IETF 104 HTTP/QUIC 関連

後藤浩之 (グリー)

自己紹介

- 後藤 浩之 (グリー)
  - インフラ担当
- ISOC-JP インターネット標準化推進委員会
- 興味：Web, HTTP・QUIC関連

目次

- QUIC WG
  - WG状況
  - アップデートについて
- HTTP WG
  - NewWork

用語集 1

HTTP, QUIC

- HTTP:単にHTTPと言った場合は、セマンティクスを指す
  - フォーマットや表現形式、トランスポートとは関係ない
- HTTP/2: TCPコネクション上でHTTPリクエストを多重化する
- QUIC: UDP上で整合性と暗号化の機能を提供するトランスポート
- HTTP/3: QUICをトランスポートとするHTTP次期バージョン
- ストリーム: HTTP/2やHTTP/3 コネクション上の仮想通信単位
- フレーム: ストリーム上のメッセージ単位
- 0RTT: ハンドシェイク中にアプリケーションデータも送信する

用語集2

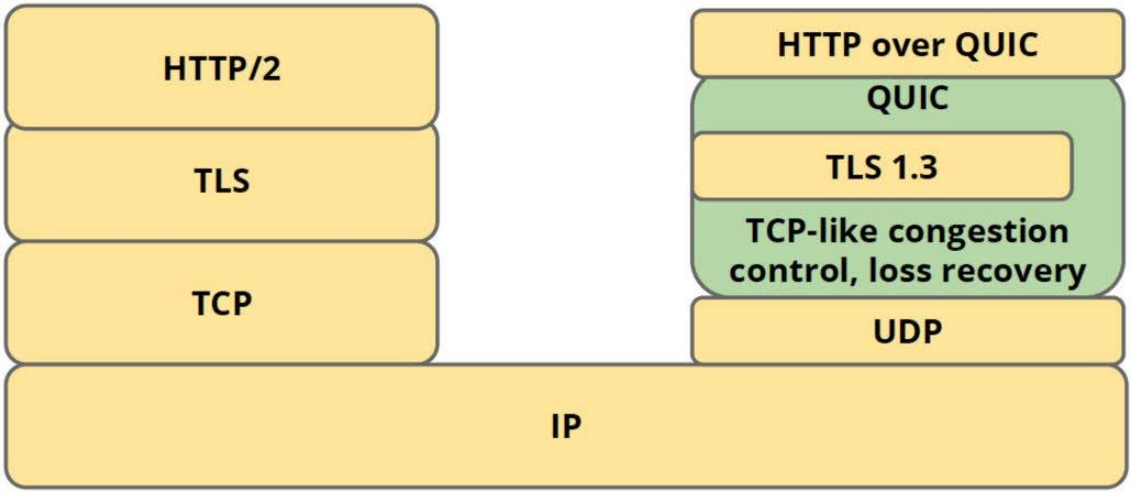
プロセス

- Github
  - Issue: Github上のIssue
  - PullRequest (PR): Github上で行われる、変更の提案方式
  - ラベル: IssueやPullRequestにつけるタグのようなもの
- IETF
  - Interim: WGが開催している中間会議。IETF本会合とIETF本会合とちょうど中間の次期に実施される

