

A Study of FDDI and Ethernet Traffic on the LANL Network Backbone

Eric Weigle[†] and Wu-chun Feng^{†‡}
{ehw, feng}@lanl.gov

[†]Computer & Computational Sciences Division
Los Alamos National Laboratory
Los Alamos, NM 87545

[‡]Department of Computer & Information Science
Ohio State University
Columbus, OH 43210

Abstract

We discuss several network traffic traces taken on the Los Alamos National Laboratory backbone between 2001 and 2002. These traces, approximately 1.2 terabytes of binary data, will soon be released for public study in an anonymous form. This paper will discuss the collection methodology, tools, and an analysis showing some interesting features of the traffic.

Keywords: network traffic collection, traffic traces, TICKET, tcpdump, libpcap

1. Introduction

Between February 2001 and February 2002 we had permission to collect headers from any network packets traversing the LANL backbone. We collected several traces during this period, spanning a total of approximately 90 days. During this period backbone moved from being a FDDI network to a Gigabit Ethernet network and we have data before and after the switch. The tap point was on the link just inside the LANL firewall, and thus most of the observed packets are ingress/egress traffic.

Traffic from the FDDI network was collected using a modified version of `tcpdump` [1]. Our version saved only the subset of packet headers in which we were interested, primarily for privacy and security reasons but also to reduce the size of the output file.

We intended to capture the Ethernet data in a similar way, but unfortunately (at the time) several issues forced us to come up with a “home-brew” mechanism. The solution was a prerelease version of the TICKET [3] software. TICKET is a passive monitor implemented within the Linux kernel to enable high-speed traffic collection on commodity hardware and was presented in PAM2002.

2. Trace Overview

We give a high-level overview of the data, its format, and a few caveats. In-depth analysis is impossible given space constraints.

2.1. Trace Data and Format

The traces are a set of fixed length records in a flat file format with an 8-byte header per file. This contains a 32-bit ‘magic’ number to identify the file and configuration information including:

- what type of data is in the file (FDDI, Ethernet, or anonymization mappings)
- what type of timestamps are used (cycles, nanoseconds, microseconds)
- whether or not the file is anonymized
- the endianness of the data (big-endian or little-endian)
- what version of the file format is being used.

The records are simply a binary `write()` of the C structure used to represent the data in memory. Table 1 lists the fields in this structure, which are exactly the subset of packet headers we collected, along with their sizes in bytes. Both the FDDI and TICKET traces use the Ethernet data structure, which is 42 bytes in size. Unused fields are set to zero and the wasted space is reclaimed via compression. Memory-mapping these files makes them easy to work with.

Level	Field	Size	FDDI	Eth.
Raw	Timestamp	8	•	•
Raw	Packet Length	2	•	•
MAC	TOS/Length	2	•	•
MAC	Source	6	•	•
MAC	Destination	6	•	•
IP	Source Address	4	•	•
IP	Dest. Address	4	•	•
IP	Packet Length	2		•
IP	Header Length	1		•
IP	Protocol	1		•
UDP/TCP	Source Port	2	•	•
UDP/TCP	Dest. Port	2	•	•
UDP/TCP	Data Length	2		•

