Awk script to get end-to-end delays from *ns*-2 tracefiles

Lloyd Wood

Lloyd Wood wrote a simple awk script that parses the packet tracefiles generated by the network simulator *ns*-2. The script computes end-to-end delays from source to destination for a packet that crosses the network and is received at its intended destination, and produces a latency time for each packet. This is useful to show how delay changes significantly due to routing or other changes, if say a packet is sent at regular intervals across the network, say via a CBR agent firing once a second.

Lloyd used his script to model changes in delay due to movement of non-geostationary satellites, showing how orbital motion affected end-to-end latency of packet arrivals. Graphing results created with this script are included in:

> **Revisiting elliptical satellite orbits to enhance the *O3b* constellation,** Lloyd Wood, Yuxuan Loua and Opeoluwa Olusola, Journal of the British Interplanetary Society, vol. 67, pp. 110-118, March 2014.
>
> **Satellite constellation networks**, Lloyd Wood, Chapter 2, Internetworking and Computing over Satellite Networks, Yongguang Zhang (ed.), Kluwer Academic Press, ISBN 1-4020-7424-7, pp. 13-34, March 2003.
>
> **Internetworking with satellite constellations**, Lloyd Wood, thesis for award of Doctor of Philosophy (PhD), University of Surrey, June 2001.
>
> **Managing diversity with handover to provide classes of service in satellite constellation networks**, Lloyd Wood, George Pavlou and Barry Evans, 19th International Communication Satellite Systems Conference (ICSSC '01), vol. 3, session 35, no. 194, Toulouse, France, April 2001.
>
> **Effects on TCP of routing strategies in satellite constellations**, Lloyd Wood, George Pavlou and Barry Evans, IEEE Communications Magazine, vol. 39, no. 3, pp. 172-181, March 2001.

Lloyd originally made this script public on the *ns*-users mailing list:
http://www.isi.edu/nsnam/archive/ns-users/webarch/1999/msg02093.html
Lloyd Wood ‘Re: About packet delays’, ns-users mailing list, 30 July 1999.

Others have since found it useful, and many requests have been made for the script in the years since, which is why this introduction has been written. The `delay.awk` script is on the following page ready to copy into a textfile. To run it on the tracefile `out.tr`, you would type e.g. `awk -f delay.awk < out.tr`

```
BEGIN {
   # simple awk script to generate end-to-end packet lifetime statistics
   # in a form suitable for plotting with xgraph.
   # Lloyd Wood, July 1999 and December 2014.
   # http://personal.ee.surrey.ac.uk/Personal/L.Wood/ns/

   highest_packet_id = 0;
}

{
   action = $1;
   time = $2;
   node_1 = $3;
   node_2 = $4;
   src = $5;
   flow_id = $8;
   node_1_address = $9;
   node_2_address = $10;
   seq_no = $11;
   packet_id = $12;


   if ( packet_id > highest_packet_id ) highest_packet_id = packet_id;

   # getting start time is not a problem, provided you're not starting
   # traffic at 0.0.
   # could test for sending node_1_address or flow_id here.
   if ( start_time[packet_id] == 0 )  start_time[packet_id] = time;

   # only useful for small unicast where packet_id doesn't wrap.
   # checking receive means avoiding recording drops
   if ( action != "d" ) {
      if ( action == "r" ) {
         # could test for receiving node_2_address or flow_id here.
         end_time[packet_id] = time;
      }
   } else {
      end_time[packet_id] = -1;
   }
}
END {
    for ( packet_id = 0; packet_id <= highest_packet_id; packet_id++ ) {
       start = start_time[packet_id];
       end = end_time[packet_id];
       packet_duration = end - start;

       if ( start < end ) printf("%f %f\n", start, packet_duration);
   }
}
```