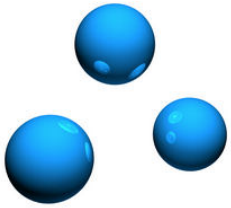


EWSDN

European Workshop on
Software Defined Networks

ERICSSON

# Resource Optimization for Service Chain Monitoring in Software-Defined Networks

Ming Xia, Meral Shirazipour, Heikki Mahkonen,
Ravi Manghirmalani and Attila Takacs
Ericsson Research

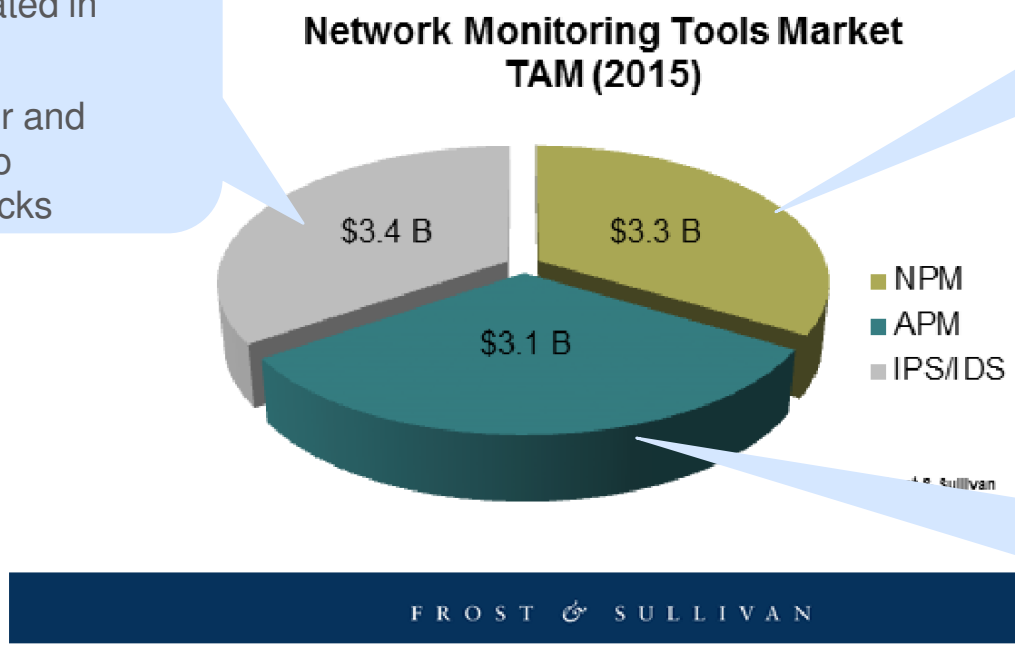
*The 4th European Workshop on SDN*
September 30 – October 2, 2015

# Traditional Monitoring

- Traditional Monitoring Tools :
  - Network Performance Monitoring (NPM)
  - Application Performance Monitoring (APM)
  - Intrusion Detection/Prevention (IPS/IDS), Security monitoring

-typically standalone, but could be integrated in applications

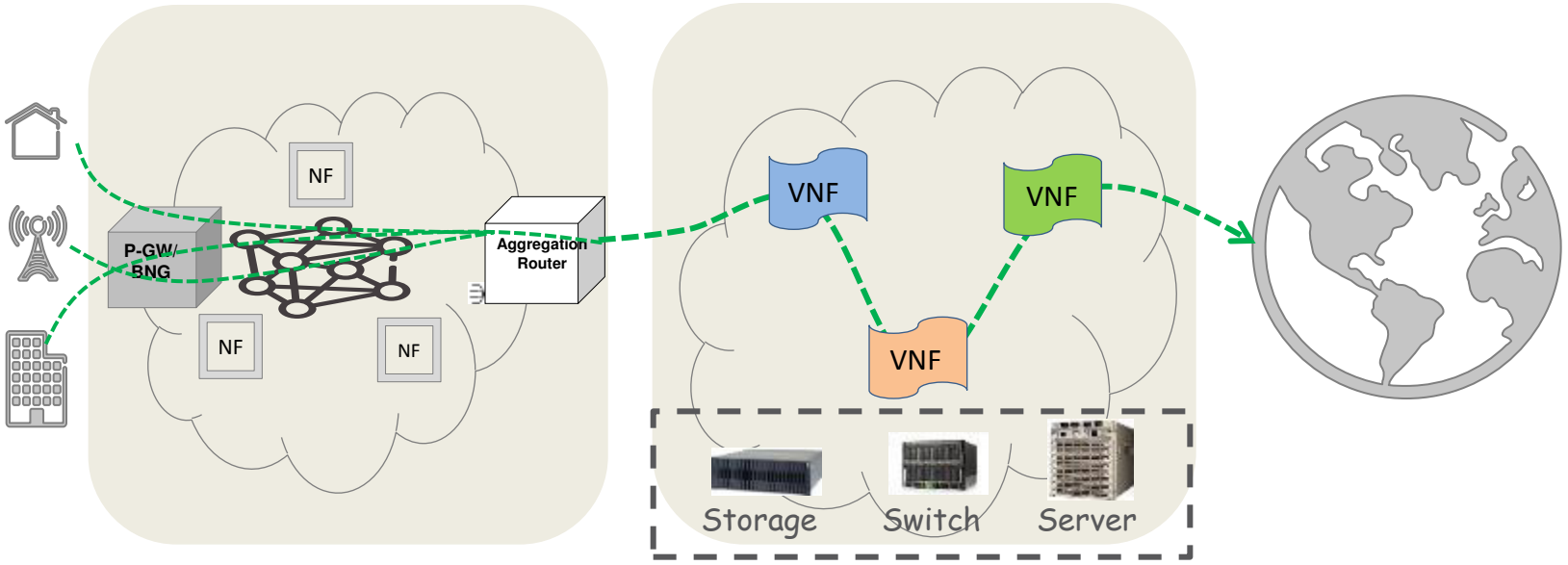
-actively monitor and analyze traffic to detect/stop attacks




