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
- Multi-classification

Traffic Classification

- Need to associate flows to the applications that generate them

  - *{UDP, IP$_{SRC}$:10.0.0.1, PORT$_{SRC}$:31215, IP$_{DST}$:212.48.72.19, PORT$_{DST}$:80} → SKYPE!*

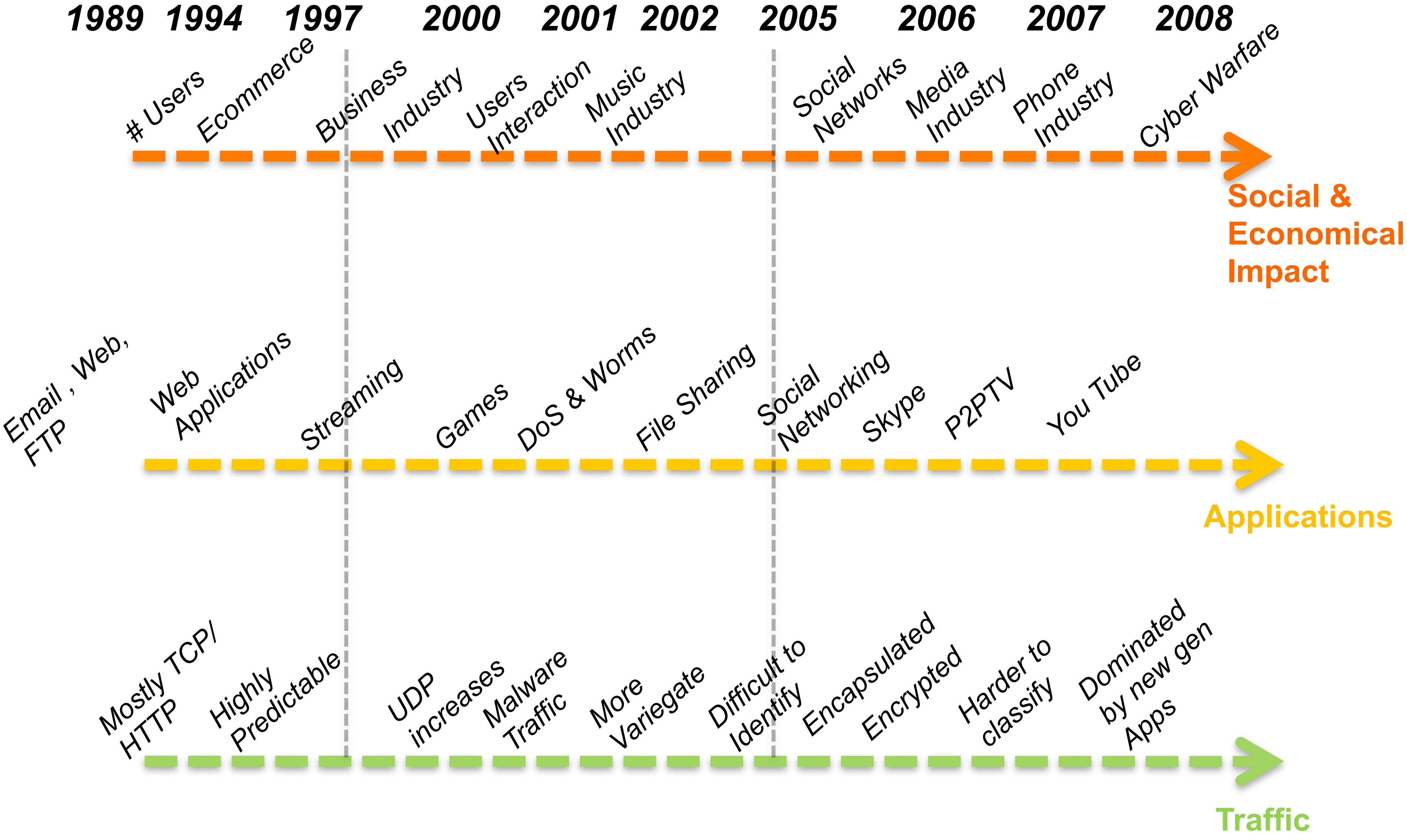
  - *{TCP, IP$_{SRC}$:10.0.0.1, PORT$_{SRC}$:2233, IP$_{DST}$: 13.29.10.199, PORT$_{DST}$:25} → SMTP!*

- Mellia et al., "Traffic classification and its applications to modern networks", Elsevier Computer Networks, Dec. 2008
- Callado et al., "A survey on internet traffic identification", IEEE Communications Surveys & Tutorials, July 2009.

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



1989 1994 1997 2000 2001 2002 2005 2006 2007 2008

Users
Ecommerce
Business
Industry
Users Interaction
Music Industry
Social Networks
Media Industry
Phone Industry
Cyber Warfare

Social & Economical Impact

Email , Web, FTP
Web Applications
Streaming
Games
DoS & Worms
File Sharing
Social Networking
Skype
P2PTV
You Tube

Applications

Mostly TCP/ HTTP
Highly Predictable
UDP increases
Malware Traffic
More Variegate
Difficult to Identify
Encapsulated
Encrypted
Harder to classify
Dominated by new gen Apps

Traffic

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)

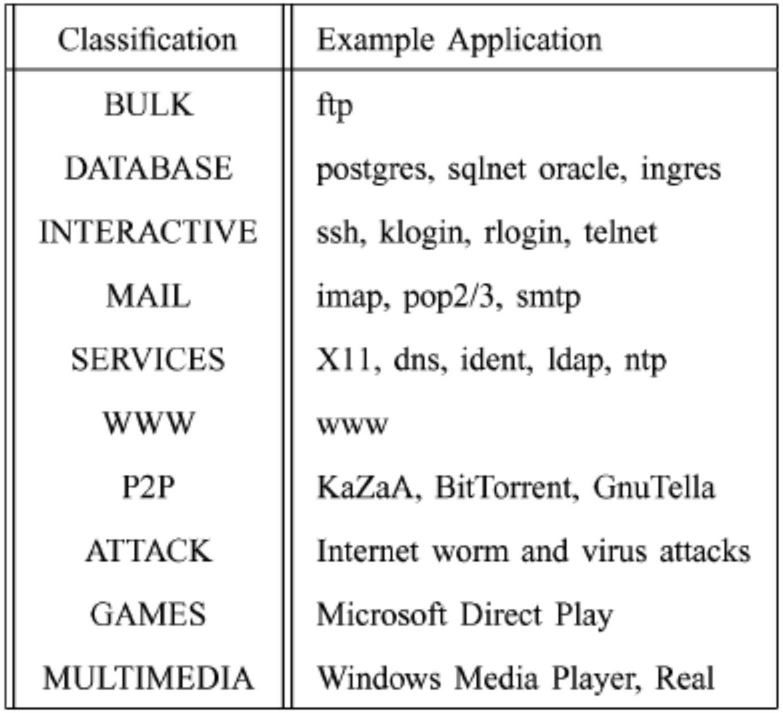
- New applications with undisclosed proprietary protocols (e.g. *Skype*)
  - New applications emerge continuously and it is difficult to investigate each of them in order to update approaches and/or signatures.
- Protocol encapsulation
  - E.g. over HTTP (*MSN, Kazaa,* …)
- Encryption
  - Application 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



Classification	Example Application
BULK	ftp
DATABASE	postgres, sqlnet oracle, ingres
INTERACTIVE	ssh, klogin, rlogin, telnet
MAIL	imap, pop2/3, smtp
SERVICES	X11, dns, ident, ldap, ntp
WWW	www
P2P	KaZaA, BitTorrent, GnuTella
ATTACK	Internet worm and virus attacks
GAMES	Microsoft Direct Play
MULTIMEDIA	Windows Media Player, Real

