IETF 103 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

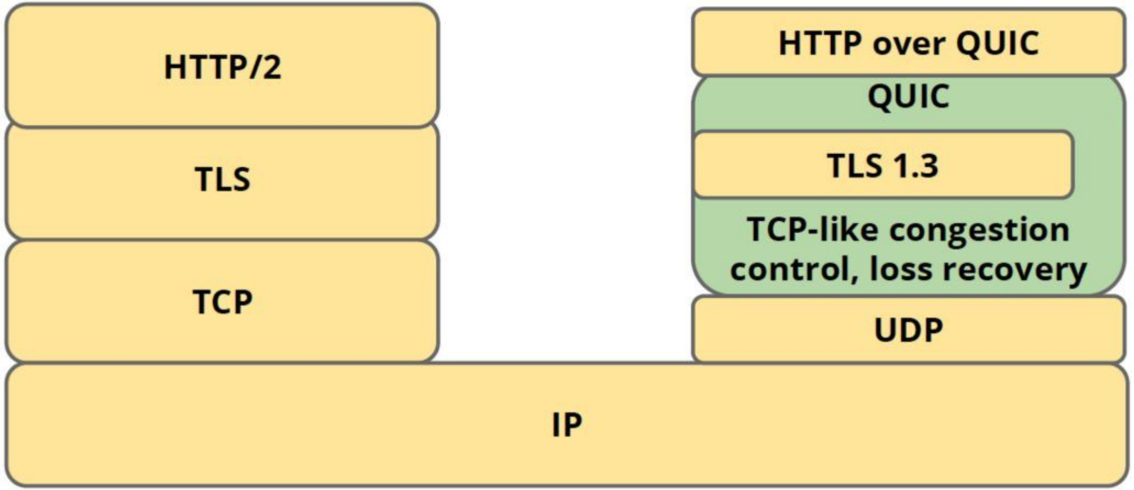
目次

- QUIC WG
  - 状況
  - HTTP/3
  - SpinBit
- HTTP WG

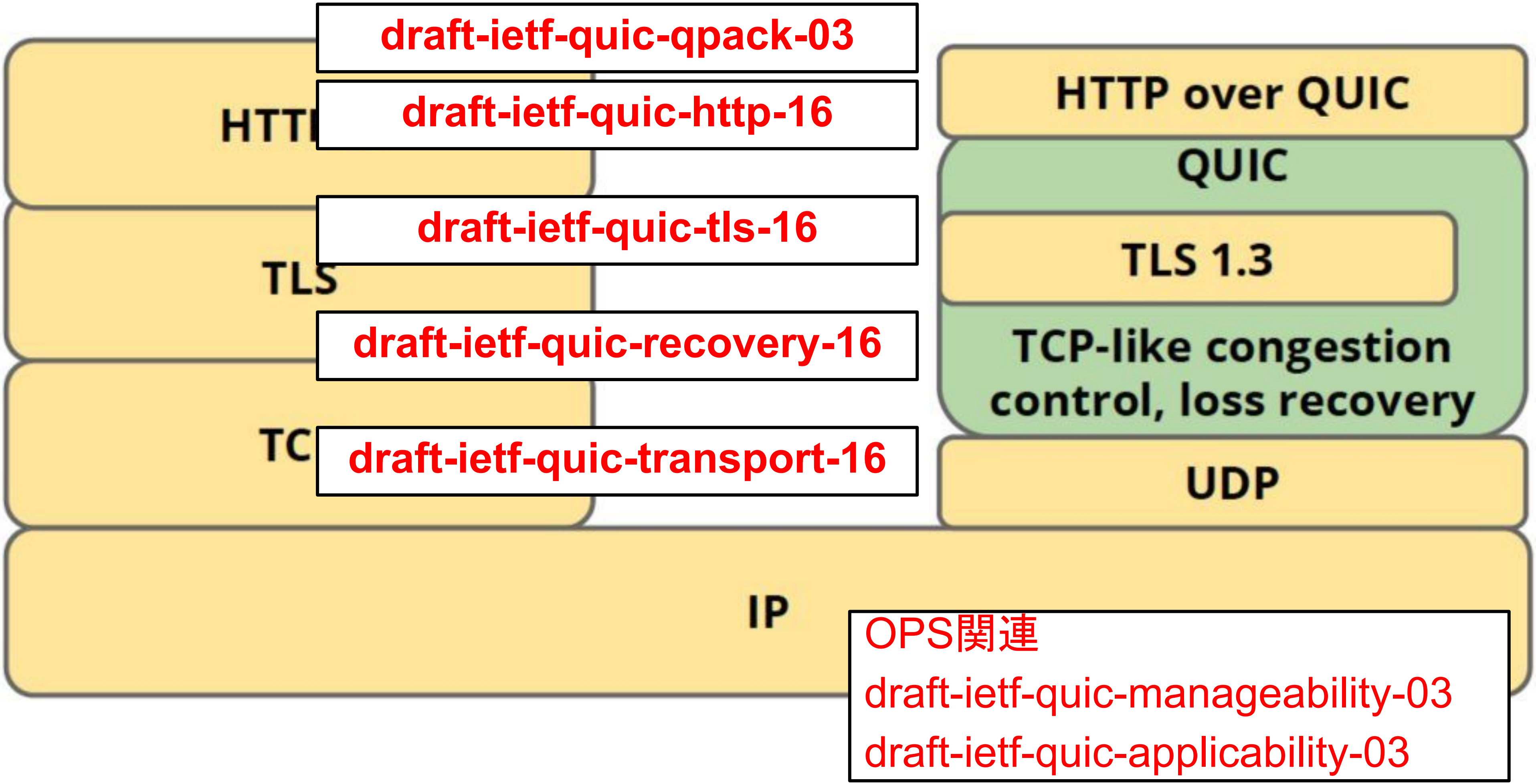
QUIC関連

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-03
HTTP
draft-ietf-quic-http-16
draft-ietf-quic-tls-16
TLS
draft-ietf-quic-recovery-16
TC
draft-ietf-quic-transport-16
HTTP over QUIC
QUIC
TLS 1.3
TCP-like congestion control, loss recovery
UDP
IP
OPS関連
draft-ietf-quic-manageability-03
draft-ietf-quic-applicability-03

