JACOBS
UNIVERSITY

Flow Signatures of Popular Applications

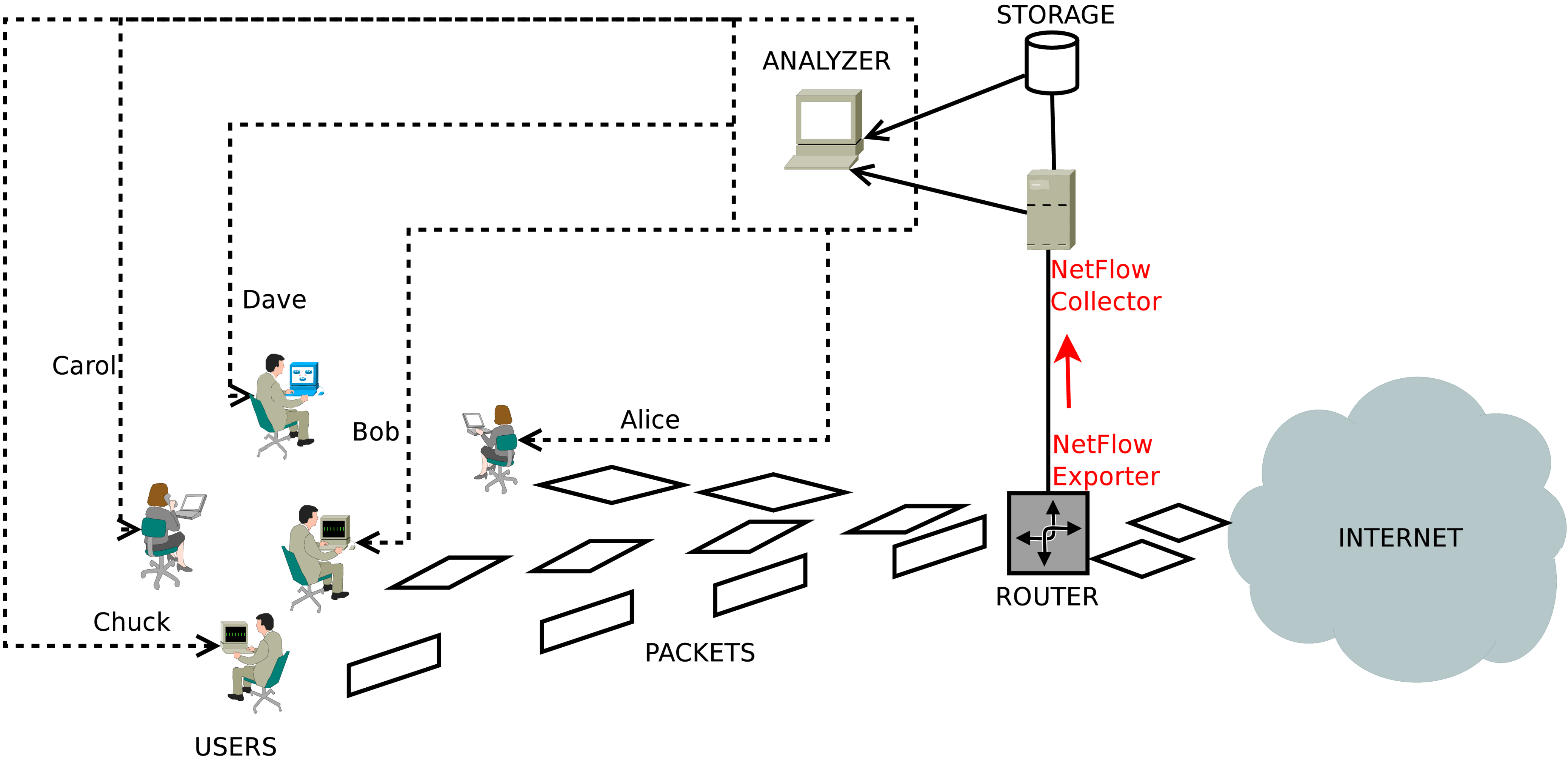
Computer Networks and Distributed Systems (CNDS) group
Presentation date: 30.07.2010

