Network Traffic Classification

Alberto Dainotti
alberto@unina.it

Dipartimento di Informatica e Sistemistica
COMICS Research Group

Outline

• Introduction
• Motivations
• Why is it difficult
• Definitions
• State of Art
• TIE
Traffic Classification: Intro

- TC: Associating traffic flows to network applications that generate them
- Recent interest of Research & Industry
  - Ports are not reliable anymore
  - Payload-based approaches have issues
  - New applications
  - Encryption
  - No perfect solution up to today

The Net before and during last years

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#Users</td>
<td>E-commerce</td>
<td>Business</td>
<td>Users</td>
<td>Interaction</td>
<td>Music</td>
<td>Industry</td>
<td>Social</td>
<td>Networks</td>
<td>Media</td>
</tr>
<tr>
<td>Email, Web, FTP</td>
<td>Web Applications</td>
<td>Streaming</td>
<td>Games</td>
<td>P2P</td>
<td>File Sharing</td>
<td>Social</td>
<td>Networking</td>
<td>Skype</td>
<td>P2P TV</td>
</tr>
<tr>
<td>Mostly TCP/HTTP</td>
<td>Highly Predictable</td>
<td>UDP increases</td>
<td>Malware Traffic</td>
<td>More - Viruses</td>
<td>Difficulty to identify</td>
<td>Encrypted</td>
<td>Hinder to classify</td>
<td>Dominated by new gen Apps</td>
<td>Traffic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Social & Economical Impact

Applications
TC Motivations

What if we cannot classify traffic?

• We have no clue of what our links carry
  – How is people using the Internet?
  – What’s the killer application?
  – Does it really matter to model this or that?
  – Is something “strange” happening and we don’t know it?

• We cannot
  – do provisioning
  – perform resource allocation and offer QoS
  – enforce security policies (e.g. Firewalling)
  – do accounting based on typology of traffic
  – study network traffic if we cannot retrace phenomena to specific applications and protocols (e.g. congestion)

TC: Why is it difficult? (1/4)

• Traditional approach: transport-level ports
• The Internet Assigned Numbers Authority (IANA)
  – assigns the well-known ports from 0-1023
  – registers port numbers in the range from 1024-49151 to applications
  – defines ports from 49152 through 65535 as “dynamic and/or private”

• This association is not reliable anymore!
TC: Why is it difficult? (2/4)

• Ports
  – many applications have no IANA registered ports while they use numbers already registered by others
  – many applications use random ports numbers or allow users to define any port number
  – often applications are configured to use well-known ports to disguise their traffic and circumvent security and network-usage policy enforcement
  – sometimes several servers share a single IP address, thus they need to offer their services through different ports by using network (and port) address translation.

TC: Why is it difficult? (3/4)

• New applications with undisclosed proprietary protocols (e.g. Skype)
  – New applications emerge continuously and it is difficult to investigate each of them in order to update approaches and/or signatures.

• Protocol encapsulation
  – E.g. over HTTP (MSN, Kazaa, …)

• Encryption
  – Application payload
  – Application protocol encapsulation (SSL, SSH, …)
  – Network level (IPSec Tunnels, …)
TC: Why is it difficult? (4/4)

- **Link speed**
  - We often need to do classification online
  - Speed / computational complexity of algorithms
    - Payload inspection (complexity)
    - Other approaches (how much data do we need?)
  - Storage
  - Manual inspection
  - Logistics in general

- **Privacy**
  - How invading a technique is?
  - Access to full payload may be not allowed
  - Storage may be not allowed
  - Trace anonymization (issues)

TC: Definitions (1/6)

- **Classes** (detail-level of classification)
  - traffic classes (e.g. bulk, interactive, ...)
  - (application categories (e.g. chat, streaming, web, mail, file sharing, etc.)
  - applications (e.g. KaZaa, Edonkey, IMAP, POP, SMTP, ...)
  - a single application