ハッカソン

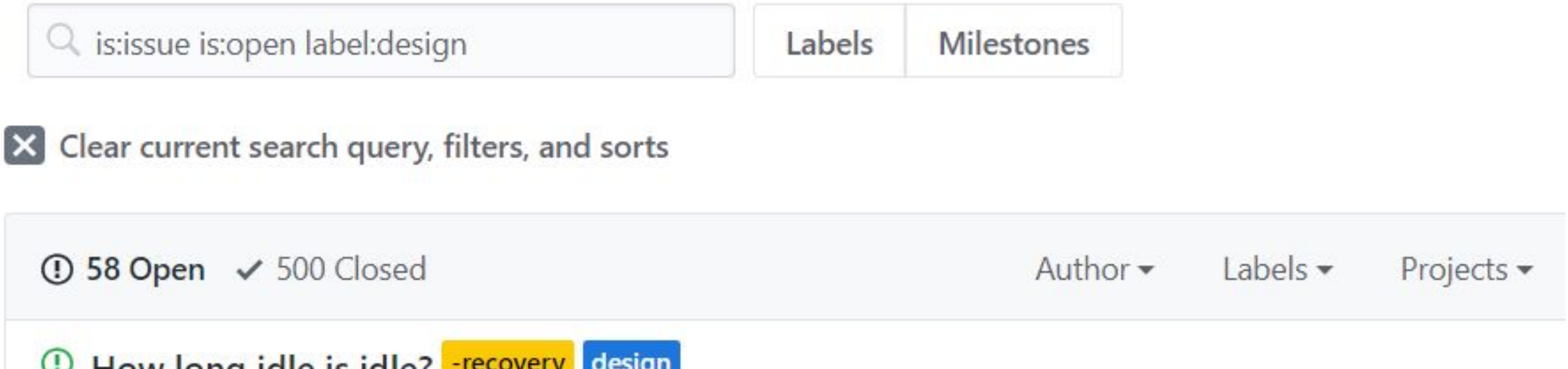
- 9th Implementation Draft (draft-15, tls1.3)

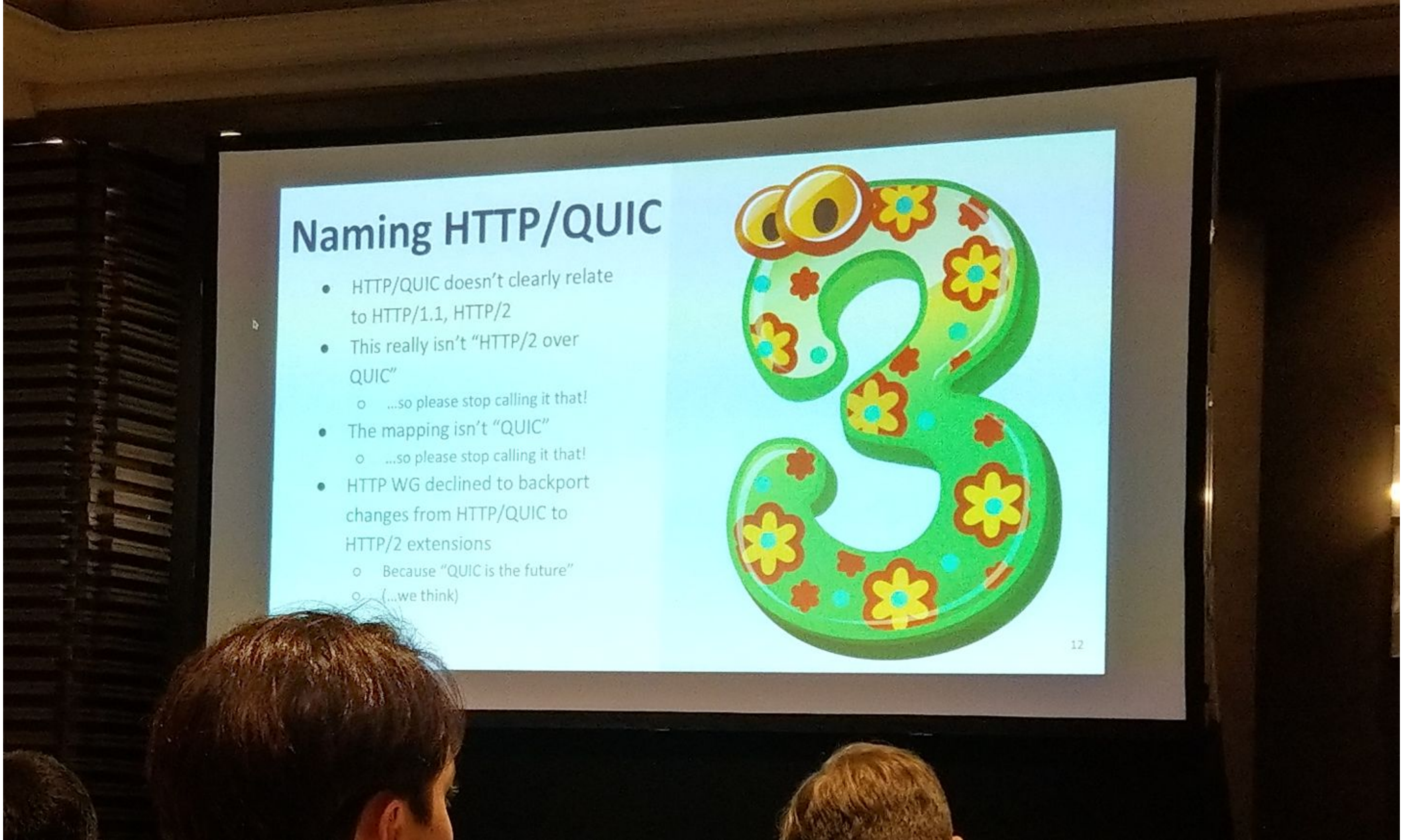
H: handshake
D: Stream Data
C: Connection Close