AULD *et al.*: BAYESIAN NEURAL NETWORKS FOR INTERNET TRAFFIC CLASSIFICATION

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

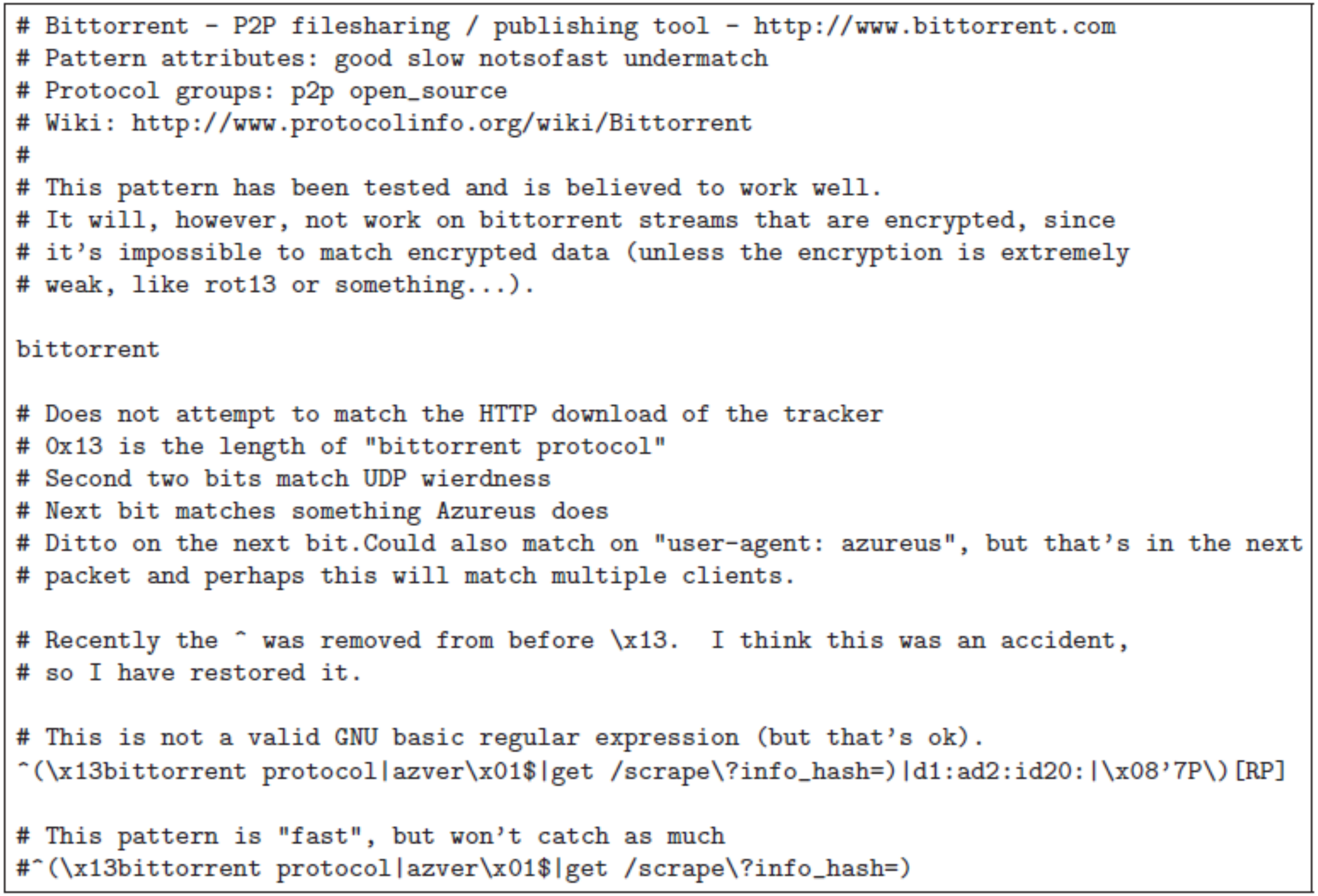
TC: State of Art (1/7)

- 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 *http://www.caida.org/tools/measurement/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, …

TC: State of Art (3/7)



```
# Bittorrent - P2P filesharing / publishing tool - http://www.bittorrent.com
# Pattern attributes: good slow notsofast undermatch
# Protocol groups: p2p open_source
# Wiki: http://www.protocolinfo.org/wiki/Bittorrent
#
# This pattern has been tested and is believed to work well.
# It will, however, not work on bittorrent streams that are encrypted, since
# it's impossible to match encrypted data (unless the encryption is extremely
# weak, like rot13 or something...).

bittorrent

# Does not attempt to match the HTTP download of the tracker
# 0x13 is the length of "bittorrent protocol"
# Second two bits match UDP wierdness
# Next bit matches something Azureus does
# Ditto on the next bit.Could also match on "user-agent: azureus", but that's in the next
# packet and perhaps this will match multiple clients.

# Recently the ^ was removed from before \x13.  I think this was an accident,
# so I have restored it.

# This is not a valid GNU basic regular expression (but that's ok).
^(\x13bittorrent protocol|azver\x01$|get /scrape\?info_hash=)|d1:ad2:id20:|\x08'7P\)[RP]

# This pattern is "fast", but won't catch as much
#^(\x13bittorrent protocol|azver\x01$|get /scrape\?info_hash=)
```

L7-filter Bittorrent pattern file

TC: State of Art (4/7)

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

TC: State of Art (5/7)

- Flow-features based (*continued*)
  - Some references:
    - Tom Auld, Andrew W. Moore, and Stephen F. Gull. Bayesian neural networks for internet traffic classification. *IEEE Transactions on Neural Networks*, 18(1):223–239, January 2007.
    - Laurent Bernaille, Renata Teixeira, and Kave Salamatian. Early application identification*. In ACM CoNEXT*, December 2006.
    - Jeffrey Erman, Anirban Mahanti, Martin Arlitt, Ira Cohen, and Carey Williamson. Offline/realtime traffic classification using semi-supervised learning. *In IFIP Performance,* October 2007.
    - A. Dainotti, W. De Donato, A. Pescapè, P. Salvo Rossi, Classification of network traffic via packet-level hidden markov models. *In IEEE GLOBECOM 2008*, December 2008.

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:
    - Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and kc claffy.Transport layer identification of p2p traffic. In ACM IMC, October 2004.
    - Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos. Blinc: Multilevel traffic classification in the dark. In ACM SIGCOMM, August 2005.

TC: State of Art (7/7)

- Identification of a single application
  - Some references on Skype identification:
    - J. Kurose D. Towsley K. Suh, D.R. Figueiredo. Characterizing and detecting skype-relayed traffic. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications*, April 2006
    - Dario Bonfiglio, Marco Mellia, Michela Meo, Dario Rossi, and Paolo Tofanelli. Revealing skype traffic: when randomness plays with you. *In ACM SIGCOMM '07*:, pages 37–48, New York, NY, USA, 2007.
    - Marcell Perenyi and Sandor Molnar. Enhanced skype traffic identification. *In ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, pages 1–9, ICST, Brussels, Belgium, Belgium, 2007
    - D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca, and D. Rossi. Tracking down skype traffic*. In INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 261–265, 2008.

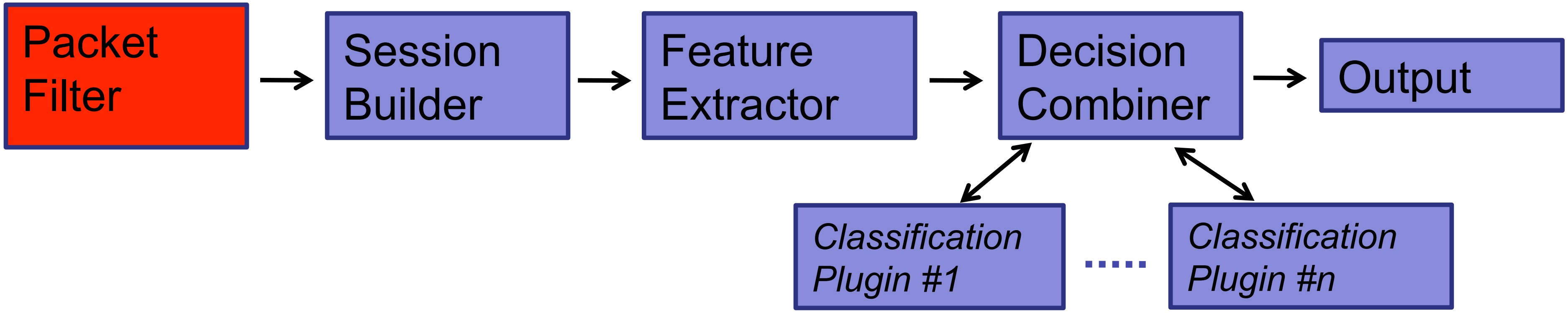
TIE: Traffic Identification Engine (1/2)

- An open-source software platform to allow the research community to work with *shared tools and data*
  - Supporting multiple approaches and techniques
  - Allowing comparison of different techniques
  - Able to act as a multi-classifier
  - Three available operating modes: *Offline*, *Realtime* and *Cyclic* mode.
  - Written in C, runs on Linux and FreeBSD platforms
  - ***http://tie.comics.unina.it***

TIE: Traffic Identification Engine (2/2)

- Easy to add: classification *features*, classification *techniques*, combination strategies
- Support for different definitions of *objects*
  - *Flows, Bi-flows, TCP connections, Hosts*
- Support for different definitions of *classes*
  - *Application IDs, Sub-IDs, Group IDs*
- Defined format of Output & Input Tables
- Tools for numerical and graphical analysis and comparison
  - Several common *metrics*: Accuracy, Byte-Accuracy, Precision, F-Measure, Recall
  - Confusion Matrices

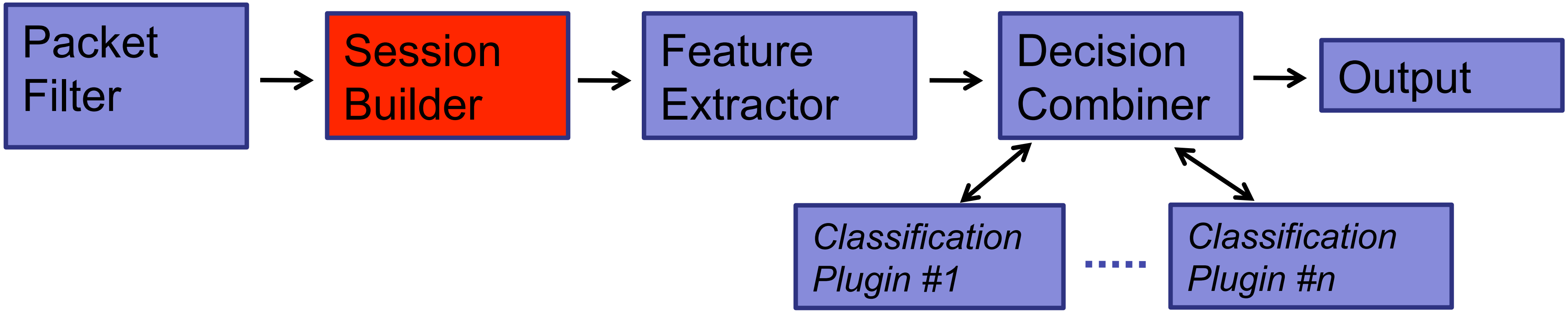
Tie’s Components: Packet filter



- Based on the *pcap** library
- Input can be either live traffic or a traffic trace
- Can operate packet filtering and validation both at kernel-level (using BPF) and user-level (e.g. skipping the first *m* packets, stopping the analysis after *n* packets, selecting traffic within a specified time range, checking for header integrity …)