<table>
<thead>
<tr>
<th>Classification</th>
<th>Example Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>BULK</td>
<td>ftp</td>
</tr>
<tr>
<td>DATABASE</td>
<td>postgres, sqlnet, oracle, ingex</td>
</tr>
<tr>
<td>INTERACTIVE</td>
<td>ssh, klogin, login, telnet</td>
</tr>
<tr>
<td>MAIL</td>
<td>imap, pop3, smtp</td>
</tr>
<tr>
<td>SERVICES</td>
<td>X11, dns, ident, ldap, smtp</td>
</tr>
<tr>
<td>WWW</td>
<td>www</td>
</tr>
<tr>
<td>P2P</td>
<td>KaZaa, BitTorrent, Gnutella</td>
</tr>
<tr>
<td>ATTACK</td>
<td>Internet worm and virus attacks</td>
</tr>
<tr>
<td>GAMES</td>
<td>Microsoft: Direct Play</td>
</tr>
<tr>
<td>MULTIMEDIA</td>
<td>Windows Media Player, Real</td>
</tr>
</tbody>
</table>
TC: Definitions (2/6)

• Classification Objects
  – TCP Connections
  – Flows
    • 5-tuple plus timeout
  – Bidirectional Flows (*biflows*)
    • 5-tuple, bidirectional, timeout
  – Hosts
    • Host main behavior

TC: Definitions (3/6)

• Approaches
  – Port-based: based on IANA port assignment and on common knowledge of ports typically used by applications.
  – Payload-based: inspect payload content at transport level to identify strings related to the application-level protocol (and in general to the application) matching a set of pre-defined rules.
TC: Definitions (4/6)

• Approaches (continued)
  – Flow-features-based: typically based on machine-learning classification techniques applied to features extracted from traffic flows.
    • Features: flow-level, pkt-level, … In general, they need header-only access.
    • Machine-learning approaches
      – Supervised Learning
      – Unsupervised Learning (Clustering)

TC: Definitions (5/6)

• Approaches (continued)
  – Behavioral and host-based: based on the interactions of the host under observation with the rest of the world, usually in terms of number of connections opened, ports used, and also by using mixes of the above techniques to sketch a typical profile of the host to be compared against profiles previously stored.

• Approaches can be combined!
TC: Definitions (6/6)

- **Online vs Offline**
  - Lightweight and fast
  - Hardware-based
  - Limited data

- **Ground truth**
  - Payload-based
  - Heuristics
  - Manual Inspection
  - Alternative techniques requiring user collaboration

---

TC: State of Art (1/7)

- **Port-based**
  - Perform poorly
    - e.g. year 2005: between 50% and 70% accuracy in classifying flows
    - Recent experiments (year 2008): around 20%
  - The fastest and simplest
  - Still used
    - E.g. continuous monitoring with realtime reporting
  - Several implementations available
    - CoralReef
TC: State of Art (2/7)

• Payload-based
  – Drawbacks
    • Privacy concerns
    • Computationally heavy
    • Can be tricked
    • Constant updates (automated approaches to signature creation have been proposed)
    • Encryption
  – Plus
    • Still very reliable (used for ground-truth)
  – Implementations
    • Proprietary: Cisco NBAR, Juniper AI, …
    • Open: L7-filter (http://l7-filter.sourceforge.net), BRO, …

L7-filter Bittorrent pattern file
Flow-features based

- Drawbacks
  - Still very experimental
    - Literature is confusing: traces, objects, classes, metrics, gt, …
    - Lack of real implementations
  - Implementation

- Plus
  - Promising with respect to:
    - Encryption, obfuscation, encapsulation, etc.
    - Privacy
    - Online classification

- Implementations
  - NetAI: http://caia.swin.edu.au/urp/dstc/netai
  - Tstat 2.0: http://tstat.tlc.polito.it
  - TIE: http://tie.comics.unina.it

