Towards Flow-based Cloud Service Monitoring

Idilio Drago

DACS - Design and Analysis of Communication Systems

March, 2012
Motivation

Outsourcing to “the cloud”:

- Key applications - e.g. cloud storage
- Need for performance/health measurement (accounting, etc.)
Motivation

**Outsourcing to “the cloud”:**
- Key applications - e.g. cloud storage
- Need for performance/health measurement (accounting, etc.)

**Does “the cloud” fail?**
- Amazon EC2 (3x) in 2011; Microsoft Azure last Feb.

**Possible approaches:**
- Active probing
- Client instrumentation
- Network-based?
Scenario
Scenario

- Transport layer
Scenario

- **Transport layer**
- Upper layer protocol flows (HTTP, DNS flows using DPI)
- Complex protocol interactions (Web applications in CDNs)
- Complex applications, no DPI (Encrypted L7 protocols)
Raw flow records - Dropbox’s control servers

No sampling

Intuitive reasoning:
- TCP is bi-directional
- Outgoing vs. incoming
Raw flow records - Dropbox’s control servers

No sampling

Intuitive reasoning:
- TCP is bi-directional
- Outgoing vs. incoming

Hypotheses:
- Server unreachable?
- Exporter problems?
- Coincidence?
Exporter problems?

- 24-hours PCAP
- YAF

Setup:

- Similar parameters
- uniflow etc.

What is a “flow”?
Post-processing flow records (revisited)

NetFlow-based TCP connection summaries:

• Sommer, R., Feldmann, A.: NetFlow: Information loss or win?

Validation Setup:

Goals:

• Establish a "ground truth"
• Check the effects of different flow export setups
• Update results
Post-processing flow records (revisited)

Original heuristic:

- Group/match records (time composition)
- Long inactive timeout (300s) - reduced in case of FINs (30s)
- Determine originator/responder and classify connections
Post-processing flow records (revisited)

Original heuristic:

- Group/match records (time composition)
- Long inactive timeout (300s) - reduced in case of FINs (30s)
- Determine originator/responder and classify connections

<table>
<thead>
<tr>
<th></th>
<th>Conns (%)</th>
<th>Packets (%)</th>
<th>P.</th>
<th>R.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>57.6</td>
<td>53.5</td>
<td>1.00</td>
<td>0.93</td>
</tr>
<tr>
<td>Aborted</td>
<td>26.6</td>
<td>27.8</td>
<td>0.88</td>
<td>0.99</td>
</tr>
<tr>
<td>Ongoing</td>
<td>1.3</td>
<td>14.4</td>
<td>0.95</td>
<td>0.66</td>
</tr>
<tr>
<td>Closing</td>
<td>0.7</td>
<td>0.4</td>
<td>0.99</td>
<td>0.95</td>
</tr>
<tr>
<td>Failed</td>
<td>13.8</td>
<td>3.9</td>
<td>0.92</td>
<td>0.76</td>
</tr>
</tbody>
</table>

- **Under-counting** failed connections
- Almost invariant to flow export parameters
Post-processing flow records (improved)

Flow record semantics:

- Start times, flags per record etc.
- How exporters handle termination by TCP flags
Post-processing flow records (improved)

**Flow record semantics:**

- Start times, flags per record etc.
- How exporters handle termination by TCP flags

<table>
<thead>
<tr>
<th></th>
<th>Orig.</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P.</td>
<td>R.</td>
<td>P.</td>
</tr>
<tr>
<td>Complete</td>
<td>1.00</td>
<td>0.93</td>
<td>1.00</td>
</tr>
<tr>
<td>Aborted</td>
<td>0.88</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>Ongoing</td>
<td>0.95</td>
<td>0.66</td>
<td>0.99</td>
</tr>
<tr>
<td>Closing</td>
<td>0.99</td>
<td>0.95</td>
<td>0.99</td>
</tr>
<tr>
<td>Failed</td>
<td>0.92</td>
<td>0.76</td>
<td>0.99</td>
</tr>
</tbody>
</table>

- Fast reuse of sockets, “spontaneous” RST after FINs+ACKs etc.
Example: Measuring unhealthy TCP traffic

(a) Dropbox’s control servers

(b) All traffic at a UT building
What if packet-sampling is applied?

Example:

- No sampling
- Facebook (dynamic)
- Short “local” outage
What if packet-sampling is applied?

**Example:**
- No sampling
- Facebook (dynamic)
- Short “local” outage

**Reattempts:**
- Originators
- New socket
- SYN retransmissions
What if packet-sampling is applied?

Assuming:
- 1 SYN from orig.
- 0/1 SYN from resp.
- Random sampling

Estimate:
- Max. likelihood
- Confidence intervals

Over-estimating
Handling SYN retransmissions

New counter:
- \textit{tcpSynTotalCount}
- Bytes/packets
- No other flags

Accuracy:
- Sampling probability
Summary

Raw flow records (sampled/non-sampled) might be misleading
Summary

Raw flow records (sampled/non-sampled) might be misleading

Simple time composition of non-sampled flow records:

- Under-count failed connections
- Discard information
Summary

Raw flow records (sampled/non-sampled) might be misleading

Simple time composition of non-sampled flow records:
  - Under-count failed connections
  - Discard information

Inversion of packet-sampled flows assuming 1-SYN per connection:
  - Over-estimate failed connections
Thank you

Questions?