-typically tools to check network integrity, policy compliance, and SLA violations

-tools to monitor the application performance from end user experience.
-monitoring of application components and RCA

# SDN/NFV-Based Service Chaining




**Operator's Access Network with Legacy NFs (e.g. middleboxes)**

**Operator's DC/Cloud with virtualized network functions**

**Internet**

**Service Chaining**: The operation of steering a traffic flow through a number of network functions in a specific order based on the operator's policy and service level agreement (SLA)

# Literature Review:

## Selected Monitoring Tools and Products

Network (infra and virtualized):

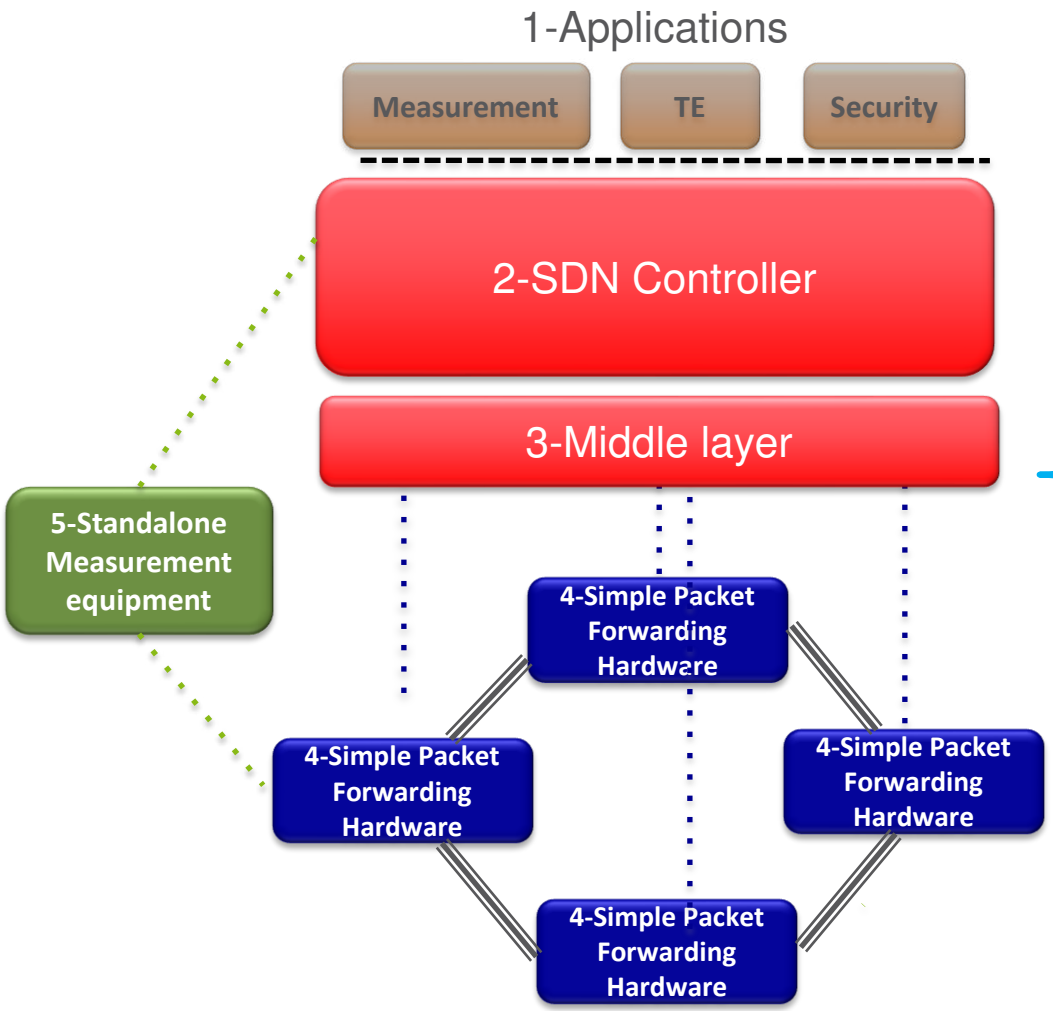
- Netflow, sFlow
- Commercial:
  - cPacket
  - APCON
  - Net Optics
  - BigSwitch (BigTap)
  - Gigamon
  - ThousandEyes
  - ESI by Overture (for NFV)
  - ONPath and Simena
  - Anue
  - VSS
  - Virtela
  - Vistapointe
- SDN based proposals
  - Shown in the next slide.

