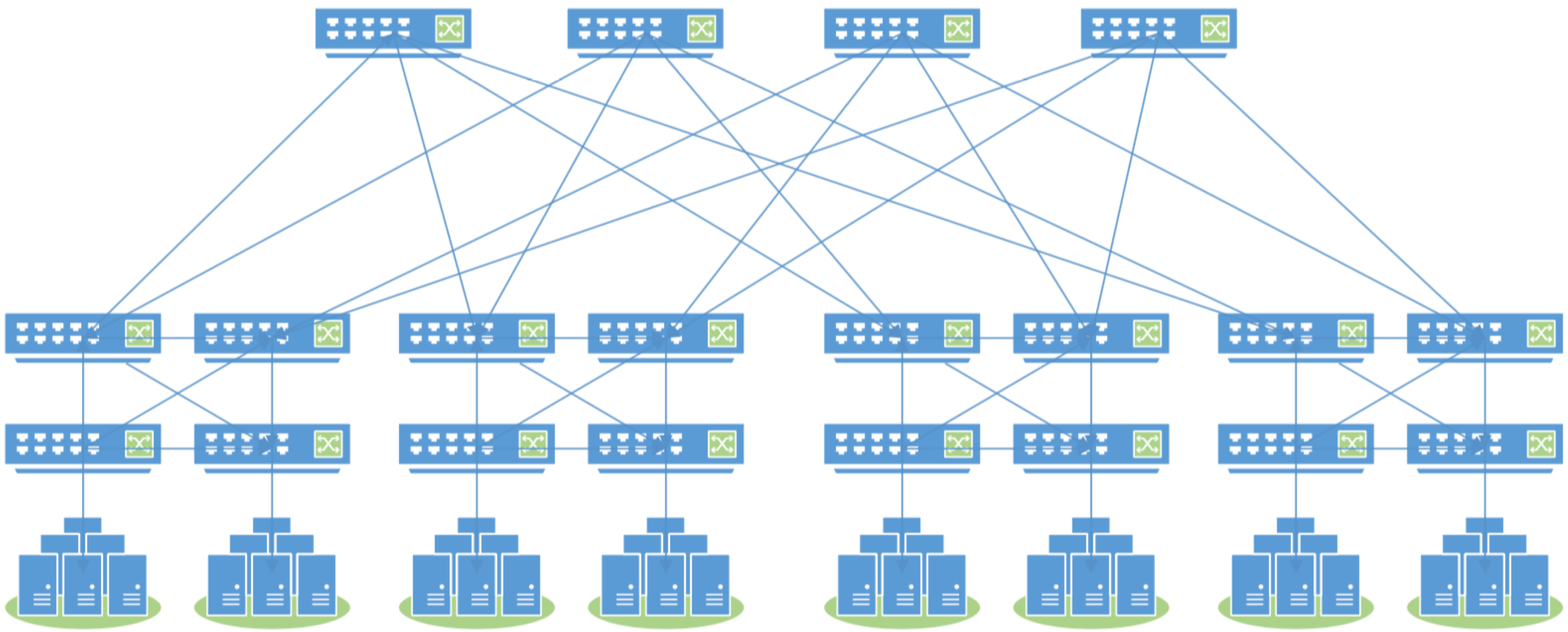
Network Tomography in SDNs

- Another proposed solution is to use the *iterative rank maximization algorithm*
  - An iterative method to construct the matrix D with a guarantee of maximum rank based on the fast rank computation algorithm
  - Gradually picks up a row that satisfies the constraints, add it to D and test if rank is higher
  - Complexity of $O(mn(\log \min\{m, n\})^2)$ each time
- The solution found by the algorithms may be impossible (or highly expensive) to aggregate with current standard of SDN

Network Tomography in SDNs

- Topology aware heuristic

- Let's assume a 4-ary fat-tree topology (k=4)

