

# WebTransport

QUIC Interim  
May 2019, London

# What is WebTransport?

It is an API that allows Web applications to connect to remote servers use various transport protocols interchangeably.

- Uses a stream as a basic abstraction
- Allows sending datagrams where available
- All underlying transports for WebTransports must:
  - Have origin checks
  - Require explicit opt-in from the server
  - Have mandatory congestion control
  - Use TLS

# Design goals

- Transport flexibility
  - WebTransport aims to make transports pluggable, by providing uniform APIs for streams and datagrams
    - Allows easy upgrade to a new transport
    - Allows easy fallback when a better transport does not work
- Ease of server implementation
  - Vital for adoption
  - Achieved by relying on technologies with large amount of implementation experience (QUIC, WebSocket)
- Constrained by Web security model
  - Servers have to explicitly opt into accepting web clients
  - Origin checks in all transports

# What are the transports abstracted?

- QuicTransport (draft-vvv-webtransport-quic)
  - Creates a dedicated QUIC connection
  - Allows application a greater level of control over what being sent
  - Very easy to implement on top of QUIC
- Http3Transport (draft-vvv-webtransport-http3)
  - Creates a “logical connection” inside an existing HTTP/3 connection
  - Removes the overhead from creating new connection:
- FallbackTransport (draft coming soon)
  - Simulates multiplexed streams over WebSocket
  - Entirely polyfillable in JavaScript
  - Works with proxies and on TCP-only networks
  - Poor performance due to head-of-line blocking

# QuicTransport

- Defines an ALPN value for WebTransport
- Defines a special transport parameter to perform an origin check
- Streams are QUIC streams
- Datagrams are QUIC datagrams (draft-pauly-quic-datagram)

That's it!

# Http3Transport

- Defines a special CONNECT-based handshake that creates a session within an HTTP/3 connection
  - A session ID is assigned as a result of that handshake
  - Multiple instances of Http3Transport can use the same HTTP/3 connection
- After a session is established, QUIC datagrams can be exchanged
  - Datagrams are prefixed with session ID that the recipient can use to multiplex different transports
- Both peers can create streams (prefixed by session ID), both bidirectional and unidirectional

# Related technologies

WebTransport is like

- ...WebSockets, but over QUIC
  - Both aim to expose full set of features of underlying transport
  - WebTransport is actually more than that, since it's a framework that can accept any stream-based transport, and has backpressure in the API
- ...RTCDataChannel, but client-server and over QUIC
  - Both aim to provide Web applications with unreliable, unordered data transport
  - The complexity of implementing ICE, DTLS and SCTP in userspace has been limiting its adoption in client-server case
- ...TAPS, but for the Web
  - Both aim to abstract multiple transports into a unified framework
  - Due to security constraints, the transports and capabilities provided by both do not overlap

# Links

## Web API:

- [WICG Discourse summary](#)
- [Explainer](#)
- [API spec](#)

## Wire format:

- [Overview](#)
- [QuicTransport](#)
- [Http3Transport](#)