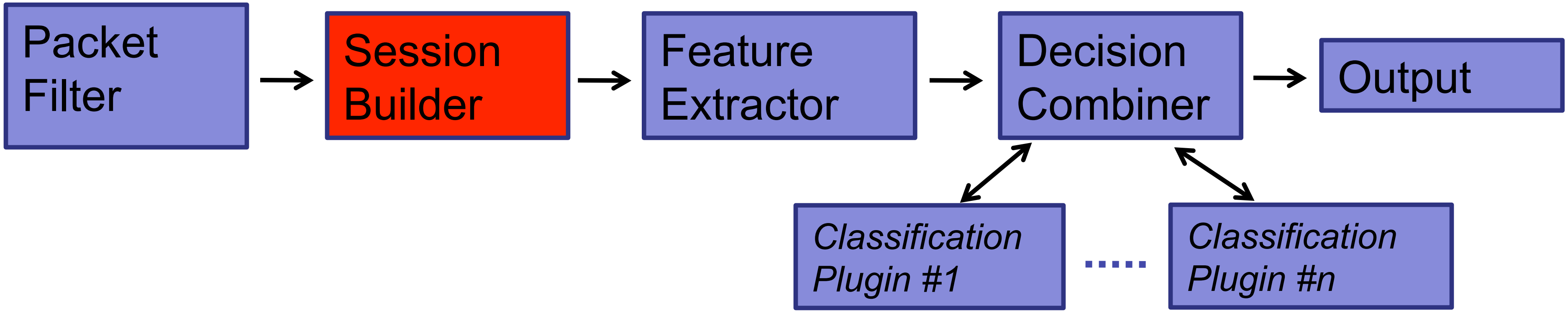
* *http://ww.tcpdump.org*

Tie’s Components: Session Builder



- Different definitions of “sessions” are allowed
  - **Flows**
    - *<L4Proto, IP_{src}, $Port_{src}$, IP_{dst}, $Port_{dst}$> + timeout*
  - **Biflows**
    - Same as above but *src* and *dst* swappable
    - Support for **TCP connections** through simple heuristics based on TCP flags
  - **Hosts**
    - Under development
- It keeps updated the status of each session (Status Information, Flags, Counters, …)

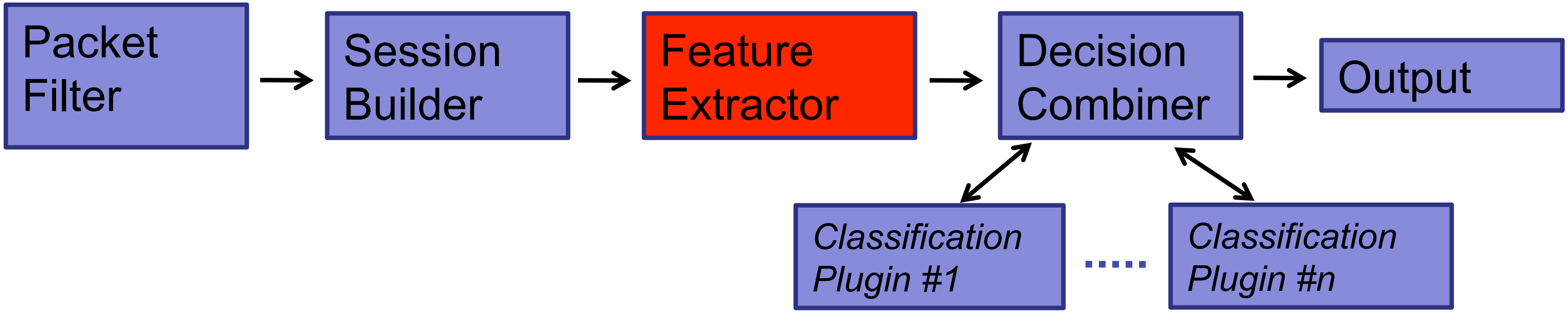
Tie's Components: Session Builder



TCP connections heuristics:

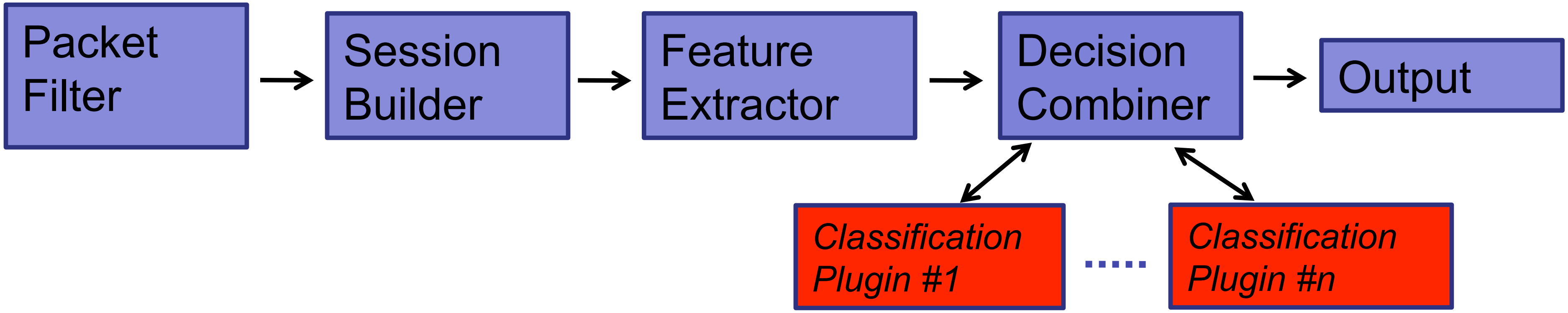
- If the 1st packet of a TCP biflow does not contain a SYN flag then it is skipped.
- The creation of a new biflow is forced if a TCP packet containing only a SYN flag is received (if a TCP biflow with the same tuple was active then it is forced to expire).
- A biflow is forced to expire if a FIN flag has been detected in both directions.
- The *inactivity timeout* is disabled on TCP biflows (they expire only if FIN flags are detected).

Tie's Components: Feature Extractor



- Features can be enabled/disabled at compile time
- Features
  - Portions of payload
  - Pkt/byte count
  - PS vector
  - IPT vector
  - …

Tie’s Components: Classification Plugins (1/2)



Packet Filter → Session Builder → Feature Extractor → Decision Combiner → Output

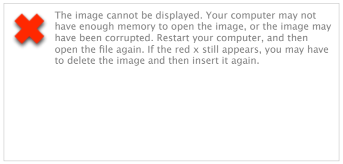
Classification Plugin #1 ….. *Classification Plugin #n*

- Each plugin
  - implements a specific classification technique
  - operates on a session
  - returns a result that includes a confidence value
- “dummy” plugin source available

```
typedef struct {
                    int (* disable)();
                    int (* enable)();
                    int (* load_signatures)(char *);
                    int (* train)(char *);
                    class_output* (* classify_session)(void *session);
                    int (* dump_statistics)(FILE *);
                    bool (* is_session_classifiable)(void *session);
                    int (* session_sign)(session *, class_output *);
                  char *name;
                    u_int32_t *flags;
} classifier;
```

Tie’s Components: Classification Plugins (2/2)

Name	***Based on***	***Status***	***Contributor***
Port	L4 Ports	Available	UNINA (signatures from CAIDA)
L7	Deep Payload Inspection	Available	UNINA (signatures/code from Linux L7-filter)
NBC	Lightweight Payload Inspection	Under test	UNINA
GMM-PS	Statistical Approach: PS	Under test	UNINA
HMM	Statistical Approach: PS, IPT	Under test	UNINA
FPT	Statistical Approach: PS, IPT	Under devel.	UNIBS
Joint	Machine Learning	Under devel.	UNINA-CAIDA-CENS
GT	Information from Hosts	Under devel.	UNIBS-UNINA-CAIDA



