Resource/Accuracy Tradeoffs in Software-Defined Measurement

Presenter: Kyriakos Zarifis

Masoud Moshref, Minlan Yu, Ramesh Govindan
Management policies need measurement

Traffic Engineering:
- Find elephant flows to route [Hedera]
- Estimate rack-to-rack traffic matrix [MicroTE]

Accounting:
Tiered pricing based on network usage

Troubleshooting:
- Find network bottlenecks of an application
- Incast problem
Software Defined Measurement

1. (Re)Configure resources
2. Fetch statistics

Controller
Traffic Engineering

Motivation  Tradeoff  Case study  Discussion
Challenges

Limited resources
- Limited CPU/memory in switches
- Limited control network bandwidth

Complexity of different primitives in switches
- Counting by prefix matching (TCAM)
- Counting based on hashes
- Programming

Different measurements
- Resource usage depends on measurement
Using Different Primitives for Measurement

Comparing primitives in performing a measurement task

Example: Finding heavy hitter IP when traffic from 10.0.1.0/24 goes beyond a threshold
Motivation

- Large time-scale measurement
- Measure-install loop latency
- Controller link bandwidth
- Slowly varying traffic

Discussion

Counting TCAM matches

<table>
<thead>
<tr>
<th>Filters</th>
<th>Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0 * * * *</td>
<td>s 18</td>
</tr>
</tbody>
</table>

- Monitor 10.0.1.0/24
- When goes beyond threshold, iteratively search (e.g., binary search)

- Large time-scale measurement
- Measure-install loop latency
- Controller link bandwidth
- Slowly varying traffic

Tradeoff

Experiment

- Fetch
- Install

Case study

0 1 1 0 * * * *
Sketches using hash based counters

Src: 10.0.1.5, Size: 2

- Uses cheaper resources (SRAM)
- Not dependent on traffic history
  - Use for varying traffic
  - Still history may help in large time scale to optimize parameters
Programming switches

Use more complex data structures.commands:
- Heap for finding large flows

Uses CPU resources
- Can it keep up with line rate?
Make sure controller can still do the global task
## Summary of tradeoffs of primitives

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Prefix matching</th>
<th>Hash-based counting</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory usage</td>
<td>TCAM</td>
<td>SRAM</td>
<td>SRAM</td>
</tr>
<tr>
<td>Bandwidth usage</td>
<td>Medium</td>
<td>Large</td>
<td>Small</td>
</tr>
<tr>
<td>Time-scale</td>
<td>Large</td>
<td>Small</td>
<td>Small</td>
</tr>
<tr>
<td>Traffic history sensitivity</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Concerns</td>
<td><strong>Accuracy vs Resource</strong></td>
<td><strong>Accuracy vs Resource</strong></td>
<td>Global tasks, Speed</td>
</tr>
</tbody>
</table>
Case Study: Hierarchical Heavy Hitter Detection
Hierarchical Heavy Hitters

Motivation

Case study

Discussion

11
Hierarchical Heavy Hitters

- Longest IP prefixes
- Contribute a large amount of traffic
- After excluding any HHH descendants in the prefix tree
Algorithms for single switch HHH detection

- **Prefix-based counting**
  - Proposed an iterative algorithm at controller
  - Given TCAM capacity, monitor prefixes that maximize accuracy

- **Hash-based counting**
  - Use Hierarchical Count-Min sketch
  - Approximate traffic in all levels using sketches
  - Efficiently traverse IP tree at controller to find HHHs
Evaluation

Max-Cover saves bandwidth for large number of counters and large thresholds.

Sketches have better accuracy for small threshold (longer prefixes, higher variability).

Max-Cover saves bandwidth for large number of counters and large thresholds.

Normalize switch cost
80 SRAM ≡ 1 TCAM
Future work

- Use joint information
- Guarantee accuracy

- Select primitive
- Assign switch resources
- Run global measurement

Controller

Traffic Engineering
Accounting
Anomaly Detection

Software Defined Measurement

North-bound
South-bound