

QUIC for WebRTC

The Practice of Adapting QUIC in WebRTC-based Real-time Streaming System
在基于WebRTC的实时流系统中使用QUIC

Jianjun Zhu 诸剑俊
jianjun.zhu@intel.com
Intel Open Source Technology Center

Agenda 议程

- QUIC overview 概述
- QUIC and WebRTC
- Practice 实践分享

QUIC

Quick **U**DP Internet **C**onnections

Key Strengths 优点

- Low-latency connection establishment 快速建立连接
- Multiplexing without head-of-line blocking 多路复用缓解对头拥塞
- Authenticated and encrypted header and payload 为头部和内容加密
- Rich signaling for congestion control and loss recovery 丰富的拥塞控制和丢包恢复信令
- Stream and connection flow control 基于流和连接的流量控制
- Connection migration and resilience to NAT rebinding 连接迁移和NAT重新绑定

Low-latency connection establishment 快速建立连接

- TCP handshake
- TLS handshake
- Transport over UDP

Ref: https://docs.google.com/presentation/d/15e1bLKYeN56GL1oTJSF9OZiUsI-rcxisLo9dEyDkWQs/edit#slide=id.gaf8802b44_0_87

Multiplexing 多路复用

- A webpage requests multiple resources (HTTP/2)
- 一个页面访问多个资源

- When a TCP packet is lost
- TCP包丢失会导致后续的包等待

- Only specific stream is impacted when a QUIC packet is lost
- QUIC包丢失只影响对应的流

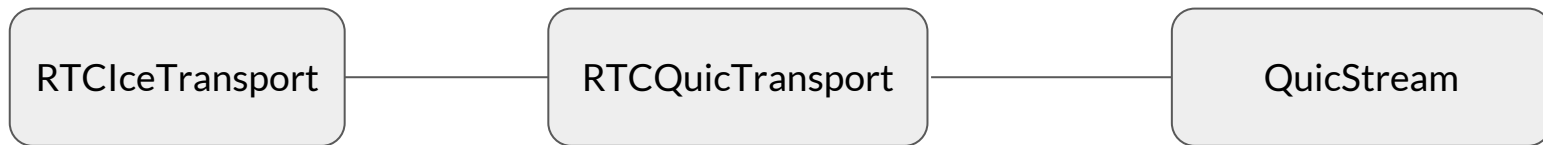
Connection Migration

- TCP: 5-tuple, identified by IP/port 五元组,IP和端口来标识
- QUIC: identified by a 64 bit connection ID 连接ID来标识

QUIC and WebRTC (P2P)

- QUIC API for Peer-to-peer Connections (P2P Quic)

<https://w3c.github.io/webrtc-quic/>



- Use cases 使用场景
 - A replacement of SCTP data channel 基于QUIC的数据流,类似data channel
 - Media over QUIC 使用QUIC传输媒体数据

QUIC and WebRTC (Client/Server)

- QUIC API for Client-to-Server Connections (C/S Quic)

<https://w3c.github.io/webrtc-quic/cs.html>

- Use cases 使用场景
 - Remove ICE which is designed for P2P 取消对ICE的依赖
 - Share the same connection with HTTP 与HTTP共用一个连接

