Virtual Network Diagnosis as a Service

Wenfei Wu (UW-Madison)

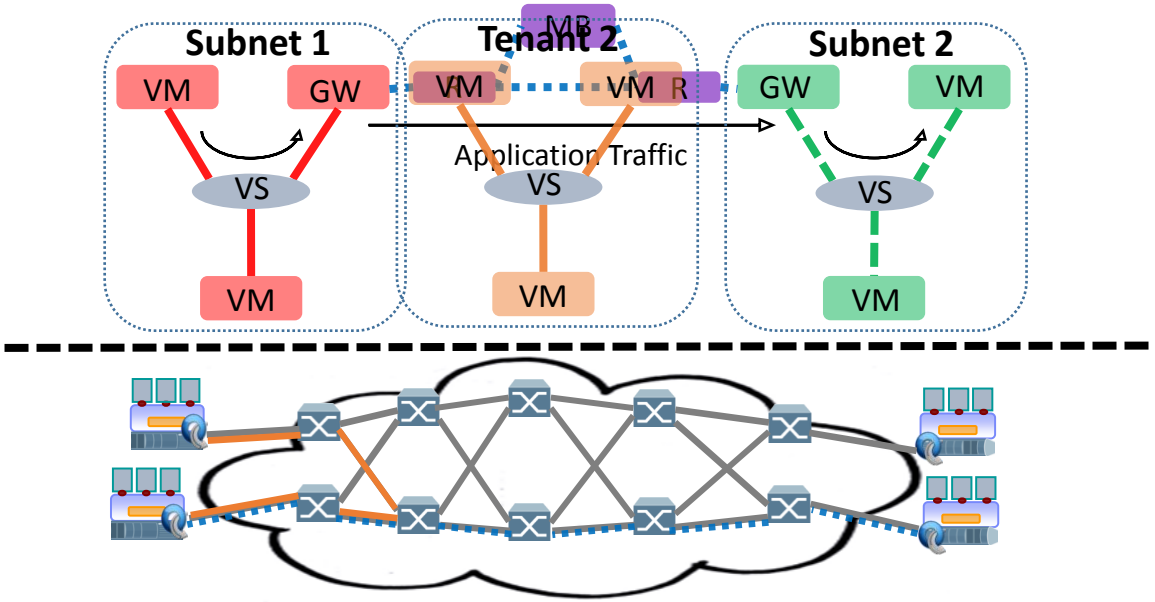
Guohui Wang (Facebook)

Aditya Akella (UW-Madison)

Anees Shaikh (Google)

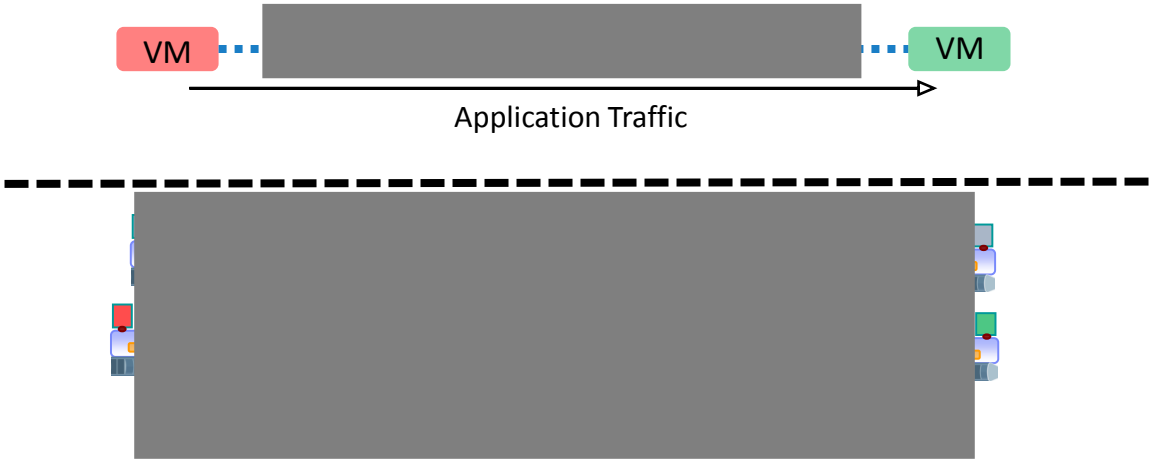
Virtual Networks in the Cloud

- Data center infrastructure
- Virtual networks
- Virtual-to-Physical Mapping
- Network services
- Sharing, Isolation



