

# QUIC And HTTP/3 Protocols

## Transport/App Protocol, Connection, Packet, Frame, TLS 1.3, Multiplexing, Flow-Control, Implementation

QUIC and HTTP/3 are respectively the much anticipated next-generation transport protocol and next generation application protocol. HTTP/3 runs on top of QUIC. Both are currently undergoing standardization through the IETF.

Both protocols are being designed in unison to work extremely well together to achieve an elevated level of performance and security. They offer a number of very interesting new features and continue the work we first see in HTTP/2 of adding multiplexing for HTTP connections.

“The QUIC transport protocol incorporates stream multiplexing and per-stream flow control, .. By providing reliability at the stream level and congestion control across the entire connection, it has the capability to improve the performance of HTTP compared to a TCP mapping. QUIC also incorporates TLS 1.3 at the transport layer, offering comparable security to running TLS over TCP, but with improved connection setup latency. [\[link\]](#)”

At the end of this course, attendees will understand how both protocols work and why they are important.

<b>Contents of One-Day Training Course</b>	
<p><b>Target Audience</b> Networking professionals and senior software engineers who require a deeper understanding of these new protocols that will play a very significant role in the future of the web and the Internet</p> <p><b>Prerequisites</b> Programming experience in any low-level language. Attendees will develop server and client-side implementations of both protocols as part of the labs.</p> <p>Good all-round networking knowledge; attendance at our <i>Fundamentals Of TCP/IP Networking</i> course or similar experience.</p>	<p><b>Next Gen Protocols</b> What are we trying to achieve? Can we not reach these goals with TCP and HTTP/1.1 or HTTP/2? What do QUIC and HTTP/3 offer?</p> <p><b>QUIC Overview</b> General tour of how QUIC works Based on UDP (a foundation that is already supported everywhere) QUIC plays role of TCP in protocol stack Message flows Connection set up and tear down Intro to security Available QUIC implementations Extensions</p> <p><b>Connections</b> Reasons for low-latency connection setup What impact this has? Connection migration Error correction Packet layout</p> <p><b>QUIC Streams &amp; Multiplexing</b> Unidirectional and bidirectional streams How multiple streams are multiplexed onto a single connection</p> <p><b>Flow Control</b> Connection flow control Stream flow control</p> <p><b>QUIC And Security</b> Modern TLS 1.3 built into QUIC Security is not an add-on option Review of security architecture Responding to NAT rebinding</p> <p><b>Frames</b> PADDING, RST_STREAM, [CONN APP] _CLOSE, MAX_DATA, MAX_STREAM_DATA, MAX_STREAM_ID, PING, BLOCKED, STREAM_BLOCKED, STREAM_ID_BLOCKED, NEW_CONN_ID, RETIRE_CONNECTION_ID, STOP_SENDING, ACK, PATH_[CHALLENGE] RESPONSE], NEW_TOKEN, STREAM, CRYPTO</p> <p><b>HTTP/3 Overview</b> Overall architecture HTTP/3 endpoints Options for discovery Types of streams (control, push, reserved) Available HTTP/3 implementations</p> <p><b>HTTP Framing</b> DATA, HEADER, PRIORITY Settings Framing architecture</p> <p><b>HTTP Message Exchanges</b> Message has one HEADERS frame and a number of DATA frames and optionally a concluding HEADERS frame Message flows</p> <p><b>Connection Management</b> Cancellation, Compression, Prioritization Server Push Error management</p> <p><b>Impact On Application Design</b> How these new protocols will influence application design (esp. multiplexing)</p> <p><b>Project</b> Create simple implementations of QUIC and HTTP/3 and see how application code could benefit</p>