The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

COST-TMA

TIE-L7: a payload inspection plugin

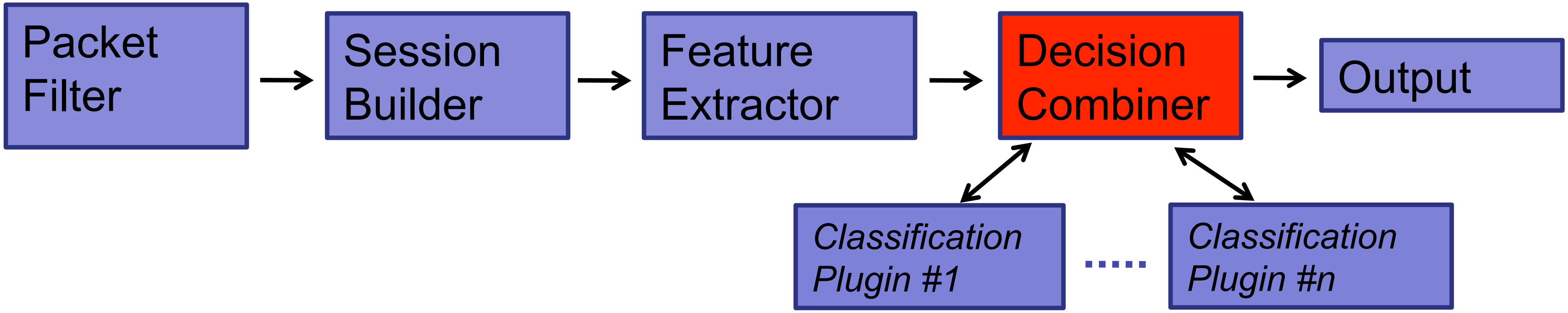
- Linux L7-filter*:
  - Performs traffic classification through deep payload inspection
  - State of the art technique (more than 120 applications supported)
  - Based on *pattern matching* through *regular expressions*

```
vnc ^rfb 00[1-9]\.00[0-9]\x0a
```

- TIE-L7
  - Allows the comparison against other approaches
  - Can be used as ground-truth technique in TIE
  - Runs both *offline* and *online* and on multiple OSs (e.g. FreeBSD)
  - We are improving the rules, and adding new ones

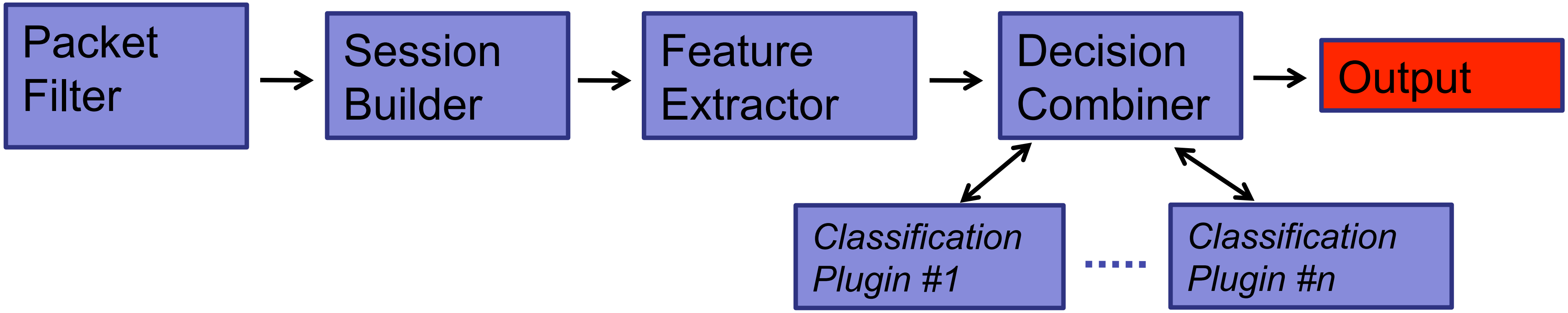
http://l7-filter.sourceforge.net/

Tie's Components: Decision Combiner



- The decision combiner determines the combination strategy
  - When to attempt classification
  - When each classifier is invoked
  - When the final decision is taken
  - How to combine the classification outputs from the classification plugins into the final decision

Tie's Components: Output



- Main output contains information about the sessions processed and their classification
- Output format is one, semantics change depending on session type (flow, biflow) and working mode (offline, realtime, cyclic,…)
- A collection of utilities are distributed with TIE for the post-processing of the output (e.g. overall stats, confusion matrix, …)

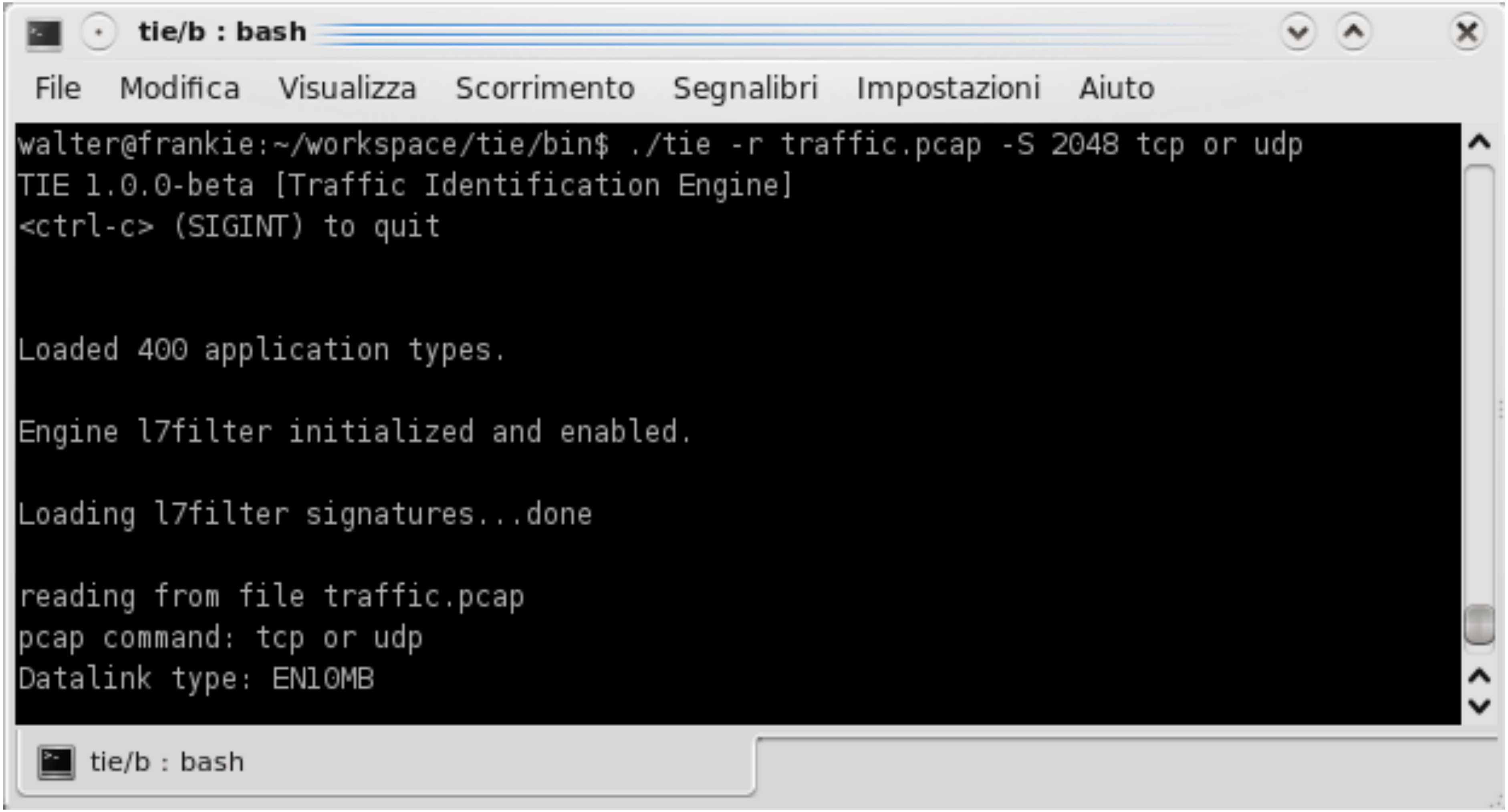
Operating Modes

- Offline
  - the classification of a session is generated only when the session ends or at the end of TIE execution
- Realtime
  - the classification of a session is generated as soon as it is available. This operating mode implements online classification.
- Cyclic
  - the classification of all live sessions is generated at regular intervals (e.g. each 5 minutes). Automated Web Reports using CAIDA’s CoralReef tools.