Virtual Network Problems

- Multiple layers may have various problems
  - Connectivity/Performance issues in applications
- Isolation and abstraction prevent tenants from diagnosing their virtual networks

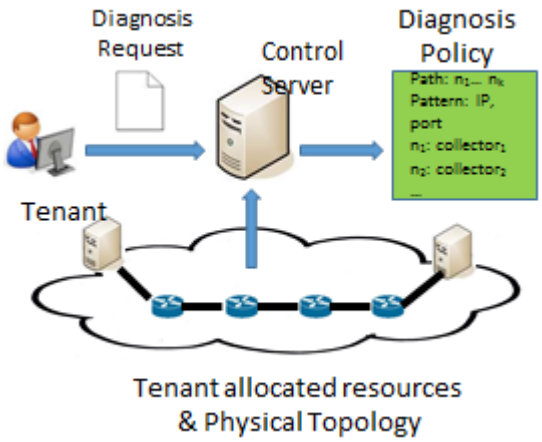


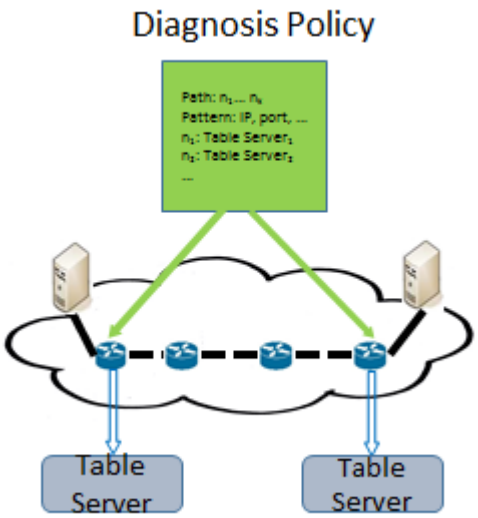
Existing Solutions

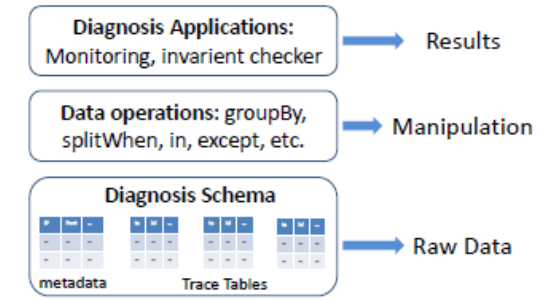
- Infrastructure-layer tools may expose the physical network to tenants
  - sFlow, NetFlow
  - OFRewind, NDB, etc.
- Tools in VMs are difficult to deploy in some virtual components (e.g., middleboxes)
  - Tcpdump, Xtrace, SNAP

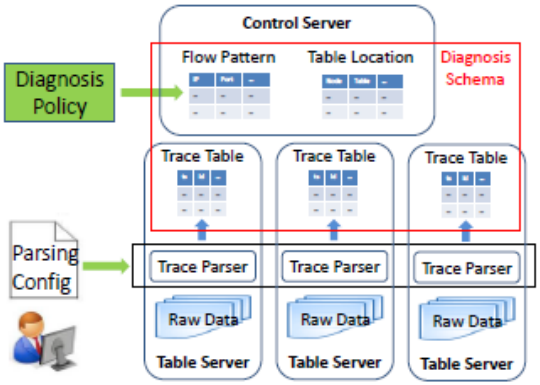
VND Proposal

- The cloud provider should offer a virtual network diagnostic *service* (VND) to the tenants
  1. Diagnostic request
  2. Trace collection
  3. Trace parse
  4. Data query