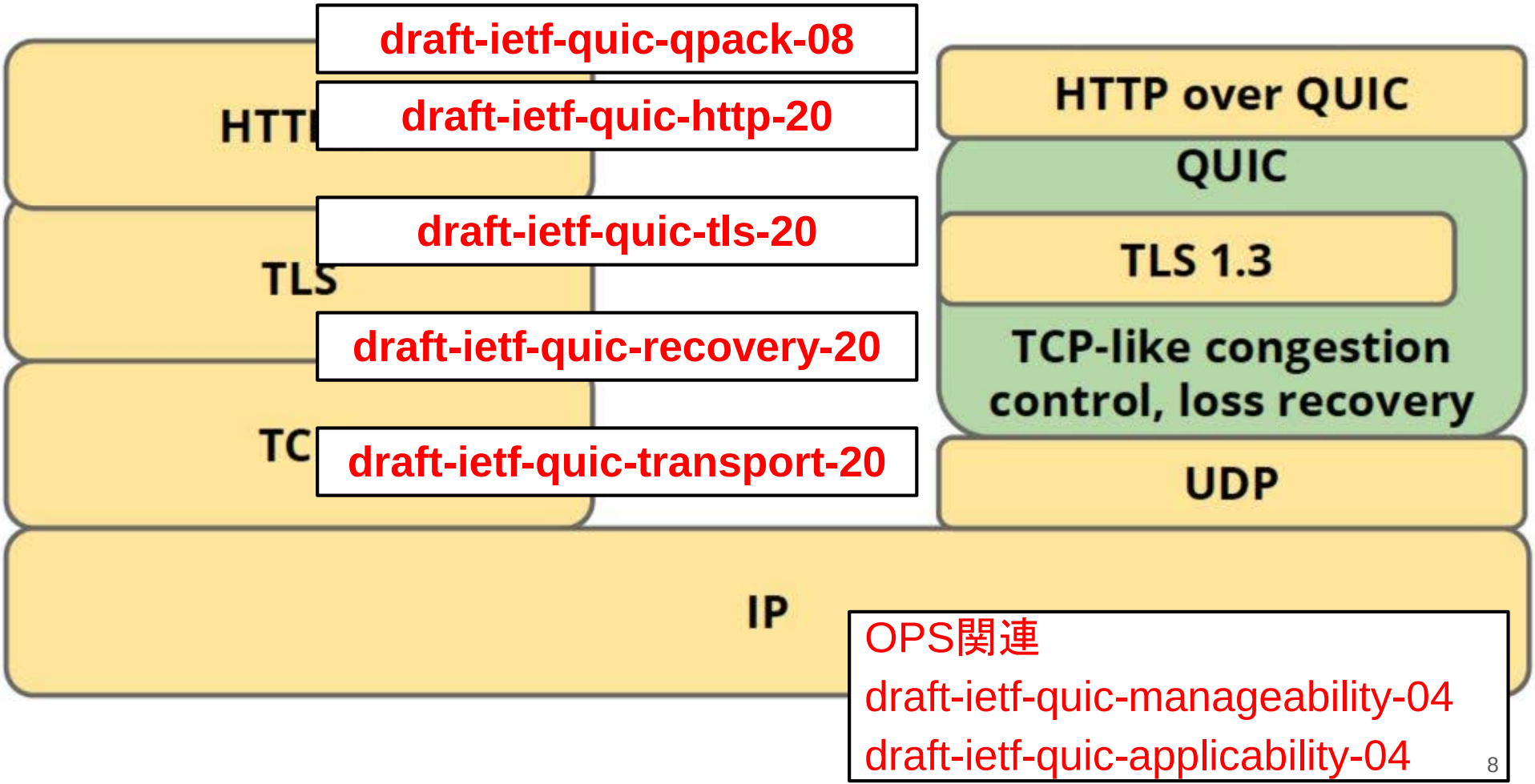
QUIC関連

QUICとは

QUICとは、Google社が考案したプロトコル
UDP上で動作し、TCPのような信頼性と、TLSのように暗号化された通信を提供する。上位プロトコルとして、HTTPを想定しているが限定はされない



引用 IETF98 QUIC Tutorial 14p
https://datatracker.ietf.org/meeting/98/materials/slides-98-edu-sessf-quic-tutorial


draft-ietf-quic-qpack-08
draft-ietf-quic-http-20
draft-ietf-quic-tls-20
draft-ietf-quic-recovery-20
draft-ietf-quic-transport-20
HTTP over QUIC
QUIC
TLS 1.3
TCP-like congestion control, loss recovery
UDP
IP
OPS関連
draft-ietf-quic-manageability-04
draft-ietf-quic-applicability-04

ハッカソン

- **前回:** 9th Implementation Draft (draft-19)

H: handshake
D: Stream Data
C: Connection Close

server → / client ↓	Minq	Mozquic	Quicly	quant	ngtcp2	mvfst	picoQUIC	winquic	f5	ATS	quiche	lsquic	ngx_quic	quicker	quicr	AppleQUIC
Minq																
Mozquic																
Quicly			VHDRZ	VHDRZ	VHDRZ		VHDRZ		VHD	VHDRZ	VHD					
quant	V		VHDCRZ	VHDCRZSMU	VHDCRZS		VHDCRZSMU	VHDCRZS	VHDCS	VHDCRZM	VHDC	VHDC	V	VHDCRZ	V	
ngtcp2			VHDCRZ	VHDCRZS	VHDCRZS		VHDCRZS	VHDCRZS	VHDC	VHDCRZ	VHDC					
mvfst			VHD	VHDC	VHDC	VHDCR						VHDC				
picoQUIC			VHDCRZ	VHDCRZSMU	VHDCRZS	VH	VHDCRZSMU	VHDCRZS	VHDCS	VHDCRZ	VHDC	VHC		VHDCRZ		V
winquic			VHDCRZ	VHDCRZS	VHDCRZS		VHDCRZS	VHDCRZS	VHDC	VHDCRZS				VHDC		
f5			VHD	HDCS	V	V	HDCS	VHS	HDCS	VHDC	VHDC	HC		VH	V	
ATS				V	VHD		VHDC	V	VH	VHD	VHD	VH		VHD		
quiche				VHDC	VHDC		VHDC		VHDC		VHDC					
lsquic				VHRS	VH	VHDC	VHS	V	VHS	VH	VH	VHDCRS				
ngx_quic																
quicker			HDRZ	VHDCRZ	HDCRZ	VHD	HDCRZ	HDCRZ	HDC	HDCRZ	HDC	VH		VHDCRZ		
quicr																
AppleQUIC				VHDCS	VS		VHD	VHDS	VHD	VHDS	VHD	H				
quic-go								VHDS								
Quinn																
Pandora																
f5_test			V	VHDC	VHDC		VHDC	VHDC	VHDC	V	VHDC		V		V	

ハッカソン

- 今回: 12th Implementation Draft (draft-20)

