Traffic classification under sampling

Dario Rossi
dario.rossi@enst.fr
http://www.enst.fr/~drossi

Joint work with
Silvio Valenti
Davide Tammaro
Antonio Pescape’

4th NMRG workshop, IETF83, 31 March 2012
Agenda

• Traffic classification taxonomy
• Sampling
• Methodology
  – Dataset, tools, workflow, metrics, etc.
  – Sampling strategies
• Experimental results
  – Feature distortion
  – Classification accuracy
• Conclusions
• Advertisement
• Further advertisement
  – if time allows and audience interested :)


Traffic classification

• Problem
  – Look at packets in the network and guess which application has generated them

• Applications
  – Intrusion Detection System
  – Quality of Service
  – Lawful interception

• Challenges
  – Applications try to cheat (well-known ports no longer reliable),
  – Applications evolve (proprietary protocols, encryption...)
  – Lightweight to keep up with modern network speed
## Traffic classification taxonomy

<table>
<thead>
<tr>
<th>Approach</th>
<th>Subcategory</th>
<th>Granularity</th>
<th>Timeliness</th>
<th>Complexity</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload Based</td>
<td>Deep Packet Inspection (DPI) [1,2]</td>
<td>Fine-grained individual applications</td>
<td>Early (first few packets)</td>
<td>Access to packet payload of first few packets.</td>
<td>Deterministic technique;</td>
</tr>
<tr>
<td></td>
<td>KISS[ToN’10]</td>
<td>Fine-grained individual applications</td>
<td>Online (100s packets windows)</td>
<td>Access to packet payload of several packets.</td>
<td>Robust technique</td>
</tr>
<tr>
<td></td>
<td>Stochastic Packet Inspection</td>
<td></td>
<td></td>
<td>High cost</td>
<td></td>
</tr>
<tr>
<td>Statistical</td>
<td>[4,5,6,7]</td>
<td>Coarse-grained, class of application</td>
<td>Late (after the flow end)</td>
<td>Access to flow-level information</td>
<td>Post-mortem analysis</td>
</tr>
<tr>
<td>Analysis</td>
<td></td>
<td></td>
<td></td>
<td>Lightweight cost</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[8,9]</td>
<td>Fine-grained individual applications</td>
<td>Early (first 5 packets)</td>
<td>Access to first few packets</td>
<td>On the fly classification</td>
</tr>
<tr>
<td>Behavioral</td>
<td>[10,11]</td>
<td>Coarse-grained, class of application</td>
<td>Late (after the flow end)</td>
<td>Lightweight</td>
<td>Post-mortem analysis</td>
</tr>
<tr>
<td>Analysis</td>
<td>Abacus [ComNet’11]</td>
<td>Fine-grained, individual P2P applications</td>
<td>Online (1s-5s seconds windows)</td>
<td>Lightweight</td>
<td>Online classification Limited to P2P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Traffic classification taxonomy (refs)

• Why sample Internet traffic?
  – To reduce computation & storage
• How much information do we loose?
  – Monitoring [TC22]
  – Classification [JNM’12, TRAC’11]
• In the reminder of this talk [JNM’12]
  – (see advertisement)
## Dataset