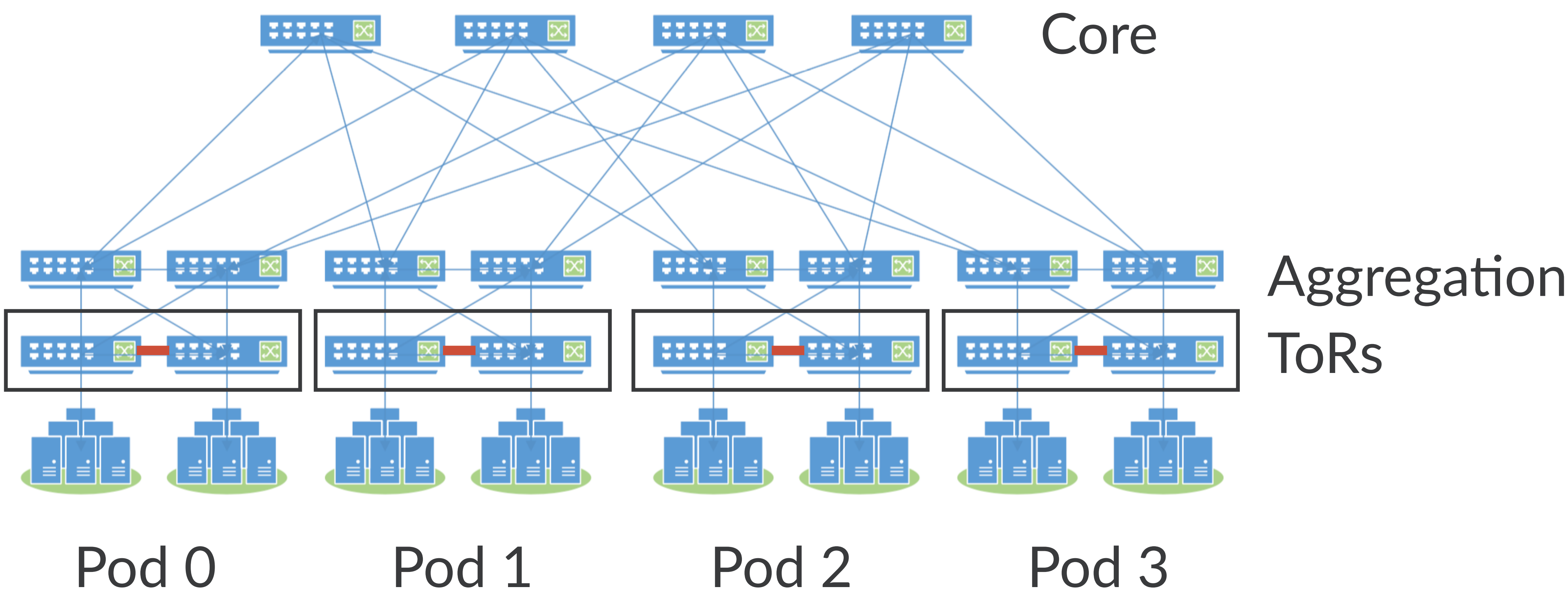


Proposition 1

- "If we set up the SDN rules for the ToR pairs within every pod in fat-tree architecture first and then the ToR pairs across different pods both in an lexicographical order, then the SDN rules for the following ToR pairs would not increase the rank of the network tomography problem"