Flow-features based (continued)

- Some references:
TC: State of Art (6/7)

• Behavioral and host-based:
  – Exploit correlations and other information
  – Host-based approaches can work well on edge networks, not in backbones
  – Some references:

TC: State of Art (7/7)

• Identification of a single application
  – Some references on Skype identification:
An approach based on traffic modeling (1/2)

- From a Simple PDF to a more complicated, but more realistic, stochastic process
  - A HMM able to capture PS and IPT mutual and temporal dependencies

- Applied to more categories of Traffic
- Models usable for
  - Performance Evaluation
  - Traffic Generation
  - Prediction
  - Classification

An approach based on traffic modeling (2/2)

- Classify flows generated by sources (unidirectional traffic from hosts)
- Based on previous study on traffic modeling at packet level
- Overall accuracy: 91.3%
- Accuracy decreases when considering more classes

<table>
<thead>
<tr>
<th>Classification Results: Confusion Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>AoM</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>AoM</td>
</tr>
<tr>
<td>CS</td>
</tr>
<tr>
<td>Edonkey</td>
</tr>
<tr>
<td>HTTP</td>
</tr>
<tr>
<td>MSN</td>
</tr>
<tr>
<td>PPlive</td>
</tr>
<tr>
<td>SMTP</td>
</tr>
</tbody>
</table>
TIE: Traffic Identification Engine

- An open-source software platform working as a multiple classifier system
- Purpose: to allow the community to work with *shared tools and data* to investigate several aspects of traffic classification
  - Offline, Online, historical web reports
  - Easy to add: classification techniques, classification features, combination strategies
  - Well-documented API
  - Anonymized traces with ground-truth data
  - Code to the data
- *Elected reference tool for TC inside PRIN RECIPE and Cost-TMA EU projects*

http://tie.comics.unina.it

Computer Networks II – Network Traffic Classification

TIE’s Components

- Well-defined portions of code allow easy modifications and extensions
- Processing revolves around a sessions table. Each session structure in the table contains
  - Status Information
  - Flags
  - Counters
  - Features
TIE framework

- Application IDs, Sub-IDs, Groups
- Output & Input Tables
- Classification Plugin API
- Sessions
  - Flows, Biflows, Hosts, etc..
- Scripts for numerical and graphical analysis and comparison

TIE’s Classification Plugins

<table>
<thead>
<tr>
<th>Name</th>
<th>Based on</th>
<th>Status</th>
<th>Contributor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>L4 Ports</td>
<td>Available</td>
<td>UNINA (signatures from CAIDA)</td>
</tr>
<tr>
<td>L7</td>
<td>Deep Payload Inspection</td>
<td>Available</td>
<td>UNINA (signatures/code from Linux L7-filter)</td>
</tr>
<tr>
<td>NBC</td>
<td>Lightweight Payload Inspection</td>
<td>Under test</td>
<td>UNINA</td>
</tr>
<tr>
<td>GMM-PS</td>
<td>Statistical Approach: PS</td>
<td>Under test</td>
<td>UNINA</td>
</tr>
<tr>
<td>HMM</td>
<td>Statistical Approach*: PS, IPT</td>
<td>Under test</td>
<td>UNINA</td>
</tr>
<tr>
<td>FPT</td>
<td>Statistical Approach**: PS, IPT</td>
<td>Under devel.</td>
<td>UNIBS</td>
</tr>
<tr>
<td>Joint</td>
<td>Machine Learning</td>
<td>Under devel.</td>
<td>UNINA-CAIDA-CENS</td>
</tr>
<tr>
<td>???</td>
<td>???</td>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>

More research activities thanks to TIE

- Studying *deepness* of payload inspection techniques
- Comparison of classification performance and computational requirements of different approaches
- Proposing a *lightweight* payload inspection approach for online classification
- Study (and improvement) of *ground truth* state of art

Thanks for the attention

Any Questions?