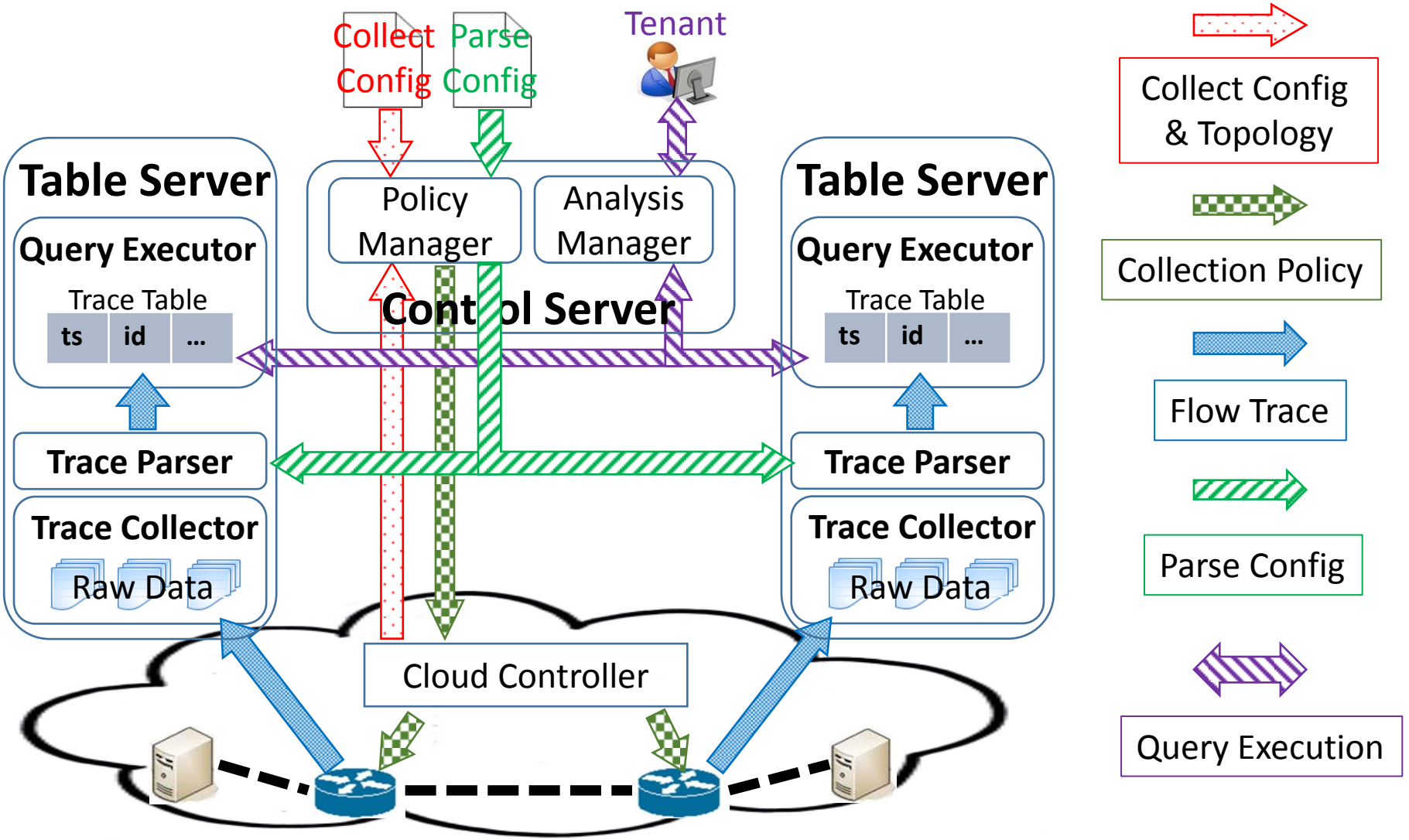
VND Challenges

- Preserve isolation and abstractions
- Low overhead
- Scalability

Contents

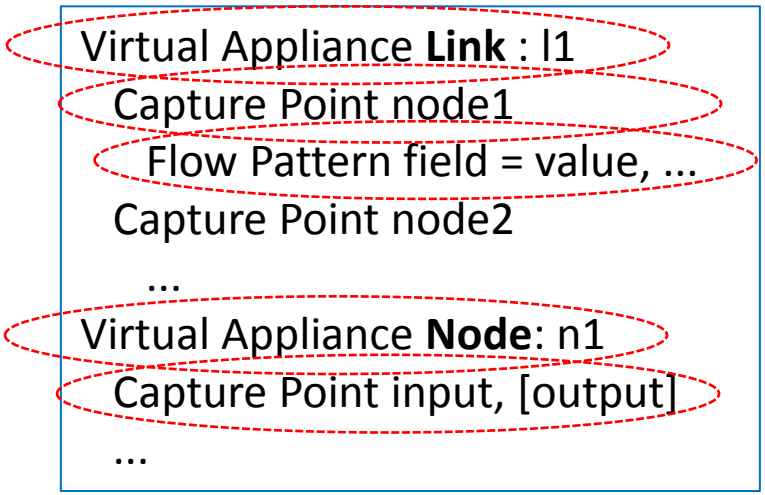
- Motivation
- VND Design & Implementation
- Evaluation
- Conclusions