Table II. Subset of the dataset used for classification, and application breakdown.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>UniBS</th>
<th></th>
<th>Campus</th>
<th></th>
<th>Auckland</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Flow</td>
<td>Byte</td>
<td>Flow</td>
<td>Byte</td>
<td>Flow</td>
<td>Byte</td>
</tr>
<tr>
<td>HTTP</td>
<td>49.3</td>
<td>5.6</td>
<td>41.8</td>
<td>62.7</td>
<td>34.8</td>
<td>25.3</td>
</tr>
<tr>
<td>HTTPS</td>
<td>1.5</td>
<td>1.2</td>
<td>41.8</td>
<td>30.6</td>
<td>34.8</td>
<td>23.4</td>
</tr>
<tr>
<td>FTP</td>
<td>-</td>
<td>-</td>
<td>4.8</td>
<td>0.03</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IMAPS</td>
<td>3.7</td>
<td>0.1</td>
<td>0.2</td>
<td>3.9</td>
<td>0.6</td>
<td>0.9</td>
</tr>
<tr>
<td>POP3</td>
<td>1</td>
<td>0.01</td>
<td>-</td>
<td>-</td>
<td>5.6</td>
<td>2.8</td>
</tr>
<tr>
<td>SMTP</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>23.9</td>
<td>47.5</td>
</tr>
<tr>
<td>Skype</td>
<td>1</td>
<td>0.7</td>
<td>11.1</td>
<td>2.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>eDonkey</td>
<td>40.1</td>
<td>87.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BitTorrent</td>
<td>3.3</td>
<td>5.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IP's</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Available at</td>
<td>410K</td>
<td>61K</td>
<td>81K</td>
<td>6.59K</td>
<td></td>
</tr>
<tr>
<td>Ground truth</td>
<td>Port-based</td>
<td>DPI [32]</td>
<td>gt [33]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Sampling**

**Classification**

SYN-sampling  
= Systematic + SYN set

SYN needed to have a recall for all flows!

**Monitoring**

Systematic

Stratified sampling

Random sampling

\[ k = \text{sampling period and } p = \frac{1}{k} \]
Features

Tstat is a L4 traffic analyzer, produces >255 packet/flow level features
Unsampled Classification in Tstat

Steep increase of BitTorrent UDP...

...Juventus lost the match!

ISP1

ISP5
Analysis/Classification

• Traffic monitoring
  – Distance of feature distribution for aggregate [ITC22]
    • Hellinger Distance, Kullback-Leibler, Fleiss Chi-square
  – Distortion of individual flow features [IJNM’12]
    • Relative error, correlation coefficient
    • Instrumental for traffic classification

• Traffic classification
  – C45 trees (in this talk) and SVM
  – Accuracy w.r.t ground-truth
Figure 3. Example of distortion of per-flow features (Campus dataset): scatter plot of TCP source port (a) and average packet size (b) for unsampled vs sampled traffic, along with statistical indexes of correlation.
Feature distortion (2/2)

Figure 8. CDF of (left) Err\% and (right) $\rho$ for UniBS trace and different sampling step.
Expected impact on classification?

Some samples can still discriminate

Most samples can’t
(pkt size==0 => only SYN seen)

Expect poor performance, unless... any guess?

Figure 9. Scatter plot of features values for unsampled and SYN Sampling $k = 100$ for UniBS trace, contrasting peer-to-peer (P2P) and traditional client-server (CS) applications.
Features at different sampling step K

Single packet feature
Multiple packets features

Most relevant
Least relevant

Figure 10. (a) Parallel coordinates plot for most-relevant features and (b) scatter plot of information gain for all features with \( k = 2, 10 \).
# Most relevant features at different K

Table V. Feature Information gain for UniBS trace at different sampling rates.

