

Foo over Internet Transport

Gorry Fairhurst
Mike Bishop
(Help from many other people)



22 July 2026

Aim of this Talk

Introduction to IETF Transports for new apps/protocols

- Outline
 - IETF Transports
 - UDP, QUIC, H3 and Beyond
- Open discussion

Transport Mechanisms

Not all transports are equal!

- **Delivery:** Unicast/Multicast, Stream v datagram; Multi-streaming v. HOL blocking, Ordering
- **Security:** Data security; control security, DOS protection
- **Reliability:** Datagrams v. recovery of loss, timeouts; Integrity of header and payload
- **Flow control** (receiver back-pressure)
- **Sharing without harm** (network back pressure): Congestion Control (CC) / Circuit Breakers
- **Choosing and using a path:** ICMP, HAPPY, NAT traversal, Firewall keep-alives
- **Choosing Packet Size:** PMTU discovery, Fragmentation / Reassembly (resistance to attack)
- **Extensibility:** Extension points, feature & version negotiation
- **Logging and reporting**

Some IETF Transports

Protocol	UDP RFC768	TCP RFC 9293	SCTP RFC 9260	QUIC RFC 9000	RTP RFC 3550	MoQ	HTTP RFC 9113 RFC 9114
Delivery	Small Message	Stream	Message	Stream or Datagram	Message	Message	Message
Security	Payload?	Payload?	Payload?	Yes	Payload?	Yes	Yes
Reliable	No	Yes	Yes	Data may be resent	No	?	Yes
CC	No	Yes	Yes	Yes	No	Yes	Yes
Ordered	No	Yes	Configured	Configured	Configured	Configured	?
Multipath	No	MPTCP	Yes	Migration, MP-QUIC	?	?	?

UDP is RFC 768, what does that mean?

- UDP provides a small set of features [RFC 8304]
- UDP is often seen to reduce implementation and load...
- What does **your** application need above UDP?... [RFC 8085]
- Think about how the application maps to UDP, with edge cases there be dragons!
 - If you can use QUIC/H3/TAPS/etc you can leverage a lot of the solved problems

QUIC is RFC 9000, what does that mean?

- Does your application need multiplexing?
 - Will it actually benefit from head-of-line-blocking avoidance?
 - The more concurrent streams, the more decisions a QUIC implementation has to make!
 - Every connection can give some flow control at the start and update it later
 - DATAGRAMS are subject to congestion control, acknowledged, but not retransmitted
 - There is no standard defined for exchanging stream priority
 - Implementations are only recommended to offer a local API
- Think about how **your** application maps to QUIC; with edge cases there be dragons!
- If you can use HTTP/3, you can leverage a lot of the solved problems

QUIC Connections et al.

- Single TLS 1.3 handshake
 - Inherits PQC, ECH, etc. as these are enabled by TLS stacks
 - 0-RTT connection setup can accelerate subsequent reconnections
- Single congestion control context
- Connection + stream-level flow control
- Other potential benefits highly dependent on deployment work, not protocol:
 - Connection ID routing
 - Connection migration between paths

QUIC Bytestreams et al.

- Client- and Server-Initiated
- Bidirectional and Unidirectional
- Ordered* and reliable delivery
 - Stream data is ordered within the stream
 - No ordering *or priority* guarantees between streams
- Two usage models:
 - Each stream is independent message / exchange (HTTP style)
 - Each stream is independent data flow (BGP Monitoring Protocol style)

QUIC Datagrams

- Congestion-controlled, not flow controlled
- Unordered, unreliable delivery
 - No priority guarantees between datagrams
- Need unordered reliable delivery? Use a stream as a message instead.
- Need unreliable ordered delivery? ... Sorry.
- Not quite native to QUIC:
 - DATAGRAM frame is a widely-implemented QUIC extension
 - HTTP DATAGRAMs define a datagram-to-stream binding that associate datagrams to parent streams in HTTP/3

Cross-Transport API Surface

- QUIC interacts with certain application primitives:
 - Connection
 - Bytestreams
 - Messages
- (Mostly) the same application interface can run over:
 - QUIC
 - WebTransport
 - over HTTP/2 [TCP] or HTTP/3 [QUIC]
 - Qmux
 - QUIC primitives over TCP underlay
- TAPS provides a consistent model across multiple transports

RFCs that could help

- Transport Features (RFC 8303; RFC 8304)
- UDP Guidelines (BCP145: RFC 8085)
- Building Protocols with HTTP (BCP56: RFC 9205)
- TAPS Interface (RFC 9621, RFC 9622, RFC 9623)

Transport Mechanisms:

- CC Requirements (BCP 41: RFC 2914, BCP 133: RFC 9743, BCP 208: RFC 8084)
- NAT/ICE (BCP127: RFC 4787, BCP142: RFC 5382; RFC 5766; RFC 5928; RFC 8445)
- Ports (RFC 6056), Timers (RFC 8961), ICMP (RFC 5927)

Examples of Foo over QUIC*...

HTTP3 (RFC 9114)

DNS over QUIC (DoQ) (RFC 9250)

MASQUE Proxy - HTTP, TCP UDP, Ether Frame (various, MASQUE WG)

Media (various MoQ, draft-ietf-moq-transport)

RTP over QUIC (AVTcore: draft-ietf-avtcore-rtp-over-quic)

Storage over QUIC (draft-cel-nfsv4-rpc-over-quicv) (no current SMB proposal)

BGP (draft-retana-idr-bgp-quic)

BMP (draft-liu-grow-bmp-over-quic)

RPKI (draft-liu-sidrops-rpki-rtr-over-quic)

IPFIX (draft-llg-opsawg-ipfix-over-quic)

NETCONF (draft-ietf-netconf-over-quic)

RADIUS (draft-yl-radext-quic-transport)

PCEP (draft-yang-pce-pcep-over-quic)

REGEXT (draft-ietf-regext-epp-quic)

Network Telemetry (draft-netana-opsawg-telemetry-over-quic)

... etc

*** Do people here know more? ***

* Individual and Working Group I-Ds

OPSAREA

Many of the drafts on the list will be discussed at IETF-126 on Thursday, see:

“Operational considerations when using QUIC as a transport protocol”

... do also join/follow the discussion there!

16:30 - 18:30	Thursday Session IV				
—	Grand Klimt Hall 1	ART	atp	Authenticated Transfer	        
—	Grand Klimt Hall 2	IRTF	maprg	Measurement and Analysis for Protocols	        
—	Park Suite 6	IRTF	rasprg	Research and Analysis of Standard-Setting Processes Research Group	        
—	Grand Park Hall 1	OPS	opsarea	Operations & Management Area Open Meeting	        
—	Grand Park Hall 2	RTG	rtgwg	Routing Area Working Group	        
—	Park Suite 8	SEC	acme	Automated Certificate Management Environment	        
—	Park Suite 7	SEC	cose	CBOR Object Signing and Encryption	        
—	Grand Park Hall 3	WIT	moq	Media Over QUIC	        
18:45 - 19:45	Lenz Restaurant			New Participants' Social Hour	