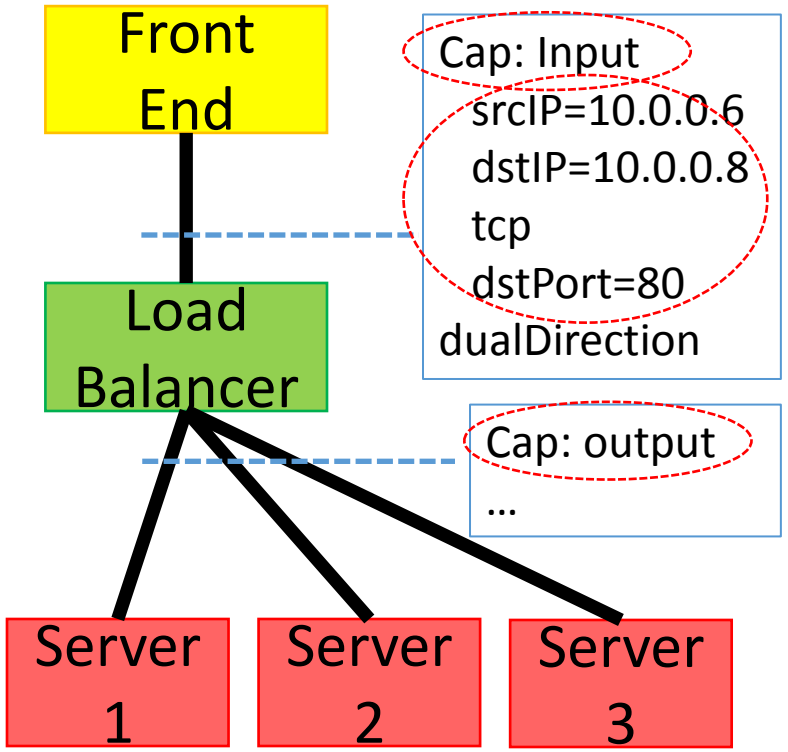
VND Architecture



Data Collection (1)

- The tenant submits a Data Collection Configuration

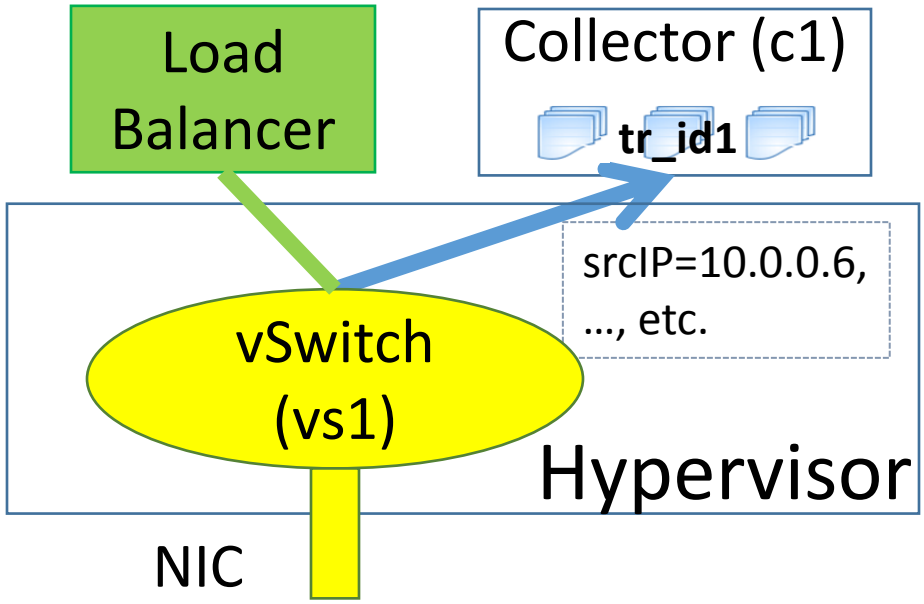




Data Collection (2)

- Policy Manager generates a Data Collection Policy

Trace ID
tr_id1
Physical Cap Point
vs1
Collector
c1 at h1
Pattern
field = value, ...
Path
vs1, ..., h1, c1



Data Parse

- The tenant submits a Data Parse Configuration

Table ID tab_id1
Filter exp
Fields field_list

exp = exp and exp |
exp or exp | not exp
| (exp) | prim
prim = field in value_set