H: handshake
D: Stream Data
C: Connection Close

client ↓	h2o/quicly	quant	ngtcp2	mvfst	picoQUIC	winquic	f5	ATS	quiche	lsquic	ngx_quic	AppleQUIC	quic-go	Quinn	Pandora
h2o/quicly	VHDCRZS 3	VHDCRZ	VHDCRZS	HDC 3	VHDCRZS 3	HDC	VHDCRS 3	VHDCRZS	VHDCS 3				VHDCS		
quant	VHDCRZ	VHDCRZS MBUPE	VHDCRZS MBU	VHDCRZ 3	VHDCRZS MBU3P	VHDCRZS P3	VHDCRS U3E	VHDCRZS M3	VHDCRS 3	VHDCRZS M3PE	VHDCR 3	VHDC M	VHDCRS	VHDCRZS MBUPE	
ngtcp2	VHDCRZS	VHDCRZS U	VHDCRZS MBU	VHDCRZ	VHDCRZS MBU	VHDCRZS	VHDCRS U	VHDCRZS MB	VHDCRS	VHDCS	VHDC			VHDCRZS MBU	
mvfst		HDC	VHDCRZ	VHDCRZ 3	VHDCRZ 3	VHDCRZ 3	HDC 3	VHDCRZ 3		VHDCRZ 3	HDC 3			HDC	
picoQUIC	VHDCRZS 3	VHDCRZS MBUP	VHDCRZS MBU	VHDCRZ	VHDCRZS MBU3P	VHDCRZS P	VHDCRS U3	VHDCRZS MB3	VHDCRS 3	VHDCRZS MB3	VHDCRZ 3	VHDC	VHDCS	VHDCRS MBUP	
winquic		VHDCRZS	VHDCRZS	VHDCRZ	VHDCRZS	VHDCRZS P	VHDCS	VHDCRZS	VHDCRS	VHS	VH	VHDC		VHDCRS	
f5	HDCS	HDCS E	HDCS	HDC	HDCS	HDCS	HDCS E	HDCS	HDCS	HCS E	HDC E	HDC		HDCS E	
ATS	VHDCRS 3	VHDCRS M	VHDCRS M	VHDCR 3	VHDCRS M3	VHDCRS 3	VHDCRS 3	VHDCRS M3	VHDCRS 3	VHDCRS 3	VHDCR 3	VHDC		VHDCRS M	
quiche	VHDCS 3	VHDCS	VHDCS	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS 3		VHDCS 3	VHDCS	
lsquic				VHDCS 3	VHDCRS 3P	VHDCS 3P	VHDCRS 3	VHDCRS 3		VHDCRS 3PE	VHDC 3				
ngx_quic	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
AppleQUIC		VHDCS	VHDCS	VHDC 3	VHDCS 3	VHDCS	VHDCS 3	VHDCS 3	VHDCS	VHDCS 3		VHDCS		VHDCS	
quic-go	VHD				VHDCS		VHDCS								
Quinn		VHDCRZS U	VHDCRZS BU		VHDCRZS BU	VHDCRZS	VHDCS	VHDCRZS 3		VHS				VHDCRZS BUP	
Pandora		VHDC													

実装について

今までコード非公開だったものが公開され始めている
https://github.com/quicwg/base-drafts/wiki/Implementations

- Facebook: mvfst (https://github.com/facebookincubator/mvfst)
- Google: quiche (https://quiche.googlesource.com/)
- Cloudflare: quiche (https://github.com/cloudflare/quiche)

また、HTTP/3への対応が始まっており、実際に通信できるところまで来ている

QUIC の状況

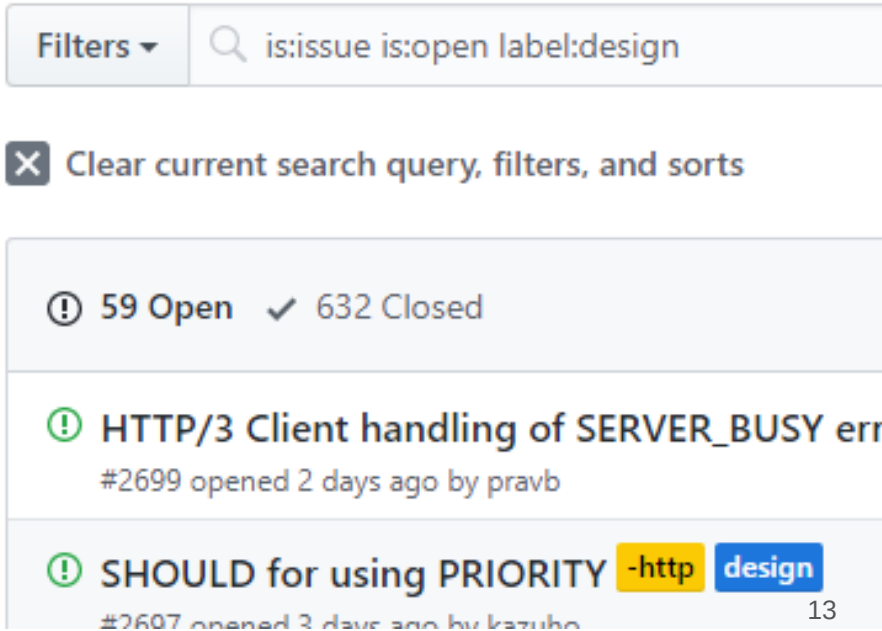
- マイルストン: 2019年7月 Coreドキュメント のIESG送り
- 「Early-Stage Process」 -> 「Late-Stage Process」
  - https://github.com/quicwg/base-drafts/blob/master/CONTRIBUTING.md
- IETF104 での議論はUpdate / Issue整理 / 関連報告 (2セッション)
  - Recovery Update
  - Discarding QUIC Old 1-RTT Keys
  - HW Offloading
  - Migration
  - Issue整理

