Network Traffic Evolution

Prof. Anja Feldmann, Ph.D.
Dr. Nikolaos Chatzis
boa.net.t-labs.tu-berlin.de
Oilsnake
Python

Username: measurement13
Password: Meter13
## Example trace

<table>
<thead>
<tr>
<th>Name</th>
<th>port</th>
<th>% bytes</th>
<th>% packets</th>
<th>bytes per packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>world-wide-web</td>
<td>80</td>
<td>???</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>netnews</td>
<td>119</td>
<td>???</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>pop-3 mail</td>
<td>110</td>
<td>???</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>. . .</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- How?????
Passive measurements

Definition:
- Observing traffic into the network
- Computing metrics on the monitored traffic
  - In our case: Application Mix

Packet monitors

- Available data:
  - All protocol information
  - All content
# IP header format

<table>
<thead>
<tr>
<th>4-bit Version</th>
<th>4-bit Header Length</th>
<th>8-bit Type of Service (TOS)</th>
<th>16-bit Total Length (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16-bit Identification</td>
<td>3-bit Flags</td>
<td>13-bit Fragment Offset</td>
<td></td>
</tr>
<tr>
<td>8-bit Time to Live (TTL)</td>
<td>8-bit Protocol</td>
<td>16-bit Header Checksum</td>
<td></td>
</tr>
<tr>
<td>32-bit Source IP Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-bit Destination IP Address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Options (if any)

Payload

20-byte header
## TCP header format

<table>
<thead>
<tr>
<th>16-bit source port number</th>
<th>16-bit destination port number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>32-bit sequence number</td>
</tr>
<tr>
<td></td>
<td>32-bit acknowledgement number</td>
</tr>
<tr>
<td>4-bit header length</td>
<td>U A P R S Y F</td>
</tr>
<tr>
<td></td>
<td>16-bit window size</td>
</tr>
<tr>
<td></td>
<td>16-bit TCP checksum</td>
</tr>
<tr>
<td></td>
<td>16-bit urgent pointer</td>
</tr>
<tr>
<td>Options (if any)</td>
<td>Payload</td>
</tr>
</tbody>
</table>

20-byte header
Tools

- ipsumdump
  - Good for quick summaries

- tcpdump
  - Good for in depth details
  - Basis for wireshark

- wireshark
  - Good for visual inspection of in depth details

- Bro
  - Good for in depth scripted analysis
  - Security analysis
  - Application analysis
Ipsmumdump (subset)

anja% ipsumdump -h
'Ipsmumdump' reads IP packets from tcpdump(1) files, or network interfaces, and summarizes their contents in an ASCII log.

Usage: ipsumdump [CONTENT OPTIONS] [-i DEVNAMES | FILES] > LOGFILE

-t, --timestamp Include packet timestamps.
-s, --src Include IP source addresses.
-d, --dst Include IP destination addresses.
-S, --sport Include TCP/UDP source ports.
-D, --dport Include TCP/UDP destination ports.
-l, --length Include IP lengths.
-p, --protocol Include IP protocols.
   --id Include IP IDs.
-g, --fragment Include IP fragment flags ('F' or '.').
-F, --tcp-flags Include TCP flags word.
-Q, --tcp-seq Include TCP sequence numbers.
-K, --tcp-ack Include TCP acknowledgement numbers.
-W, --tcp-window Include TCP receive window (unscaled).
   --udp-length Include UDP lengths.
-L, --payload-length Include payload lengths (no IP/UDP/TCP headers).
   --payload Include packet payloads as quoted strings.
   --payload-md5 Include MD5 checksum of packet payloads.
   --capture-length Include lengths of captured IP data.

Data source options (give exactly one):
-r, --tcpdump Read tcpdump(1) FILES (default).
-i, --interface Read network devices DEVNAMES until interrupted.
Tcpdump (subset)

Usage: tcpdump [options] [filter expression]

general options:
    [-c packetcount ]

input options:
    [-i interface ] | [-r input dumpfile name
                   [ -P passphrase or - for stdin ]]
    [-F filterfile ] [ -s snaplength ]

binary-output options:
    [-w dumpfile(base) [ -W dumpfile slice size ]
     [ -P passphrase or - for stdin]]

ASCII output options:
    [-n]: do not resolve hostnames
    [-M]: output in machine-readable format
    [-v]: increase verbosity (e.g. prints checksums)
    [-e]: print linklayer information
    [-X]: full-packet output in hex-format
    [-A]: print packet payload as ASCII
    [-S]: absolute TCP sequence numbers

...
Tcpdump output
(three-way TCP handshake and HTTP request message)

617756405:617756405 (0) win 32120 <mss 1460,sackOK,timestamp 46339
0,nop,wscale 0> (DF)

2598794605:2598794605 (0) ack 617756406 win 16384 <mss 512>

1:1(0) ack 1 win 32120 (DF)

1:1(0) ack 513 win 16384

513:676(163) ack 1 win 32256 (DF)

1:179(178) ack 676 win 16384
Wireshark (subset)

Usage: wireshark [options] ... [ <infile> ]

Capture interface:
- `i <interface>`  name or idx of interface (def: first non-loopback)
- `f <capture filter>`  packet filter in libpcap filter syntax
- `s <snaplen>`  packet snapshot length (def: 65535)
- `S`  update packet display when new packets are captured

Capture stop conditions:
- `c <packet count>`  stop after n packets (def: infinite)

Input file:
- `r <infile>`  set the filename to read from (no pipes or stdin!)

Processing:
- `R <read filter>`  packet filter in Wireshark display filter syntax
- `n`  disable all name resolutions (def: all enabled)

User interface:
- `g <packet number>`  go to specified packet number after "-r"
- `m <font>`  set the font name used for most text
- `t ad|a|r|d|dd|e`  output format of time stamps (def: r: rel. to first)
- `X <key>:<value>`  eXtension options, see man page for details
- `z <statistics>`  show various statistics, see man page for details
Selecting traffic

- Filter to focus on a subset of the packets
  - IP addresses/prefixes (e.g., to/from specific Web sites, client machines, DNS servers, mail servers)
  - Protocol (e.g., TCP, UDP, or ICMP)
  - Port numbers (e.g., HTTP, DNS, BGP, Napster)
- Collect first n bytes of packet (snap length)
  - Medium access control header (if present)
  - IP header (typically 20 bytes)
  - IP+UDP header (typically 28 bytes)
  - IP+TCP header (typically 40 bytes)
  - Application-layer message (entire packet)
Monitoring a LAN link

Shared media (Ethernet, wireless)

Host A  Host B  Monitor

Multicast switch

Host A  Switch  Host C

Host B  Monitor

Monitor integrated with a bridge

Host A  Bridge/Monitor  Host B
Monitoring a WAN link

Splitting a point-to-point link

Router A  Monitor  Router B

Line card that does packet sampling

Router A