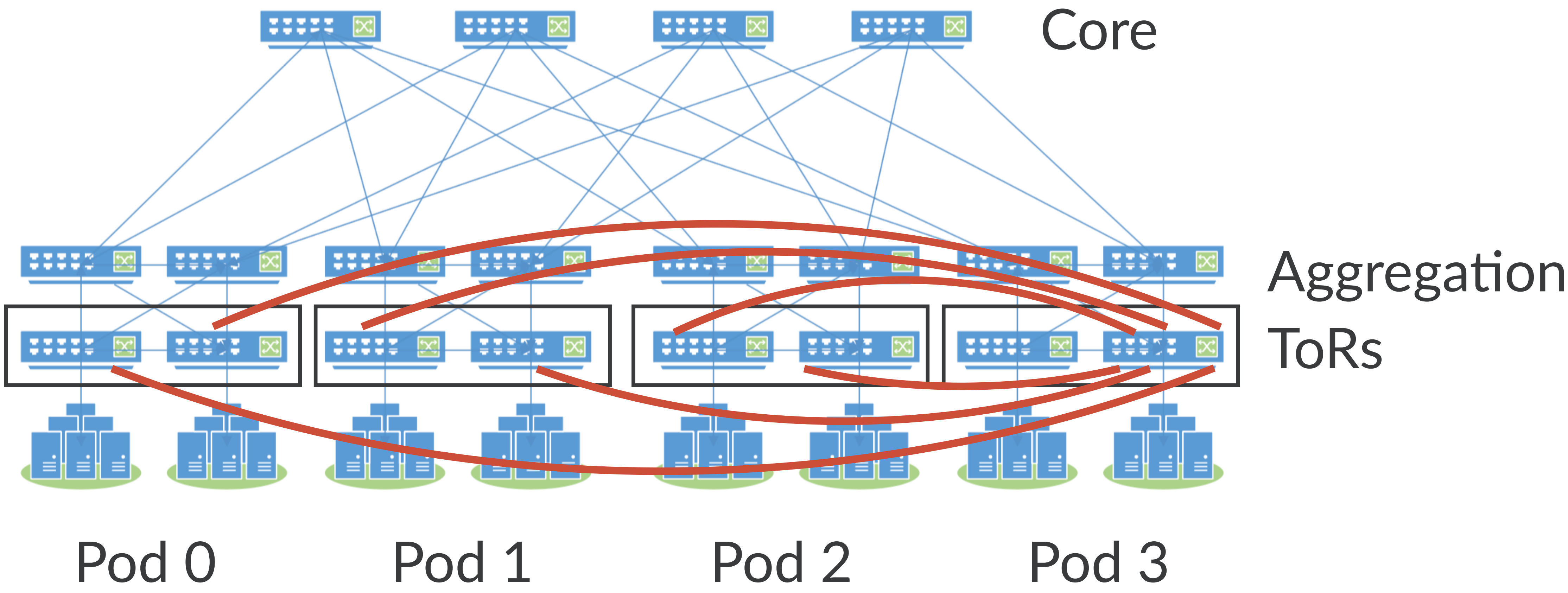
Case 1

❖ "For the ToR pairs within each pod in lexicographical order, the last pair in each pod"