QUIC の状況

- Issue 状況 (design ラベルのもの)

前回

今回



「Early-Stage Process」 -> 「Late-Stage Process」

WG Draftの進め方

Early-Stage Process (HTTP/3, QPACK, Recovery)

- Author に権限を与え、Draftが柔軟に進みオーバーヘッドをs苦なくする
- Draftにデザイン変更を加える形で、修正をProposeする。新しいDraftが出ると、該当デザインIssueがMLでピックアップされ、コンセンサスが得られるとhas-consensusとなる

Late-Stage Process (Invariants, Transport, TLS)

- プロトコルのデザイン変更を抑え、Draftはコンセンサスを正確に反映する
- WGで議論され、チェアがコンセンサスをジャッジする

Recovery Update

Revoverey Specのアップデートについて

- TLO and RTO をあわせてPTOとする
- ack delay を Max Acl Delayまでにする
- など

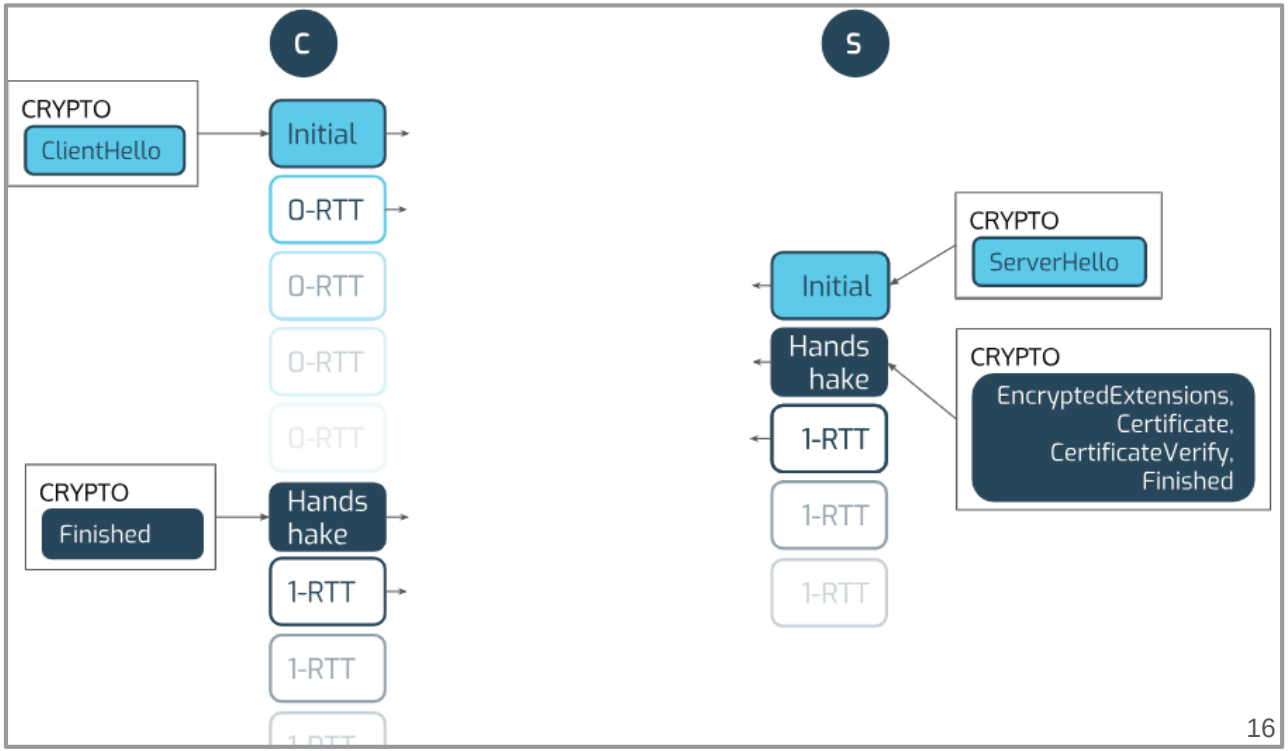
Since -16

- Unify TLP and RTO into a single PTO; eliminate min RTO, min TLP and min crypto timeouts; eliminate timeout validation (#2114, #2166, #2168, #1017)
- Redefine how congestion avoidance in terms of when the period starts (#1928, #1930)
- Document what needs to be tracked for packets that are in flight (#765, #1724, #1939)
- Integrate both time and packet thresholds into loss detection (#1969, #1212, #934, #1974)
- Reduce congestion window after idle, unless pacing is used (#2007, #2023)
- Disable RTT calculation for packets that don't elicit acknowledgment (#2060, #2078)
- Limit ack_delay by max_ack_delay (#2060, #2099)
- Initial keys are discarded once Handshake are available (#1951, #2045)
- Reorder ECN and loss detection in pseudocode (#2142)
- Only cancel loss detection timer if ack-eliciting packets are in flight (#2093, #2117)