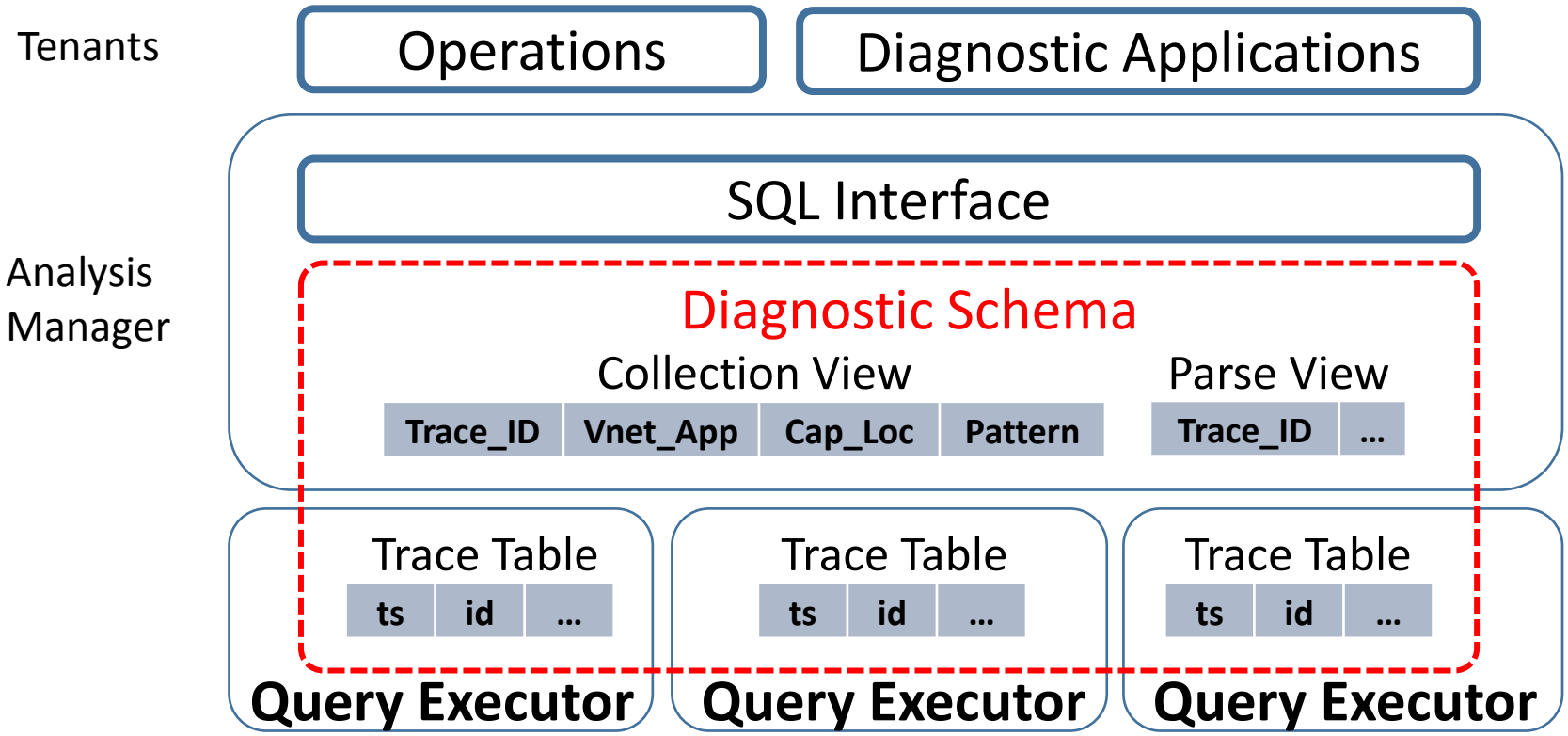
field_list = field (as name)
(, field (as name))*

Trace ID all
Filter: ip.proto = tcp
or ip.proto = udp
Fields: timestamp as ts,
ip.src as src_ip,
ip.dst as dst_ip,
ip.proto as proto,
tcp.src as src_port,
tcp.dst as dst_port,
udp.src as src_port,
udp.dst as dst_port

ts	src_IP	dst_IP	proto	src_port	dst_port
…	…	…	…	…	…

Data Analysis

- Trace Tables form a distributed database



Data Analysis Examples

ts	src_IP	dst_IP	seq	ack	length
t1	IP1	IP2	1000	1	1400
t2	IP2	IP1	0	2400	0
t3	IP1	IP2	2400	1	1400
t4	IP2	IP1	0	3800	0