Unreliable Transport 非可靠传输

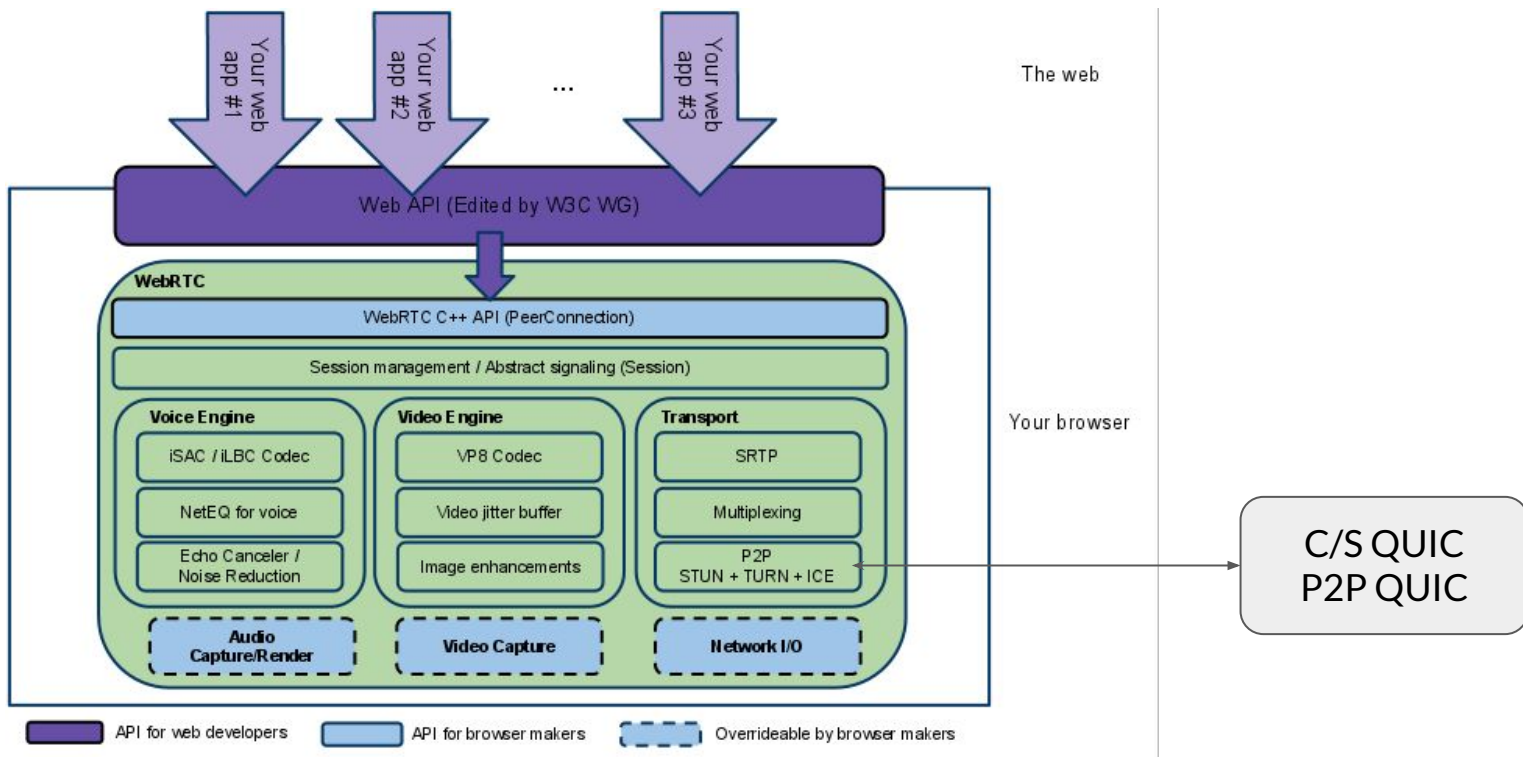
An Unreliable Datagram Extension to QUIC

