Sampled Streaming

draft-gray-sampled-streaming-02

IETF 106, Singapore
November 2019

Authors:
Andrew Gray (Charter Communications)
LJ Wobker (Cisco Systems)
Problem trying to solve

We need a way to perform packet captures of links in a consistent and configurable manner.

- Inline sampling devices have to be upgraded every time link speed changes, typically lag behind, and are very expensive.
- Traditional methods of doing this tend to be limited in capabilities, and require significant on-box processing and/or control plane punting.
- Data planes are getting much faster – control planes not so much.
- We have huge datacenter infrastructures we can throw lots and lots of data at and do lots of compute – we just need to get it to them.
- We need to be able to have the device filter, and have sampling rates be variable between 1:1 and 1:<big number> sampling rates.
- We must be able to capture as many headers and as much of the payload as possible.
Nomenclature from the draft

Client

Receiver

Replicator

Point

Control/Negotiation

Sampled Packets

Normal Traffic Flow
Must haves

• Must be very light on the on-box CPU and control plane (preferably no work at all done at the on-box control plane after set up)
  • Processing terabits of traffic on the data plane – control planes can’t keep up, even at very low sampling rates (1:20,000 or worse).

• Must avoid ASIC recirculation or other issues that degrade performance as much as possible.

• We want one method to cover multiple vendors, capturing on servers, etc.
Nice to haves

- Generally a device forwarding traffic knows a lot more about a given packet than just the raw packet – getting that information off-box is useful.
  - Precise receive or transmit timestamp.
  - Ingress and/or egress ports or tunnels, or other actions taken.
- We would also like to be able to capture/sample packets that are dropped, in addition to those being forwarded.
  - Is QoS dropping the right things?
  - Are we seeing drops for other reasons?
  - Why is traffic being dropped or punted?
Why not IPFIX extensions, OAM extensions, etc.? 

- We need the raw packets.
- We would like to have a chance at getting packets that are being dropped, as well as forwarded.
- We need to be able to have different capture parameters for different types of traffic, and these can change relatively quickly.
- We are interested in whatever additional information we can get out of forwarding ASICs.
- Generally too dependent on control plane or on-box CPU.
Why not a fixed / specified metadata format?

We want to do this with “really fast” data planes...

• mutex: “really fast” and “mangle headers into a really specific format”
• different data planes will have different metadata available
  • possible also at different bit widths ... 
  • or different ordering ... or different other stuff ...

We want to be somewhat future-proof ...

• no way to know what stuff will be interesting in the future 
• ‘fast’ data planes will have more metadata tomorrow than today

A Replicator that can self-describe it’s data output

• can encode the data according to its own capabilities
• can tell the listener what that format is
Next steps

• The -01 version had code written to support (Replicator was a server, Point was effectively tcpdump, Client and Receiver were simple applications). The -02 version has a fair number of updates to the negotiation based on a lot of feedback, which has not been written into code yet.

• We have gotten a lot of good feedback after announcements on the mailing list, but it has generally been unicast.

• Looking for suggestions for improvement, additional use cases, or other votes of support.

• Some items being worked on/discussed:
  • The capture filter syntax is unique to this draft, which probably isn’t ideal.
  • Additional options to communicate what a forwarding ASIC is doing with a packet.
Q&A