Snoop Version 2 Packet Capture File Format

Status of this Memo
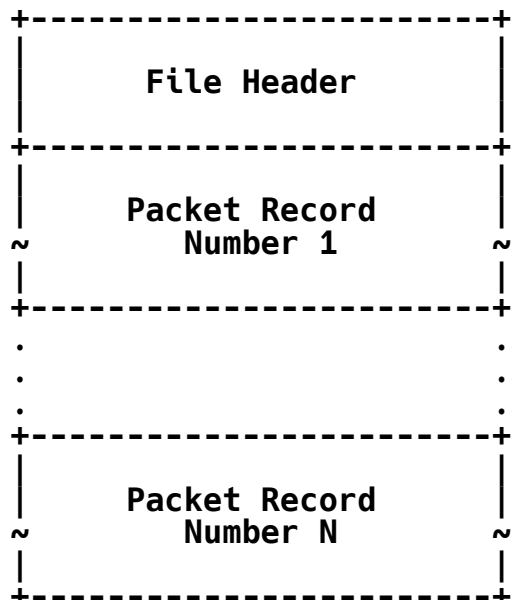
Abstract

   This paper describes the file format used by "snoop", a packet
   monitoring and capture program developed by Sun. This paper is
   provided so that people can write compatible programs to generate and
   interpret snoop packet capture files.

1. Introduction

   The availability of tools to capture, display and interpret packets
   traversing a network has proven extremely useful in debugging
   networking problems. The ability to capture packets and store them
   for later analysis allows one to de-couple the tasks of collecting
   information about a network problem and analysing that information.
   The "snoop" program, developed by Sun, has the ability to capture
   packets and store them in a file, and can interpret the packets
   stored in capture files. This RFC describes the file format that the
   snoop program uses to store captured packets. This paper was written
   so that others may write programs to interpret the capture files
   generated by snoop, or create capture files that can be interpreted
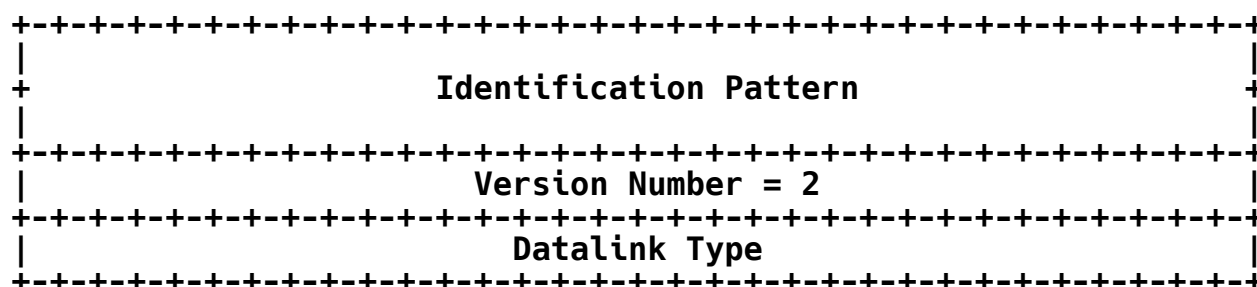   by snoop.

2. File Format

    The snoop packet capture file is an array of octets structured as
    follows:

```
            +------------------------+
            |                        |
            |      File Header        |
            |                        |
            +------------------------+
            |                        |
            |    Packet Record        |
            ~       Number 1          ~
            |                        |
            +------------------------+
            .                        .
            .                        .
            .                        .
            +------------------------+
            |                        |
            |    Packet Record        |
            ~       Number N          ~
            |                        |
            +------------------------+
```

    The File Header is a fixed-length field containing general
    information about the packet file and the format of the packet
    records it contains. One or more variable-length Packet Record
    fields follow the File Header field. Each Packet Record field holds
    the data of one captured packet.