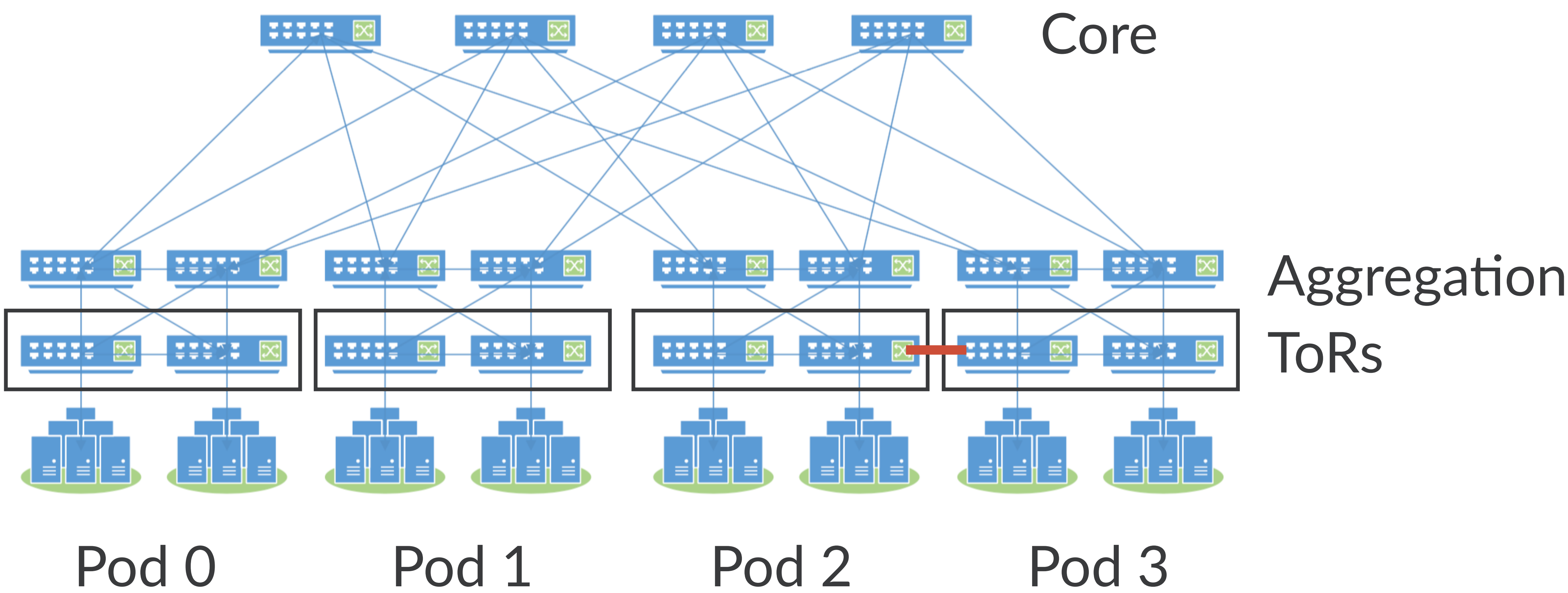
Case 2

- "The pairs between all the ToRs (except the last ToR) and the last ToR"



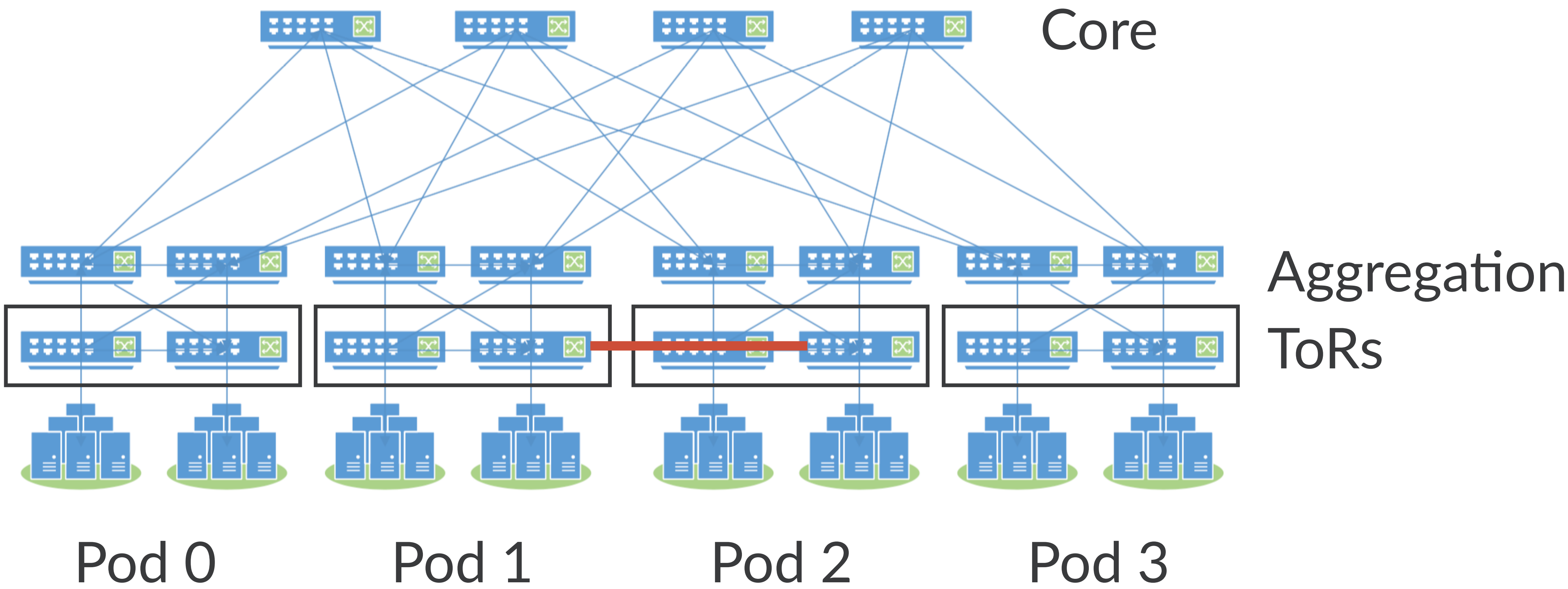
Case 3

❖ "The pairs between the last ToR in the penultimate pod and the ToRs in the last pod (except the last ToR, which is included in the last case)"



