Transport Layer Identification of P2P Traffic
by T. Karagiannis, A. Broido, M. Faloutsos, and k. claffy

Hadrien Croubois

Computer Science Department of the ENS de Lyon

May 14, 2012
1 Introduction

2 Data Description

3 Payload Method

4 Non Payload Method

5 Conclusion
Transport Layer Identification of P2P Traffic
1. Introduction

2. Data Description

3. Payload Method

4. Non Payload Method

5. Conclusion
Data captured at an OC-48 link of a Tier 1 US ISP connection (2,48 Gbits/s)
- Data captured at an OC-48 link of a Tier 1 US ISP connection (2.48 Gbits/s)
Data captured at an OC-48 link of a Tier 1 US ISP connection (2.48 Gbits/s)

- 4 datasets, from May 2003 to April 2004 (60-122 minutes each)
Datasets description

D09 - D10: 44 bytes for each packet
Datasets description

**D09 - D10**: 44 bytes for each packet
- IP & TCP/UDP headers
Datasets description

D09 - D10: 44 bytes for each packet
- IP & TCP/UDP headers
- 4 bytes of payload
Datasets description

**D09 - D10**: 44 bytes for each packet
- IP & TCP/UDP headers
- 4 bytes of payload

**D11 - D13**: 58 bytes for each packet
Datasets description

D09 - D10: 44 bytes for each packet
- IP & TCP/UDP headers
- 4 bytes of payload

D11 - D13: 58 bytes for each packet
- 16 bytes of TCP/UDP payload
1. Introduction
2. Data Description
3. Payload Method
4. Non Payload Method
5. Conclusion
Method Description

- Identification of P2P traffic based on characteristic bit string in packet payload.
Identification of P2P traffic based on characteristic bit string in packet payload.

<table>
<thead>
<tr>
<th>P2P protocol</th>
<th>String</th>
<th>Trans. prot.</th>
<th>Def. ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>eDonkey2000</td>
<td>0xe319010000 0xc53f010000</td>
<td>TCP/UDP</td>
<td>4661-4665</td>
</tr>
<tr>
<td>Fasttrack</td>
<td>&quot;Get /.hash&quot; 0x2700000002980</td>
<td>TCP/UDP</td>
<td>1214</td>
</tr>
<tr>
<td>BitTorrent</td>
<td>&quot;0x13Bit&quot;</td>
<td>TCP</td>
<td>6881-6889</td>
</tr>
<tr>
<td>Gnutella</td>
<td>&quot;GNUT&quot;, &quot;GIV&quot; &quot;GND&quot;</td>
<td>TCP/UDP</td>
<td>6346-6347</td>
</tr>
</tbody>
</table>
M1: Check source/destination port with table
M1: Check source/destination port with table
- Port matches $\rightarrow$ Flow tagged as P2P
M1: Check source/destination port with table
   - Port matches → Flow tagged as P2P

M2: Check the payload of each packet with table
M1: Check source/destination port with table
  - Port matches → Flow tagged as P2P
M2: Check the payload of each packet with table
  - String matches → Flow tagged as P2P
**M1:** Check source/destination port with table
- Port matches → Flow tagged as P2P

**M2:** Check the payload of each packet with table
- String matches → Flow tagged as P2P
- No packet matches → Flow tagged as non-P2P
M1 : Check source/destination port with table
  - Port matches → Flow tagged as P2P

M2 : Check the payload of each packet with table
  - String matches → Flow tagged as P2P
  - No packet matches → Flow tagged as non-P2P

M3 : For P2P flow identified at step M2, record sources & destination IP
M1: Check source/destination port with table
   - Port matches → Flow tagged as P2P

M2: Check the payload of each packet with table
   - String matches → Flow tagged as P2P
   - No packet matches → Flow tagged as non-P2P