Cloud (infra & application):

- System tools:
  - collectd
  - Graphite
  - Cacti
- Cloud Tools:
  - Ganglia
  - Nagios
  - zenoss
  - Ceilometer
  - Sensu
  - Monasca (Openstack)
  - Gigamon
  - Satelliz
  - IBM smart cloud monitoring
  - Boundary
  - Datadog

# Literature Review

## SDN Monitoring Classification




Courtesy to Ying Zhang (was with Ericsson Research when this review was performed)

- **1-Network measurement as applications**
  - DREAM [Moshref: SIGCOMM14]
  - OpenWatch [Zhang: CoNext 13]
  - Intentional Network Monitoring [Donovan: HotNets14]
  - Compiling queries [Narayana:HotSDN14]
  - DDoS prevention [Brocade]
  - SENSS [Yu: ONS14]
- **2-Controller instrumentation**
  - NICE [Canini: NSDI12]
  - HotSwap[Vanbever:HotSDN13]
  - Assertion language [Beckett: HotSDN14]
  - LegoSDN [Chandrasekaran:HotNets14]
- **3-Middle layer debugging**
  - VeriFlow [Khurshid: HotSDN12]
  - NetPlummer[Kazemian:NSDI13]
  - Header space analysis [Kazemian:NSDI12]
  - Virtual Network diagnosis [Wu: SoCC13]
  - Forward Networks
- **4-Switch extensions**
  - OpenSketch[Yu: NSDI13]
  - State transition[Moshref:HotSDN14]
  - P4 [Bosshart: SIGCOMM13]
  - TPP [Jeyakumar:SIGCOMM14]
- **5-Standalone measurement box**
  - NetSight[Handigol: NSDI14]
  - MCS[Scott:SIGCOMM14]
  - Y![Wu: SIGCOMM14]
  - BigTap[BigSwitch]