Table 1: Packet Header Data Collected

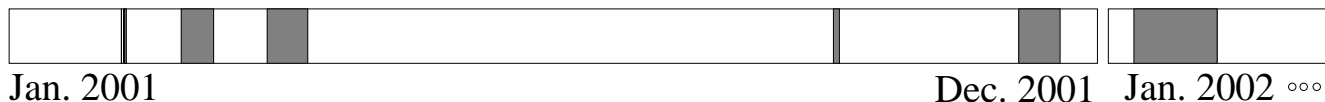


Figure 1: Graphical view of Trace Periods

#	Type	Year	Time			Size			
			Start	Stop	Length	Files	Raw	Compressed	Ratio
1	FDDI	2001	02/06 16:01	02/06 16:24	0:22:41	4	622.4 MB	101.5 MB	16.3%
2	FDDI	2001	02/07 12:29	02/07 13:20	0:50:50	6	1148 MB	199.7 MB	17.4%
3	FDDI	2001	02/07 13:49	02/07 16:34	2:44:52	19	3656 MB	632.5 MB	17.3%
4	FDDI	2001	02/07 17:32	02/08 07:39	14:06:37	16	3126 MB	513.0 MB	16.4%
5	FDDI	2001	02/26 17:30	03/09 14:05	10d 20:34:24	686	134.2 GB	23.08 GB	17.2%
6	FDDI	2001	03/27 13:33	04/10 04:14	13d 14:41:23	702	137.3 GB	22.74 GB	16.6%
7	Enet	2001	10/03 11:37	10/05 11:46	2d 00:08:50	185	34.27 GB	6.675 GB	19.5%
8	Enet	2001	12/04 15:05	12/18 12:17	13d 21:11:26	1092	203.4 GB	51.17 GB	25.2%
9	Enet	2002	01/08 14:00	02/25 12:06	47d 22:05:38	4000	744.1 GB	151.2 GB	20.3%

Table 2: Trace Periods and Lengths

Tools are included in the publicly available TICKET software distribution [2] to read and write these files, as is a tool to convert between this format and the more complete but less efficient `libpcap` file format used by `tcpdump`.

Table 2 and Figure 1 give further high-level information about the traces.

2.2. Good and Bad Aspects of Traces

Taking these traces was a learning experience. While we avoided many problems we observed in prior traces, we encountered several other issues. These relate to the way traffic was multiplexed through the hub, the NIC handles small packets, PC hardware is designed, timestamping was performed, and the way host clocks are synchronized to a global 'true' clock. Each could skew the data slightly; but we believe it is still highly accurate.

3. Trace Features

Trace 8 is a convenient size for analysis—two weeks long, and the compressed version fits on a single 80GB hard drive. It contains packets to or from 744,266 unique IP addresses. Of these 639,696 were external, 65,533 were internal on the main LANL class B network, and 37,037 were on other traffic blocks allocated to LANL.

Figure 2 shows the data observed over the course of this trace. The data has been reduced by a factor of about 10^7 to a granularity of one minute.

The 65,533 figure is interesting as it shows an almost 100% utilization of the address space. This is from aggressive internal network security scans to detect and eliminate potential holes. Only a subset of hosts is actually 'alive.'

4. Conclusion and Future Work

We have discussed a large set of traces (soon to be made public in anonymized form) and a few of their aspects. Due to space constraints we have not fleshed out the collection methodology, motivation, anonymization process, or discussed many of the interesting features in the data. These will, of course, be expanded upon in the final version.

References

- [1] Lawrence Berkeley National Laboratory Network Research. TCPDump: the Protocol Packet Capture and Dumper Program. <http://www.tcpdump.org/>.
- [2] E. Weigle. The TICKET Software Distribution, December 2002. <http://public.lanl.gov/radiant/software/ticket.html>.
- [3] E. Weigle and W. Feng. TICKETing High-Speed Traffic with Commodity Hardware and Software. In *Proceedings of the Third Annual Passive and Active Measurement Workshop (PAM2002)*, March 2002.

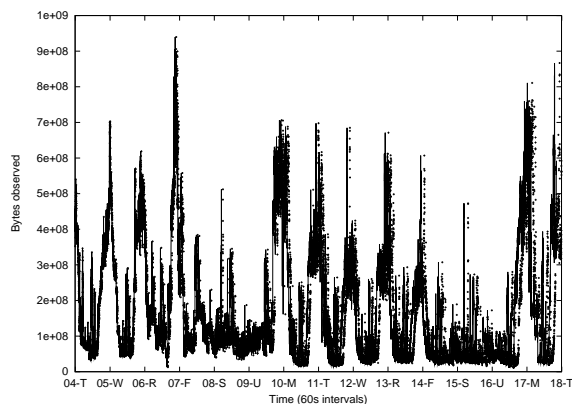


Figure 2: Bytes Observed in Trace 8