<table>
<thead>
<tr>
<th>Features</th>
<th>Unsampler Score</th>
<th>Unsampler Rank</th>
<th>Sampled k=2 Score</th>
<th>Sampled k=2 Rank</th>
<th>Sampled k=10 Score</th>
<th>Sampled k=10 Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server-IP-address</td>
<td>1.68</td>
<td>1</td>
<td>1.68</td>
<td>1</td>
<td>1.68</td>
<td>1</td>
</tr>
<tr>
<td>cwin-min-c2s</td>
<td>1.49</td>
<td>2</td>
<td>1.20</td>
<td>6</td>
<td>0.60</td>
<td>14</td>
</tr>
<tr>
<td>min-seg-size-c2s</td>
<td>1.48</td>
<td>3</td>
<td>1.22</td>
<td>5</td>
<td>0.47</td>
<td>23</td>
</tr>
<tr>
<td>cwin-max-c2s</td>
<td>1.47</td>
<td>4</td>
<td>1.11</td>
<td>8</td>
<td>0.56</td>
<td>15</td>
</tr>
<tr>
<td>max-seg-size-c2s</td>
<td>1.43</td>
<td>5</td>
<td>1.17</td>
<td>7</td>
<td>0.46</td>
<td>24</td>
</tr>
<tr>
<td>initial-cwin-c2s</td>
<td>1.41</td>
<td>6</td>
<td>0.71</td>
<td>26</td>
<td>0.29</td>
<td>32</td>
</tr>
<tr>
<td>First-time</td>
<td>1.37</td>
<td>7</td>
<td>1.37</td>
<td>2</td>
<td>1.37</td>
<td>2</td>
</tr>
<tr>
<td>cwin-min-s2c</td>
<td>1.35</td>
<td>8</td>
<td>1.06</td>
<td>11</td>
<td>0.53</td>
<td>16</td>
</tr>
<tr>
<td>Server-TCP-port</td>
<td>1.34</td>
<td>9</td>
<td>1.34</td>
<td>3</td>
<td>1.34</td>
<td>3</td>
</tr>
<tr>
<td>initial-cwin-s2c</td>
<td>1.33</td>
<td>10</td>
<td>0.77</td>
<td>22</td>
<td>0.30</td>
<td>31</td>
</tr>
<tr>
<td>Client-IP-address</td>
<td>1.31</td>
<td>11</td>
<td>1.31</td>
<td>4</td>
<td>1.31</td>
<td>4</td>
</tr>
<tr>
<td>cwin-max-s2c</td>
<td>1.28</td>
<td>12</td>
<td>0.99</td>
<td>14</td>
<td>0.49</td>
<td>21</td>
</tr>
<tr>
<td>min-seg-size-s2c</td>
<td>1.22</td>
<td>13</td>
<td>0.96</td>
<td>16</td>
<td>0.51</td>
<td>19</td>
</tr>
<tr>
<td>max-seg-size-s2c</td>
<td>1.21</td>
<td>14</td>
<td>1.03</td>
<td>12</td>
<td>0.50</td>
<td>20</td>
</tr>
<tr>
<td>Last-time</td>
<td>1.14</td>
<td>15</td>
<td>1.09</td>
<td>9</td>
<td>1.02</td>
<td>5</td>
</tr>
<tr>
<td>win-max-s2c</td>
<td>1.08</td>
<td>16</td>
<td>1.07</td>
<td>10</td>
<td>0.98</td>
<td>6</td>
</tr>
<tr>
<td>Completion-time</td>
<td>1.03</td>
<td>17</td>
<td>0.97</td>
<td>15</td>
<td>0.42</td>
<td>25</td>
</tr>
<tr>
<td>win-min-s2c</td>
<td>1.02</td>
<td>18</td>
<td>1.01</td>
<td>13</td>
<td>0.94</td>
<td>7</td>
</tr>
<tr>
<td>unique-byte-s2c</td>
<td>1.02</td>
<td>19</td>
<td>0.74</td>
<td>23</td>
<td>0.42</td>
<td>27</td>
</tr>
<tr>
<td>data-byte-s2c</td>
<td>1.01</td>
<td>20</td>
<td>0.74</td>
<td>24</td>
<td>0.42</td>
<td>26</td>
</tr>
</tbody>
</table>
Training policy matters

- Classification accuracy remains if we train and classify with the same sampling step!

Figure 12. Impact of Homogeneous vs Heterogeneous training set policies at varying sampling rates in terms of flow and byte accuracy.
Conclusions

• Traffic sampling
  – Heavy distortion on aggregated features
    Details in [ITC22]
  – Heavy distortion on individual features
    Details in [IJNM’12]
  – Classification still accurate if training and testing
    at homogeneous sampling rates [IJNM’12]
    • Distorted features preserve distance in some non
      geometric space (e.g., gaussian SVM, or information
      gain amount for C45 trees)
Advertisement: selected publications

Classification


Classification & Sampling


- **[ITC’22]** Pescape, D. Rossi, D. Tammaro and S. Valenti, *On the impact of sampling on traffic monitoring and analysis*. In ITC22, Amsterdam, The Netherlands, September 7 - 9 2010