Overview

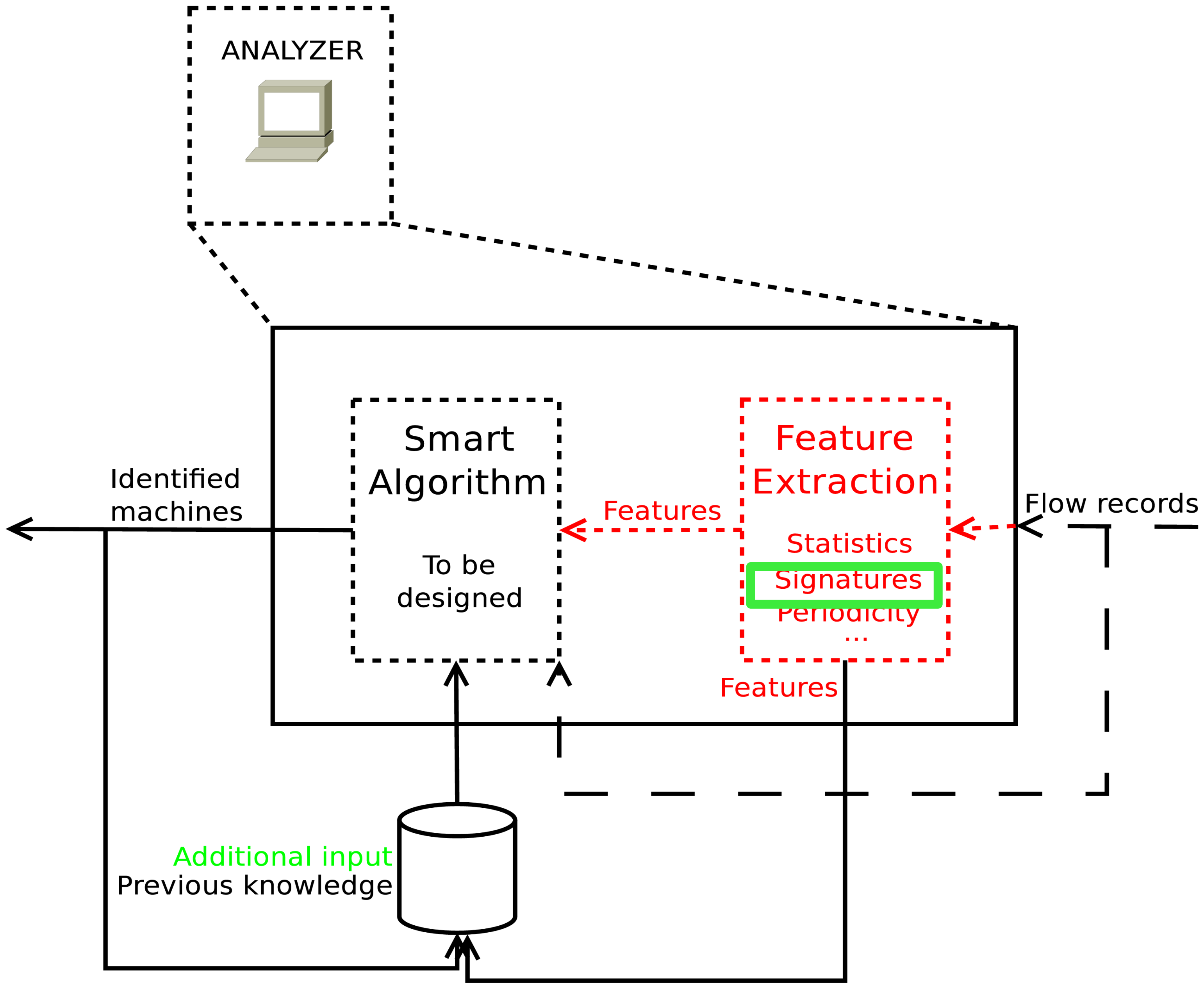
- Problem statement
- Applications & signatures
- Results
- Future research plan

Problem Statement

Identify users based on their network activity (expressed in flow records)



Problem Statement



Applications & Signatures

Linux and Windows versions (when possible)

- *Web browsers:*
  - *Firefox, Opera, Google Chrome*
- *Instant messaging clients*
  - *Skype, Yahoo, ICQ, MSN*
- *Mail clients*
  - *Mozilla Thunderbird, Microsoft Outlook, Windows Live*
- *Media players*
  - *Amarok, iTunes, Windows Media Player*

Browsers

- *Google Chrome v5.0*

Google Chrome v4.X showed no activity, V5.0 is different

- *Within a few seconds:*

DNS type A queries to the local DNS server

rsobazcuyh.students.jacobs-university.de 10 letter _random_ strings

To determine if unknown intranet DNS names are redirected to a special site, and which?