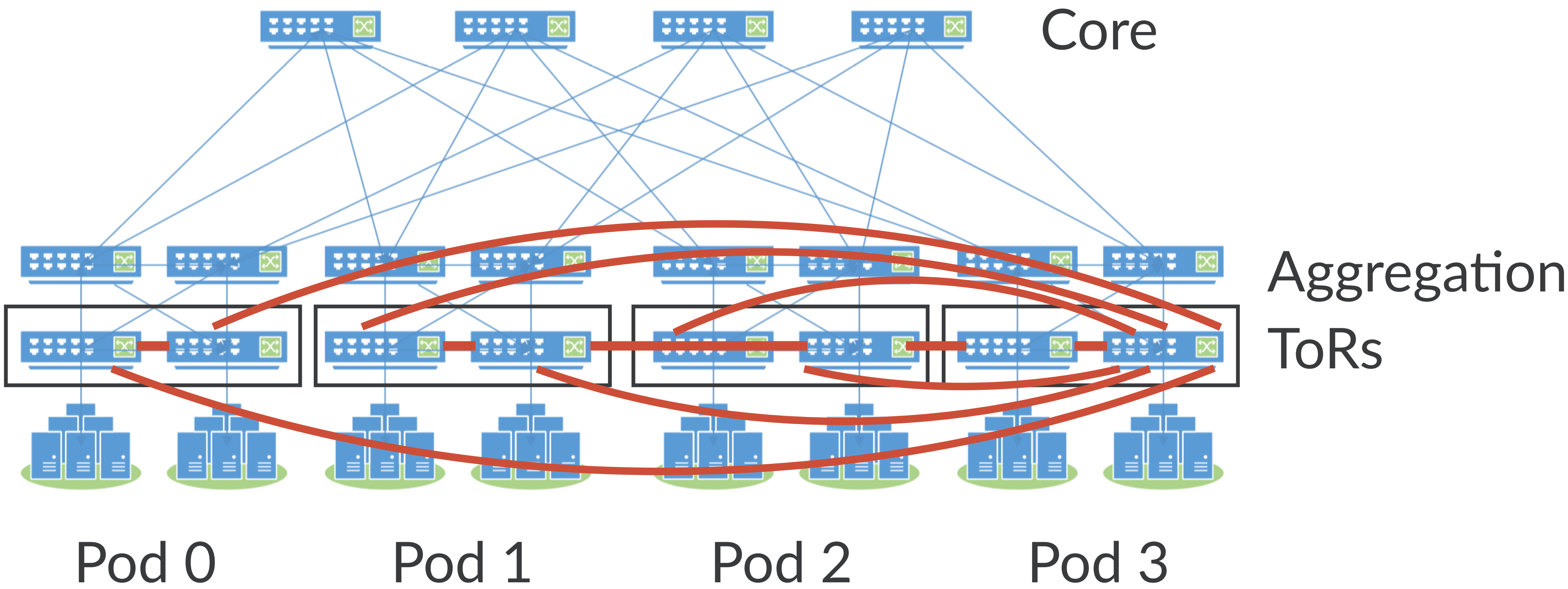
Case 4

- "The pair between the last ToR in the last third pod and the last ToR in the penultimate pod"



All the cases

- The 12 rules (in a 4-ary fat-tree architecture) will not increase the rank of the network tomography problem