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	

QUIC WG

- 2セッション開催
  - やはり、大部屋開催
- ドキュメント
  - draft-16 で大きなeditorial修正が入った
  - 大きなデザイン変更は落ち着いてきている
- マイルストーンが後ろ倒しになり、2018/11 => 2019/7 (IESG)




Naming HTTP/QUIC
HTTP/QUIC doesn't clearly relate to HTTP/1.1, HTTP/2
This really isn't "HTTP/2 over QUIC"
...so please stop calling it that!
The mapping isn't "QUIC"
...so please stop calling it that!
HTTP WG declined to backport changes from HTTP/QUIC to HTTP/2 extensions
Because "QUIC is the future"
(...we think)
12

Naming HTTP/QUIC (1/2)

- HTTP over QUIC の名称を HTTP/3に変更する議論
- 背景
    - gQUIC (フルスタック), iQUIC (トランスポート) の違いに伴う混乱、明確に名前をつける
    - HTTP/2 over QUIC ではないことの明確化
    - HTTP/2よりも大きい番号をつけ、そちらへの移行を明確化
- 議論
    - ML「Identifying our deliverables」(10/28)
    - QUIC WG (11/6)
    - HTTP WG (11/8)

Naming HTTP/QUIC (2/2)

- QUIC WGとHTTP WGの両方でHumを実施
- QUIC WG (11/6)
  - プロトコル内で使用する識別子もH3とする
  - **名前の変更 => 70:30 ほどで賛成**
  - **最終的な決定はHTTP WGの決定に委ねる => ほぼ賛成のみ**
- HTTP WG (11/8)
  - 標準化後HTTP/3やQPACKのメンテナンスを行う
  - **名前の変更 => ほぼ賛成**

HTTP/3への名称変更の合意が得られる。すでにdraftの修正が行われている

HTTP/3 紹介

HTTP/3 (HTTP over QUIC)について、簡単に紹介。

HTTPのセマンティクスはそのまま、HTTPリクエスト･レスポンスを効率よくやり取りします。

HTTP/2をそのままQUIC上で扱うと不都合がある(順番が保証されないなど)
そこで、いくつかの改善が入っている。

- QUICで提供されているストリーム管理に乗っかる形で簡略化
- HPACKからQPACKへ
- 優先処理の改善
- サーバプッシュの改善

Spin Bit (passive RTT measurement) (1/4)

- IETF99頃から議論されている
- トラブルシュート等の目的で、経路上でRTTを測定したい
  - QUICではAckも暗号化されている
    ＝＞ ハンドシェイク時しかRTTが測定できない
- Spin Bit
  - パケットが往復する度にトグルするbitをヘッダで1bit経路上に露出する

```
+--------+    1   0   0   0   0    +--------+
|        |                         |        |
| Client |      -------->          | Server |
|        |      <--------          |        |
+--------+    0   0   0   0   0    +--------+
```

Figure 4: The bit begins spinning

```
+--------+    1   1   1   1   1    +--------+
|        |                         |        |
| Client |      -------->          | Server |
|        |      <--------          |        |
+--------+    0   0   0   0   1    +--------+
```

Figure 5: Server reflects the spin edge

Spin Bit (passive RTT measurement) (2/4)

- 今まで、Spin Bitの有用性・セキュリティ上の懸念などが議論されてきた
- この仕様をコアドキュメントに入れるかどうかのHumが行われた

Spin Bit (passive RTT measurement) (3/4)

入念な確認が行われた

- Confirm: decision will be limited to the single-bit variant
- Confirm: The spin bit fit for purpose
- Confirm: it must be possible not to spin
- Confirm: the privacy/security/ossification aspects of spin are wll-understood
- Confirm: intent to immplement and deploy
- **Consensus**: include the spin bit in the specification
- **Consensus**: requirement level / negotiation mechanism for the spin bit

Spin Bit (passive RTT measurement) (2/4)

実装及びデプロイするかは意見が割れた

- ブラウザベンダー等は否定的
- CDNなどのサーバサイドとしては半々くらい

Humの結果

- コアドキュメント側はspin bit用に1bit予約する
  - 任意の値をして良い
- Spin Bitは拡張仕様、ネゴシエーションは行わない

HTTP関連

HTTP WG

- 2セッション開催
  - HTTPtre Issue 整理 / その他
- 扱っているItemは多い
  - 今回は継続のworkが多め

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-03 - **HTTP Semantics**
draft-ietf-httpbis-cache-03 - **HTTP Caching**
draft-ietf-httpbis-messaging-03 - **HTTP/1.1 Messaging**

Woring Items

粛々とIssueの議論

- CDN Loop Prevention
- Variants
- BCP56bis
- Secondary Certificates
- Structured Headers
- Cache Digests for HTTP/2
- Client Hints
- RFC6265bis: Cookies

New Work

「The Cache HTTP Response Header」

- CDNなどで、’x-cache’ というヘッダで、このレスポンスがキャッシュにhitしたかや、エンドポイント情報を示される。標準化されておらず各社バラバラ

```
X-Cache: TCP_MEM_HIT from a118-214-191-15 (AkamaiGHost/6.7.0.2-9183648)
```

これに対し、ちゃんと標準化されたcacheヘッダを定義する提案

おわりに

- QUIC WG
  - Spin Bitと長く議論されていたトピックに終始が打たれた
  - HTTP over QUICからHTTP/3へ
  - マイルストーンは後ろ倒しに
- HTTP WG
  - HTTPtreのメンテナンス
  - 引き続きのWorking Itemが多い印象

