Potential Updates to UDPCL

BRIAN SIPOS

RKF ENGINEERING SOLUTIONS

IETF110

Goals for UDPCLbis

Maintain backward compatibility: one bundle—one datagram

- Allow existing implementations to be adapted.
- Should legacy logic to handle BPv6 be kept in?

Improve unspecified or weak points of UDPCL:

- Handle bundles which are barely larger than the path MTU.
- Defines how unicast and multicast are to be used.

Add long-term extensibility and interoperable CL security:

- Provide a place to put extensions, keeping the overall send-and-forget strategy

Further areas for improvement:

- Bundle packing within a single datagram? This could improve efficiencies for certain workflows.
- Can a UDP conversation be bidirectional? A “polling” conversation would allow NAT traversal.

Independent Draft

Last submission was in `draft-sipos-dtn-udpcl-00`.

- Late edits missed the Datatracker freeze before this IETF.

Behaviors:

- Clarifies IP address (multicast) and UDP port uses.
- Retains keepalive and BPv6 handling.
- Adds BPv7, CL-fragmented transfer, and optional-to-use DTLS.
- Inherits the same PKIX certificate profile and use case as TCPCLv4.

Open issues discussed on next slide.

Next Steps

Issues documented at https://github.com/BSipos-RKF/dtn-bpbis-udpcl/issues

- Allow bundle packing in one datagram? All of the message encodings except keepalive support this.
- Recommend an upper bound on fragmented transfers? Could use `draft-templin-dtn-ltpfrag` as guidance. Loss of any one fragment leads to loss of the whole transfer.
- Possible extension to allow path-MTU discovery (RFC 8899) separate from transfers. This could be extended to bundle-level MTU discovery; is that a necessary thing?
- Possible extension to advertise return-path bundle transfers, allowing use behind NAT like TCPCL.
- Could recommend to use same local and peer address-and-port to simplify conversation tracking.

Adoption as WG Draft?