SIGCOMM 17 Preview Session: Network Monitoring

Ying Zhang
Software Engineer, Facebook
Network monitoring is important!

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Diagnose long delay/loss problems</td>
</tr>
<tr>
<td>Utilization</td>
<td>Traffic engineering</td>
</tr>
<tr>
<td>Availability</td>
<td>Identify and diagnose failure</td>
</tr>
<tr>
<td>Security</td>
<td>Timely attack detection</td>
</tr>
</tbody>
</table>
Network monitoring is unsolved

Market research reports a $2.32 B network analytics market by 2020

Startups in network analytics

- Pluribus Networks
- LogicMonitor
- Big Monitoring Fabric
- Netsil
- Luminus
- Enlightened networking
State of the art: bottom up approach

Too much information: Hard to find needles in a haystack

Lack network-wide view:
Ad-hoc integration of data across switches & hosts

Too little information: aggregated, sampled data is not enough
Key challenges

Flexible: diverse, real-time queries

Scale: thousands of devices, millions of flows, terabytes of traffic

Performance: limited resource/support on hardware switches
Challenge 1: lack of abstraction

Need to support diverse, customized, real-time query

- Attack detection
- Performance diagnosis
- Traffic engineering

Network analytics on the collector
Challenge 2: Scalability

Scale to a large amount of traffic in a large network

Network analytics on the collector

10,000s switches

100,000s servers; 1,000,000s VMs

A lot of data to measure per device

100K-1M concurrent flows per switch

10Gbps link per host

Challenge 3: limited hardware support

Limited resources, lack of programmability

Switches
- Small, on chip memory
- Control functions (forwarding, firewalls, ...)
- Monitoring

Hosts
- CPU
- Real apps
- Monitoring

Limited packet processing time
- 12 ns per packet on 40G port*
- 70 ns per packet on 10G link ^

State of the art: problems

Too much information: Hard to find needles in a haystack

Network analytics on the collector

Lack network-wide view:
Ad-hoc integration of data across switches & hosts

Too little information: aggregated, sampled data is not enough
How do we make it better?

Programmable abstractions

Network analytics on the collector

Algorithmic and system design

Programmable hardware support

Quantitative Network Monitoring with NetQRE
Yifei Yuan (University of Pennsylvania), Dong Lin (LinkedIn Inc.), and Ankit Mishra, Sajal Harwaha, Rajeev Alur, and Boon Thau Loo (University of Pennsylvania)

SketchVisor: Robust Network Measurement for Software Packet Processing
Qun Huang (Huawei Future Network Theory Lab), Xin Jin (Johns Hopkins University), Patrick C. Lee (The Chinese University of Hong Kong), Runhui Li (Huawei Future Network Theory Lab), Lu Tang (The Chinese University of Hong Kong), and Yi-Chao Chen and Gong Zhang (Huawei Future Network Theory Lab)

Constant Time Updates in Hierarchical Heavy Hitters
Ran Ben Basat (Technion), Gil Einziger (Nokia Bell Labs), Roy Friedman (Technion), Marcelo Cagniani Luizelli (Federal University of Rio Grande do Sul), and Erez Waisbard (Nokia Bell Labs)

Language-directed hardware design for network performance monitoring
Srinivas Narayana, Anirudh Sivaraman, Vikram Nathan, and Prateesh Goyal (MIT CSAIL), Venkat Arun (IIT Guwahati), Mohammad Alizadeh (MIT CSAIL), Vimalkumar Jeyakumar (Cisco Tetration Analytics), and Changhoon Kim (Barefoot Networks)
Paper preview: programmable abstraction

Quantitative Network Monitoring with NetQRE
Yifei Yuan (University of Pennsylvania), Dong Lin (LinkedIn Inc.), and Ankit Mishra, Sajal Marwaha, Rajeev Alur, and Boon Thau Loo (University of Pennsylvania)

- Background: existing languages provide low-level abstractions, hard to monitor application-level/session-level events: e.g. monitor VoIP usage exceeding a quota

- Challenges: flow level measurement fails to capture application semantics, hard to generalize or customize

- Core idea: NetQRE language built on top of quantitative regular expressions, compile to efficient implementation with low memory footprint
Language-directed hardware design for network performance monitoring
Srinivas Narayana, Anirudh Sivaraman, Vikram Nathan, and Prateesh Goyal (MIT CSAIL), Venkat Arun (IIT Guwahati), Mohammad Alizadeh (MIT CSAIL), Vimalkumar Jeyakumar (Cisco Tetration Analytics), and Changhoon Kim (Barefoot Networks)

- Background: existing switch monitoring primitives (sampling, mirroring, counting) are restrictive; new technologies (in-band network telemetry, Tetration) lack flexibility
- Challenges: adding fixed function switch monitoring is unsustainable, need hardware to support expressive language
- Core idea: define Marple language, compiler to translate to programmable switches, use aggregation and filtering to improve performance
• Background: **Sketch** summarizes traffic stats of all packets with fixed-size memory, with low errors.

• Challenges: Using sketches have overhead, fail to keep up with high speed

• Core idea: use normal path and fast path in data plane, use control plane to recover from the error
Paper preview: efficient algorithm

• Background: Hierarchical Heavy Hitters (HHH) aggregate of flows in a address block, used for detecting prefixes suddenly responsible for large traffic.

• Challenges: HHH update time is long, cannot keep up with line speed

• Core idea: randomized update with O(1) complexity

Constant Time Updates in Hierarchical Heavy Hitters
Ran Ben Basat (Technion), Gil Einziger (Nokia Bell Labs), Roy Friedman (Technion), Marcelo Caggiani Luizelli (Federal University of Rio Grande do Sul), and Erez Waisbard (Nokia Bell Labs)

Figure 1: A high level overview of this work. Previous algorithms' update requires \( \Omega(H) \) run time, while we perform at most a single \( O(1) \) update.
Summary

- Network monitoring is an important topic
- Traditionally an afterthought
- It requires coordination in multiple layers

Tuesday 4:15 - 5:55!