Playing with TIE

- Ground-Truth
  - Evaluation of Ground-Truth techniques
  - Alternative techniques requiring user collaboration
- Anonymized traces with ground-truth data
- Approaches based on pattern recognition
  - Traffic classification through novel packet-level features
  - Unsupervised approaches
- Approaches based on payload inspection
  - Lightweight Payload Inspection
- Combination techniques/strategies

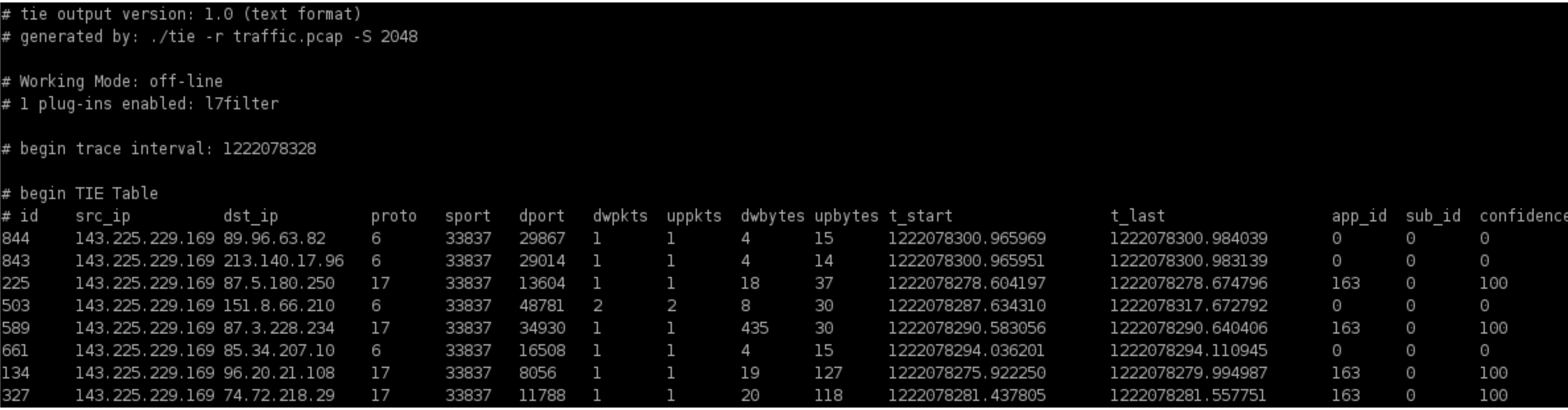
Screenshots



tie/b : bash
File Modifica Visualizza Scorrimento Segnalibri Impostazioni Aiuto

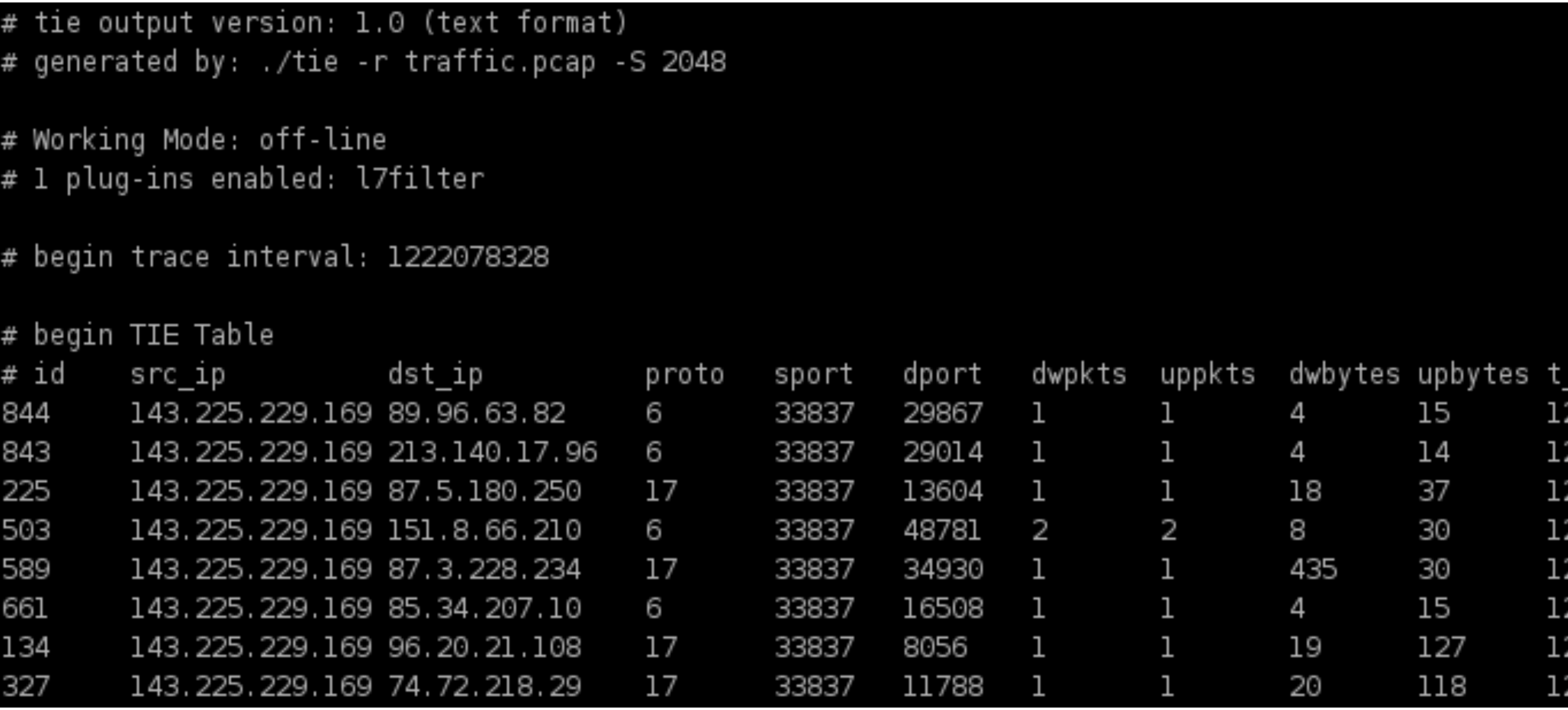
```
walter@frankie:~/workspace/tie/bin$ ./tie -r traffic.pcap -S 2048 tcp or udp
TIE 1.0.0-beta [Traffic Identification Engine]
<ctrl-c> (SIGINT) to quit



Loaded 400 application types.

Engine l7filter initialized and enabled.

Loading l7filter signatures...done

reading from file traffic.pcap
pcap command: tcp or udp
Datalink type: EN10MB
```

Screenshots



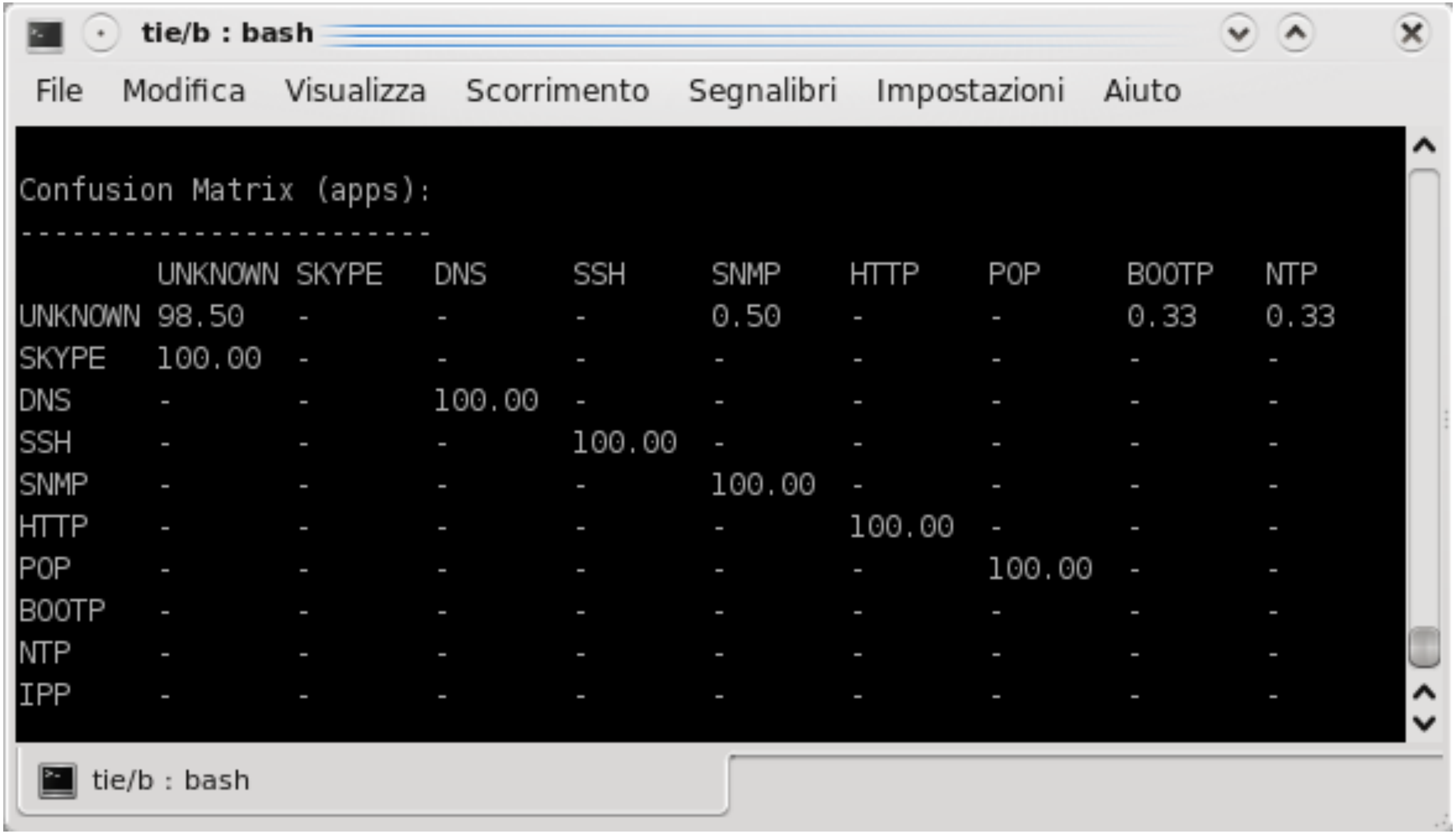
```
# tie output version: 1.0 (text format)
# generated by: ./tie -r traffic.pcap -S 2048

# Working Mode: off-line
# 1 plug-ins enabled: l7filter

# begin trace interval: 1222078328

# begin TIE Table
# id    src_ip          dst_ip          proto   sport   dport   dwpkts  uppkts  dwbytes upbytes t_start                 t_last                  app_id  sub_id  confidence
844     143.225.229.169 89.96.63.82     6       33837   29867   1       1       4       15      1222078300.965969       1222078300.984039       0       0       0
843     143.225.229.169 213.140.17.96   6       33837   29014   1       1       4       14      1222078300.965951       1222078300.983139       0       0       0
225     143.225.229.169 87.5.180.250    17      33837   13604   1       1       18      37      1222078278.604197       1222078278.674796       163     0       100
503     143.225.229.169 151.8.66.210    6       33837   48781   2       2       8       30      1222078287.634310       1222078317.672792       0       0       0
589     143.225.229.169 87.3.228.234    17      33837   34930   1       1       435     30      1222078290.583056       1222078290.640406       163     0       100
661     143.225.229.169 85.34.207.10    6       33837   16508   1       1       4       15      1222078294.036201       1222078294.110945       0       0       0
134     143.225.229.169 96.20.21.108    17      33837   8056    1       1       19      127     1222078275.922250       1222078279.994987       163     0       100
327     143.225.229.169 74.72.218.29    17      33837   11788   1       1       20      118     1222078281.437805       1222078281.557751       163     0       100
```

