Packet Stats Design Requirements

Author: Jad Naous jnaous@stanford.edu

1.0 Goal

The goal of the project is to produce a simple design that can be incorporated into a switch to collect fairly accurate statistics on arrival/departure/drop times of packets on output queues. The data is aggregated into a raw Ethernet packet which is then forwarded to a specified MAC address or broadcast to be processed further.

2.0 Specifications

This section lists the requirements in detail.

2.1 Definitions

Event: a record of a packet arrival/departure/drop consisting of the event time and packet length, and any other information necessary to process the event.

User: the host CPU performing operations on behalf of an administrator

Time difference: Time elapsed since last event

Timestamp: Time elapsed since last timer reset

2.2 Statistics Capture and Timing

1- The design shall be able to capture all events on all output queues in the switch.
2- The user shall be able to set which output queues to monitor
3- The user shall be able to reset the timer
4- Timer resolution: The user shall be able to set the timer resolution/precision as (2^c)*8ns where ‘c’ can be 0 to 7 (i.e. resolutions from 8 ns to 1.024us)
5- Packet stats capture shall be able to run a very long time (years) without any intervention from the user
6- The user shall be able to see the number of events that were missed/could not be recorded.

2.3 Events Packet and Transmission

1- The captured statistics shall be written into a UDP packet

2- The user shall be able to program the packet's Ethernet, IP and UDP headers
3- The packet shall be self contained, needing no information from another packet
4- The user shall be able to select the port(s) to send the events packets from
5- The user shall be able to force the packet to be transmitted even if it is not complete

2.4 Events Format

There will be 4 event types:

1- Arrival/Departure/Drop (short) Events will have the following format

Event Type	Output Queue	packet len in 64-bit words	Time difference
31 30	29 27	26 19	18 0

In the case of 128ns timer resolution, this format will allow time differences of 52.4 ms before wrap-around. Also note that the packet length here is in words of 64-bits.

2- Timestamp Event: used to maintain independence between packets and to solve the Time Difference wrap-around problem

Event Type	Timestamp
63 62	61 0

2.5 Packet Format

Following the Ethernet, IP, and UDP headers, there are 4 reserved bits, then 4 bits indicating the version of the event packet capture system, then 8 bits indicating the number of event types excluding the Timestamp event type. This is followed by a 32 bit sequence number (starting at 0) and a list of the queue sizes before the first short event. A series of short events then follows. If the time difference since the last event is longer than 19 bits, a new Timestamp event is inserted.

<table>
<tr><th colspan="8">63</th><th colspan="8">0</th></tr>
<tr><td colspan="12">dst mac
48</td><td colspan="4">src mac hi
16</td></tr>
<tr><td colspan="8">src mac lo
32</td><td colspan="4">ethertype
16</td><td colspan="1">V
4</td><td colspan="1">L
4</td><td colspan="2">TOS
8</td></tr>
<tr><td colspan="4">total length
16</td><td colspan="4">id
16</td><td colspan="4">flags+frag off
16</td><td colspan="2">TTL
8</td><td colspan="2">PROT
8</td></tr>
<tr><td colspan="4">chksum
16</td><td colspan="8">src ip
32</td><td colspan="4">dst ip hi
16</td></tr>
<tr><td colspan="4">dst ip lo
16</td><td colspan="4">UDP src
port - 16</td><td colspan="4">UDP dst
port - 16</td><td colspan="4">UDP len
16</td></tr>
<tr><td colspan="4">UDP cksum
16 (cleared)</td><td colspan="1">0</td><td colspan="1">E
V</td><td colspan="2">num
mon
evts</td><td colspan="8">pkt sequence number</td></tr>
<tr><td colspan="8">output queue size 0</td><td colspan="8">output queue size 1</td></tr>
<tr><td colspan="8">output queue size 2</td><td colspan="8">output queue size 3</td></tr>
<tr><td colspan="8">output queue size 4</td><td colspan="8">output queue size 5</td></tr>
<tr><td colspan="8">output queue size 6</td><td colspan="8">output queue size 7</td></tr>
<tr><td colspan="16">Timestamp event</td></tr>
<tr><td colspan="8">Short event</td><td colspan="8">Short event</td></tr>
<tr><td colspan="8">Short event</td><td colspan="8">Short event</td></tr>
<tr><td colspan="8">Short event</td><td colspan="8">Timestamp event high</td></tr>
<tr><td colspan="8">Timestamp event low</td><td colspan="8">Short event</td></tr>
<tr><td colspan="8">Short event</td><td colspan="8">Short event</td></tr>
</table>

2.6 Bandwidth requirements

A single 1500 bytes events packet will have enough space for 1 timestamp event and ~350 short events. If there are 2 events every 1 us for every queue, then every events packet will have enough space for around 184 us per queue. If all queues are being monitored, then every events packet will have around 46 us worth of statistics.

In the single queue case, the bandwidth requirement is 1500*8/184us=65.2 Mbps. When monitoring all queues, the bandwidth requirement is 261 Mbps <u>on average</u>. Monitoring all queues can stress the links when there is a lot of activity.