Discarding QUIC Old 1-RTT Keys

QUICには4つの鍵がある

- Initial Key
- 0-RTT Key
- Handshake Key
- 1RTT Key

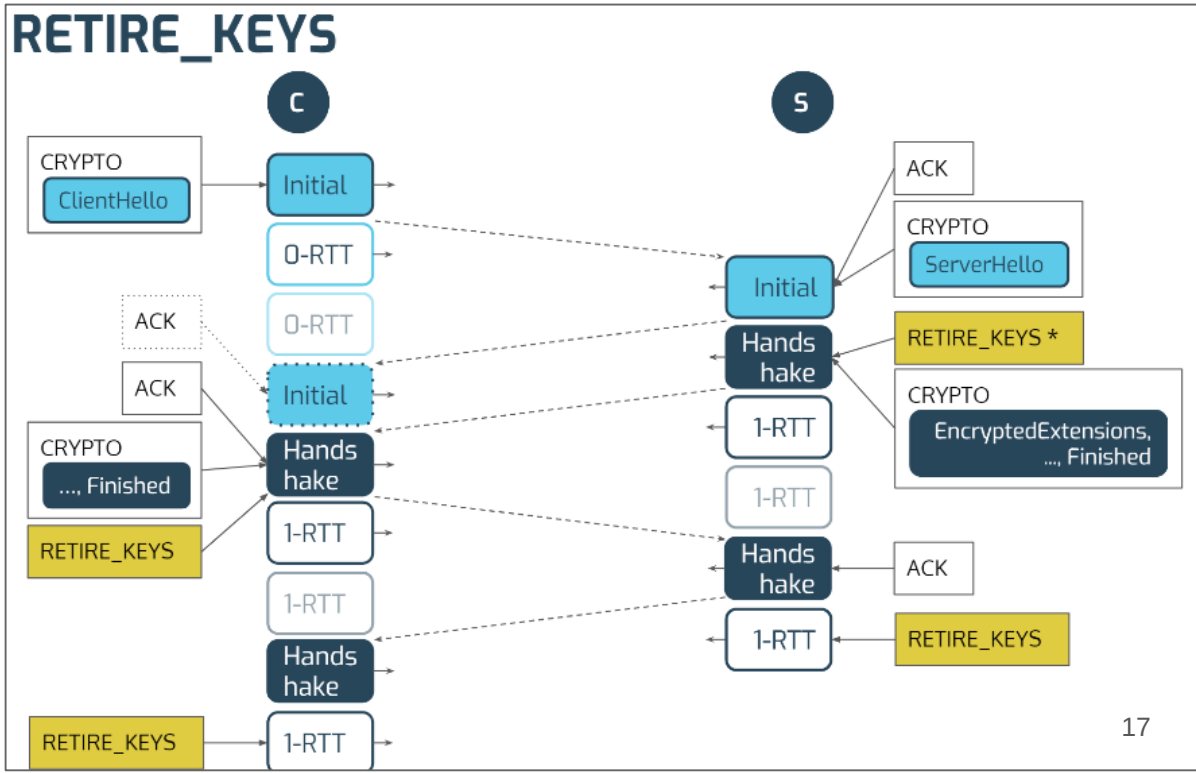


Discarding QUIC Old 1-RTT Keys

いらない鍵を破棄できるようにシグナルする提案が3つ

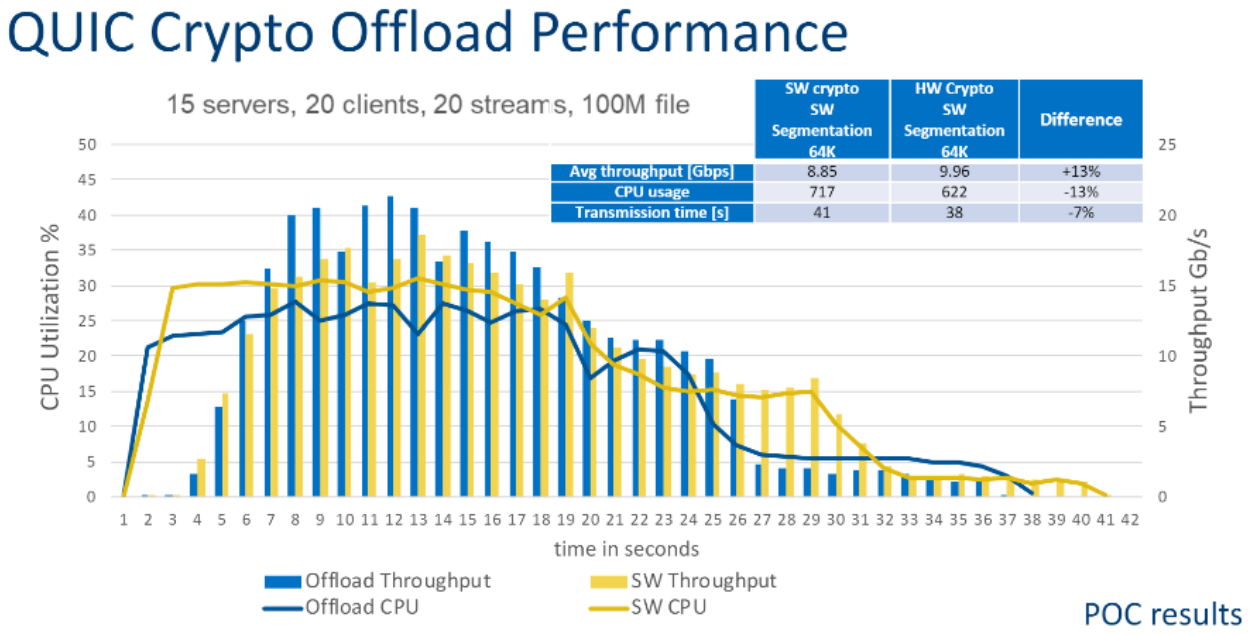
- Key_Ready
- Retire_Keys
- Max_KEY_UPDATES