Screenshots



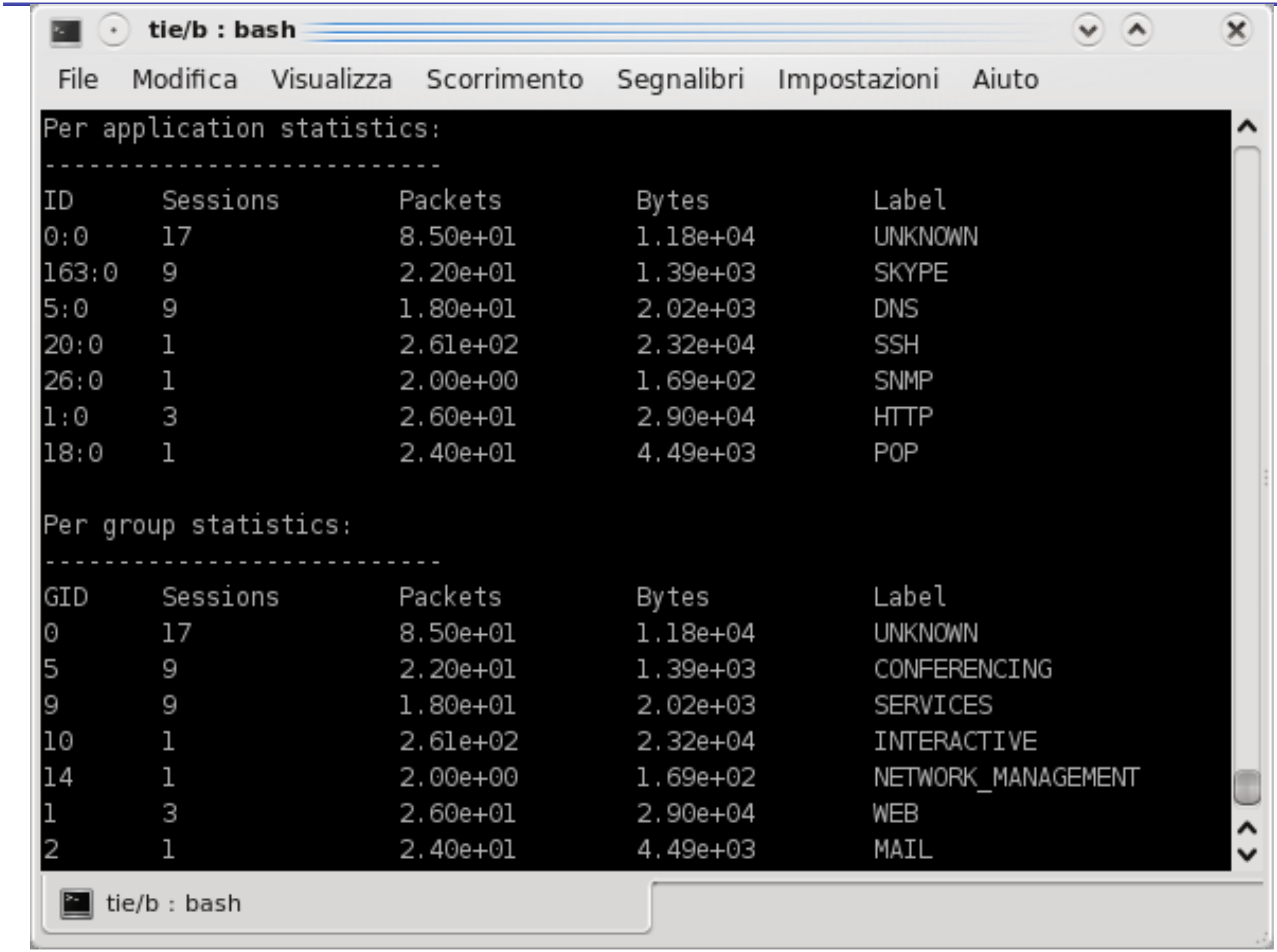
```
# tie output version: 1.0 (text format)
# generated by: ./tie -r traffic.pcap -S 2048

# Working Mode: off-line
# 1 plug-ins enabled: l7filter

# begin trace interval: 1222078328

# begin TIE Table
# id    src_ip          dst_ip          proto   sport   dport   dwpkts  uppkts  dwbytes upbytes t_
844     143.225.229.169 89.96.63.82     6       33837   29867   1       1       4       15      12
843     143.225.229.169 213.140.17.96   6       33837   29014   1       1       4       14      12
225     143.225.229.169 87.5.180.250    17      33837   13604   1       1       18      37      12
503     143.225.229.169 151.8.66.210    6       33837   48781   2       2       8       30      12
589     143.225.229.169 87.3.228.234    17      33837   34930   1       1       435     30      12
661     143.225.229.169 85.34.207.10    6       33837   16508   1       1       4       15      12
134     143.225.229.169 96.20.21.108    17      33837   8056    1       1       19      127     12
327     143.225.229.169 74.72.218.29    17      33837   11788   1       1       20      118     12
```

Screenshots



```
Confusion Matrix (apps):
------------------------
        UNKNOWN SKYPE   DNS     SSH     SNMP    HTTP    POP     BOOTP   NTP
UNKNOWN 98.50   -       -       -       0.50    -       -       0.33    0.33
SKYPE   100.00  -       -       -       -       -       -       -       -
DNS     -       -       100.00  -       -       -       -       -       -
SSH     -       -       -       100.00  -       -       -       -       -
SNMP    -       -       -       -       100.00  -       -       -       -
HTTP    -       -       -       -       -       100.00  -       -       -
POP     -       -       -       -       -       -       100.00  -       -
BOOTP   -       -       -       -       -       -       -       -       -
NTP     -       -       -       -       -       -       -       -       -
IPP     -       -       -       -       -       -       -       -       -
```

Screenshots

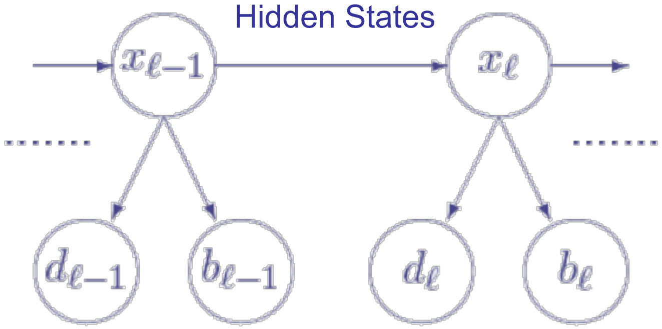


tie/b : bash
File Modifica Visualizza Scorrimento Segnalibri Impostazioni Aiuto

```
Per application statistics:
---------------------------
ID      Sessions        Packets         Bytes           Label
0:0     17              8.50e+01        1.18e+04        UNKNOWN
163:0   9               2.20e+01        1.39e+03        SKYPE
5:0     9               1.80e+01        2.02e+03        DNS
20:0    1               2.61e+02        2.32e+04        SSH
26:0    1               2.00e+00        1.69e+02        SNMP
1:0     3               2.60e+01        2.90e+04        HTTP
18:0    1               2.40e+01        4.49e+03        POP

Per group statistics:
---------------------------
GID     Sessions        Packets         Bytes           Label
0       17              8.50e+01        1.18e+04        UNKNOWN
5       9               2.20e+01        1.39e+03        CONFERENCING
9       9               1.80e+01        2.02e+03        SERVICES
10      1               2.61e+02        2.32e+04        INTERACTIVE
14      1               2.00e+00        1.69e+02        NETWORK_MANAGEMENT
1       3               2.60e+01        2.90e+04        WEB
2       1               2.40e+01        4.49e+03        MAIL
```

