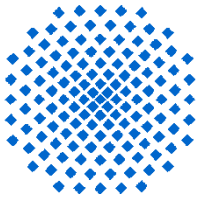


Universität Stuttgart

Institute of Parallel and
Distributed Systems (IPVS)

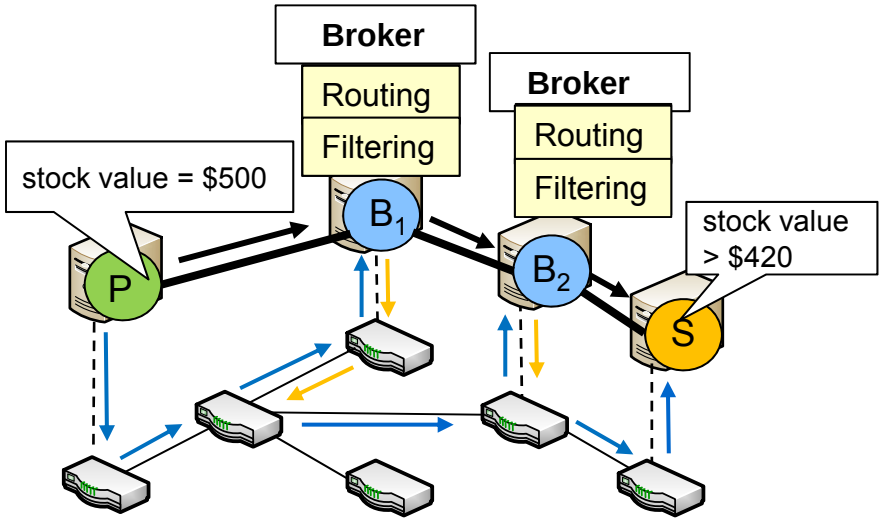
Universitätsstraße 38
D-70569 Stuttgart

Addressing TCAM Limitations of Software-Defined Networks for Content-Based Routing

Sukanya Bhowmik, M. Adnan Tariq, Alexander Balogh,
Kurt Rothermel

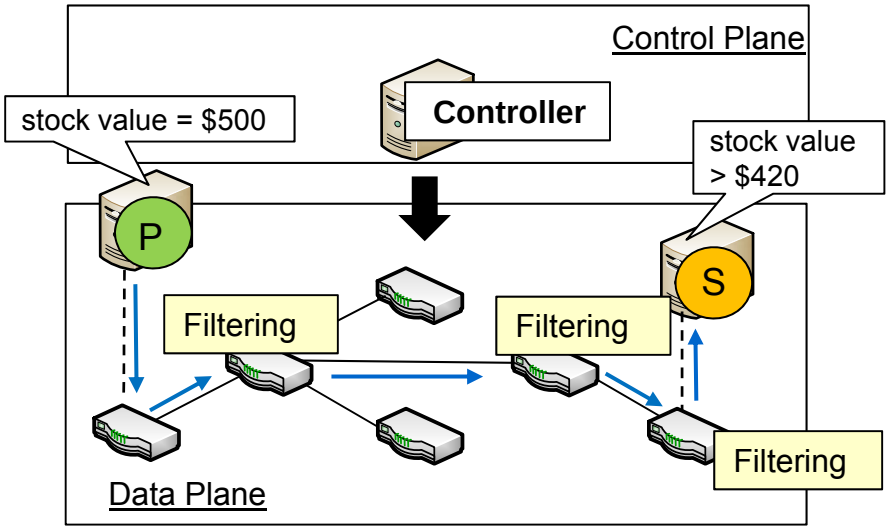
University of Stuttgart

High Performance Publish/Subscribe Middleware



Publish/subscribe middleware so far ...