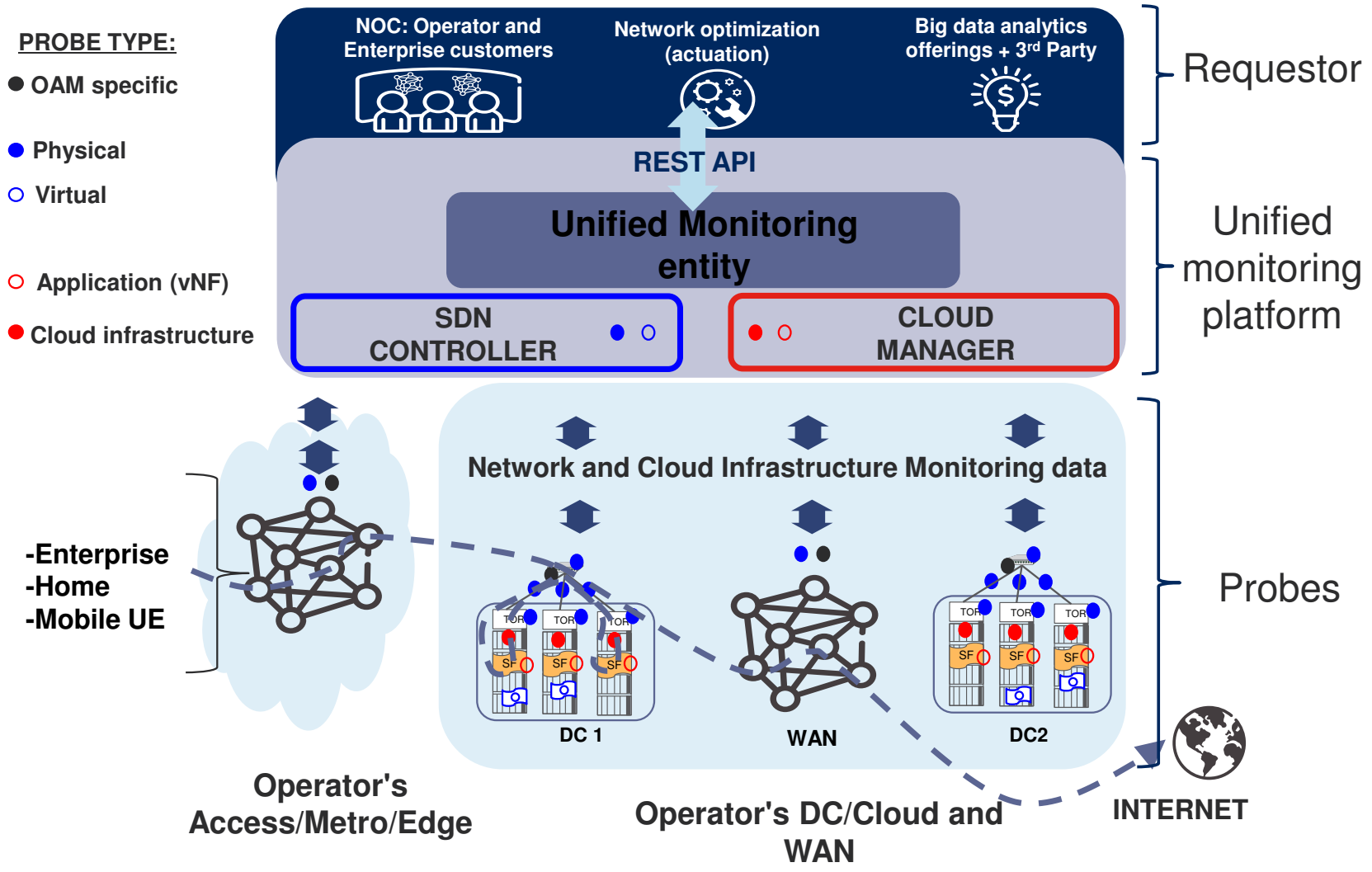
# Counting & Monitoring for NFV

- Traditional network monitoring goals still apply :
  - i.e. OAM/Network Performance Monitoring (NPM) , Application Performance Monitoring (APM), Intrusion Detection/Prevention (IPS/IDS)
- New opportunities:
  - Monitoring as a Service (MaaS) concept
    - i.e. Dynamic: on demand and scalable (not a standalone product)
      - C&M can grow as traffic grows (e.g. for faster network upgrades)
  - General C&M applications adapted to specific use cases
    - With fast TTM
  - Enhance performance through real-time feedback based monitoring:
    - Close-Loop concept and Actuation, use of analytics.
  - Optimized use of C&M resources with a unified network(SDN) and cloud monitoring.

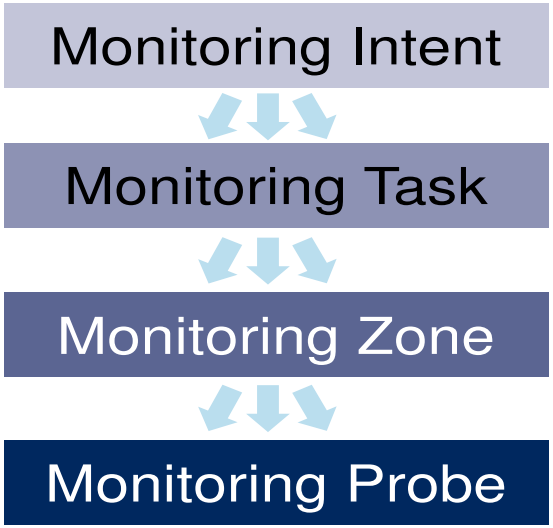
# Categorization of Monitoring Probe

| **Network monitoring** -Physical Probes | **Network monitoring** -Virtual probes in form of vSwitch etc. | **Cloud Infrastructure** | **Cloud Application** (e.g. NFV) |
|---|---|---|---|
| **Network metrics:** *(cumulative and rates)* › Packet count › Latency › Jitter › Bandwidth › Packet loss › … | **Network metrics:** *(cumulative and rates)* › Packet count › Latency › Jitter › Bandwidth › Packet loss › … | **Cloud metrics:** *(cumulative & rates)* › CPU › RAM › Bandwidth › Storage › … | **Examples for NFV Apps KPIs:** › Processing delay (using VM Rx/Tx interface timestamps) › Throughput › Functional Correctness *(not necessarily from monitoring probes)* |
| **OAM** › CFM/ Service OAM: -IEEE 802.3ag › Link OAM: - IEEE 802.3ah › OAM Functions: -ITU-T Y.1731 › Other: BFD, TWAMP, SDN versions. | **OAM** › Same as physical network probes | **OAM** › Tools such as Zabbix, already used for infrastructure monitoring | **OAM** › Packet loss rate, latency, delay variation of flows, › Maximum time to detect and recover from faults. (availability) |

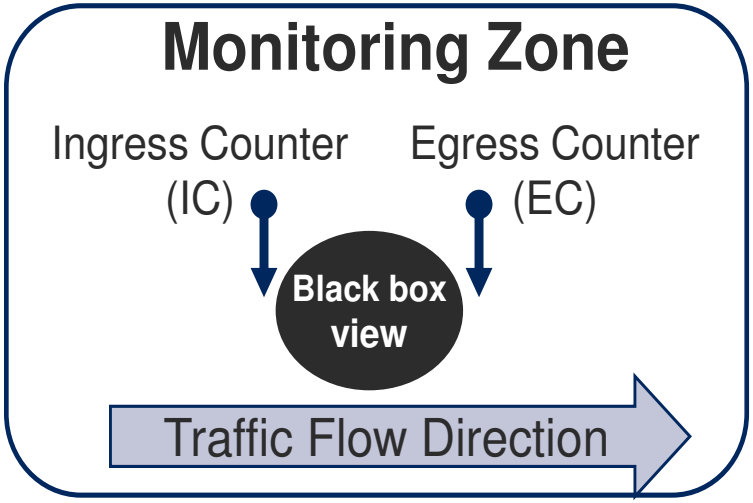
# Monitoring Framework (Diamond)



