

oftrace:
An OpenFlow Debugging and
Analysis Tool

Rob Sherwood

Yiannis Yiakoumis

Stanford Clean Slate Lab

June 4, 2009

Problem 1/2: Wireshark Hurts My Brain

The screenshot shows the Wireshark interface with a list of network packets. The interface includes a menu bar (Edit, View, Go, Capture, Analyze, Statistics, Help), a toolbar with various icons, and a search bar with 'Expression...', 'Clear', and 'Apply' buttons.

Time	Source	Destination	Protocol	Info
1 0.000000	171.67.74.62	171.67.74.56	TCP	60198 > 6633 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 WS=0 TSV=0
2 0.000009	171.67.74.56	171.67.74.62	TCP	6633 > 60198 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=0 TSV=0
3 0.001000	171.67.74.62	171.67.74.56	TCP	60197 > 6633 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 WS=0 TSV=0
4 0.001003	171.67.74.56	171.67.74.62	TCP	6633 > 60197 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=0 TSV=0
5 0.001005	171.67.74.62	171.67.74.56	TCP	60198 > 6633 [ACK] Seq=1 Ack=1 Win=33580 Len=0
6 0.001052	171.67.74.56	171.67.74.62	OFFP	Hello (SM) (8B)
7 0.001999	171.67.74.62	171.67.74.56	TCP	60197 > 6633 [ACK] Seq=1 Ack=1 Win=33580 Len=0
8 0.002028	171.67.74.56	171.67.74.62	OFFP	Hello (SM) (8B)
9 0.002998	171.67.74.62	171.67.74.56	TCP	60196 > 6633 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 WS=0 TSV=0
10 0.003002	171.67.74.56	171.67.74.62	TCP	6633 > 60196 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=0 TSV=0
11 0.003998	171.67.74.62	171.67.74.56	TCP	60196 > 6633 [ACK] Seq=1 Ack=1 Win=33580 Len=0
12 0.004029	171.67.74.56	171.67.74.62	OFFP	Hello (SM) (8B)
13 0.004997	171.67.74.62	171.67.74.56	TCP	60195 > 6633 [SYN] Seq=0 Win=32768 Len=0 MSS=1460 WS=0 TSV=0
14 0.005001	171.67.74.56	171.67.74.62	TCP	6633 > 60195 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=0 TSV=0
15 0.005996	171.67.74.62	171.67.74.56	OFFP	Hello (SM) (8B)
16 0.006002	171.67.74.56	171.67.74.62	TCP	6633 > 60198 [ACK] Seq=9 Ack=9 Win=5888 Len=0
17 0.006003	171.67.74.62	171.67.74.56	TCP	60195 > 6633 [ACK] Seq=1 Ack=1 Win=33580 Len=0
18 0.006036	171.67.74.56	171.67.74.62	OFFP	Hello (SM) (8B)
19 0.006055	171.67.74.56	171.67.74.56	OFFP	Sendmail Daemon (550) (8B)

Packet 1 (74 bytes on wire, 74 bytes captured)

Ethernet II, Src: AlaxalaN_78:31:d8 (00:12:e2:78:31:d8), Dst: Supermic_b0:61:16 (00:30:48:b0:61:16)

TCP Internet Protocol, Src: 171.67.74.62 (171.67.74.62), Dst: 171.67.74.56 (171.67.74.56)

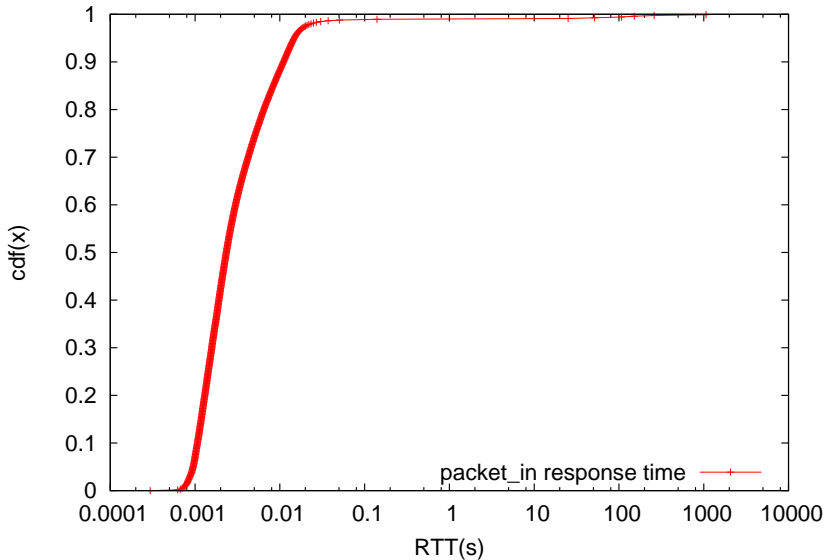
Transmission Control Protocol, Src Port: 60198 (60198), Dst Port: 6633 (6633), Seq: 0, Len: 0

```
00 30 48 b0 61 16 00 12 e2 78 31 d8 08 00 45 00 .0H.a... .x1...E.
00 3c b0 46 40 00 40 06 9f 78 ab 43 4a 3e ab 43 .<.F@.@. .x.CJ..C
4a 38 eb 26 19 e9 ce 3a ed 93 00 00 00 00 a0 02 J8.&....:.....
00 00 1f 2c 00 00 02 04 05 b4 01 03 03 00 01 01 .....:.....
08 0a 00 00 00 00 00 00 00 00 ..
```