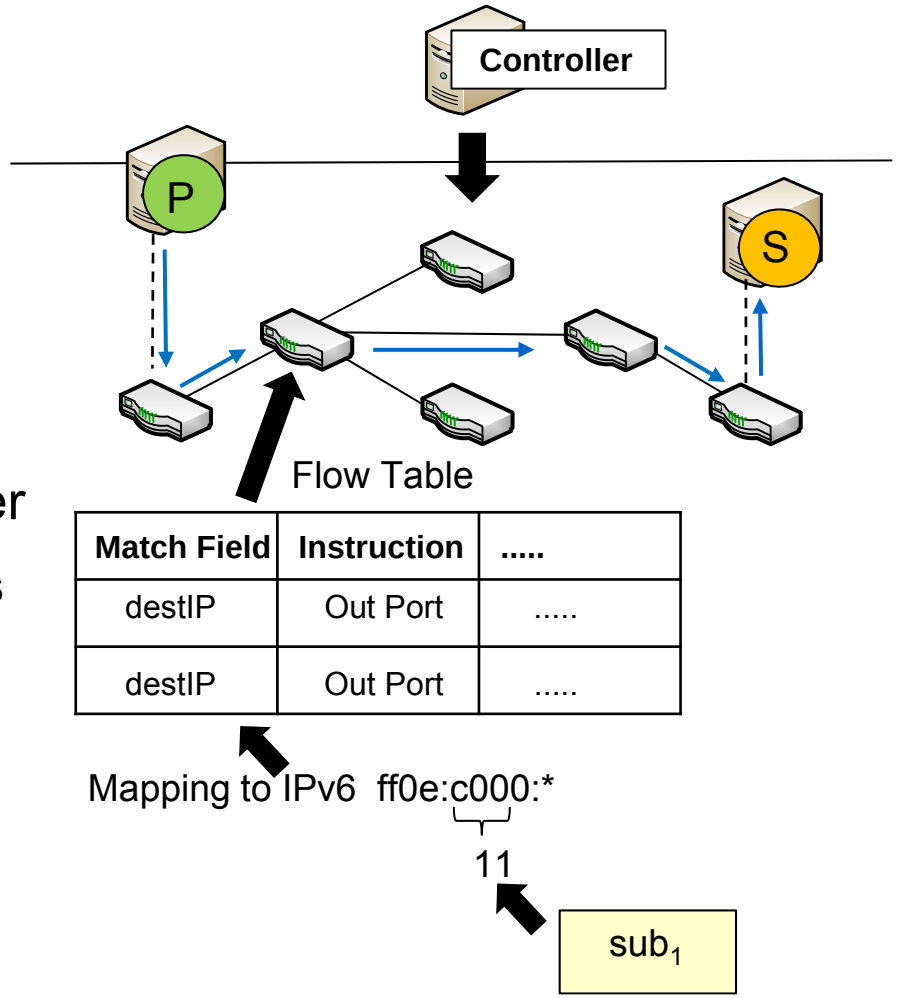
- Overlay network of brokers
- Routing and filtering in software
  - ✓ Expressive and accurate filtering of events in software
  - ✗ Reduced throughput, increased latency



SDN-based publish/subscribe...

Publish/Subscribe Middleware on SDN

- Filters pushed to network layer
  - ✓ Line-rate forwarding of events
  - ✗ Hardware limitations
    - TCAM constraint



TCAM Limitations

- TCAM is expensive and power-hungry
  - 100 times greater cost than RAM
  - 100 times greater power consumption than RAM

Cacheflow, SOSR '16

Vendors support limited no. of flow table entries in TCAM

(Typically a few thousands)

TCAM Limitations in Publish/Subscribe

- Systems may have up to millions of subscribers (content filters)
- Switches may be shared among applications
  - Fraction of flows available for pub/sub traffic

- Two possibilities
  - ➢ Drop filters/flows
    - False negatives
  - ➢ Aggregate filters/flows
    - False positives

Contributions

Expressive filtering of events despite aggregation of filters in the presence of TCAM constraint on switches

- Propose a filter aggregation algorithm that targets bandwidth efficiency in the system
- Propose methods to handle dynamics (changing subscriptions and event distribution) in the system
- Thoroughly evaluate the proposed algorithms

Filter Aggregation Problem

Given a set of switches with exceeded TCAM capacity ($\mathbb{ER}$)