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



Hidden States

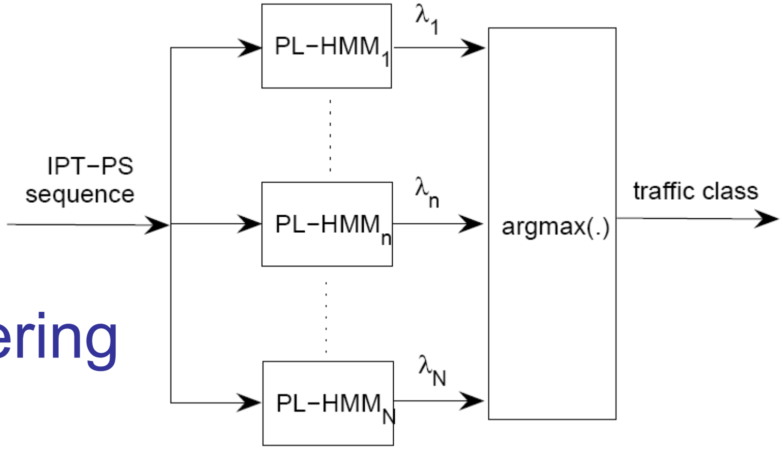
IPT and PS conditional distributions

– Applied to more categories of Traffic

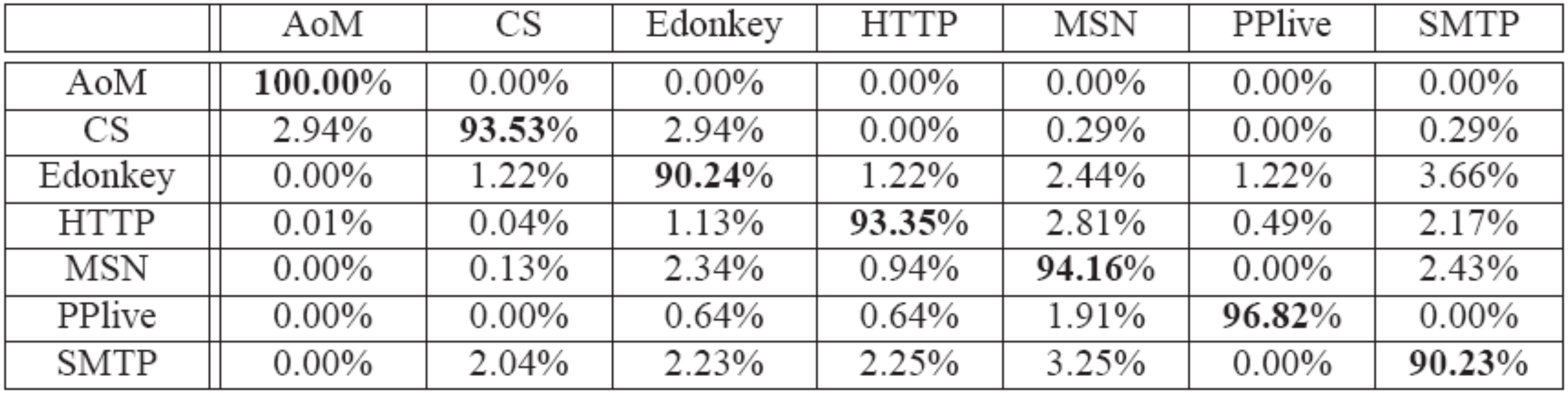
– Models usable for
  – Performance Evaluation
  – Traffic Generation
  – Prediction
  – Classification

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



CLASSIFICATION RESULTS: CONFUSION MATRIX



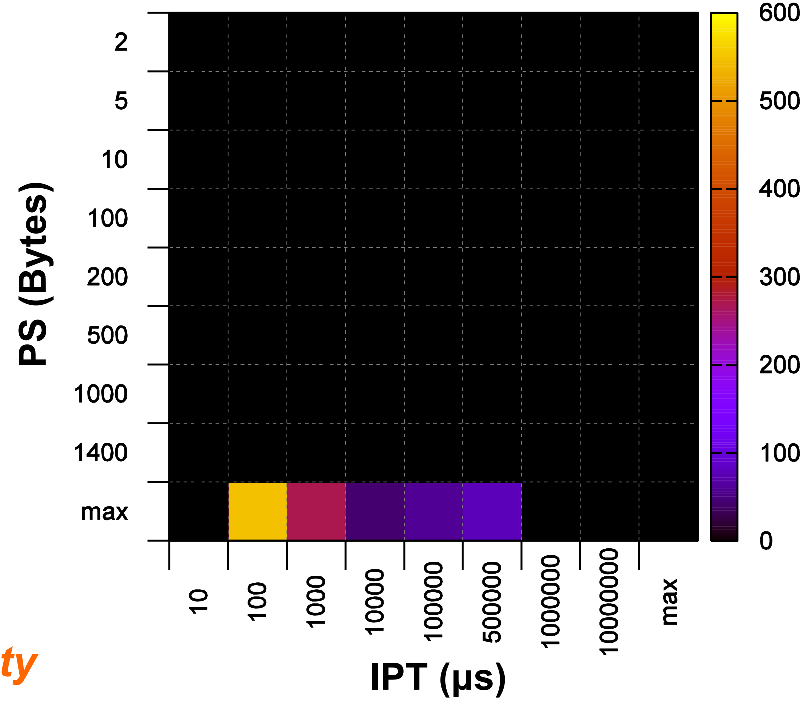
	AoM	CS	Edonkey	HTTP	MSN	PPlive	SMTP
AoM	**100.00%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CS	2.94%	**93.53%**	2.94%	0.00%	0.29%	0.00%	0.29%
Edonkey	0.00%	1.22%	**90.24%**	1.22%	2.44%	1.22%	3.66%
HTTP	0.01%	0.04%	1.13%	**93.35%**	2.81%	0.49%	2.17%
MSN	0.00%	0.13%	2.34%	0.94%	**94.16%**	0.00%	2.43%
PPlive	0.00%	0.00%	0.64%	0.64%	1.91%	**96.82%**	0.00%
SMTP	0.00%	2.04%	2.23%	2.25%	3.25%	0.00%	**90.23%**

Joint: TC w/ pkt-level joint distributions

- Based on joint distributions of PS and IPT: heavily quantized in a matrix of features
- Classify bidirectional flows (*biflows*)
- Machine Learning approach for classification: *SVM, 1-NN, 3-NN*
- Working with full traces (high number of applications)
- Traffic from different links and with different dates

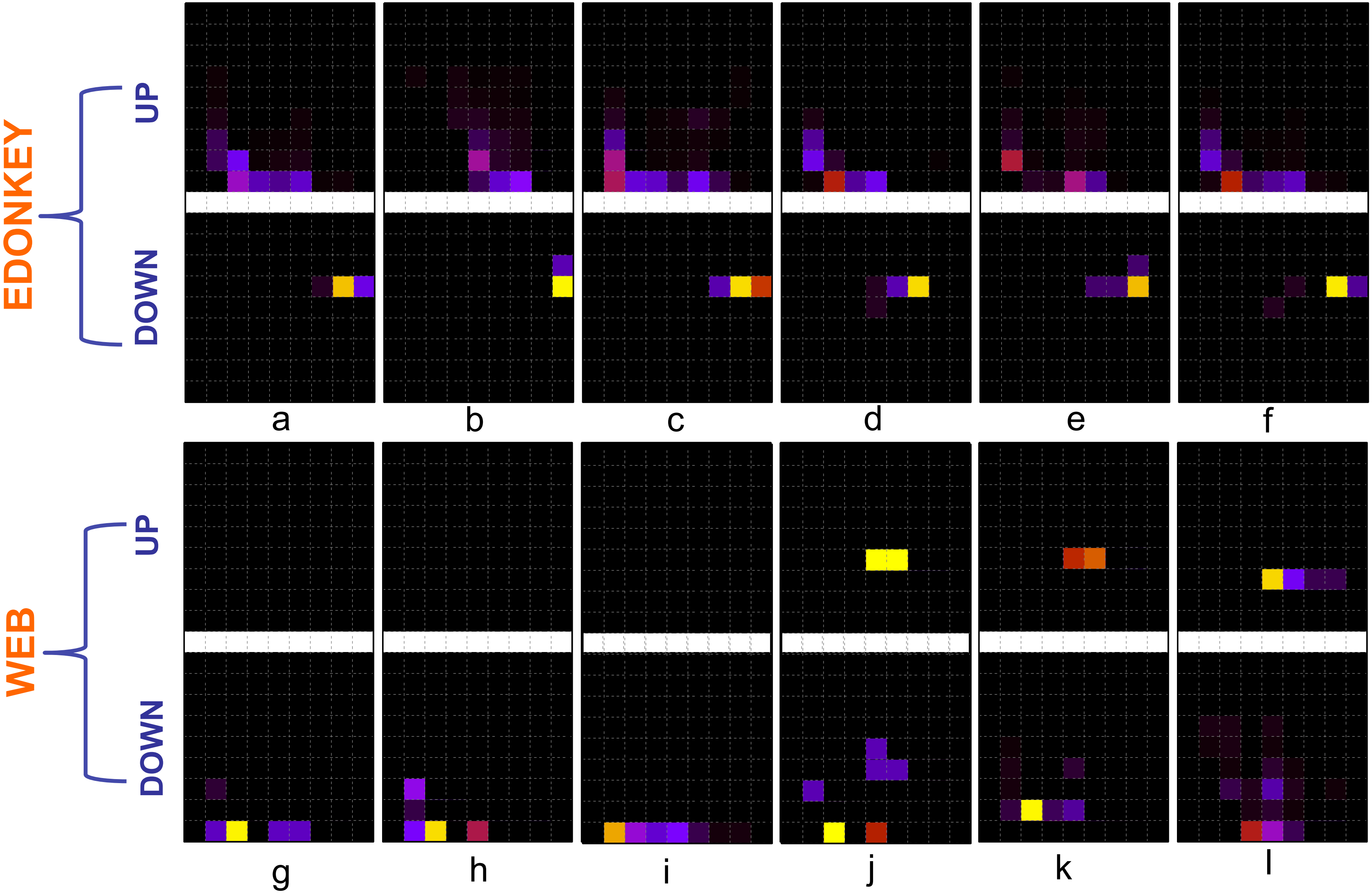
Trace	DATE	Duration	Pkts	Biflows
KAIST	14/9/2006	21hrs	357M	3.2M
UNINA	16/5/2008	20 min	52M	758k