Desginチームで検討
=> 現在PRが出ている



HW Offloading

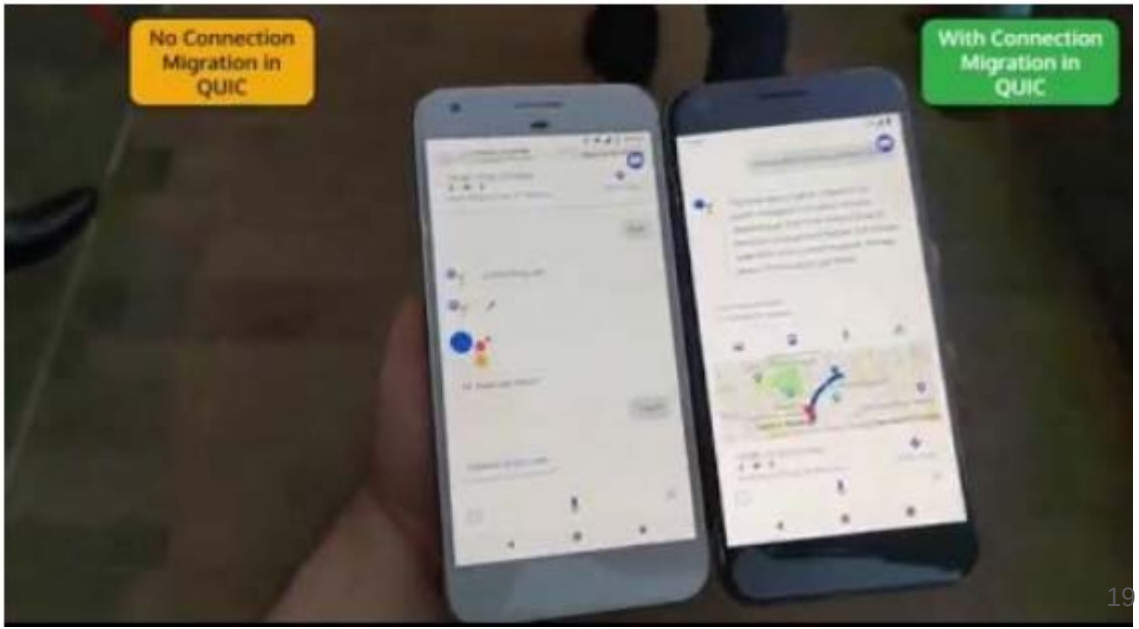
- Intelによる、QUICの暗号化処理のオフロード計測



Migration

Google のQUIC Connection Migrationの測定およびDemo

- Text search:
  - -0.7% failures
  - -0.3% cancels



Issues

- QUIC Ossification
  - 観測者にとってバージョンがランダムに見えるようにするなど、通信の観測者が不具合を起こすことを予め防ぐ(解釈できないようにする)
  - Hum
    - [stronger] Close with no action, punt to applicability statement
    - [some, but weak] Leave open and block v1
- 2473, Connection ID Length changes
  - Connection IDを更新する際に長さを維持するか (毎回Lengthを管理する必要がある)
  - Hum
    - Discussed in Prague; hum strongly supported closing with no action

非公式 Side Meeting

「Some notes about WebRTC and QUIC」
https://mailarchive.ietf.org/arch/msg/quic/Qs4QYOTmAT81BJw1amxEE1T9azo

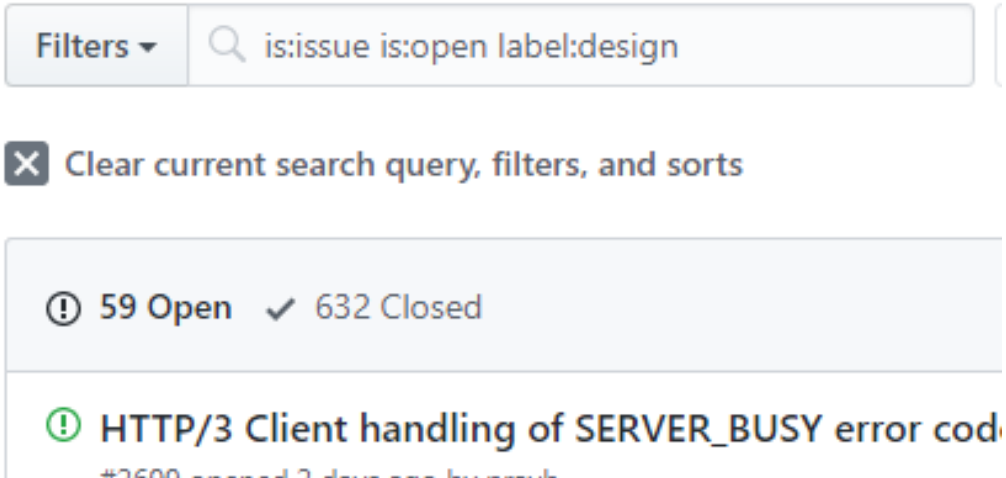
- WebRTCでQUICを使う議論
  - RTP over QUIC
  - DataChannelにSCTPではなくQUICをつかう
- API
- ICE