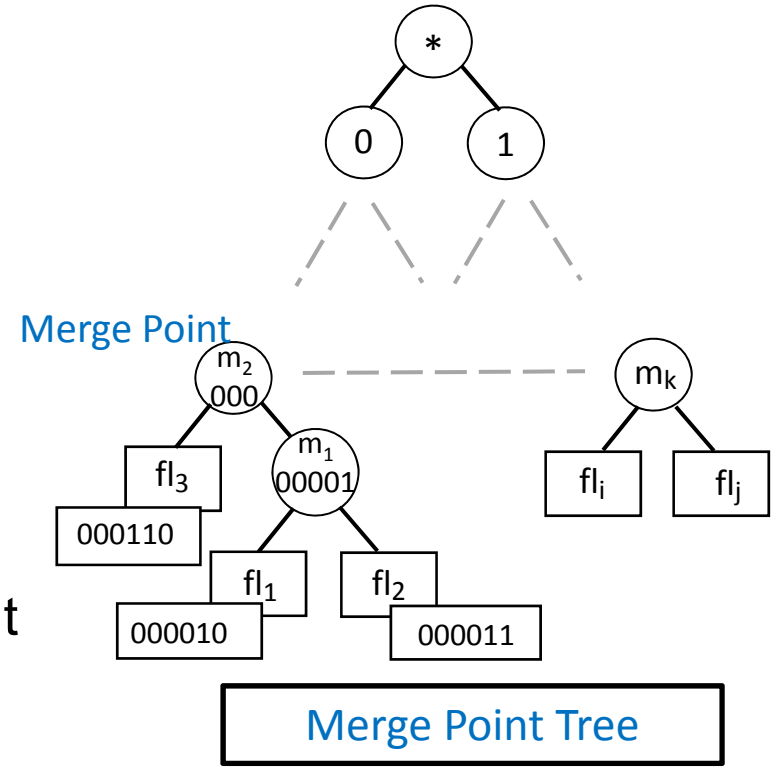
For each switch $\in \mathbb{ER}$

- Select a set of aggregated filters that
  - Limits no. of filters to the TCAM capacity
  - Keeps overall network false positives, introduced due to aggregation, to a minimum

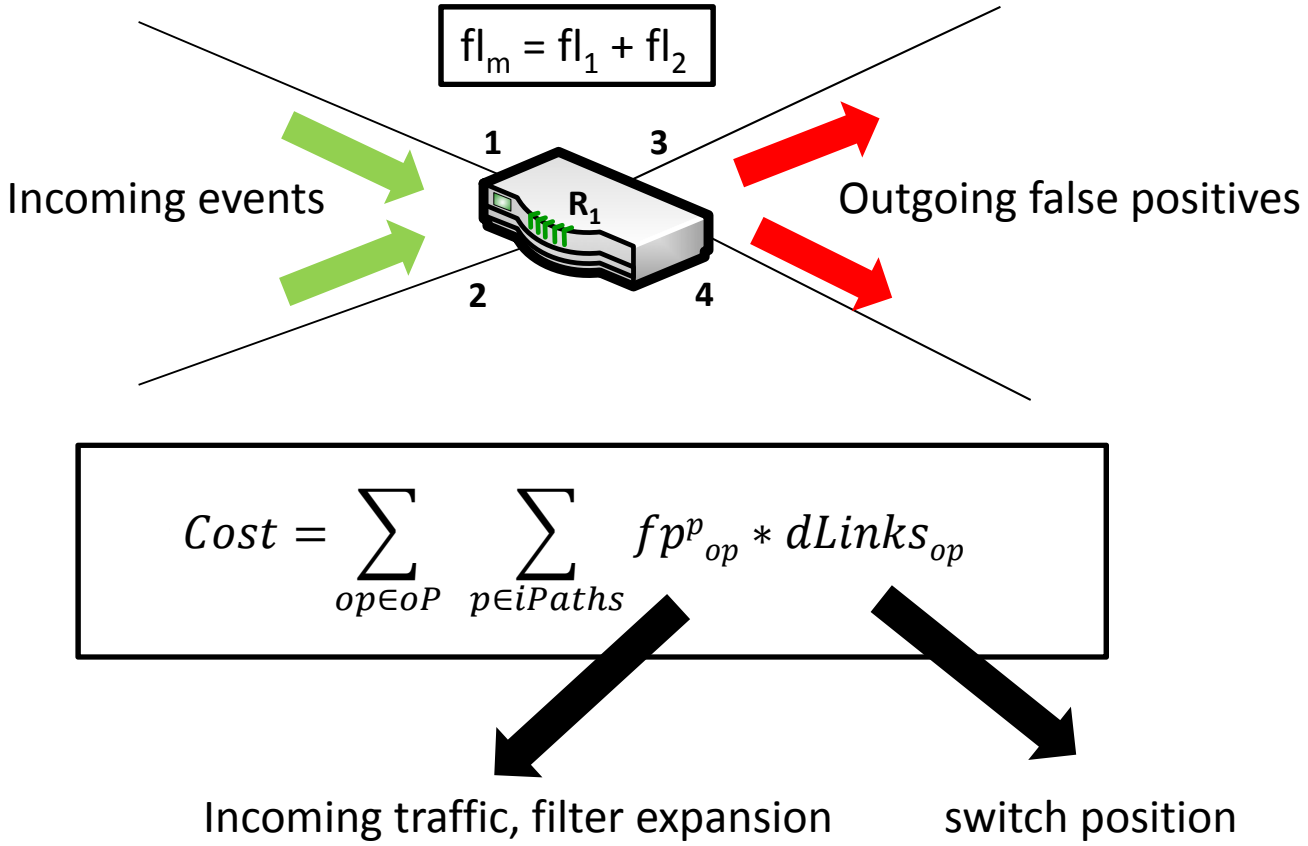
    (minimum aggregation cost)

