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, similar to that provided by the HTTP/2 framing layer. By providing reliability at the stream level and congestion control across the entire connection, it has the capability toimprove 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.

Target Audience

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

Prerequisites

Programming experience in any low-level language. Attendees will develop server and client-side implementations of both protocols as part of the labs.

Good all-round networking knowledge; attendance at our *Fundamentals Of TCP/IP Networking* course or similar experience.

Contents of One-Day Training Course

Next Gen Protocols

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?

QUIC Overview

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 implemenations
Extensions

Connections

Reasons for low-latency connection setup
What impact this has?
Connection migration
Error correction
Packet layout

QUIC Streams & Multiplexing

Unidirectional and bidirectional streams
How multiple streams are multiplexed onto a single connection

Flow Control

Connection flow control
Stream flow control

QUIC And Security

Modern TLS 1.3 built into QUIC
Security is not an add-on option
Review of security architecture
Responding to NAT rebinding

Frames

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

HTTP/3 Overview

Overall architecture
HTTP/3 endpoints
Options for discovery
Types of streams (control, push, reserved)
Available HTTP/3 implementations

HTTP Framing

DATA, HEADER, PRIORITY
Settings
Framing architecture

HTTP Message Exchanges

Message has one HEADERS frame and a number of DATA frames and optionally a concluding HEADERS frame
Message flows

Connection Management

Cancellation, Compression, Prioritization
Server Push
Error management

Impact On Application Design

How these new protocols will influence application design (esp. multiplexing)

Project

Create simple implementations of QUIC and HTTP/3 and see how application code could benefit