CSE 534
Passive Network Measurement

Vyas Sekar
Stony Brook University

With content from Cristian Estan
Internet Measurement

• Measure certain phenomena about the network

• Critical part of networking research

• Difficult things to do correctly
  – Bias, Reproducibility
Types of Measurements

**Active**
- traceroute
- ping
- UDP probes
- TCP probes
- Application-level “probes”
  - Web downloads
  - DNS queries

**Passive**
- Packet traces
  - Complete
  - Headers only
  - Specific protocols
- Flow records
- Specific data
  - Syslogs...
  - HTTP server traces
  - DHCP logs
  - Wireless association logs
  - DNSBL lookups
  - ...
- Routing data
  - BGP updates / tables, ISIS, etc.
Assigned reading

• New directions in traffic measurement
  – Estan, Varghese

• Automatically inferring patterns
  – Estan, Savage, Varghese

• Like?
Many Monitoring Applications

- Traffic Engineering
- Accounting
- Worm Detection
- Network Forensics
- Analyze new user apps
- Botnet analysis
- Anomaly Detection
- ......
Need to estimate different metrics

“Flow size distribution”
Traffic Engineering

“Heavy-hitters”
Accounting

“SuperSpreaders”
Worm Detection

Analyze new user apps

Botnet analysis

“Degree histogram”

“Entropy”, “Changes”

Network Forensics

........
How are these metrics estimated?

Traffic

Packet Processing

Counter Data Structures

Application-Level Metrics

Monitoring (on router)

Computation (off router)
What do we ideally want?

- Traffic
- Packet Processing
- Simple
- Support many applications
- Counter Data Structures
- High accuracy
- Application-Specific Metrics
- Monitoring (on router)
- Computation (off router)
Today’s solution: Packet Sampling

Sample packets uniformly

| FlowId | Pkt/Byte Counts |

Flow = Packets with same Src/Dst Addr and Ports

Compute metrics on sampled flows

Packet Processing

Counter Data Structures

Application-Level Metrics

Traffic

Monitoring (on router)

Computation (off router)

Estimation is inaccurate for fine-grained analysis
Extensive literature on limitations for many tasks!
Today’s solution: Packet Sampling

Sample packets at random, aggregate into flows

Flow = Packets with same pattern
Source and Destination Address and Ports

Flow reports

Not good for fine-grained analysis
Extensive literature on limitations for many tasks!
Flow Sampling

Hash(5-tuple)
If hash < r, update

Flow = Packets with same Src/Dst Addr and Ports

Pick flows at random; not biased by flow size
Good for “communication” patterns
Flow sampling

Pick flows at random; not biased by flow size
Good for “communication” patterns
Why is measuring streams hard?

• Cheap memories (DRAM) are too slow to count all packets
• Fast memories (SRAM) are too small to keep counters for all streams
• Opportunity: elephants matter, mice don’t
• Problem: usually we don’t know in advance which streams are large
Problem definition

• Given a fixed definition for streams, measure large streams accurately
  – Large = above 1% of link capacity over a 1 minute interval

• Assumptions
  – Mice don’t matter
  – Accuracy of results important
Sample and Hold

If flow in table, update Sample with prob p
If new, create entry

Accurate counts of large flows
Good for “volume” queries
Sample and Hold

**Algorithm**
If flow is already logged → update
Sample packet with probability $p$
If new flow → create counter

Accurate counts of large flows
Good for “volume” queries
Why is sample & hold better?

Sample and hold

Ordinary sampling

uncertainty

uncertainty

uncertainty
How much better is it?

- Comparing the relative error of the estimate for a stream at $1/F$ of the link bandwidth
- Memory limited to $M$ entries

<table>
<thead>
<tr>
<th>Measure</th>
<th>Ordinary sampling</th>
<th>Sample and hold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>$\sqrt{F/M}$</td>
<td>$F/M$</td>
</tr>
<tr>
<td>Memory accesses</td>
<td>$1/S$</td>
<td>$1$</td>
</tr>
</tbody>
</table>
Multistage filters

Characteristics:
• No large stream is ever omitted
• Very few entries are used by small streams
• Better performance but implementation and tuning is more complex
How do multistage filters work?

Array of counters

Hash(Pink)

stream memory
How do multistage filters work?

Array of counters

Hash(Green)

stream memory
How do multistage filters work?

Array of counters

Hash(Green)

stream memory
How do multistage filters work?
How do multistage filters work?

Collisions are OK
How do multistage filters work?

Reached threshold

Insert

stream memory

stream1 1
How do multistage filters work?

stream memory

stream1 1
How do multistage filters work?

stream memory

- stream1  1
- stream2  1
How do multistage filters work?