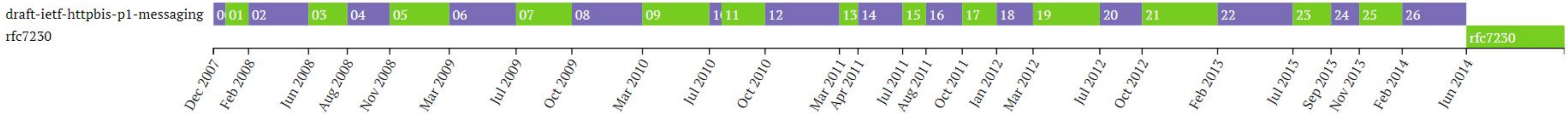
バックアップスライド

ばっくあっp

HTTPtre (HTTP-Core)

- HTTP/1.1 (RFC 7230 ～ 7235) の再改定作業。
- RFC 723x のInternet-DraftがHTTPBisと呼ばれていたので、今度は”HTTPtre”。
- RFC2616 => RFC 723x には7年を要した。HTT/2やQUICなど、様々な変更やIssueがあるためメンテナンスを行う

(HTTPbis WGの名前はそのまま)



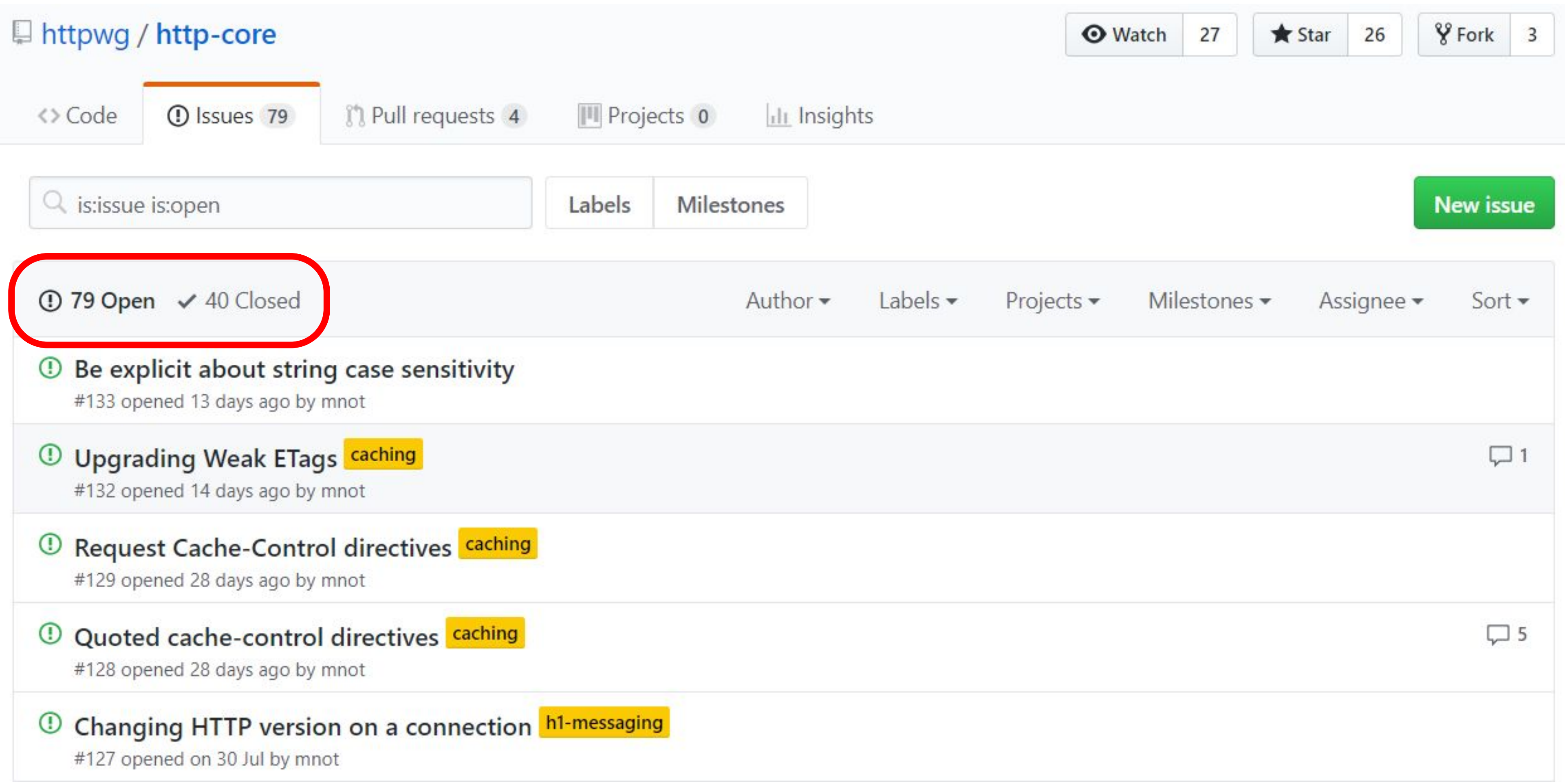
RFC723x の 7年

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-02 - **HTTP Semantics**
draft-ietf-httpbis-cache-02 - **HTTP Caching**
draft-ietf-httpbis-messaging-02 - **HTTP/1.1 Messaging**



https://github.com/httpwg/http-core/issues

CDN Loop Prevention

攻撃者がCDNに悪意ある設定・リクエストを送信することで、永遠にHTTPリエクストを CDNでforwardさせ続ける攻撃手法がある。
(Forwarding-Loop Attacks in Content Delivery Networks)

ループを防ぐためにCDN-Loopヘッダを定義し、それを見ることでループを検知する。

```
CDN-Loop: FooCDN, barcdn; host="foo123.bar.cdn"
CDN-Loop: baz-cdn; abc="123"; def="456", anotherCDN
```

=> WG Adoption (https://tools.ietf.org/html/draft-cdn-loop-prevention-00)