# Intent-Based Monitoring

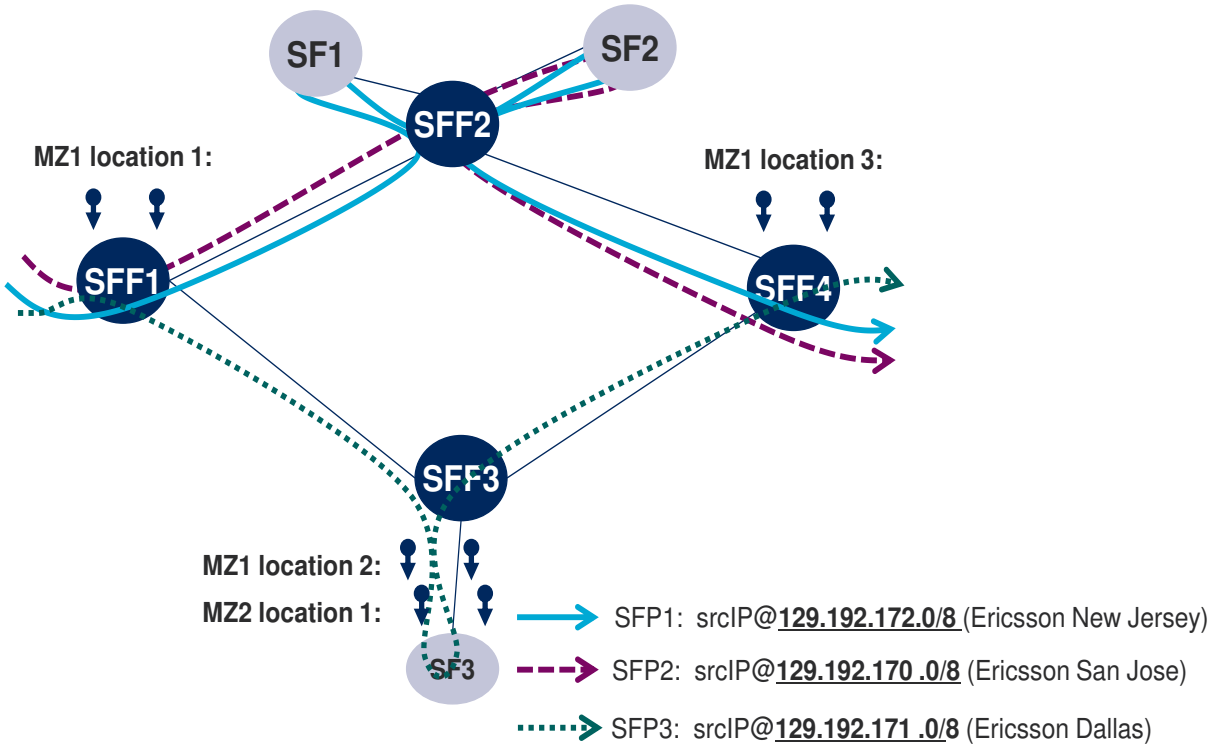



Monitoring hierarchy




Monitoring zone

# Monitoring Rule Consolidation




| | |
|---|---|
| Task 1 | MZ1: anywhere on SFP3 monitor traffic volume to destIP@74.125.0.0/16 (i.e. going to Google) |
| Task 2 | MZ2: on SFP3 monitor traffic volume through SF3 for trafficdestIP@74.125.0.0/16 (i.e. going to Google). |

# Problem Statement

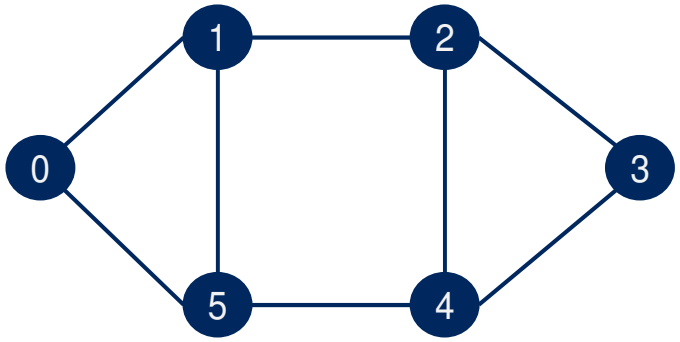
- Given:
  - a) a network topology and a set of SFPs on the topology;
  - b) a set of MZs associated with the SFPs;
  - c) a cost model for probe deployment, which can depend on number of rules, port count, etc. In this paper, we consider the cost of using a place to be inversely-proportional to the number of MZs already deployed at this place.
- Goal:
  - Deploy MZs with minimized probe-deployment cost.

# Algorithm for MZ placement

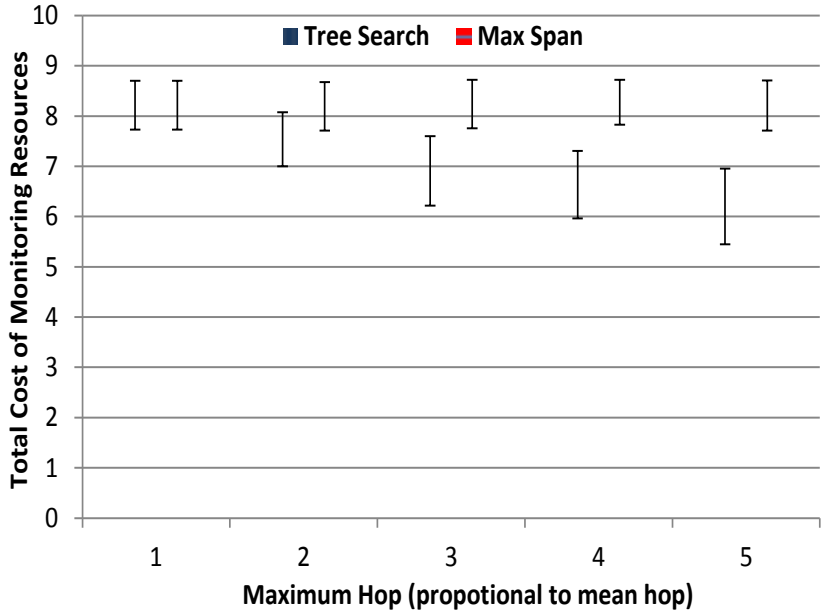
Algorithm 1. Tree search for minimum placement cost (maximizing sharing among MZs based on the cost model)

```
1        getMinCost()
2           initialize L=all MZ, P=null
3          totalCost=0, minCost=MAX_VALUE
4           search(L, P, totalCost, minCost)
5           return minCost
6
7        search(L, P, totalCost, minCost)
8           if(P==null && L==null)
9           minCost=min(minCost, totalCost)
10             return
11         else if(P!=null)
12              for each location pair (p1, p2)  in P
13                 cost=costp1/n1 + costp2/n2
14              totalCost=totalCost + cost
15                 search(L, null, totalCost, minCost)
16              totalCost = totalCost – cost
17          else
18           for MZ in L
19                 generate P: list all possible location-pairs  for IC and EC of MZ
20                 search(L-MZ, P, totalCost, minCost)
```