Proposition 2

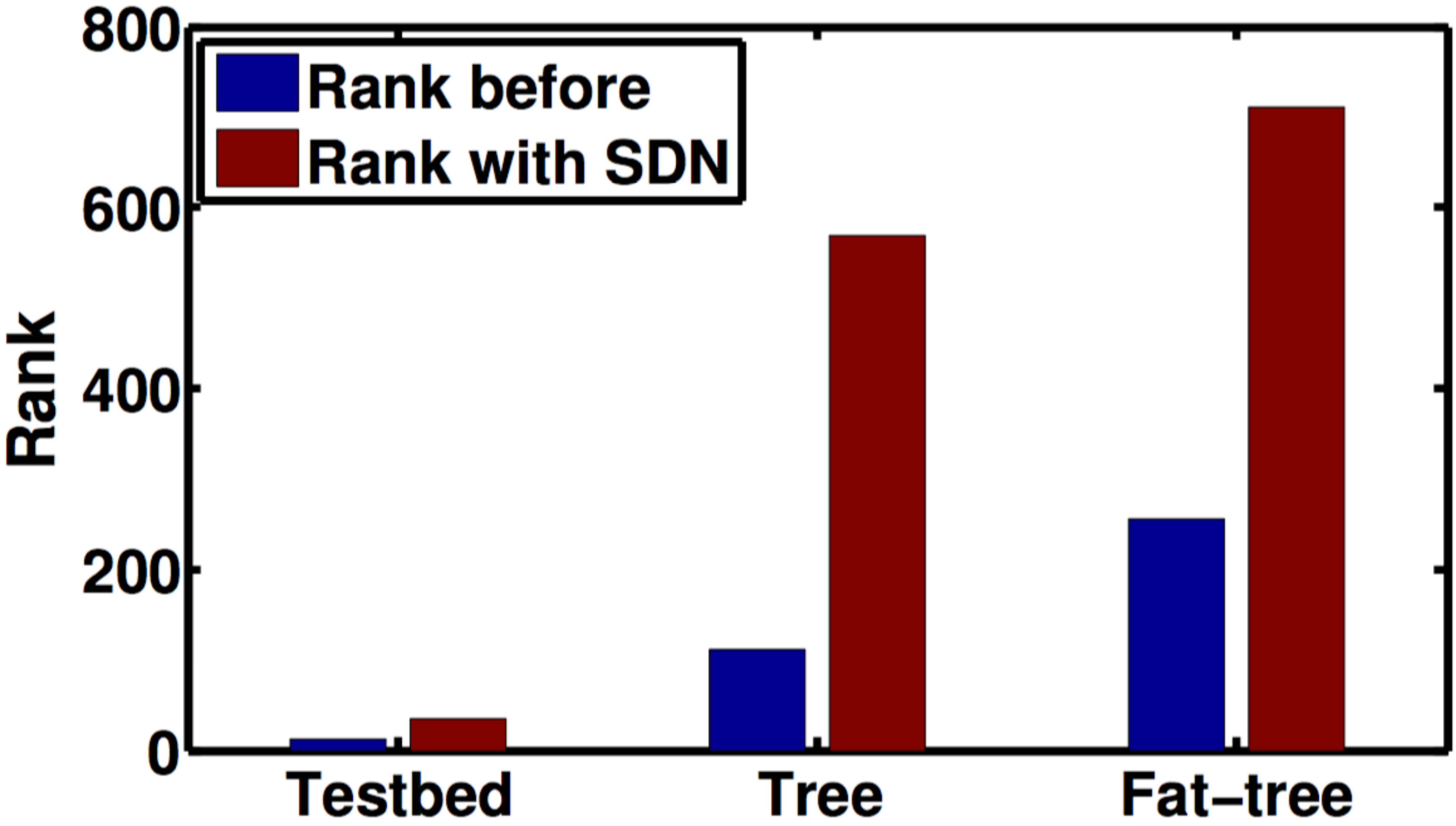
- In a fat-tree architecture, the total number of ToRs is $k^2/2$
- The number of all the possible ToRs pairs is $k^2/2 * (k^2/2 - 1) * 1/2$
- Based on proposition 1, the number of ToR pairs that can't increase the rank of D is $k^2/2 + k$
- Therefore, for a k-ary fat-tree topology, the number of SDN rules that can increase the rank is