Alt-Svc SNI and DNS ALTSVC (1/2)

alt-svcとは、Webサービスを別の別のプロトコル・サーバで提供できることをクライアントに通知する仕組み。

たとえば、example.comはexample.netのサーバでも提供できる事を示す例

```
Alt-Svc: h2="example.net:443";
```

このレスポンスヘッダを受け取ったブラウザはexample.netにつなぎに行く。このときのSNIは元のドメインの証明書(example.com)になる

Alt-Svc SNI and DNS ALTSVC (2/2)

Alt-SvcにSNIのフィールドを追加する。

```
Alt-Svc: h2="example.net:443";sni=example.net
```

TLS SNIはexample.comを使うが、HTTP/2コネクション確立後に「CERTIFICATE_REQUEST」フレームを送信しexample.netの証明書を要求する

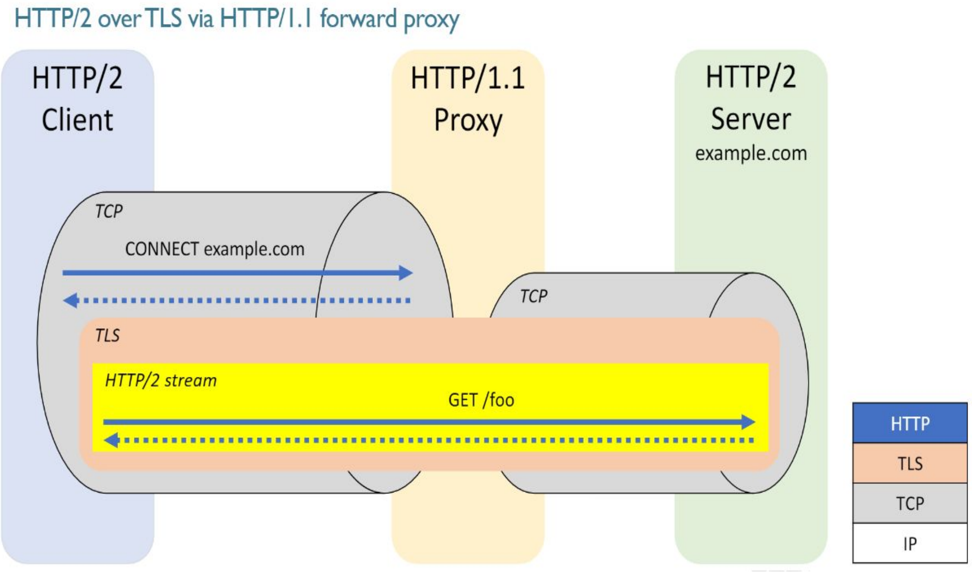
こうすることで、example.netへの接続要求であることが秘匿できる。

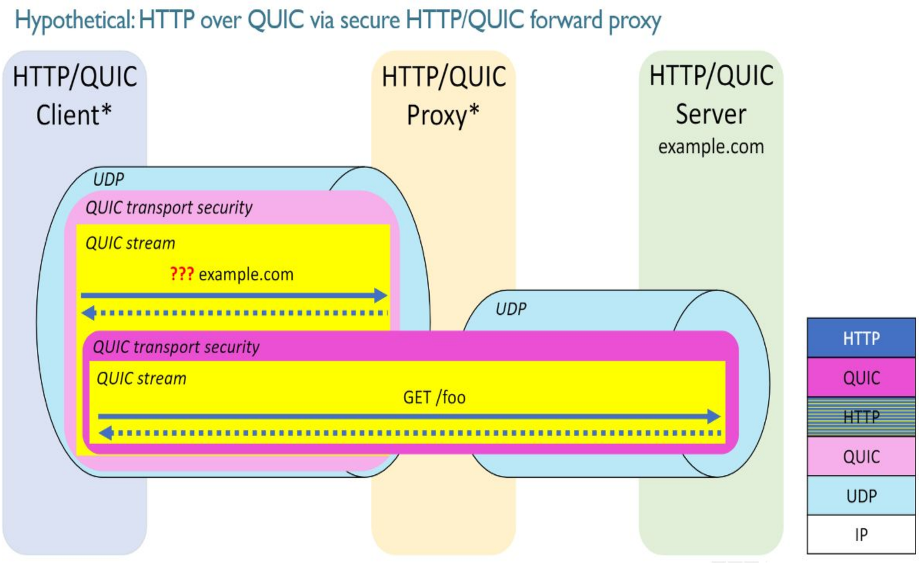
DNS ALTSVCでは、上記のalt-svc情報をDNSレコードで表現できるようになる。

＝＞TLS ESNIとの比較の議論などなど

HTTP-initiated Network Tunnelling

- QUIC(UDP)での、Proxyトンネリングの議論





wpack (Web Packaging)

- Webサイトを一つのファイルにまとめ再配送可能とする仕組み
  - オフラインでのWeb閲覧や、USBメモリによる受け渡しなどの用途
  - そのドメインのページを別のサーバから提供できる
- 2つの仕様からなる
  - Bundled HTTP Exchange: 複数のHTTPリクエスト・レスポンスを一つにする
  - Signed HTTP EXchange: 署名をし再配布可能にする
- これの仕組みによって、本来のURLとは別のところからそのURLからのレスポンスとして提供できるようになる
  - コンテンツの識別子とサーバの場所がURLで一つになっていたが、分離可能となる

wpack (Web Packaging) shide meeting

- IETF 102 でも Shide Meetingが実施される
- 主要なHTTPbisの人たちと予定が合わず、人はまばら
- 内容
  - ユースケースの確認
    - CDN
    - Service Workerとの連携
  - 進め方など
    - セキュリティ上の懸念
    - W3C/WhatWGと連携して勧めていく必要がある

