Network Traffic Evolution

Prof. Anja Feldmann, Ph.D.
Testbed

- tst1.inet.tu-berlin.de
- tst2.inet.tu-berlin.de
- user/password: measurement17/Meter17
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>
</tbody>
</table>
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></td>
<td>16-bit Identification</td>
<td>3-bit Flags</td>
<td>13-bit Fragment Offset</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>
<tr>
<td>Options (if any)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The IP header format consists of various fields that carry information about the IP packet's header and payload. The header includes fields such as version, header length, type of service, total length, identification, flags, fragment offset, time to live, protocol, header checksum, source IP address, destination IP address, options (if any), and payload. The header occupies 20 bytes in total.
### TCP header format

<table>
<thead>
<tr>
<th>Field</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit source port number</td>
<td>16</td>
</tr>
<tr>
<td>16-bit destination port number</td>
<td>16</td>
</tr>
<tr>
<td>32-bit sequence number</td>
<td>32</td>
</tr>
<tr>
<td>32-bit acknowledgement number</td>
<td>32</td>
</tr>
<tr>
<td>4-bit header length</td>
<td>4</td>
</tr>
<tr>
<td>URG</td>
<td>1</td>
</tr>
<tr>
<td>ACK</td>
<td>1</td>
</tr>
<tr>
<td>PSH</td>
<td>1</td>
</tr>
<tr>
<td>RST</td>
<td>1</td>
</tr>
<tr>
<td>SYN</td>
<td>1</td>
</tr>
<tr>
<td>FIN</td>
<td>1</td>
</tr>
<tr>
<td>16-bit window size</td>
<td>16</td>
</tr>
<tr>
<td>16-bit TCP checksum</td>
<td>16</td>
</tr>
<tr>
<td>16-bit urgent pointer</td>
<td>16</td>
</tr>
<tr>
<td>Options (if any)</td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td></td>
</tr>
</tbody>
</table>

The TCP header is 20 bytes long.
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
**ipsumdump (subset)**

'Ipsumdump' reads IP packets from tcpdump(1) files, or network interfaces, and summarizes their contents in an ASCII file.

Usage: ipsumdump [DATA OPTIONS] [-i DEVNAMES | FILES] > SUMMARYFILE

**General data options:**
- `-t, --timestamp` Include packet timestamp.
- `-T, --first-timestamp` Include flow-begin timestamp.
- `-c, --packet-count` Include packet count (usually 1).
  - `--wire-length` Include wire length (with link header/trailer).
  - `--link` Include link number (NLANR/NetFlow).

**Ethernet data options:**
- `--eth-src` Include Ethernet source address.
- `--eth-dst` Include Ethernet destination address.

**IP data options:**
- `-s, --src` Include IP source address.
- `-d, --dst` Include IP destination address.
- `-l, --length` Include IP length.
- `-p, --protocol` Include IP protocol.
- `-g, --fragment` Include IP fragment flags ('F' or '.').
- `-G, --fragment-offset` Include IP fragment offset.
  - `--ip-id` Include IP ID.
  - `--ip-sum` Include IP checksum.
  - `--ip-opt` Include IP options.
  - `--ip-ttl` Include IP time to live.
  - `--ip-tos` Include IP type of service.
  - `--ip-hl` Include IP header length.
  - `--capture-length` Include length of captured IP data.
tcpdump (subset)

tcpdump version 4.9.0
libpcap version 1.6.2
OpenSSL 1.0.1t 3 May 2016
Usage: tcpdump [-aAbdDefHIJKlLnNOPqStuUvXX#] [ -B size ] [ -c count ]
[ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
[ -i interface ] [ -j tstamptype ] [ -M secret ] [ --number ]
[ -Q in|out|inout ]
[ -r file ] [ -s snaplen ] [ --time-stamp-precision precision ]
[ --immediate-mode ] [ -T type ] [ --version ] [ -v file ]
[ -w file ] [ -W filecount ] [ -y datalinktype ] [ -z postrotate-command ]
[ -Z user ] [ expression ]
tcpdump (subset)

NAME
tcpdump - dump traffic on a network

SYNOPSIS
tcpdump [ -AbdDefHIJKlNOpqStuVwxY ] [ -B buffer_size
[ -c count ]
[ -C file_size ] [ -G rotate_seconds ] [ -F file ]
[ -i interface ] [ -J tstamp_type ] [ -m module ] [ -M secret ]
[ --number ] [ --out[format] ]
[ -r file ] [ -v file ] [ -s snaplen ] [ -T type ] [ -W file ]
[ -W filecount ]
[ -E [ipaddr algo:secret,... ]
[ -y datalinktype ] [ -z postrotate-command ] [ -z user ]
[ --time-stamp-precision=tstamp_precision ]
[ --immediate-mode ] [ --version ]
[ expression ]

description
Tcpcap prints out a description of the contents of packets on a network interface that match the boolean expression; the description is preceded by a time stamp, printed, by default, as hours, minutes, seconds, and fractions of a second since midnight. It can also be run with the -w flag, which causes it to save the packet data to a file for later analysis, and/or with the -r flag, which causes it to read from a saved packet file rather than to read packets from a network interface. It can also be run with the -v flag, which causes it to read a list of saved packet files. In all cases, only packets that match expression will be processed by tcpdump.

Tcpcap will, if not run with the -c flag, continue capturing packets until it is interrupted by a SIGINT signal (generated, for example, by typing your interrupt character, typically control-C) or a SIGTERM signal (typically generated with the kill(1) command); if run with the -c flag, it will capture packets until it is interrupted by a SIGINT or SIGTERM signal or the specified number of packets have been processed.
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)

1:513(512) ack 1 win 32256 (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)

```markdown
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)
- `p` don't capture in promiscuous mode
- `k` start capturing immediately (def: do nothing)
- `s` update packet display when new packets are captured
- `l` turn on automatic scrolling while -S is in use
- `I` capture in monitor mode, if available
- `B <buffer size>` size of kernel buffer (def: 2MB)
- `y <link type>` link layer type (def: first appropriate)
- `D` print list of interfaces and exit
- `L` print list of link-layer types of iface and exit

Capture stop conditions:
- `c <packet count>` stop after n packets (def: infinite)
- `a <autostop cond.>` ... duration:NUM - stop after NUM seconds
  filesize:NUM - stop this file after NUM KB
  files:NUM - stop after NUM files

Capture output:
- `b <ringbuffer opt.>` ... duration:NUM - switch to next file after NUM secs
  filesize:NUM - switch to next file after NUM KB
  files:NUM - ringbuffer: replace after NUM files

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)
- `N <name resolve flags>` enable specific name resolution(s): "mnlC"
```
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

Monitor integrated with a bridge

- Host A
- Bridge/Monitor
- Host B

Multicast switch

- Host A
- Host B
- Switch
- Host C
- Monitor
Splitting a point-to-point link

Router A  --  Monitor  --  Router B

Line card that does packet sampling

Router A