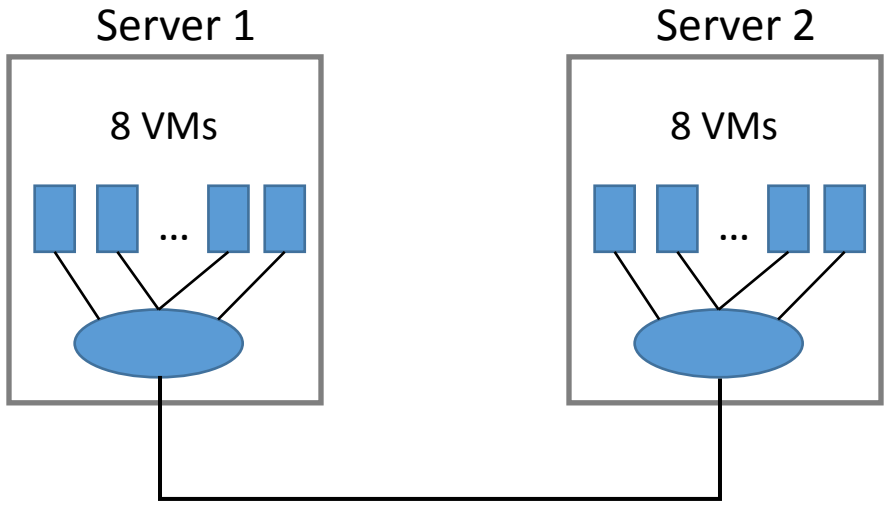
- RTT
  - Correlate data packet and ack packet
  - Compute the average time difference
- Throughput, loss, delay, statistics, etc.

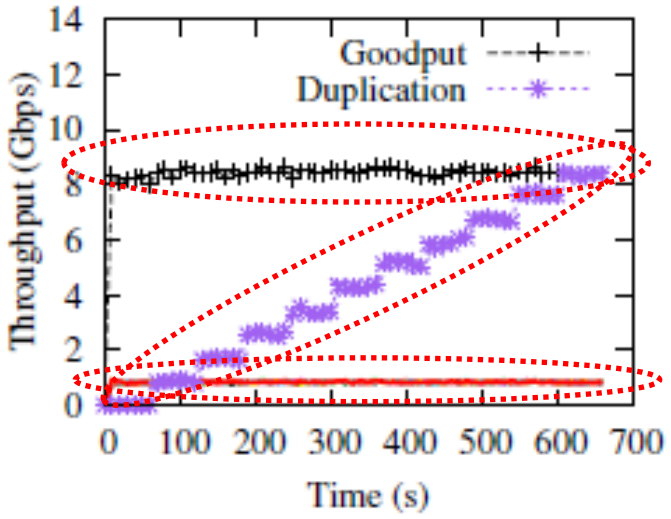
Contents

- Motivation
- VND Design & Implementation
- Evaluation
- Conclusions

Trace Collection (1)

- Network Overhead
  - Each minute, we capture one additional VM pair's traffic
  - The duplicated traffic increases as we capture more traffic
  - The original application traffic is not impacted

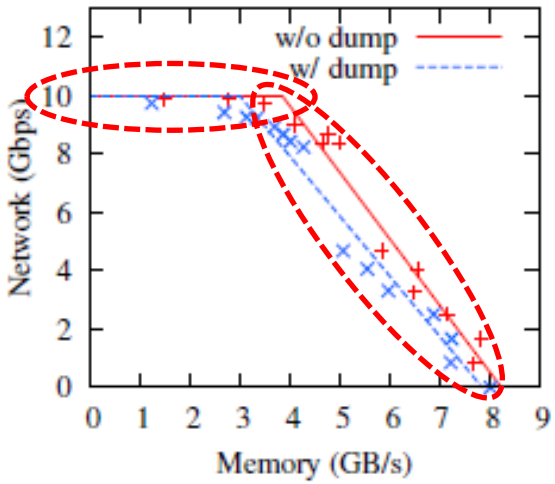




Trace Collection (2)

- Memory Overhead
  - VMs perform memory copy and data transfer simultaneously
  - We measure memory and network throughput

  - Each 1Gbps network traffic duplication causes 59 MB/s memory overhead



Data Query Overhead

- We use RTT monitoring as an example
  - 200 Mbps traffic
  - Calculate average RTT periodically
- Network overhead is negligible
- Execution time scales linearly with the data size
- VND can process 2-3 Gbps traffic in real time

Period(s)	1	3	5	7	9
Execution(s)	0.1	0.29	0.49	0.69	0.9
Traffic(MB)	<0.1	<0.1	<0.1	<0.1	<0.1

RTT

Scalability

- Control Server is a simple web server can be scaled up easily
- Data collection distributed locally to avoid being the bottleneck
- Query execution is also distributed
  - Real-time analysis
    - 2-3Gbps (RTT example, MySQL implementation)
  - Offline analysis
    - Higher volume