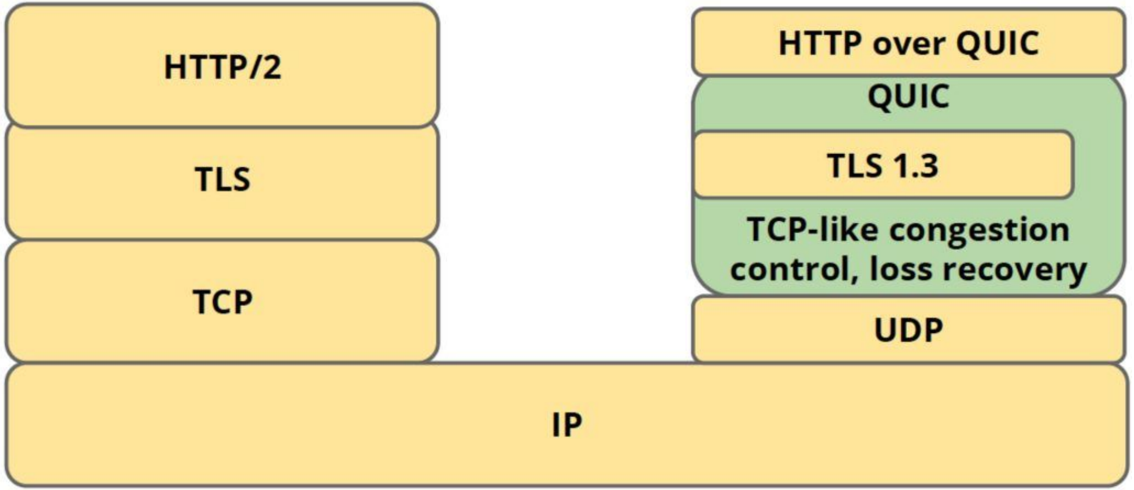
SRV and HTTP side meeting

- HTTPでのSRVの利用についての議論
- Use cases for SRV:
  - 1. Load balancing
  - 2. Service on alternate port
- HTTPではAlt-Svcを使ったりと、別の仕組みを使っている
- より良い方向性は？専用のレコード or Txtレコード?

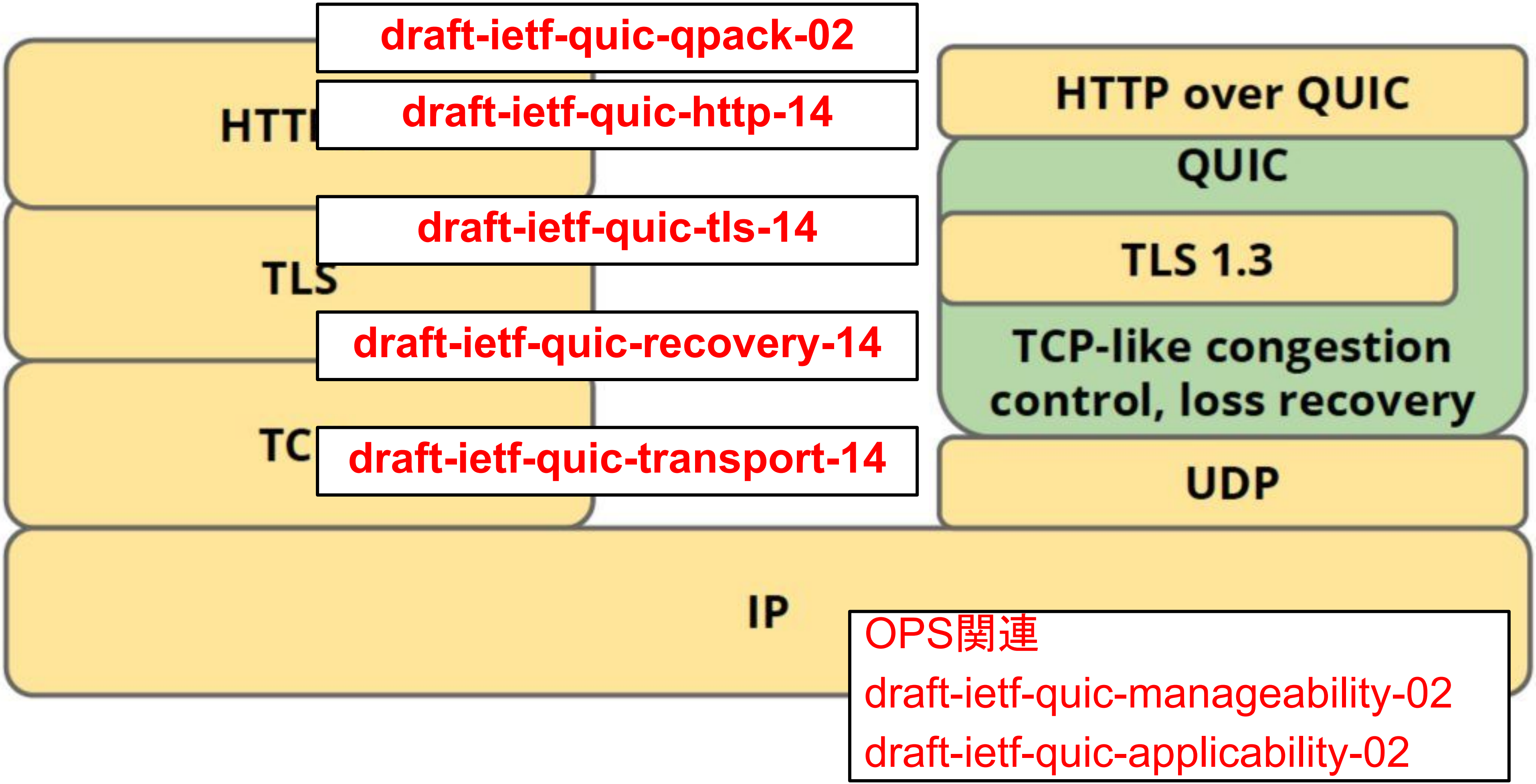
QUIC関連

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-02
HTTP
draft-ietf-quic-http-14
HTTP over QUIC
QUIC
draft-ietf-quic-tls-14
TLS
TLS 1.3
draft-ietf-quic-recovery-14
TCP-like congestion control, loss recovery
TC
draft-ietf-quic-transport-14
UDP
IP
OPS関連
draft-ietf-quic-manageability-02
draft-ietf-quic-applicability-02