In turn – avoids erroneous display of an info-bar

- *Within a few minutes:*

Contacts safebrowsing.clients.google.com & safebrowsing-cache.google.com

Pulls a list of phishing and malware web-sites

Browsers

- Opera 10.10

Uses Opera Unite technology

Discovery of local Opera Unite users is enabled by default

Uses Simple Service Discovery Protocol (SSDP); send 3 identical UDP multicast messages to 239.255.255.250 and port 1900

For the discovered Opera Unite users, communication starts on TCP port 2869

Contacts sitecheck2.opera.com; is web-site is trusted? (DNS request for it at start)

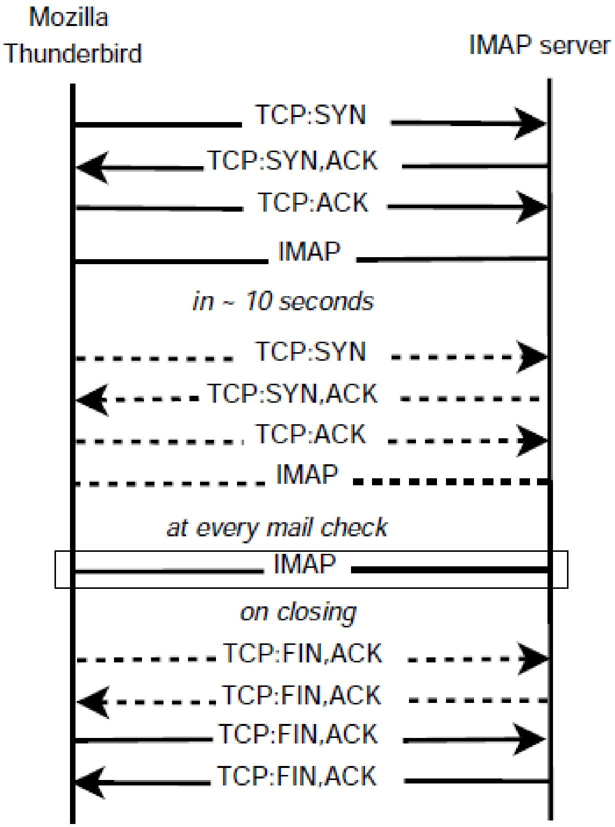
Default RSS updates every three hours

All browsers allow for extensions (i.e., AniWeather) => specific flow records

Mail Clients

- *Analyzed 3 applications for POP and IMAP servers*
- *Start-up:*
  - *Welcome page fetched http://live.mozillamessaging.com/thunderbird/start*
  - *DNS type A and AAAA queries (others – only type A)*
  - *TCP connection to retrieve folder structure, remains open*
- *In 10 seconds:*
  - *Another connection to check for new messages*
  - *Checks every 10 minutes*
  - *Connections remain open*