Approximation of a Joint PDF



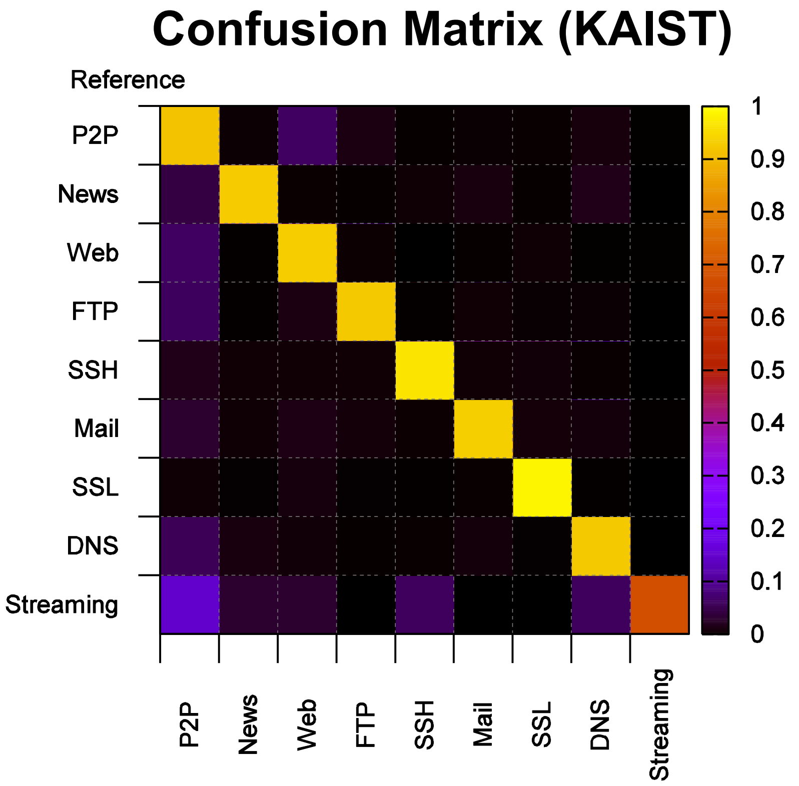
Joint activity with CAIDA and Seoul National University

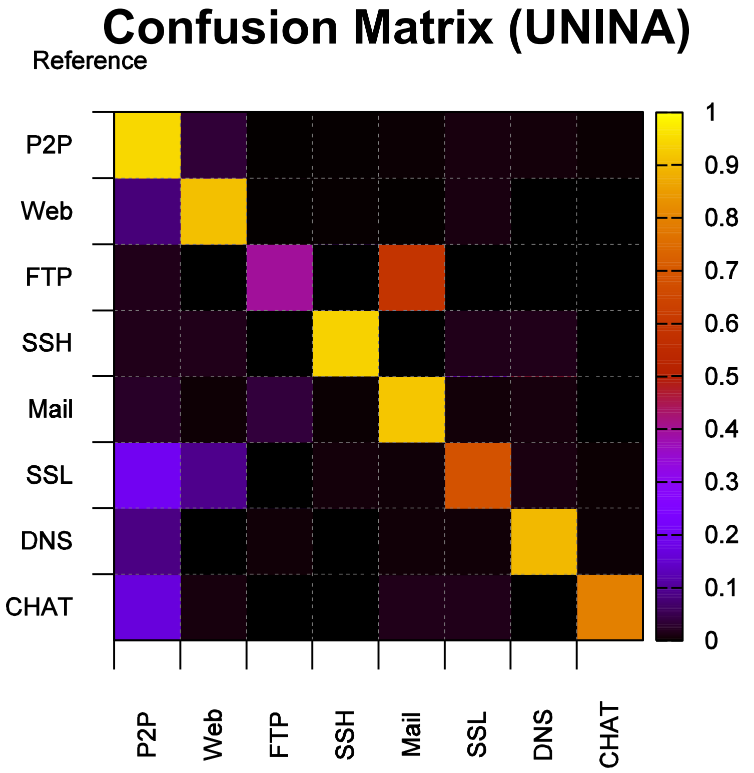
Joint: samples of *fingerprints*



Joint: classification results

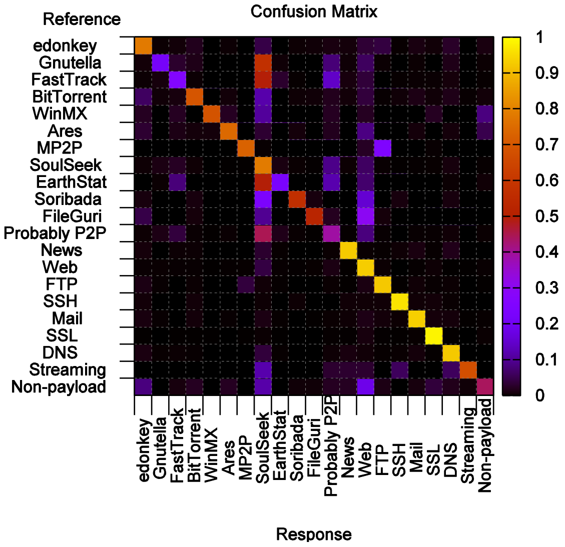
- Tests with features: Joint PDF, pkt up/down ratio, log(duration)
- Accuracy per application group:
  - KAIST: **93.2%**
  - UNINA: **92.3%**





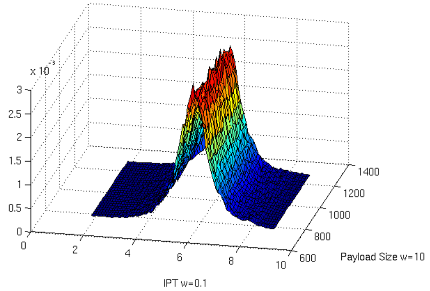
Joint: Considerations

- Confusion was higher among apps of the same category (e.g. P2P filesharing apps)
- Approach looks robust to obfuscation and encryption



Future works:

- Feature selection
- Online implementation
- Cross-testing
- Tests with better ground truth tools
- Same approach, working with *host* sessions, instead of *biflows*, may be used to detect compromised hosts (worms)



Conclusions

- Traffic Classification is important for understanding and controlling the Internet.
- Despite the large quantity of research works there are still several open issues.
- Because of the continuously evolving scenario and the emergence of new applications, research in this field will probably keep being very active in the future.
- Common tools and techniques are needed.
- Contact us if you want to use/contribute/work with TIE (***http://tie.comics.unina.it***).

Some (of our) references

- A. Dainotti, A. Pescapè, C. Sansone, *Early Classification of Network Traffic through Multi-Classification,* International Workshop on Traffic Monitoring and Analysis (TMA'11) - April 2011, Vienna (Austria)
- G. Aceto, A. Dainotti, W. de Donato, A. Pescapè, *PortLoad: taking the best of two worlds in traffic classification,* IEEE INFOCOM 2010 - WIP Track - March 2010, San Diego (CA, USA)
- V. Carela-Espanol, P. Barlet-Ros, M. Solé-Simò, A. Dainotti, W. de Donato, A. Pescapè, *K-dimensional trees for continuous traffic classification*, International Workshop on Traffic Monitoring and Analysis (TMA'10) @ PAM 2010 - April 2010, Zurich (Switzerland)
- A. Dainotti, F. Gargiulo, L. Kuncheva, A. Pescapè, C. Sansone*, Identification of traffic flows hiding behind TCP port 80*, IEEE ICC 2010 - May 2010, Capetown (South Africa)
- A. Dainotti, W. de Donato, A. Pescapè, *TIE: a Community-Oriented Traffic Classification Platform*", International Workshop on Traffic Monitoring and Analysis (TMA'09) @ IFIP Networking 2009 - May 2009, Aachen (Germany)
- A. Dainotti, W. de Donato, A. Pescapè, P. Salvo Rossi, "*Classification of Network Traffic via Packet-Level Hidden Markov Models*", IEEE GLOBECOM 2008 - Dec 2008, New Orleans (LA, USA)

Thanks for the attention

Any Questions ?

alberto@unina.it

http://www.grid.unina.it/Traffic