<https://datatracker.ietf.org/doc/draft-pauly-quic-datagram/>

WebRTC-QUIC added support for DATAGRAM

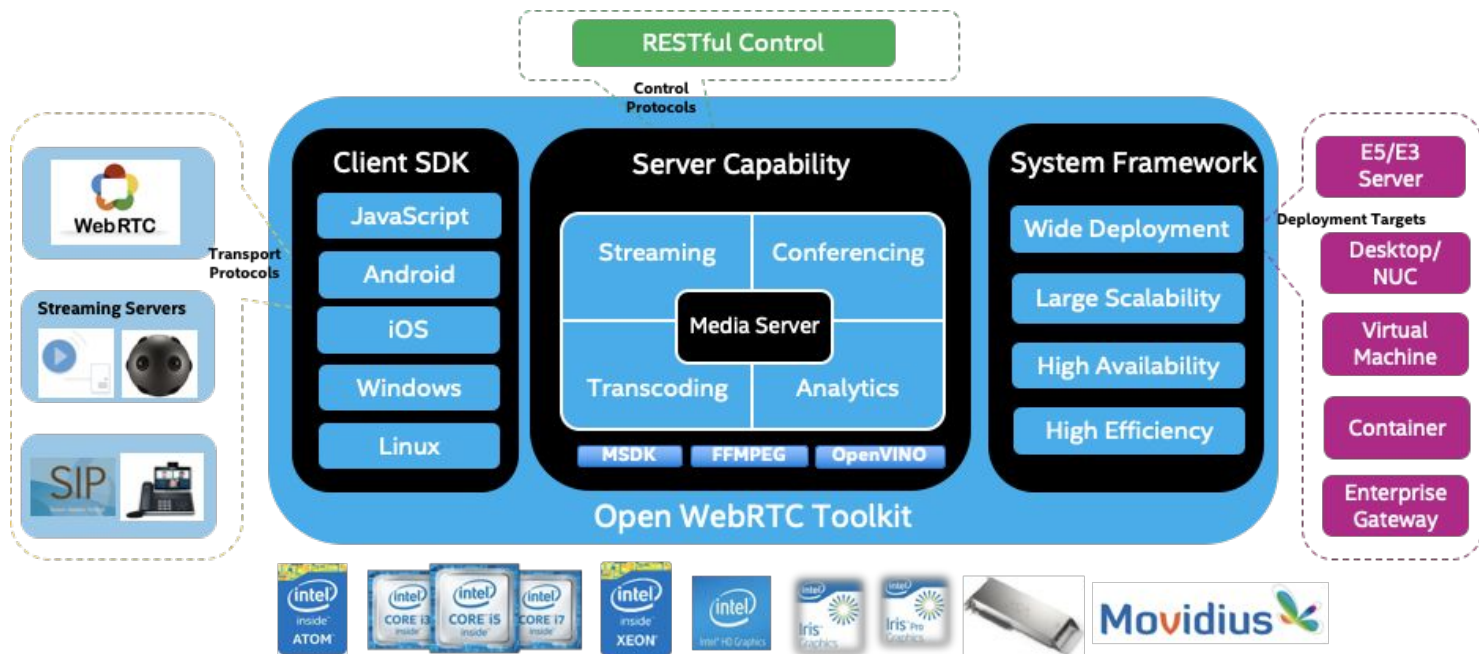
<https://github.com/w3c/webrtc-quic/issues/77>

Media over QUIC 使用QUIC传输媒体数据



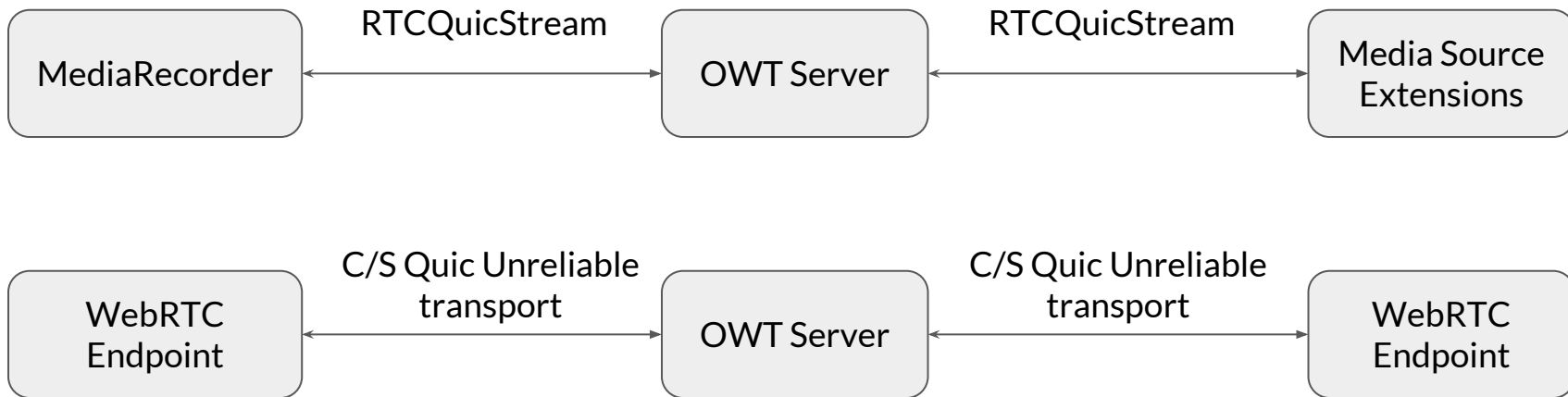
Practice - OWT Overview 实践 - OWT概述

- Open source project of Intel CS for WebRTC 开源项目
- <https://github.com/open-webrtc-toolkit>



Practice 实践

- QUIC for internal connection (inter-node) 用于节点间传输
- QUIC Connection between client and server 用于客户端和服务端连接