Applications & Signatures



Mozilla Thunderbird

IMAP server

TCP:SYN

TCP:SYN,ACK

TCP:ACK

IMAP

in ~ 10 seconds

TCP:SYN

TCP:SYN,ACK

TCP:ACK

IMAP

at every mail check

IMAP

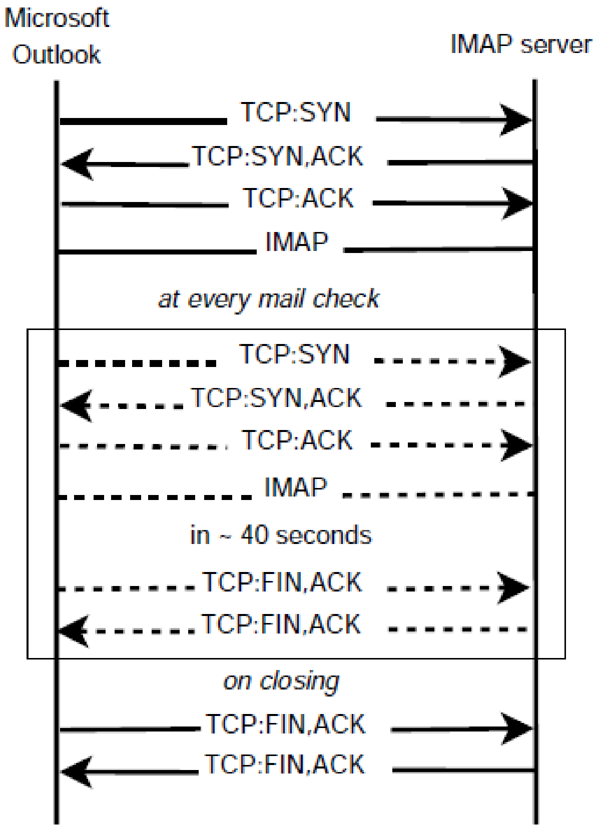
on closing

TCP:FIN,ACK

TCP:FIN,ACK

TCP:FIN,ACK

TCP:FIN,ACK



Microsoft Outlook

IMAP server

TCP:SYN

TCP:SYN,ACK

TCP:ACK

IMAP

at every mail check

TCP:SYN

TCP:SYN,ACK

TCP:ACK

IMAP

in ~ 40 seconds

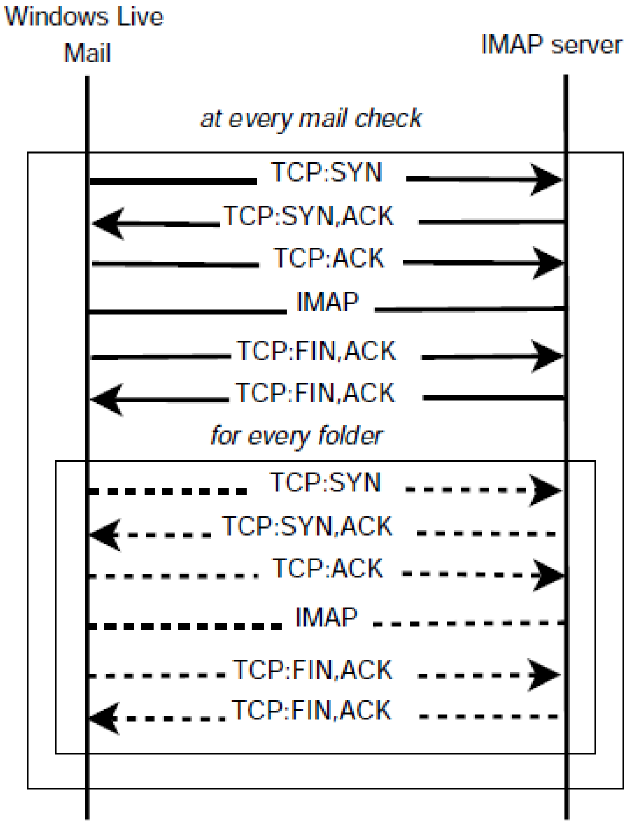
TCP:FIN,ACK

TCP:FIN,ACK

on closing

TCP:FIN,ACK

TCP:FIN,ACK



Windows Live Mail