QUIC WGの今後

- Interim: 5/20〜5/23 ロンドン
  - Inter Operability Test
- マイルストーン: Jul 2019, Coreドキュメントを IESGへ
- EPIQ conference: 8月

QUIC WGの今後

- Interim: 5/20〜5/23 ロンドン
  - Inter Operability Test
- マイルストーン: Jul 2019, Coreドキュメントを IESGへ
- EPIQ conference: 8月



HTTP関連

HTTP WG

- 2セッション開催
  - HTTPtre Issue 整理
  - Active Extension Drafts/New Work

HTTPtre (HTTP-Core)

“HTTPのセマンティクス”と”HTTP/1.1 のシンタックス”を分離

RFC 7230 - Hypertext Transfer Protocol (HTTP/1.1): **Message Syntax and Routing**
RFC 7231 - Hypertext Transfer Protocol (HTTP/1.1): **Semantics and Content**
RFC 7232 - Hypertext Transfer Protocol (HTTP/1.1): **Conditional Requests**
RFC 7233 - Hypertext Transfer Protocol (HTTP/1.1): **Range Requests**
RFC 7234 - Hypertext Transfer Protocol (HTTP/1.1): **Caching**
RFC 7235 - Hypertext Transfer Protocol (HTTP/1.1): **Authentication**

draft-ietf-httpbis-semantics-04 - **HTTP Semantics**
draft-ietf-httpbis-cache-04 - **HTTP Caching**
draft-ietf-httpbis-messaging-04 - **HTTP/1.1 Messaging**

Active Extension Drafts

- HTTP/3 and QUIC
  - 非ブラウザのHTTP/3クライアントのはなし
- Secondary Certificates
  - HTTP/2上で追加のクライアント証明書・サーバ証明書をやりとりする話

HTTP/3 and QUIC

HTTP/3の開始は、通常 Alt-Svc ヘッダでHTTP/2からHTTP/3に切り替える

- 非ブラウザにおける、HTTP/3 Onlyクライアントの場合どうするか
- Happy eyeball ?
- DNS people suggest SRV record


```
https://www.example.com:q443/
httpq://www.example.com/
https://quic@www.example.com/
```

Secondary Certificates

Secondary Certificatesの背景

クライアントからサーバへクライアント証明書の送信

- HTTPでは、コネクションを張ってから投げられたHTTPリクエストのURLによって改めてクライアント証明書を要求するユースケースがある
- HTTP/2では、HTTPリクエストが多重化されており、TLSレイヤでサーバからCertificate要求を送るにしてもどのHTTPリクエストがトリガになったかわからない (HTTP/2でTLS1.3post-handshake authenticationの禁止を別i-d化)

サーバからクライアントへサーバ証明書の送信

- HTTP/2では証明書がvalidであれば、一つのコネクション上で複数ドメインへのリクエストを送ることができる
- TLSハンドシェイクした後、追加でサーバ証明書を送信したい