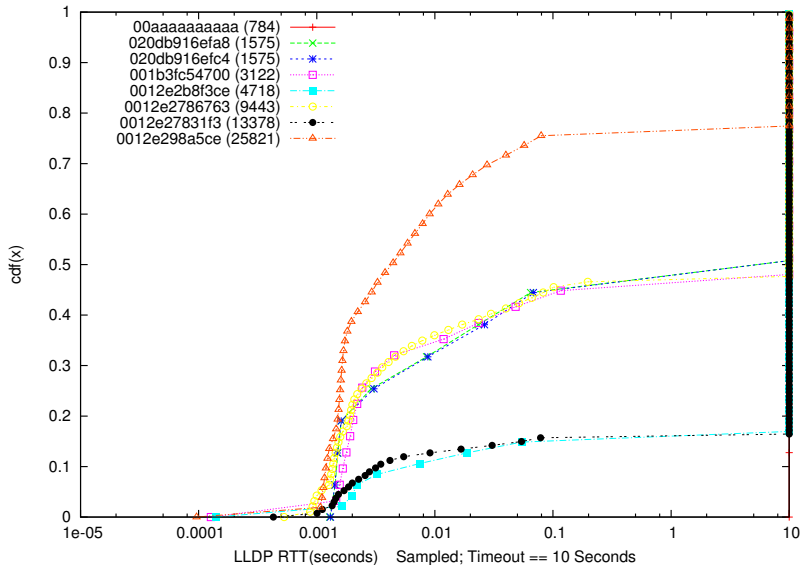
Problem 2/2: Too Much Data

- Apply logical sanity checks
 - how often do switches send bogus packet?
- Calculate aggregate statistics
- Correlate relations between packet types
 - what is NoX's processing time?
 - how do LLDP packets take?

Answer: Nox Processing Time



Answer: LLDP RTT Distribution



What Magic Is This? oftrace!

- libpcap parsing library
- Write programmatic queries
- Undoes/fixes packet reordering, duplication
- Simple 4 call API
- SWIG front end
 - use your favorite language - not mine

Design

- Measure everything, all of the time
 - I always forget to measure **something**
- No quantum effects
- libpcap has better timestamps
- Works for all controllers

- 1 oftrace * oftrace_open(char * pcapfile);
- 2 const openflow_msg * oftrace_next_msg(oftrace * oft, uint32_t ip, int port);
- 3 int oftrace_rewind(oftrace * oft);
- 4 double oftrace_progress(oftrace * oft);

Oftrace Mesg Struct

Raw Data + Convenience Pointers

- 1 ethernet header
- 2 ip header
- 3 tcp header
- 4 openflow header
- 5 (union) openflow message type
- 6 embedded packets
- 7 captured size, total size, etc.

Example: `Ildp_stats.py`

Tools

- ofdump/pyofdump : lists types and sizes of OF msgs
- ofstats/pyofstats: controller response time
 - first graph
- lldp_stats: discovery RTT
 - second graph
- ??

Implementation

- 700 lines of C (core library)
- Re-ordering is a pain
- DLT_10MB vs. DLT_LINUX_SLL
- Sequence number wrapping
- Swig : accessing raw data
 - cdata

TODO

- Better filter support
- Remove PAWS bugs (fixed?)
- Live capture mode

Conclusion

- Analyzing OpenFlow can be painful
- oftrace can help you
- I've written some tools that I think are useful
- Please contribute your own
- <http://www.openflowswitch.org/wk/Liboftrace>