Conclusions

- The cloud provider should offer a virtual network diagnostic service to the tenants
- We design VND
  - Architecture, interfaces and operations
  - Address several important challenges
- Our experiments and simulation demonstrate the feasibility of VND

END

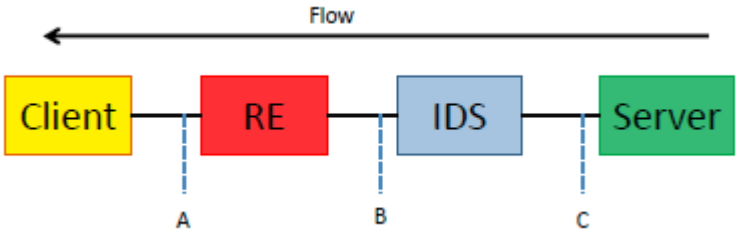
Q&A

Please contact with WISDOM

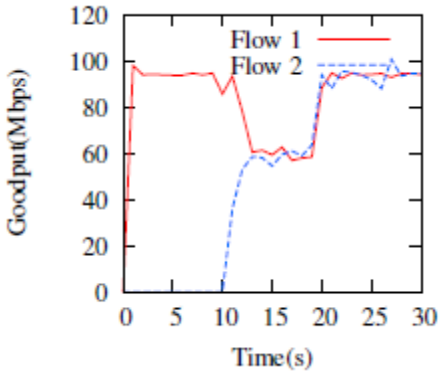
http://wisdom.cs.wisc.edu

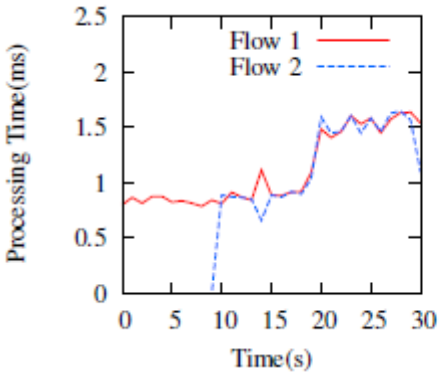
Functional Validation

- Middlebox Bottleneck Detection

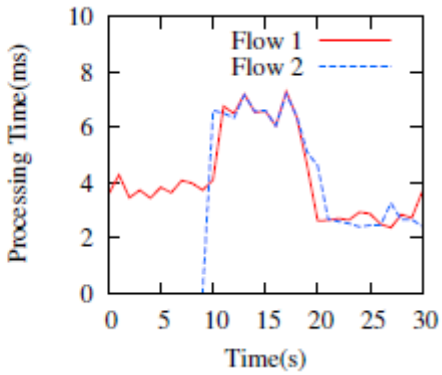


- VND use throughput and RTT time to find abnormity





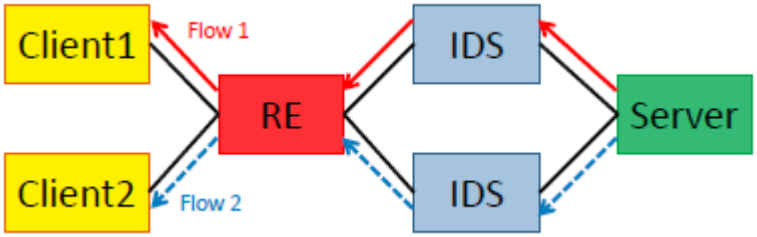
RE

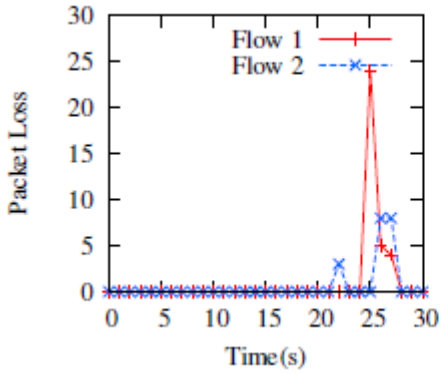


IDS

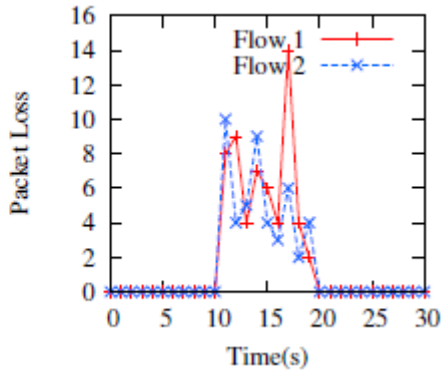
Functional Validation(2)