ハッカソン

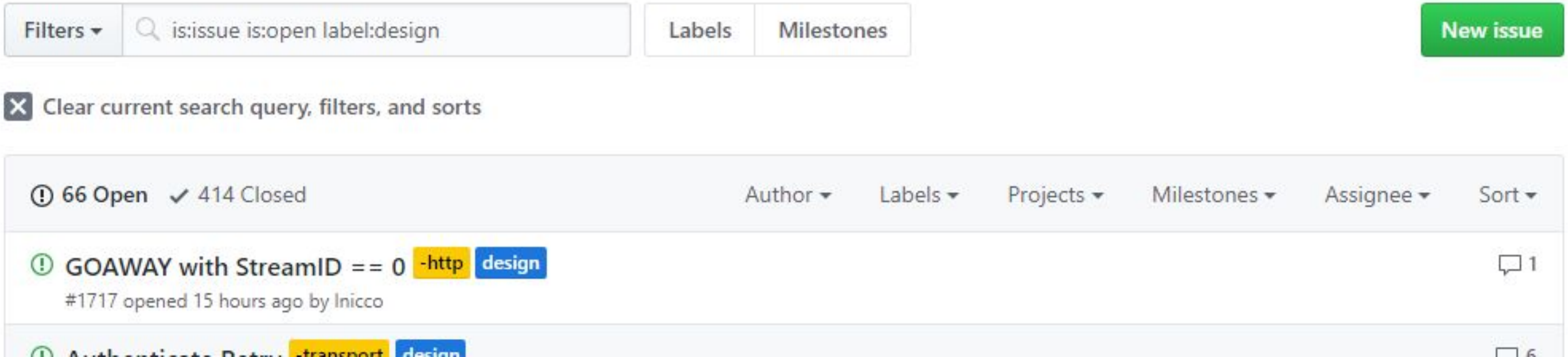
- 7th Implementation Draft

H: handshake
D: Stream Data
C: Connection Close

	Minq	Mozquic	Quicly	quant	ngtcp2	mvfst	picoQUIC	winquic	f5	ATS	Pandora	AppleQUIC	ngx_quic	!gquic	q
Minq	HDRZ		HD										HD		
Mozquic															
Quicly	HD		VHDRZ		VHDRZ								VHDR		
quant															
ngtcp2	HDC		VHDC		VHDCRZ								VHDC		
mvfst															
picoQUIC			VHDCRZ		VHDCRZ		VHDCRZS						VHDCR		
winquic			VHDCRZ		VHDCRZ		VHDCRZS	VHDCRZS		VHDCRZ			VHDCR		
f5															
ATS			VHDC		VHDC					VHDC			VHD		
Pandora															
AppleQUIC															
ngx_quic															
!gquic															

QUIC WG

- 1セッション開催
- やはり、大部屋開催
- 個人的な印象としては
  - 大きな変更は、いくつか入ったが落ち着いてきた => invariants
  - 細かい議論はまだまだたくさんある
  - V1にむけて鋭意策定中



QUIC WG

- 大きな変更
    - パケット番号の暗号化
    - **0 stream の再設計**: TLSハンドシェイク部分の変更
    - ECN in QUIC
- 議論
    - Connection IDs: Migration時のIDの記憶時間
    - Stateless Reset: 切断のループ問題
    - HTTP Priority: ゾンビストリーム
    - Spin Bit: 1bit 予約
    - QPACK

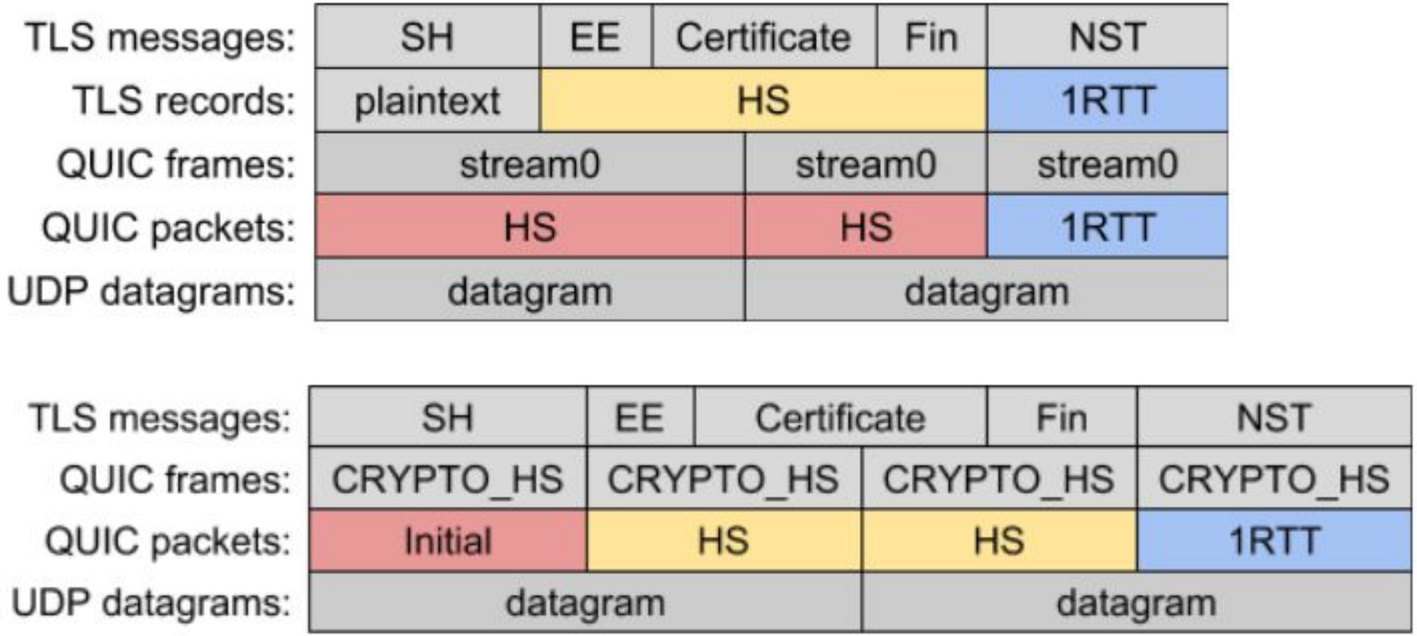
Stream 0 (1/2)