$$k^4/8 - 3 * k^2/4 - k$$

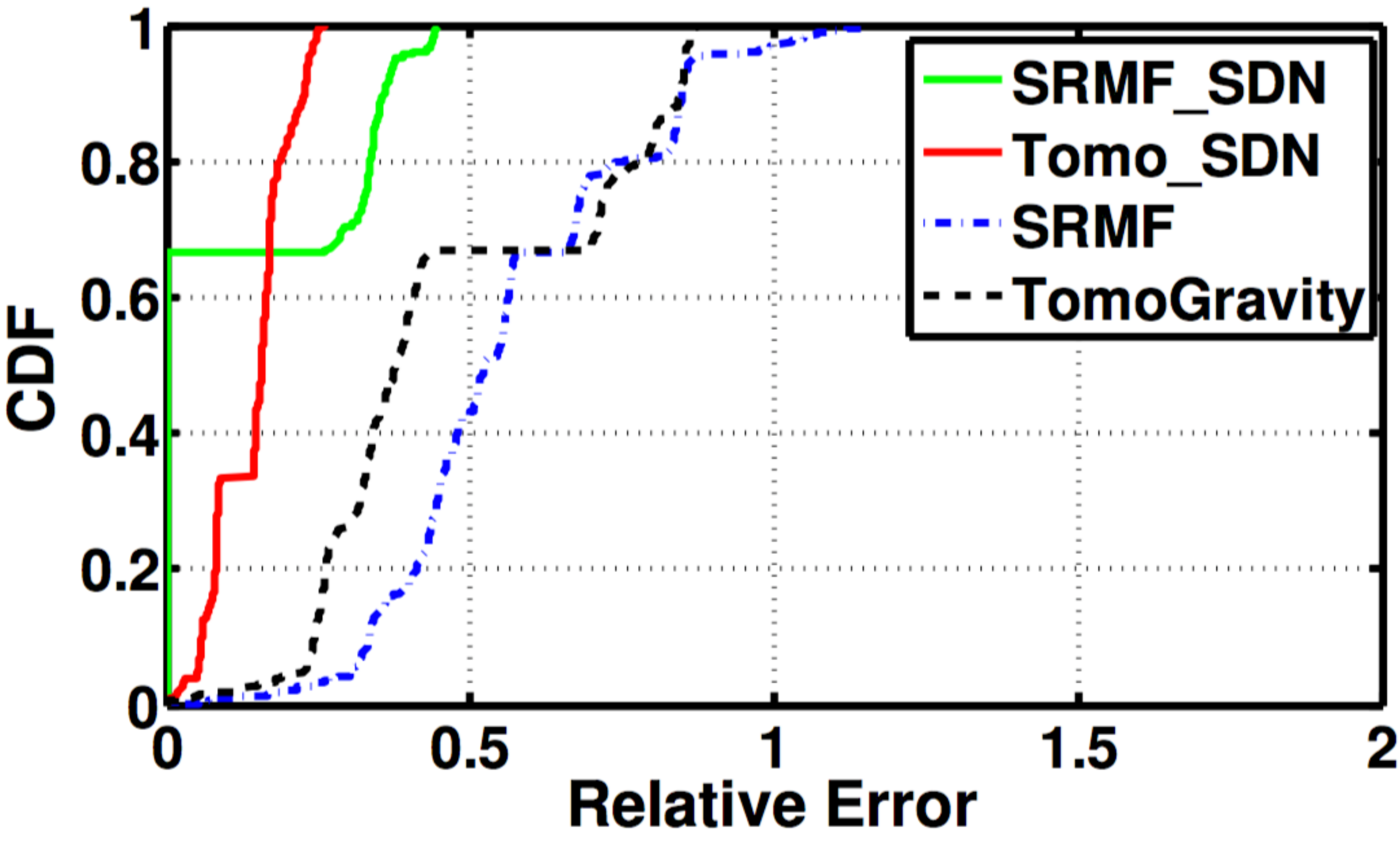
Elephant flow detection

- The proposed solution is to monitor only the flows with an high probability of being an elephant flow
- Since we know the total volume of traffic from tomography, we can proportionally divide the total number of SDN rules based on the volume of traffic from one pair to another
- Then, assign SDN rules in a greedy manner
  - Assign rules to server pairs with higher weight till there's no more rules left