IMAP server

at every mail check

TCP:SYN

TCP:SYN,ACK

TCP:ACK

IMAP

TCP:FIN,ACK

TCP:FIN,ACK

for every folder

TCP:SYN

TCP:SYN,ACK

TCP:ACK

IMAP

TCP:FIN,ACK

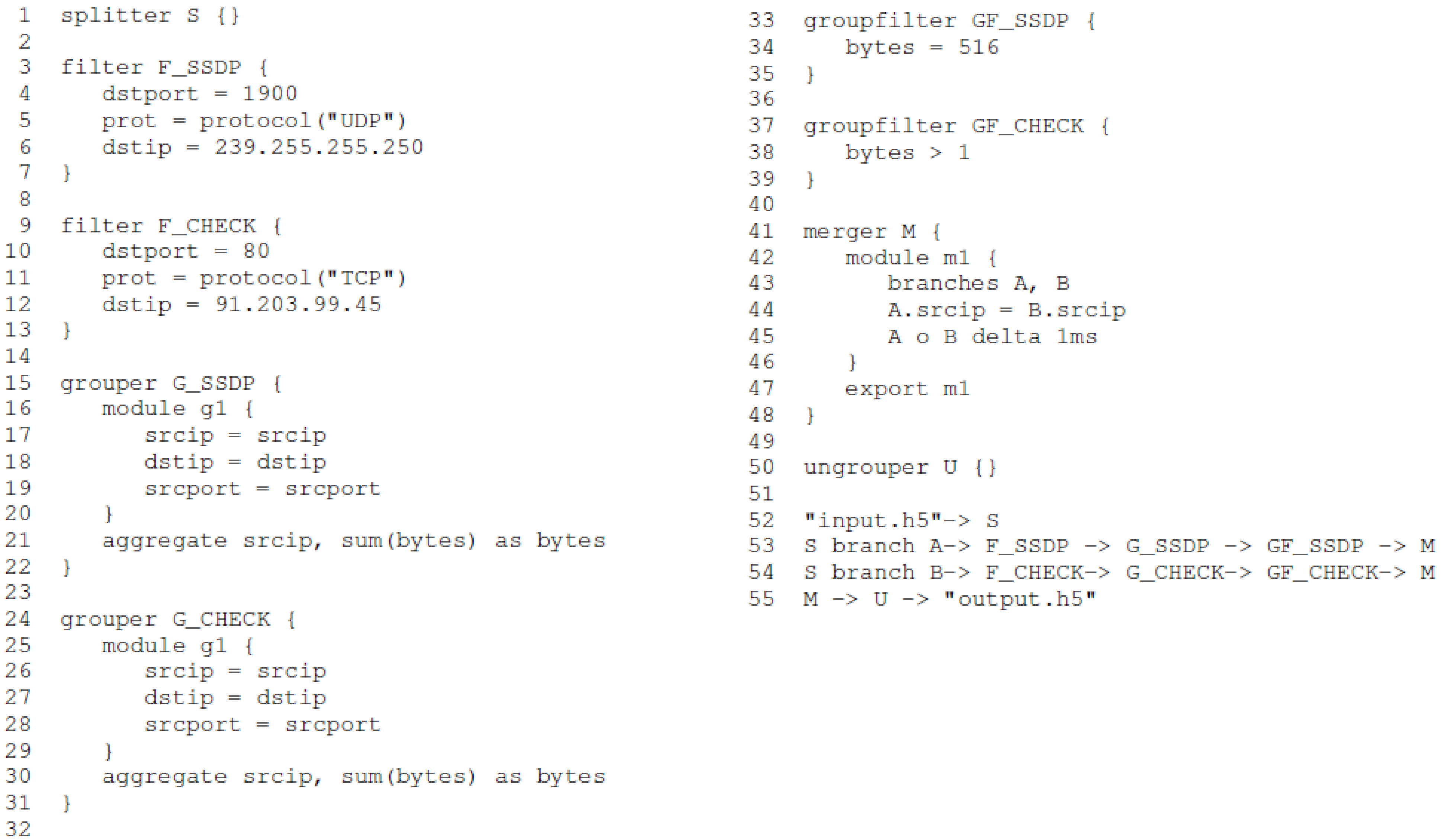
TCP:FIN,ACK

Media players

- *Amarok and iTunes use Apple's Digital Audio Access Protocol*
- *On start up:*
  - *MulticastDNS query of PTR type to 224.0.0.251:5353*
  - *For _daap._tcp.local*
  - *Repeated after [1, 2, 4, 8, ..., 1024, …] seconds*
  - *Shared Media Library discovery → connections IP_addr:3689*
  - *Lyrics and informational applets*