M3: For P2P flow identified at step M2, record sources & destination IP
   - For all non P2P flows that contain one of these IP
     → Flow tagged as possible-P2P
M1: Check source/destination port with table
   - Port matches → Flow tagged as P2P

M2: Check the payload of each packet with table
   - String matches → Flow tagged as P2P
   - No packet matches → Flow tagged as non-P2P

M3: For P2P flow identified at step M2, record sources & destination IP
   - For all non P2P flows that contain one of these IP
     → Flow tagged as possible-P2P

To minimize false positives, FTP, SSL, DNS & online gaming flows are excluded from M3
Limitations

- HTTP requests: P2P protocols using HTTP requests are not identified
- Encryption: encrypted payload is not identified
- Other P2P protocols: unreferenced P2P protocols are not identified
- Unidirectional trace: acknowledgement stream of a P2P download is not always visible because of asymmetric routing
Limitations

- HTTP requests: P2P protocols using HTTP requests are not identified
- Encryption: encrypted payload is not identified
Limitations

- **HTTP requests**: P2P protocols using HTTP requests are not identified.
- **Encryption**: Encrypted payload is not identified.
- **Other P2P protocols**: Unreferenced P2P protocols are not identified.
Limitations

- HTTP requests: P2P protocols using HTTP requests are not identified
- Encryption: encrypted payload is not identified
- Other P2P protocols: unreferenced P2P protocols are not identified
- Unidirectional trace: acknowledgement stream of a P2P download is not always visible because of asymmetric routing
The non payload method only examines packet headers to detect P2P flow.
The non payload method only examines packet headers to detect P2P flow.

As only \{IP, port\} pairs are the only available, two heuristics, based on the observation of P2P connection patterns, are used.
TCP/UDP IP pairs heuristic

- Most P2P protocols use both TCP and UDP protocols
TCP/UDP IP pairs heuristic

- Most P2P protocols use both TCP and UDP protocols
- Other applications using both TCP and UDP protocols are rare and use specific ports
Most P2P protocols use both TCP and UDP protocols

Other applications using both TCP and UDP protocols are rare and use specific ports

TCP/UDP IP pairs heuristic

{IP,port} using both TCP and UDP protocols (whose ports are not in the exclude list) are considered as P2P traffic
## Excluded ports for TCP/UDP IP pairs heuristic

<table>
<thead>
<tr>
<th>Ports</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>135,137,139,445</td>
<td>NETBIOS</td>
</tr>
<tr>
<td>53</td>
<td>DNS</td>
</tr>
<tr>
<td>123</td>
<td>NTP</td>
</tr>
<tr>
<td>500</td>
<td>ISAKMP</td>
</tr>
<tr>
<td>554,7070,1755,6970,5000,5001</td>
<td>streaming</td>
</tr>
<tr>
<td>7000,7514,6667</td>
<td>IRC</td>
</tr>
<tr>
<td>3531</td>
<td>p2pnetworking.exe</td>
</tr>
</tbody>
</table>
### {IP, port} pairs heuristic

- IPs for which the number of distinct connected IPs is equal to the number of distinct connected ports are considered P2P hosts.
- IPs for which the difference between connected IPs and ports is large (e.g., larger than 10) are considered non P2P hosts.
False positives

- Mail
False positives

- Mail
- DNS
False positives

- Mail
- DNS
- Gaming
False positives

- Mail
- DNS
- Gaming
- Malware
False positives

- Mail
- DNS
- Gaming
- Malware
- Other heuristics (One-packet pairs, MSN messenger server . . .)
1 Introduction

2 Data Description

3 Payload Method

4 Non Payload Method

5 Conclusion
Conclusion

- Easy to understand, efficient method
Conclusion

- Easy to understand, efficient method
- General method (not specific to some P2P protocols, unaffected by encryption)
Conclusion

- Easy to understand, efficient method
- General method (not specific to some P2P protocols, unaffected by encryption)
- Doesn’t need to look at payload
Any questions?