Secondary Certificates

クライアントからサーバへ
クライアント証明書の送信

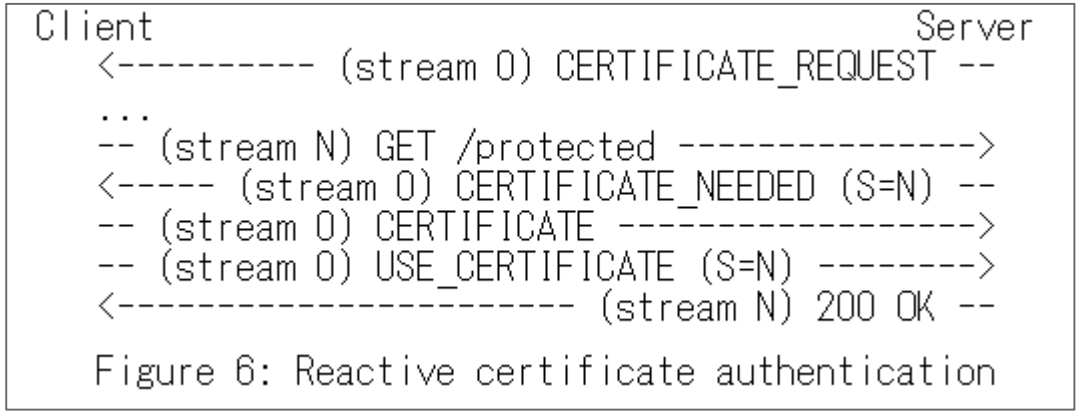


Figure 6: Reactive certificate authentication

サーバからクライアントへ
サーバ証明書の送信

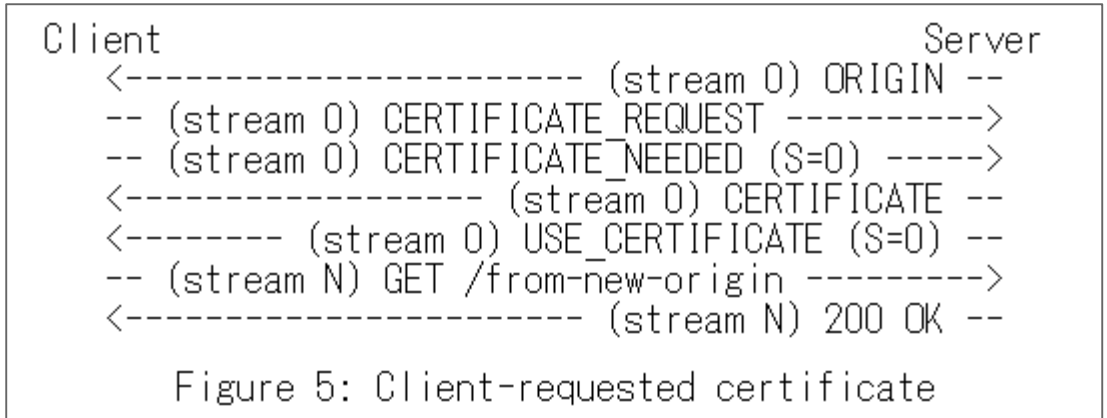


Figure 5: Client-requested certificate

Secondary Certificates

クライアントからサーバへ証明書の送信の際

- あるサーバの秘密鍵が漏れた場合に、悪意あるサーバがこの仕様を利用することでなりすましできるようになる。
- サーバから送信するサーバ証明書をそれように変更する
  - Required Domainエクステンション (IETF104後に修正Draftあり)

サーバからクライアントへサーバ証明書の送信

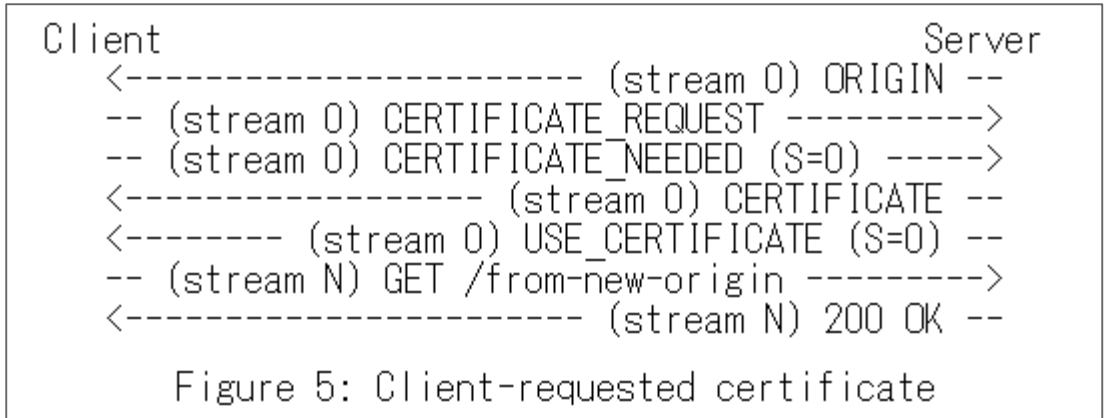


Figure 5: Client-requested certificate

New Work

- Proxy Status
  - Proxyエラーを示すヘッダの標準化
- Update on Using HTTP/2 as a Transport
  - HTTP/2上で任意のバイナリストリームを流す
- Best practices for TLS Downgrade
  - Client <--https--> CDN <--http--> Origin の問題提起
- Signed Exchanges
  - 後述

Signed Exchanges

WebPackagingという、Webサイトを署名をつけて再配布する仕組み

- **Signed HTTP Exchanges: HTTPリクエスト・レスポンスに署名する**
- **Bundled HTTP Exchanges: 複数のリクエスト・レスポンスのフォーマット**
- Loading Signed Exchanges: ブラウザの読み込み仕様

背景として、GoogleのAMPという仕組みの改善が大きい

- Web検索結果表示の改善として、WebページをAMP準拠通りに作成すると Google検索ページから該当ページを高速に表示できるようになる
- URLはGoogleのキャッシュサーバのもの
  ＝＞ オリジナルのURLを表示できるようにしたい。
  ＝＞ Signed Exchanges

Signed Exchanges

現状

- Chrome 73で使用できる
- Yahoo Japanなどでデプロイ済み
  - 「SXG（Signed HTTP Exchanges）始めました」

IETF的には

- ユースケースの整理
- セキュリティ
- W3Cなどとの連携
- Web Packaging等がウェブのエコシステムに与える影響
  => 6月にワークショップの開催
  https://www.ietf.org/blog/webpackaging/

Update on Using HTTP/2 as a Transport

HTTP/2コネクション上で、多重化されたバイトストリーム通信を行う提案仕様

モチベーション

- すでにHTTP/2は技術的にも発展しており、再利用できる
- フローコントロールがある
- 優先度制御ができる

仕様では 「:protocol : bytestream」 でHTTPリクエストヘッダを投げると以後、そのストリームがバイトストリーム通信専用になる

まとめ

QUIC WG

- 実装がInter Operatbility testではH3対応が増えている
- Late-Stage Processへ
- 7月のIESG 提出にむけてIssue整理中、デザインIssueはまだあるが...

HTTP WG

- HTTPtreのメンテナンス引き続き
- New Workも
- IETF104後に HTTP Workshopがあり
  それを踏まえて幾つかの提案あり

The HTTP Workshop

（IETFとは直接の関係はないが）

IETF104後 2-4 Aprilに、The HTTP Workshopが開催された

HTTP標準化をやってる人や、ブラウザ、ミドルウェア開発者などが議論を行った。

参加者は一部IETFとかぶっており、議論の結果が

その他のWG

- Sec Dispatch