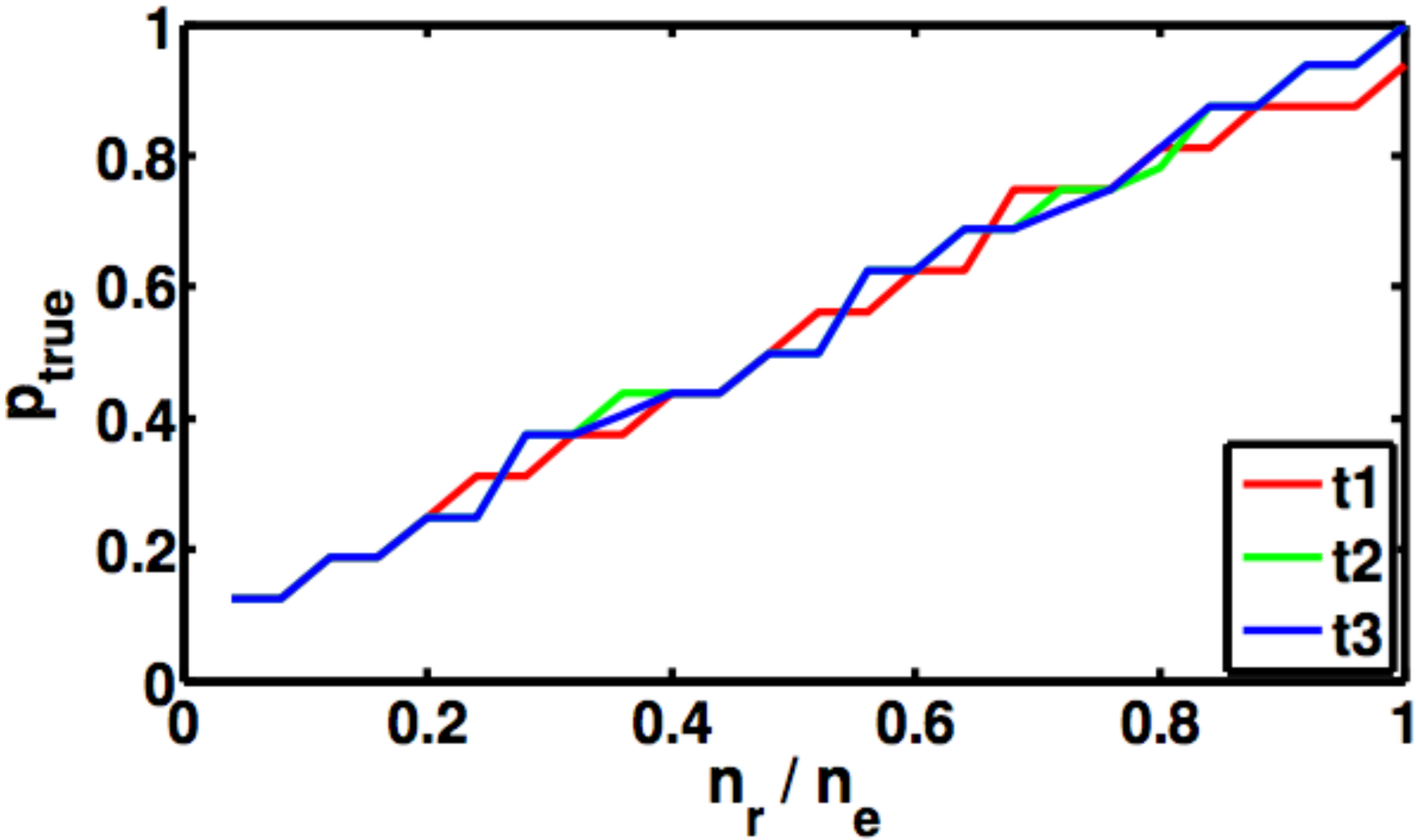
Evaluation (Rank Maximization)



Evaluation (TM Estimation in Testbed)



Evaluation (Elephant flow detection in simulation)



Conclusion

- This paper proposes a monitoring system using
  - Direct measurements with SDN rules
  - Network tomography
- to estimate the traffic matrix (TM) and detect elephant flows
- Systematic and heuristic algorithms have been used to maximize the identifiability of network tomography problems
- The evaluation of the proposed approach shows promising results

QUESTIONS?