~~Session~~ Substrate Protocol for User Datagrams (SPÜD)

spud@ietf.org

Brian Trammell (IAB IP Stack Evolution Program)

History

- We can't evolve the transport stack: narrow interfaces, even narrower constraints on packets by middleboxes on path

- We need to evolve the transport stack: low-latency services, interactive media (RTCWEB), opportunistic security (TCPINC)

- Also, we're encrypting everything, turning all those middleboxes we need to keep the network running into expensive heaters unless we come up with a Plan B.

History (in other words)

- This problem was effectively referred to us by the RTCWEB working group.
  - What draft-ietf-rtcweb-data channel says: "The encapsulation of SCTP over DTLS over ICE/UDP provides a NAT traversal solution together with confidentiality, source authentication, and integrity protected transfers."
  - What I read: "The stack is broken, please help!"

SEMI Workshop

- IAB/ISOC workshop on Stack Evolution in a Middlebox Internet, 26-27 January, Zurich

- One outcome: let's have a BoF to talk about UDP encapsulation and signaling for new transports (you are here)

- Another outcome: BarBoF on getting data about middlebox impairment in the Internet (HOPS on Sunday evening).

Initial Use Case: UDP firewall traversal

- Today UDP is often blocked ("99% of UDP is garbage") but volume of (good) UDP traffic grows, e.g. RTCWEB uses UDP for data and media.
- Hard to establish session/flow state at firewalls: block rules are stateless
- Need an *explicit* contract to establish state along the path (as today's *implicit* contract does with TCP)

Approach

- Use of identifiers ("tubes") beyond the five-tuple to link packets together
- Explicit signaling/negotiation of tube start/end
  - Indicate start/stop to middlebox
  - Confirm connection establishment by receiver

Potential Use Case: Application-Limited Flows

e.g. explicit indication of data rate for CBR traffic

- video traffic could provide maximum rate with current encoding
- network could expose rate shaping to simplify probing
- endhost/network could provide indication of sudden changes in bandwidth demand/offer

Potential Use Case: Low-Latency Services

explicit indication of loss- vs. latency-sensitive traffic

- tradeoff, *not* prioritization
- latency-sensitive traffic could be managed in a different queue or with use of AQM such as PIE/CoDel, but might see higher loss rates
- loss-sensitive traffic faces larger buffer delay, but lower loss rate
- provider decides about bandwidth sharing between both services, and might or might not expose this information

Potential Use case: Service Multiplexing

Explicit indication of relative flow priority, relative packet priority within a flow

- e.g. if service has multiple simultaneous transmission of video/audio/control data, interactive data would be prioritized within same service
- e.g. more important packets such as I-frames in video could be prioritized within same flow/tube

Generic mechanism

- Tube identifier + basic semantics on each packet
- In-band channel with extensible syntax to allow endpoints to signal traffic metadata (per-packet + per-tube) to each other, and devices on path
- Mechanism to allow on-path devices to signal back to either endpoint using the same in-band channel
- draft-hildebrand-spud-prototype defines an instance of this generic mechanism for experimentation

Constraints on information exposed

(1) **Information exposure is *declarative***

- *no negotiation*: path and endpoints expose properties independently
- lack of a2p roundtrips reduces latency impact
- no assumption what action will follow

Constraints on information exposed

(2) All entities may *trust* but *verify*

- Exposed information should be verifiable by endpoints
  - Spot checks should be sufficient
- The best way to prevent cheating is to remove the incentives to do so
- Lack of trust can be persistent

Constraints on information exposed

(3) Information must be *incrementally useful*

- i.e. need not be supported by all nodes on a path before a benefit is seen
- You must ignore (and *not* delete) what you don't understand
- You must assume you're not being understood

Haven't we been here before?

- There is a long history of path-to-endpoint and endpoint-to-path signaling in the IETF, very little of which has seen wide deployment.
  - ECN (though we're still trying!), DSCP, NSIS
- Why do this again?
  - Timing: growing use of encryption, linkage of transport evolution with limited signaling
  - Scope: keep the effort as restricted as possible.
  - Incentives: explicit attention to why endpoints (app/library/OS developers) and middleboxes would choose to play along.
  - The problems that led to past approaches haven't gone away.

Summary

- SPUD: new transport encapsulation + middlebox cooperation
- Initial use case: enable transport encaps over UDP in a middlebox+firewall-friendly way
- Constraints on additional information:
  - Declarations only, no negotiation
  - Endpoints/middleboxes may trust, but can verify
  - Incremental usefulness, no mandatory minimum vocabulary