3. File Header

    The structure of the File Header is as follows:

```
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                                                              |
  +                   Identification Pattern                     +
  |                                                              |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                    Version Number = 2                        |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                      Datalink Type                           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Identification Pattern:

> A 64-bit (8 octet) pattern used to identify the file as
> a snoop packet capture file. The Identification Pattern
> consists of the 8 hexadecimal octets:
>
>> 73 6E 6F 6F 70 00 00 00
>
> This is the ASCII string "snoop" followed by three null
> octets.

Version Number:

> A 32-bit (4 octet) unsigned integer value representing
> the version of the packet capture file being used. This
> document describes version number 2. (Version number 1
> was used in early implementations and is now obsolete.)

Datalink Type:

> A 32-bit (4 octet) field identifying the type of
> datalink header used in the packet records that follow.
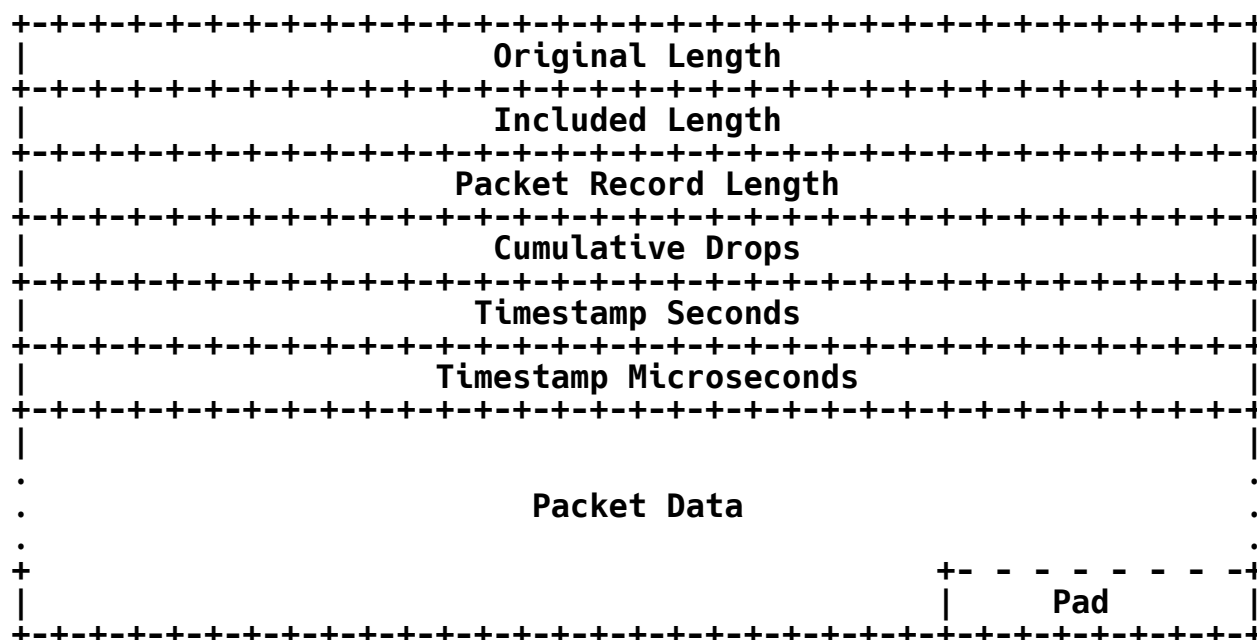> The datalink type codes are listed in the table below:

Datalink Type	Code
IEEE 802.3	0
IEEE 802.4 Token Bus	1
IEEE 802.5 Token Ring	2
IEEE 802.6 Metro Net	3
Ethernet	4
HDLC	5
Character Synchronous	6
IBM Channel-to-Channel	7
FDDI	8
Other	9
Unassigned	10 - 4294967295

4. Packet Record Format

Each packet record holds a partial or complete copy of one packet as
well as some descriptive information about that packet. The packet
may be truncated in order to limit the amount of data to be stored in
the packet file. In addition, the packet record may be padded in
order for it to align on a convenient machine-dependent boundary.
Each packet record holds 24 octets of descriptive information about
the packet, followed by the packet data, which is variable-length,
and an optional pad field. The descriptive information is structured

as six 32-bit (4-octet) integer values.

The structure of the packet record is as follows:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Original Length                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Included Length                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Packet Record Length                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Cumulative Drops                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Timestamp Seconds                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Timestamp Microseconds                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                          Packet Data                          .
.                                                               .
+                                      +- - - - - - -+
|                                      |     Pad     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Original Length

> 32-bit unsigned integer representing the length in
> octets of the captured packet as received via a network.

Included Length

> 32-bit unsigned integer representing the length of the
> Packet Data field. This is the number of octets of the
> captured packet that are included in this packet record.
> If the received packet was truncated, the Included
> Length field will be less than the Original Length
> field.

Packet Record Length

> 32-bit unsigned integer representing the total length of
> this packet record in octets. This includes the 24
> octets of descriptive information, the length of the
> Packet Data field, and the length of the Pad field.

Cumulative Drops

>   32-bit unsigned integer representing the number of
>   packets that were lost by the system that created the
>   packet file between the first packet record in the
>   file and this one. Packets may be lost because of
>   insufficient resources in the capturing system, or for
>   other reasons. Note: some implementations lack the
>   ability to count dropped packets. Those
>   implementations may set the cumulative drops value to
>   zero.

Timestamp Seconds

>   32-bit unsigned integer representing the time, in
>   seconds since January 1, 1970, when the packet arrived.

Timestamp Microseconds

>   32-bit unsigned integer representing microsecond
>   resolution of packet arrival time.

Packet Data

>   Variable-length field holding the packet that was
>   captured, beginning with its datalink header. The
>   Datalink Type field of the file header can be used to
>   determine how to decode the datalink header. The length
>   of the Packet Data field is given in the Included Length
>   field.

Pad

>   Variable-length field holding zero or more octets that
>   pads the packet record out to a convenient boundary.

5. Data Format

All integer values are stored in "big-endian" order, with the high-
order bits first.

6. Security Considerations

Security issues are not discussed in this memo.

Authors' Addresses

    Brent Callaghan
    Sun Microsystems, Inc.
    2550 Garcia Avenue
    Mailstop UMTV05-44
    Mountain View, CA 94043-1100

    Phone: 1-415-336-1051
    EMail: brent.callaghan@eng.sun.com

    Robert E. Gilligan
    Sun Microsystems, Inc.
    2550 Garcia Avenue
    Mailstop UMTV05-44
    Mountain View, CA 94043-1100

    Phone: 1-415-336-1012
    EMail: bob.gilligan@eng.sun.com