- QUICはTLS1.3のハンドシェイクをStream 0 で行っていた
  - Stream 0が非常に特殊扱いとなり、Stream 0の再送など条件などが複雑になっていた
- QUICとTLSスタックが密結合しており、レイヤリングを行いたいという議論などもあり
- IETF 101では QUIC over DTLSなどの提案もあったが、改めてデザインチームによって議論されることになった

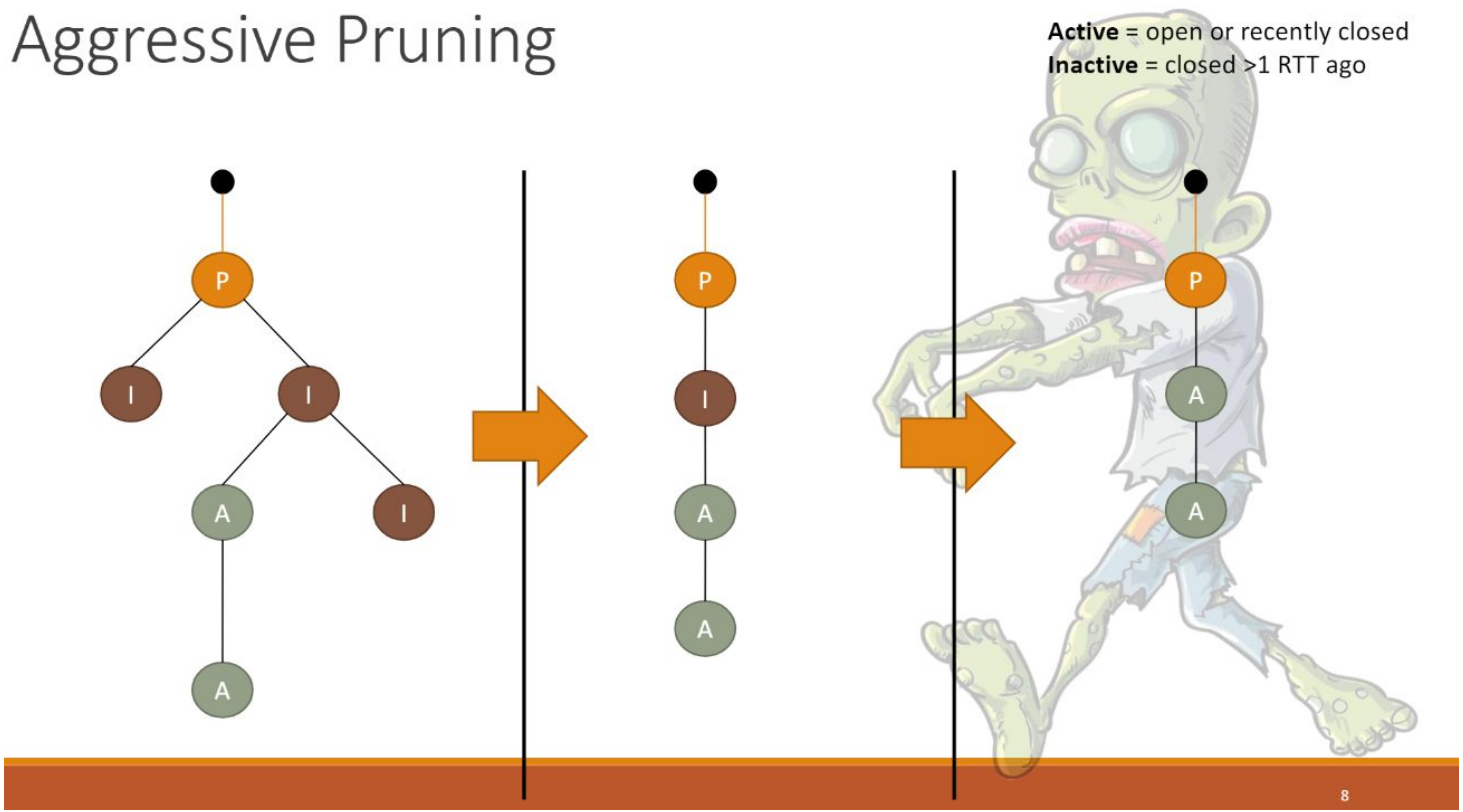
Stream 0 (2/2)

旧方式

新方式



HTTP Priority on QUIC



Other topic

- QUIC Version 44 and IETF QUIC
    - Chrome のGoogle QUICが IETF QUICへの移行をアナウンス
    - Google QUIC v44でInvariantsを対応
    - Chrome Canaryでオプション付きで有効に可能

https://groups.google.com/a/chromium.org/forum/#!msg/proto-quic/b6gZ18W5qn0/woUEYoBkDQAJ

今後

interim (September 17-18)

- 8th Implementation Draft

milestone

- Sep 2018 Version-Independent Properties of QUIC to IESG
- Nov 2018 Core Protocol document to IESG

おわりに

- HTTP
    - 拡張や新しい提案の議論もありつつ
    - メンテナンスを我慢強く進行中
- QUIC
    - 最初の頃に比べれば落ち着いてきた
    - ここの部分についてはまだまだ熱い議論がある