Further advertisement
## Traffic Classification Taxonomy

<table>
<thead>
<tr>
<th>Approach</th>
<th>Subcategory</th>
<th>Granularity</th>
<th>Timeliness</th>
<th>Complexity</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload Based</td>
<td>[1,2] Deep Packet Inspection (DPI)</td>
<td>Fine-grained individual applications</td>
<td>Early (first few packets).</td>
<td>Access to packet payload of first few packets.</td>
<td>Deterministic technique;</td>
</tr>
<tr>
<td></td>
<td><strong>KISS</strong>[Ton’10] Stochastic Packet Inspection</td>
<td>Fine-grained individual applications</td>
<td>Online (100s packets windows)</td>
<td>Access to packet payload of several packets.</td>
<td>Robust technique</td>
</tr>
<tr>
<td>Statistical Analysis</td>
<td>[4,5,6,7]</td>
<td>Coarse-grained, class of application</td>
<td>Late (after the flow end).</td>
<td>Access to flow-level information.</td>
<td>Post-mortem analysis</td>
</tr>
<tr>
<td></td>
<td>[8,9]</td>
<td>Fine-grained individual applications</td>
<td>Early (first 5 packets)</td>
<td>Access to first few packets.</td>
<td>On the fly classification</td>
</tr>
<tr>
<td>Behavioral Analysis</td>
<td>[10,11]</td>
<td>Coarse-grained, class of application</td>
<td>Late (after the flow end).</td>
<td>Lightweight</td>
<td>Post-mortem analysis</td>
</tr>
<tr>
<td></td>
<td><strong>Abacus</strong>[ComNet’11]</td>
<td>Fine-grained, individual P2P applications</td>
<td>Online (1s-5s seconds windows)</td>
<td>Lightweight</td>
<td>Online classification limitations for P2P</td>
</tr>
</tbody>
</table>
Overview

Deep Packet Inspection (DPI)
- Specific Keyword
  - GET
  - MAIL FROM:
  - BT

Stochastic Packet Inspection (KISS)
- Application syntax

Behavior analysis (Abacus)
- Algorithm design
KISS: Stochastic packet inspection

Header syntax is fixed, binary alphabet

1) Extract the first N bytes of the payload from a window of W consecutive packets
2) Divide each byte in 2 chunks of 4 bits
3) Collect the frequency distribution Oi of the values assumed by each chunk
4) Compare the distribution to a uniform distribution Ei=\(\frac{1}{2^4}\) with a \(\chi^2\)-like test

\[
X_g = \sum_{i=1}^{2^b} \frac{(O^i_g - E^i_g)^2}{E^i_g} \sim \chi^2_{g, N/2}
\]

measure the randomness of each chunk

KISS signature: \([X_1, X_2, \ldots, X_{2N}]\) over W pkts

counters

C|D = 3 bit fixed random deterministic

Y1 pkt1: cb d2 ... 02 60
Y1 pkt2: cc d5 ... 02 08
Y2 pkt1: 01 da ... 02 65
Y1 pkt3: cd c0 ... 02 d9
Y2 pkt2: 02 c1 ... 02 5c
Y2 pkt3: 03 dc ... 02 11
Y1 pkt4: ce cb ... 02 28
Y1 pkt5: cf d1 ... 02 8a
Y1 pkt6: d0 ca ... 02 3a
Y2 pkt4: 04 c2 ... 02 b7
Abacus: Behavioral signatures

Applications implement different activities (signaling, data chunks) and tuning (chunk size)

1) Count the number of packets/bytes received in a fixed time window $\Delta T$

2) Count the number of hosts sending a given number of packets/bytes (exponential binning)

3) Normalize the packet/bytewise counts to gather two probability mass functions

Example using packets

Distribution = [1, 1, 3, 0]
Signature   = [0.2, 0.2, 0.6]
Oops!

• Sorry, wrong key