Stage 1

Stage 2

stream memory

stream1   1
Automatically Inferring ..
Looking at traffic aggregates

- Aggregating on individual packet header fields gives useful results but:
  - Traffic reports are not always at the right granularity (e.g., individual IP address, subnet, etc.)
  - Cannot show aggregates defined over multiple fields (e.g., which network uses which application)
- The traffic analysis tool should automatically find aggregates over the right granularity

<table>
<thead>
<tr>
<th>Rank</th>
<th>Destination IP</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>jeff.dorm.bigU.edu</td>
<td>11.9%</td>
</tr>
<tr>
<td>2</td>
<td>tracy.dorm.bigU.edu</td>
<td>3.12%</td>
</tr>
<tr>
<td>3</td>
<td>risc.cs.bigU.edu</td>
<td>2.83%</td>
</tr>
</tbody>
</table>

Most traffic goes to the dorms …

<table>
<thead>
<tr>
<th>Rank</th>
<th>Source port</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Web</td>
<td>42.1%</td>
</tr>
<tr>
<td>2</td>
<td>Kazaa</td>
<td>6.7%</td>
</tr>
<tr>
<td>3</td>
<td>Ssh</td>
<td>6.3%</td>
</tr>
</tbody>
</table>

Which network uses web and which one Kazaa?

<table>
<thead>
<tr>
<th>Rank</th>
<th>Destination network</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>library.bigU.edu</td>
<td>27.5%</td>
</tr>
<tr>
<td>2</td>
<td>cs.bigU.edu</td>
<td>18.1%</td>
</tr>
<tr>
<td>3</td>
<td>dorm.bigU.edu</td>
<td>17.8%</td>
</tr>
</tbody>
</table>

What apps are used?

<table>
<thead>
<tr>
<th>Rank</th>
<th>Source net</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>library</td>
<td>36.6%</td>
</tr>
<tr>
<td>2</td>
<td>cs</td>
<td>28.7%</td>
</tr>
<tr>
<td>3</td>
<td>dorm</td>
<td>20.9%</td>
</tr>
</tbody>
</table>
Ideal traffic report

<table>
<thead>
<tr>
<th>Traffic aggregate</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web traffic</td>
<td>42.1%</td>
</tr>
<tr>
<td>Web traffic to library.bigU.edu</td>
<td>26.7%</td>
</tr>
<tr>
<td>Web traffic from <a href="http://www.schwarzenegger.com">www.schwarzenegger.com</a></td>
<td>13.4%</td>
</tr>
<tr>
<td>ICMP traffic from sloppynet.badU.edu to jeff.dorm.bigU.edu</td>
<td>11.9%</td>
</tr>
</tbody>
</table>

This paper is about giving the network administrator insightful traffic reports.

This is a Denial of Service attack!!
Definition: traffic clusters

- **Traffic clusters** are the **multidimensional** traffic aggregates identified by our reports
- A cluster is defined by a range for each field
- The ranges are from natural **hierarchies** (e.g. IP prefix hierarchy) – meaningful aggregates
- Example
  - **Traffic aggregate**: incoming web traffic for CS Dept.
  - **Traffic cluster**: ( SrcIP=* , DestIP in 132.239.64.0/21, Proto=TCP, SrcPort=80, DestPort in [1024,65535] )
Definition: traffic report

- **Traffic reports** = volume of chosen traffic clusters

- Keep report size manageable
  - only clusters above **threshold** (e.g. H=total of traffic/20)

- Avoid redundant data **compress**
  - Omit clusters whose traffic can be inferred (up to error H) from non-overlapping more specific clusters

- Highlight non-obvious with **unexpectedness label**
  - 50% of all traffic is web
  - Prefix B receives 20% of all traffic
  - The web traffic received by prefix B is 15% instead of 50%*20%=10%, unexpectedness label is 15%/10%=150%
Unidimensional report example
Unidimensional report example

Compression

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.0/29</td>
<td>120</td>
</tr>
<tr>
<td>10.0.0.8/29</td>
<td>380</td>
</tr>
<tr>
<td>10.0.0.8</td>
<td>160</td>
</tr>
<tr>
<td>10.0.0.9</td>
<td>110</td>
</tr>
</tbody>
</table>
Multidimensional structure ex.

Nodes (clusters) have multiple parents
Nodes (clusters) overlap

US
CA
NY

USEU
CA
GB
DE

Web

US Web
Takeaways

- Flow measurement is a critical part of n/w mgmt

- Challenges:
  - router capacity, reporting size, useful reports

- Traditional = NetFlow (packet sampling)

- Focus on elephants for some tasks (New directions paper)
  - Sample and hold, Multi stage

- Better reporting – AutoFocus
  - Multidimensional aggregation, “Interesting”-ness

- What’s missing?
  - Fine-grained monitoring, anomaly detection, security ..