Filter Aggregation Algorithm

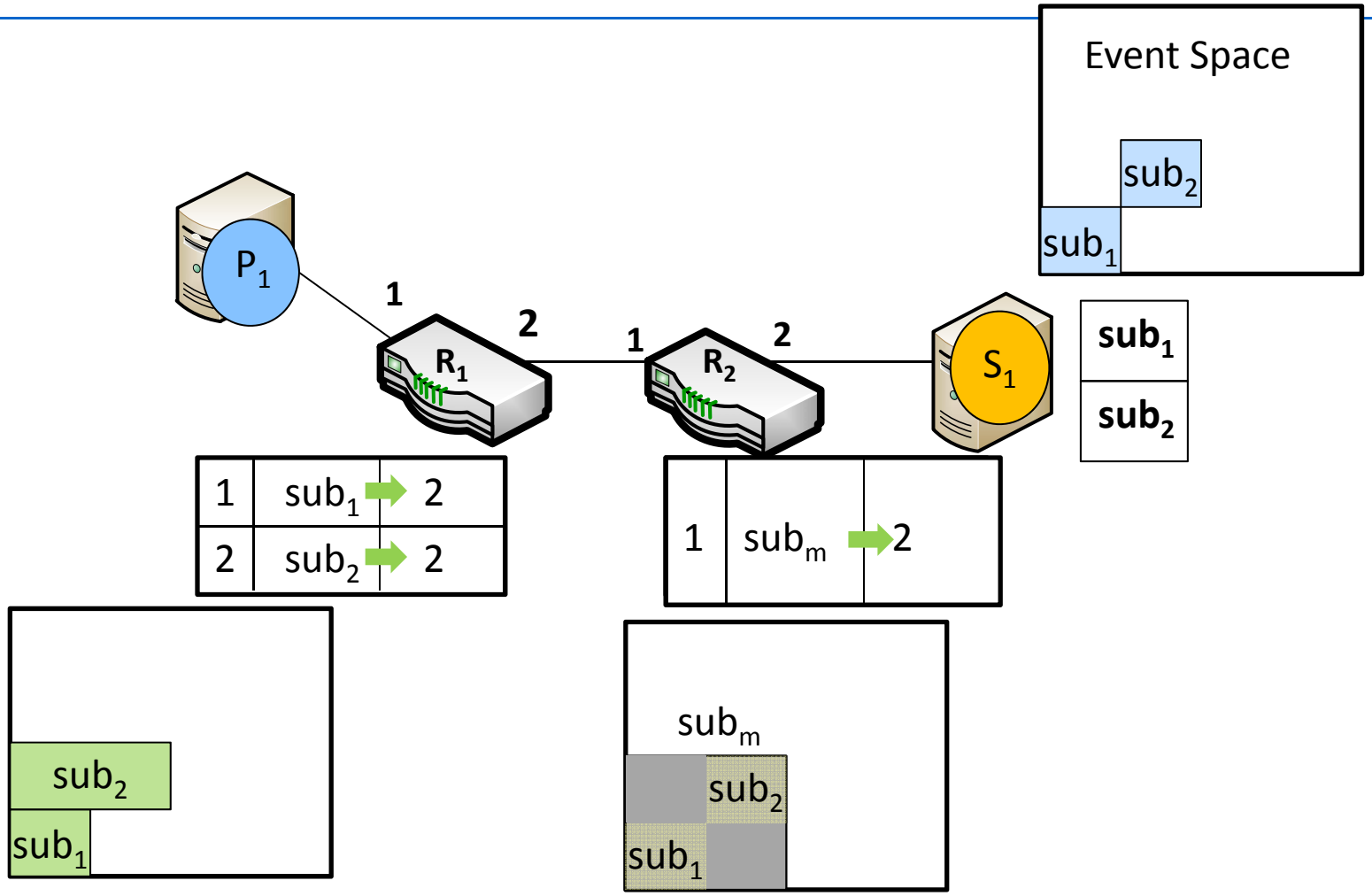
- For each switch ∈ $\mathbb{ER}$
  - Determine possible merges
    - Merge Point Tree
  - For each merge point
    - Calculate Aggregation Benefit
      - Number of reduced flows
    - Calculate Aggregation Cost
  - Greedy selection based on cost per benefit



