Invariants

QUIC @IETF 100, Singapore

Invariants - Purpose

Document the things that are QUIC **generically**

Guiding Principle: only include enough information for

a) negotiation of a QUIC version
b) routing of QUIC packets independent of version

Anything that does not contribute to those goals can change

Current Invariants (Section 5.8)

ANYTHING ELSE ~~CAN~~ WILL CHANGE!

- the location of the header form flag,
- the location of the Connection ID flag in short headers,
- the location and size of the Connection ID field in both header forms,
- the location and size of the Version field in long headers,
- the location and size of the Packet Number field in long headers, and
- the type, format and semantics of the Version Negotiation packet.

Issues

How does QUIC multiplex with RTP and friends?

- i.e., how do the current invariants affect this use case?
- I'll walk through this in more detail

We might add to invariants for middlebox things

- We will discuss this later

Multiplexing QUIC and RFC 7983

People want to use QUIC for things like WebRTC

- multiplexing with STUN, TURN and SRTP
- not using QUIC *for* RTP as in draft-rtpfolks-...

QUIC assumes use of all values of the first octet

*credit to Bernard and draft-aboba-avtcore-quic-multiplexing for the options

Option 1 - Rely on Crypto

Observation: All these protocols use crypto

All packets have some sort of authentication

- QUIC packets (except Version Negotiation) use AEAD
- STUN uses the MESSAGE-INTEGRITY attribute
- SRTP has a MAC

This might be a bad option, but it shows that multiplexing is always possible

Option 2 - Change QUIC Invariants

Different QUIC invariants might allow *these* protocols to be multiplexed with QUIC

Spend two bits and confine QUIC first octet to 192-255

Any decision needs to be made NOW

0-3	STUN
16-19	ZRTP
20-63	DTLS
64-79	TURN Channel
128-191	SRTP
192-255	QUIC

Option 3 - Shim

Observe that these cases for multiplexing QUIC include prior arrangement (such as SDP offer/answer)

Define an octet that signifies QUIC, e.g. 192

Only use that in these cases

0-3	STUN
16-19	ZRTP
20-63	DTLS
64-79	TURN Channel
128-191	SRTP
192	QUIC

Option 4 - Avoid Conflict

Observations:

- long packets are only used during the handshake
- QUIC provides the same service as DTLS
- ZRTP isn't used that much
- TURN channels ??? WTF RFC 7983 ???

0-3	STUN
0-127	QUIC short
~~20-63~~	~~DTLS~~
~~64-79~~	~~TURN Channel~~
128-191	SRTP
128-255	QUIC long

Option 4 - Avoid Conflict (cont.)

Solution:

Define QUIC-SRTP for keying SRTP (easy enough)

During the handshake only STUN and QUIC long are used

QUIC short and SRTP don't collide

Use STUN magic number

on packets that QUIC rejects

0-3	STUN
0-127	QUIC short
~~20-63~~	~~DTLS~~
~~64-79~~	~~TURN Channel~~
128-191	SRTP
128-255	QUIC long

Option 5 - New QUIC version

A new QUIC version can be designed to be friendlier

For instance, the identifiers for packet types could be selected to avoid collisions 192-255

Questions

1. Is this the right principle?
2. Are these the right invariants?
    a. What are we missing? (recognizing that we might add things based on middlebox-related discussions)
    b. Should we implement multiplexing option 2?
3. ~~Should we document QUIC invariants separate to the main draft?~~