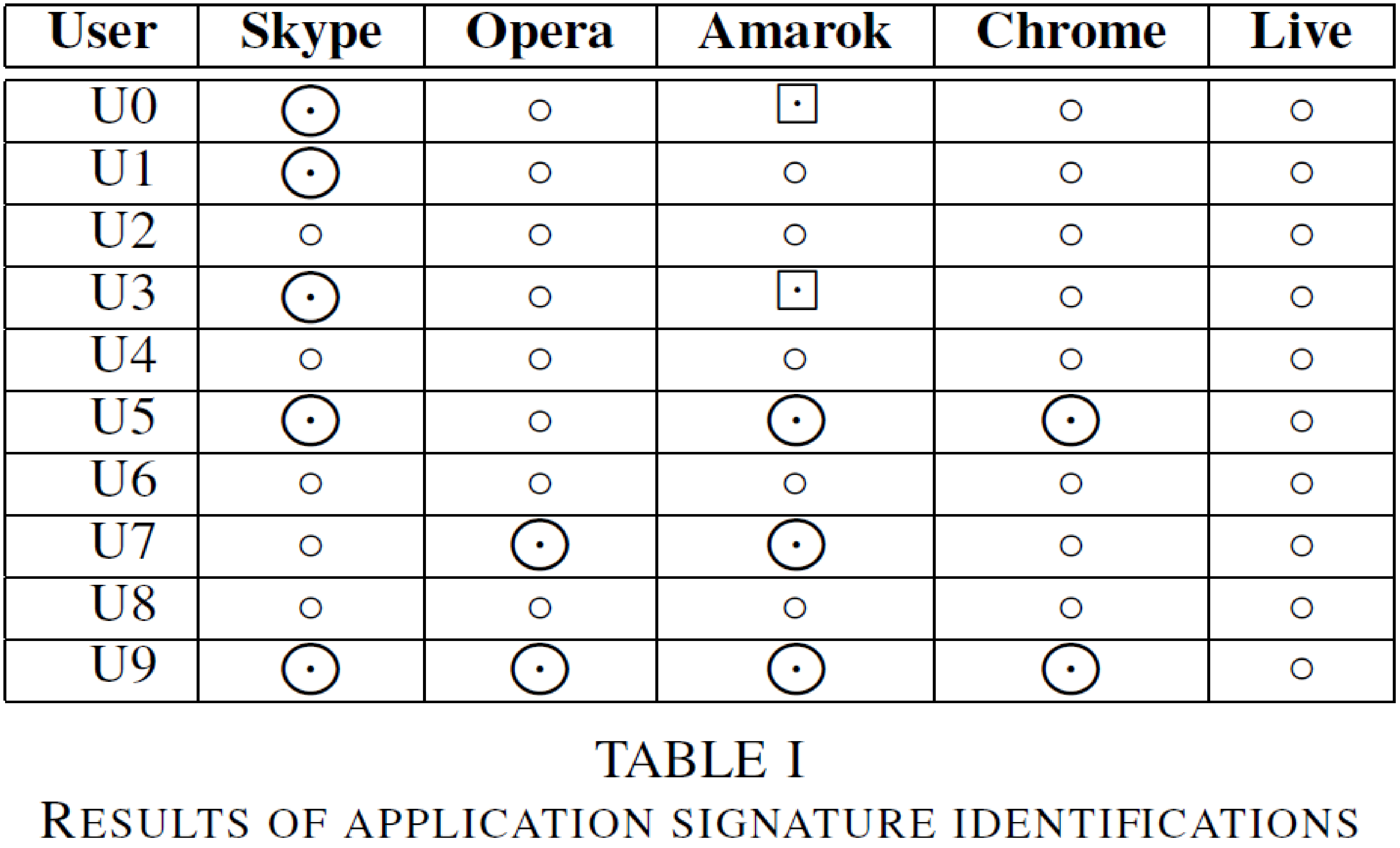
Signatures

- *We defined application signatures using a stream-based flow query language Flowy*



```
splitter S {}

filter F_SSDP {
   dstport = 1900
   prot = protocol("UDP")
   dstip = 239.255.255.250
}

filter F_CHECK {
   dstport = 80
   prot = protocol("TCP")
   dstip = 91.203.99.45
}

grouper G_SSDP {
   module g1 {
      srcip = srcip
      dstip = dstip
      srcport = srcport
   }
   aggregate srcip, sum(bytes) as bytes
}

grouper G_CHECK {
   module g1 {
      srcip = srcip
      dstip = dstip
      srcport = srcport
   }
   aggregate srcip, sum(bytes) as bytes
}

groupfilter GF_SSDP {
   bytes = 516
}

groupfilter GF_CHECK {
   bytes > 1
}

merger M {
   module m1 {
      branches A, B
      A.srcip = B.srcip
      A o B delta 1ms
   }
   export m1
}

ungrouper U {}

"input.h5"-> S
S branch A-> F_SSDP -> G_SSDP -> GF_SSDP -> M
S branch B-> F_CHECK-> G_CHECK-> GF_CHECK-> M
M -> U -> "output.h5"
```

Results



User	Skype	Opera	Amarok	Chrome	Live
U0	⊙	○	⊡	○	○
U1	⊙	○	○	○	○
U2	○	○	○	○	○
U3	⊙	○	⊡	○	○
U4	○	○	○	○	○
U5	⊙	○	⊙	⊙	○
U6	○	○	○	○	○
U7	○	⊙	⊙	○	○
U8	○	○	○	○	○
U9	⊙	⊙	⊙	⊙	○

TABLE I

RESULTS OF APPLICATION SIGNATURE IDENTIFICATIONS

Future research plan

- *Establish more application signatures for Flowy*
- *Observation phase*
  - *Observe and understand dynamics of application signatures*
- *“Uncontrolled” testing phase for signatures*
  - *Using real-life flow records*
  - *Span several days/weeks to detect correlated variations*
  - *Day-time analysis of application employment*

¡Thank you for attention!