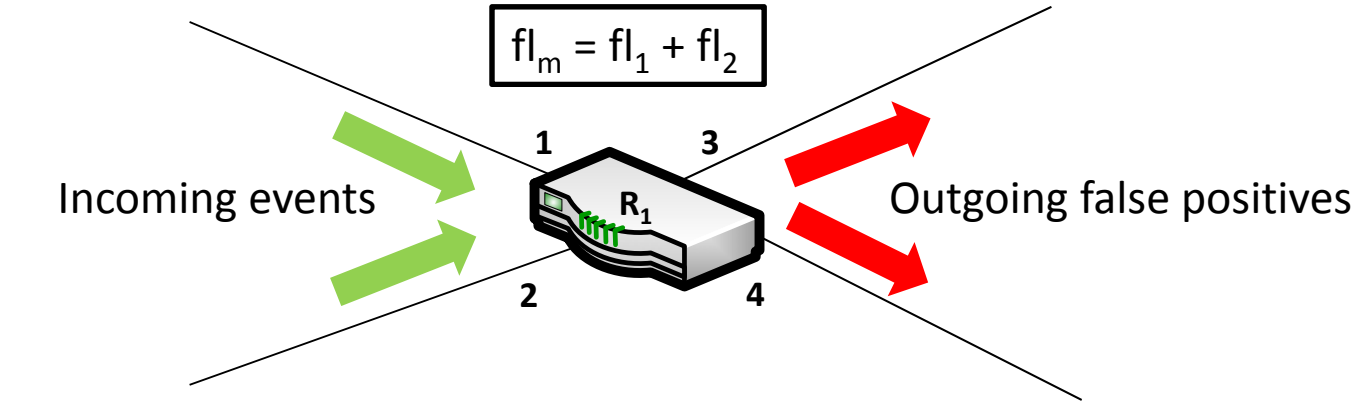
Aggregation Cost at a Merge Point



Aggregation Cost : False Positive Space



Aggregation Cost at a Merge Point

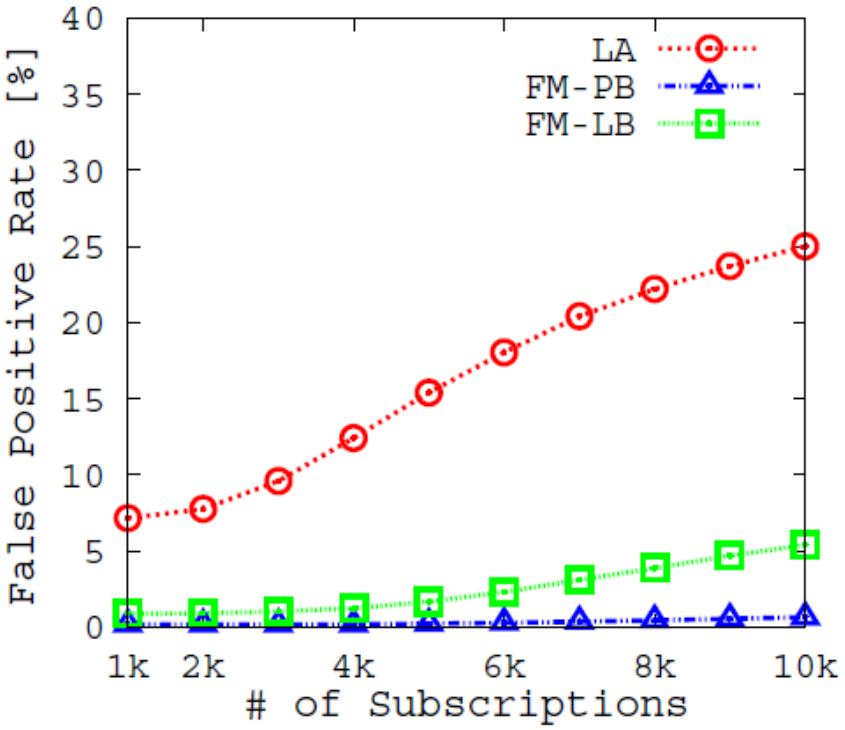


$$Cost = \sum_{op \in oP} \sum_{p \in iPaths} {fp^p}_{op} * dLinks_{op}$$

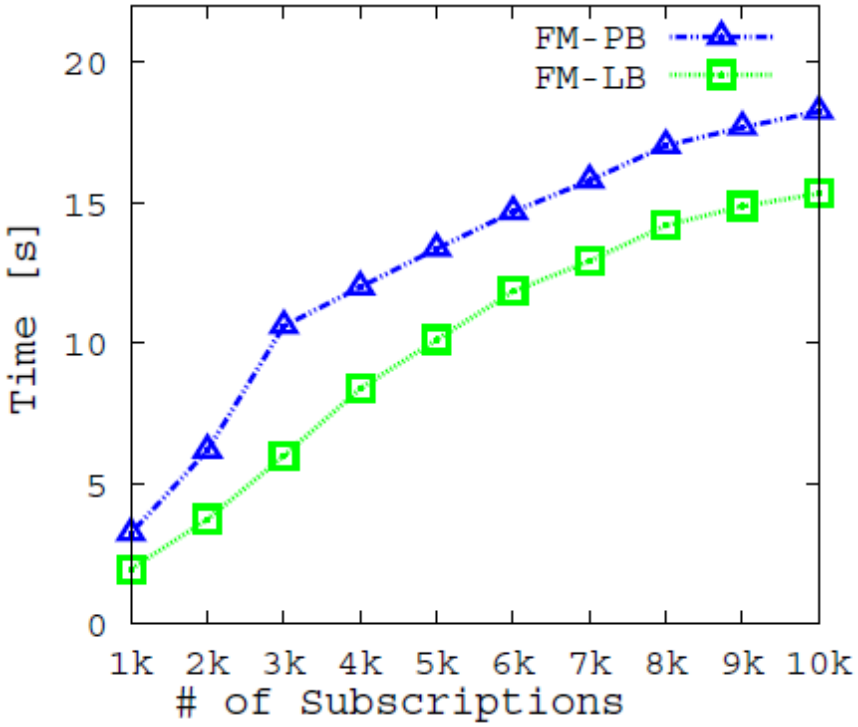
Load-based Method	Pattern-based Method
✓ Less Overhead	✓ More Efficient
✗ Less Efficient	✗ More Overhead

Performance Evaluations

False Positive Rate



Runtime Overhead



~300 switches, ~4000 hosts
Zipfian data distribution
10,000 events
TCAM constraint : 600 flows

Conclusion

- Expressive filtering of events despite aggregation of filters in the presence of TCAM constraint on switches

- Propose the Filter Aggregation Algorithm
  - Pattern-based method
  - Load-based method
  - Local Aggregation Method to handle dynamics in the system

Questions?

Thank you for your attention!

Contact:

Sukanya Bhowmik
Institute of Parallel and Distributed Systems (IPVS)
Universität Stuttgart
Universitätsstraße 38
70569 Stuttgart, Germany

Email: sukanya.bhowmik@ipvs.uni-stuttgart.de
Web: http://www.ipvs.uni-stuttgart.de/
Phone: +49 711 685 88245

dSDN
distributed
Software-Defined
Networking

http://www.d-sdn.de