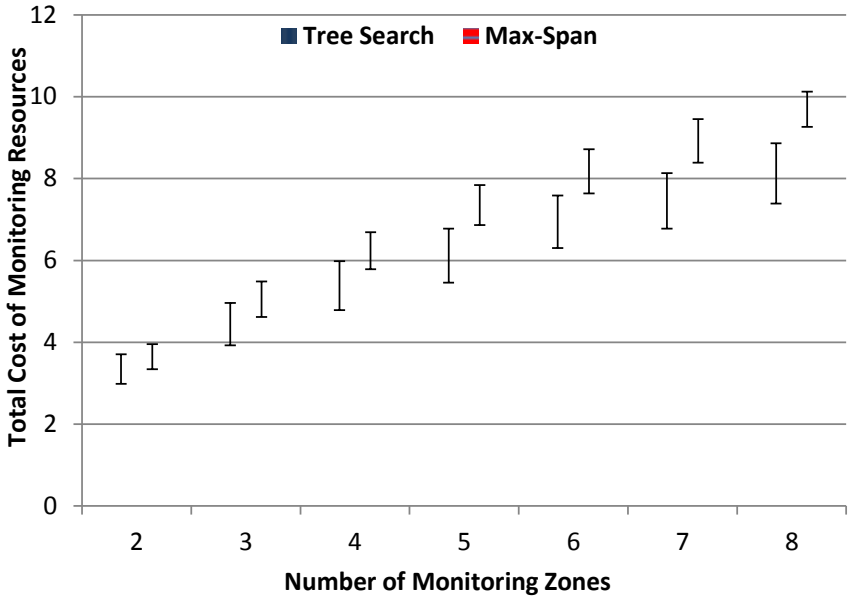
# Performance Evaluation




Simulation topology




Maximum hop count vs. total cost.



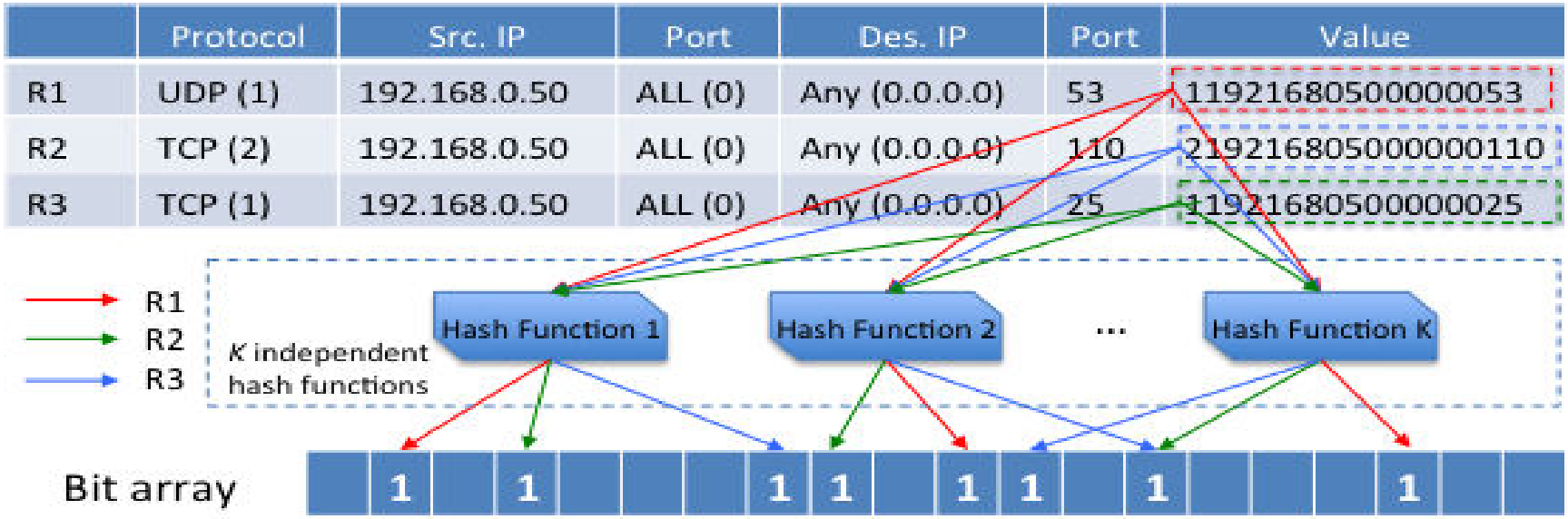

Number of monitoring zones vs. total cost.