- Middlebox scaling



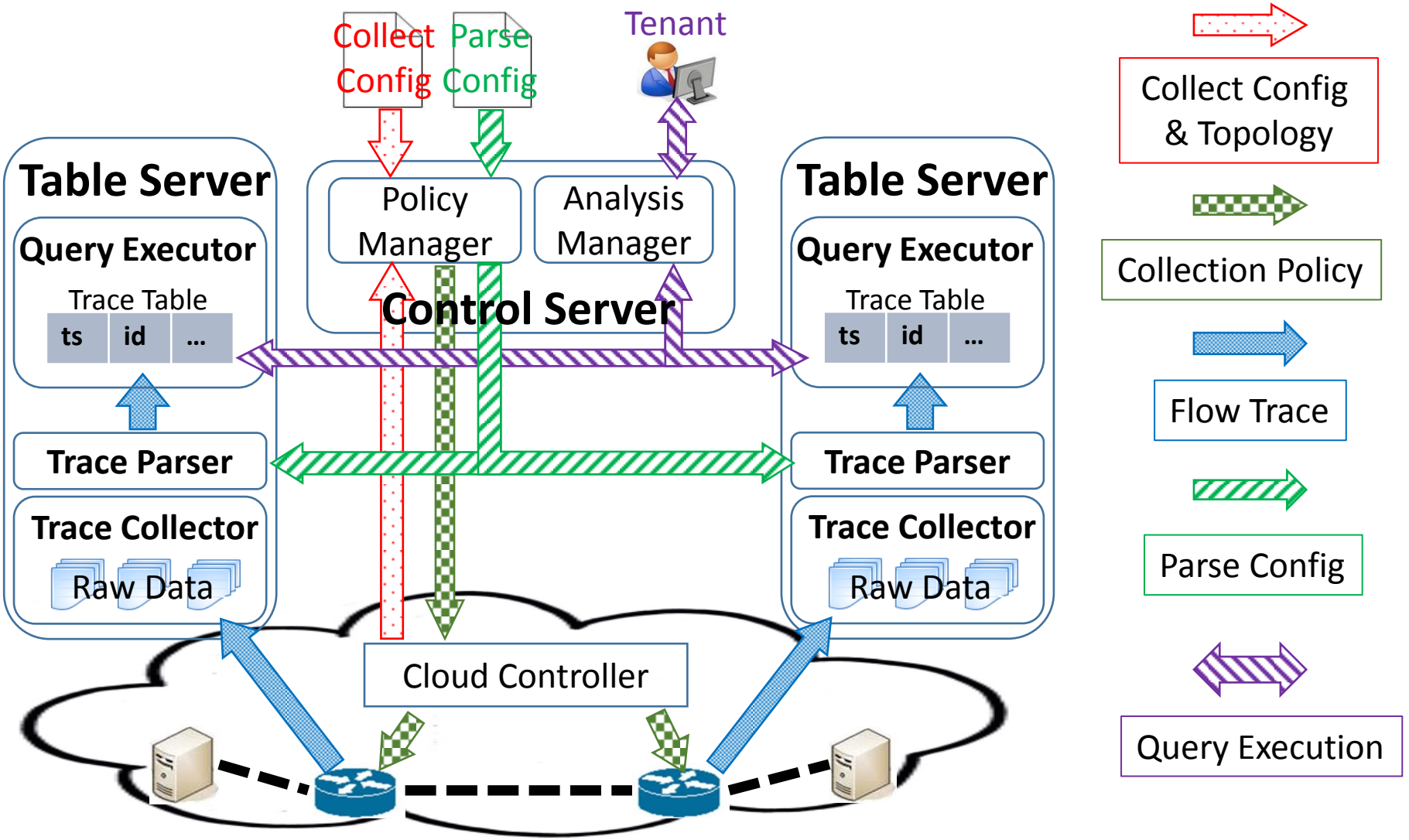


RE



IDS

VND Architecture



Optimizations

- Local Table Server placement
  - Place the collector locally with the capture points
  - Avoid the trace collection traffic traversing the network
  - Move data only when queries need it
- Avoid interference with existing rules using the multi-table feature on OVS

Scalability (2)

- Data Query simulation
  - A data center with 10,000 servers, each has a 10Gbps NIC
  - Virtual network size [2, 20]
  - Query executors can process 3 Gbps traffic in real time
  - Total link utilization [0.1, 0.9]
- Results
  - 30% of total link capacity can be queried in real time