Practice - Experiments on TCP/QUIC Comparison

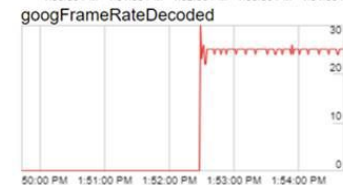
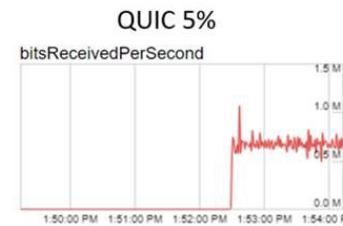
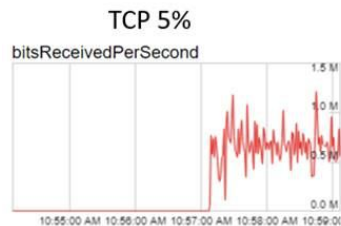
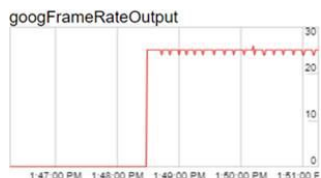
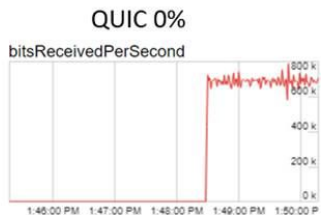
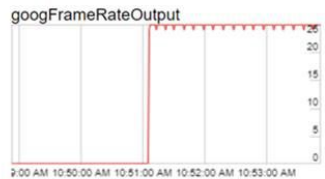
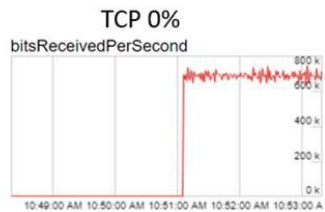
0% loss	Average(ms)	Variance	Min(ms)	Max(ms)
TCP	1(0.5)	0.02	0	4
QUIC	7.2(3.6)	483.6	1	486

5% loss	Average(ms)	Variance	Min(ms)	Max(ms)
TCP	28.8(14.4)	8716	0	1472
QUIC	2.86(1.43)	455	0	465

10% loss	Average(ms)	Variance	Min(ms)	Max(ms)
TCP	334(167)	114375	1	4599
QUIC	217(158)	373585	1	3160

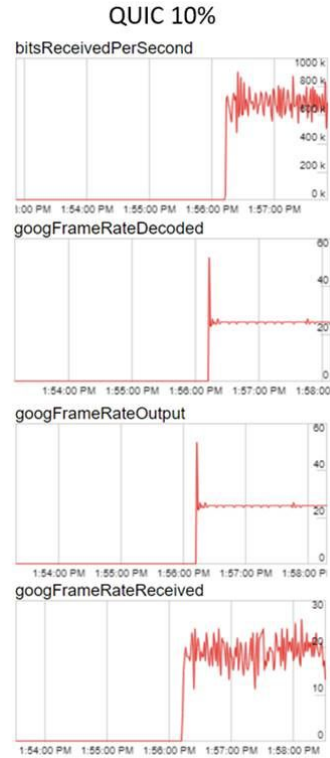
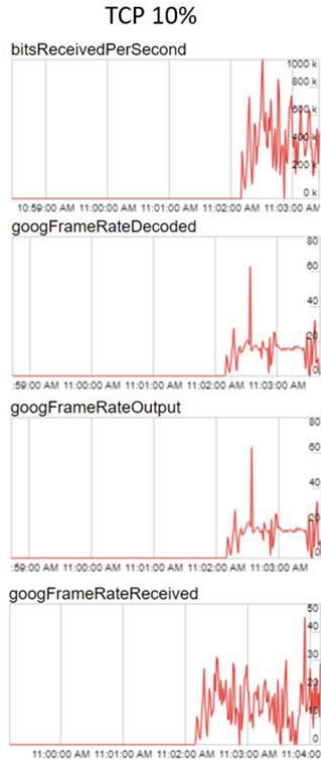
TCP & QUIC on 0ms delay network + sender loss

Practice - Experiments on OWT Internal Transport



2-way 30 ms delay applied.

Practice - Experiments on OWT Internal Transport



2-way 30 ms delay applied.

References 参考

<https://www.chromium.org/quic>

<https://quicwg.github.io/base-drafts/draft-ietf-quic-transport.html>

<https://www.youtube.com/watch?v=hQZ-0mXFmk8>