Benchmarking algorithm: *MaxSpan*

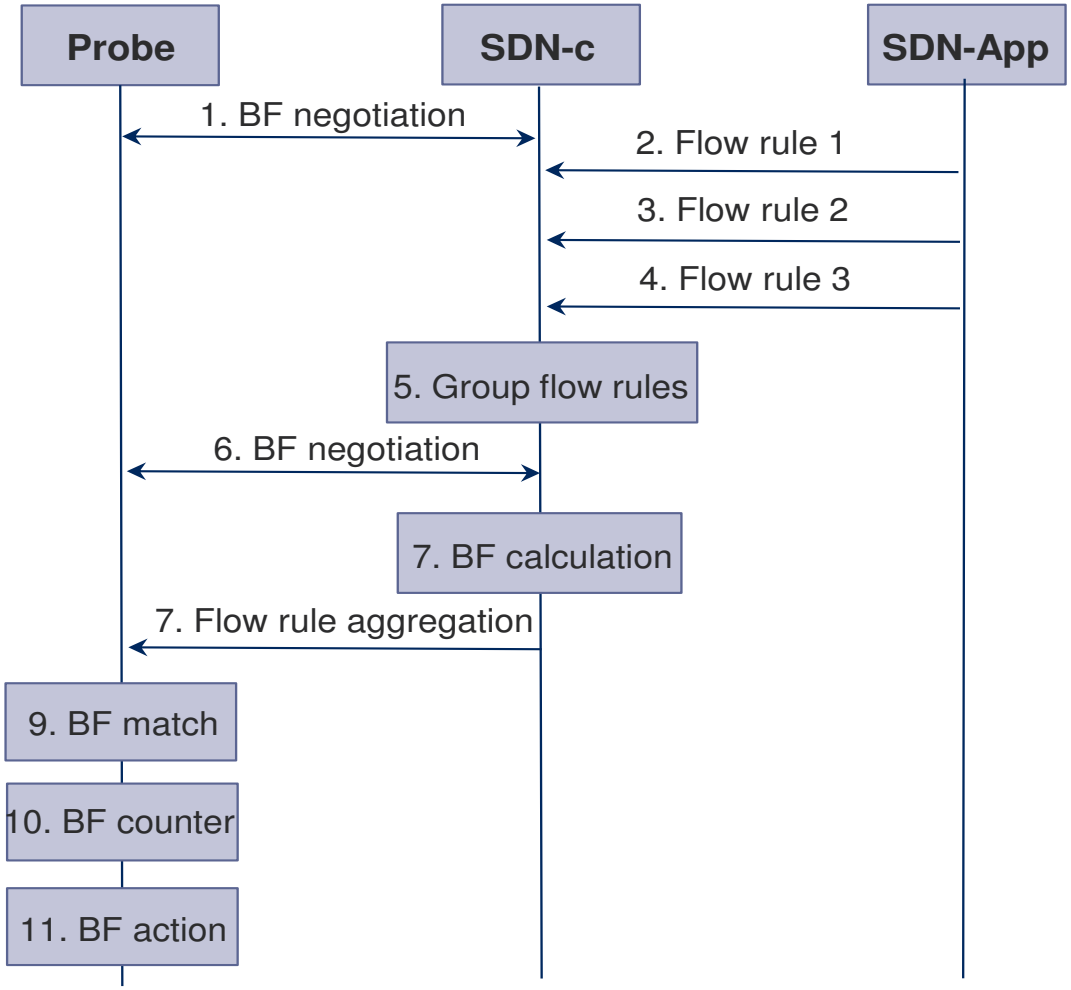
Always choose the leftmost and right most two locations as the location pair

# Further Resource Optimization using Bloom Filter

- By overlapping monitoring zones of different monitoring tasks, there is possibility for monitoring rule consolidation
- Five-tuple based monitoring rules are not efficient for wildcarding different rules
- We introduce Bloom Filter for monitoring rule consolidation (false positive rate vs. resource efficiency)



# Execution Sequence



# Resource Saving



Input:

10K rules per MZ

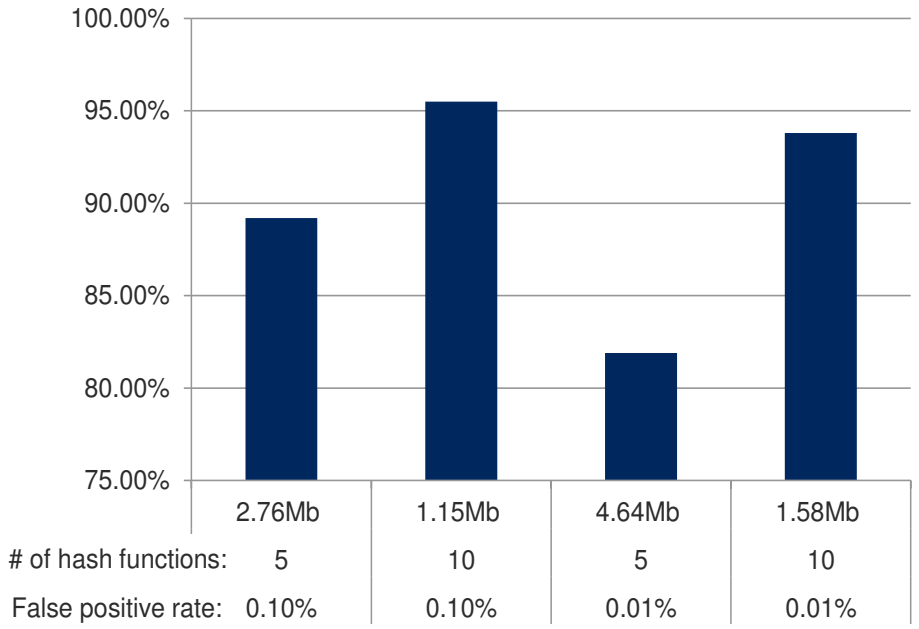
20 MZs with shared probe locations

False positive rate:

$$r = (1 - e^{-\frac{kn}{m}})^k$$

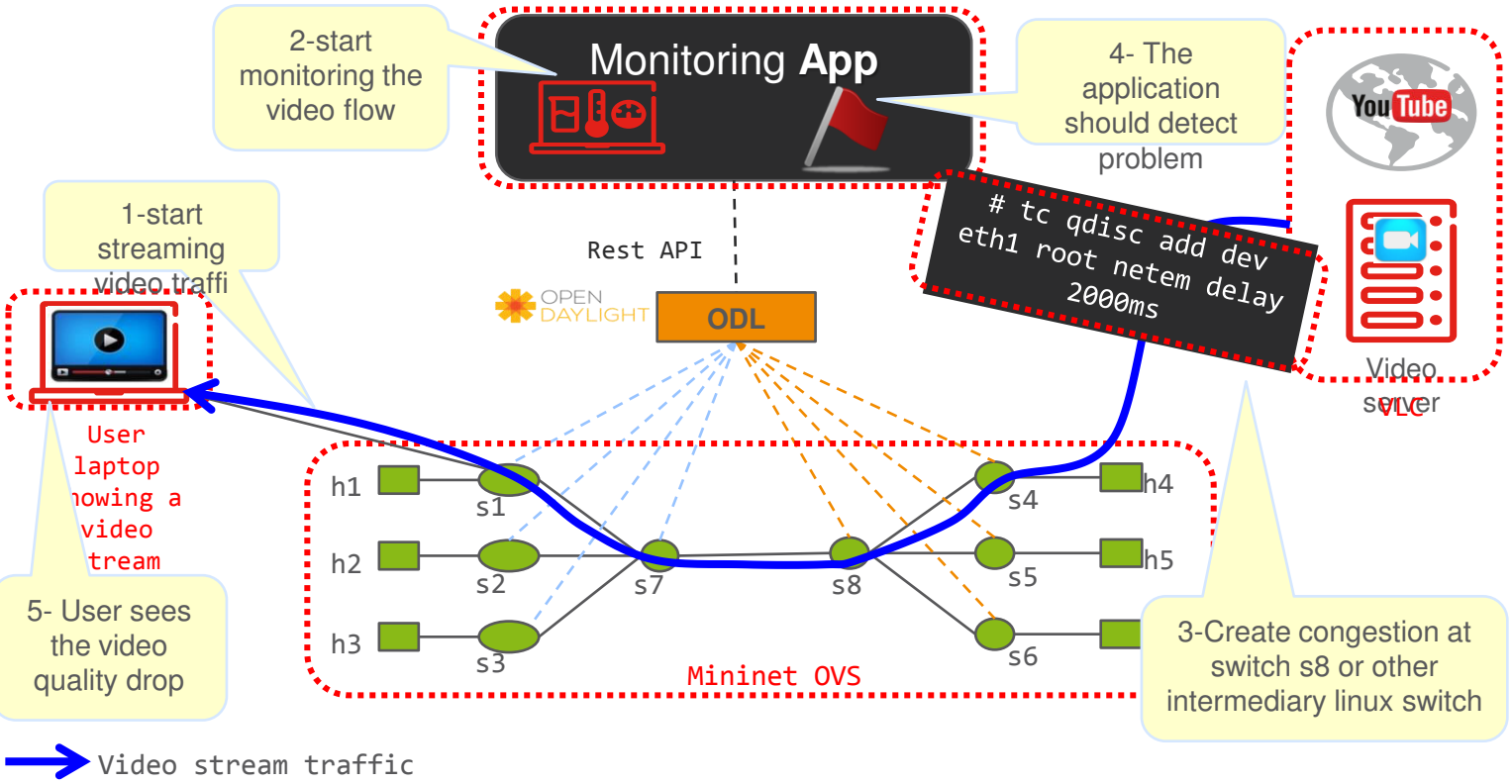
Bits requirement of five tuples.

| Field | Bits |
|---|---|
| Protocol | 8 |
| Src. address | 32 |
| Src. port | 16 |
| Des. address | 32 |
| Des. port | 16 |



# Demo

- Demo at *Open Networking Foundation* (ONF) AppFest using OpenDaylight Lithium as SDN controller
- Intent-based performance monitoring for video traffic and on-demand (i.e. programmable) troubleshooting




* Interested audience may refer to our article at Ericsson Research Blog

# Summary

- SDN & NFV will greatly reshape our networks, and bring in both complexity and controllability
- An intent-based monitoring architecture was proposed especially for service chaining, which decouples the complexity of user plane from control plane
- We identified the opportunity for monitoring resource saving among monitoring zones
- We proposed two-level optimization 1) monitoring zone level; 2) probe level. Our algorithm guarantees optimality when placing monitoring zones. Using Bloom Filter can consolidate monitoring rules to realize resource saving

# Co-authors

Meral Shirazipour

Heikki Mahkonen


Ravi Manghirmalani

Attila Takacs

ERICSSON