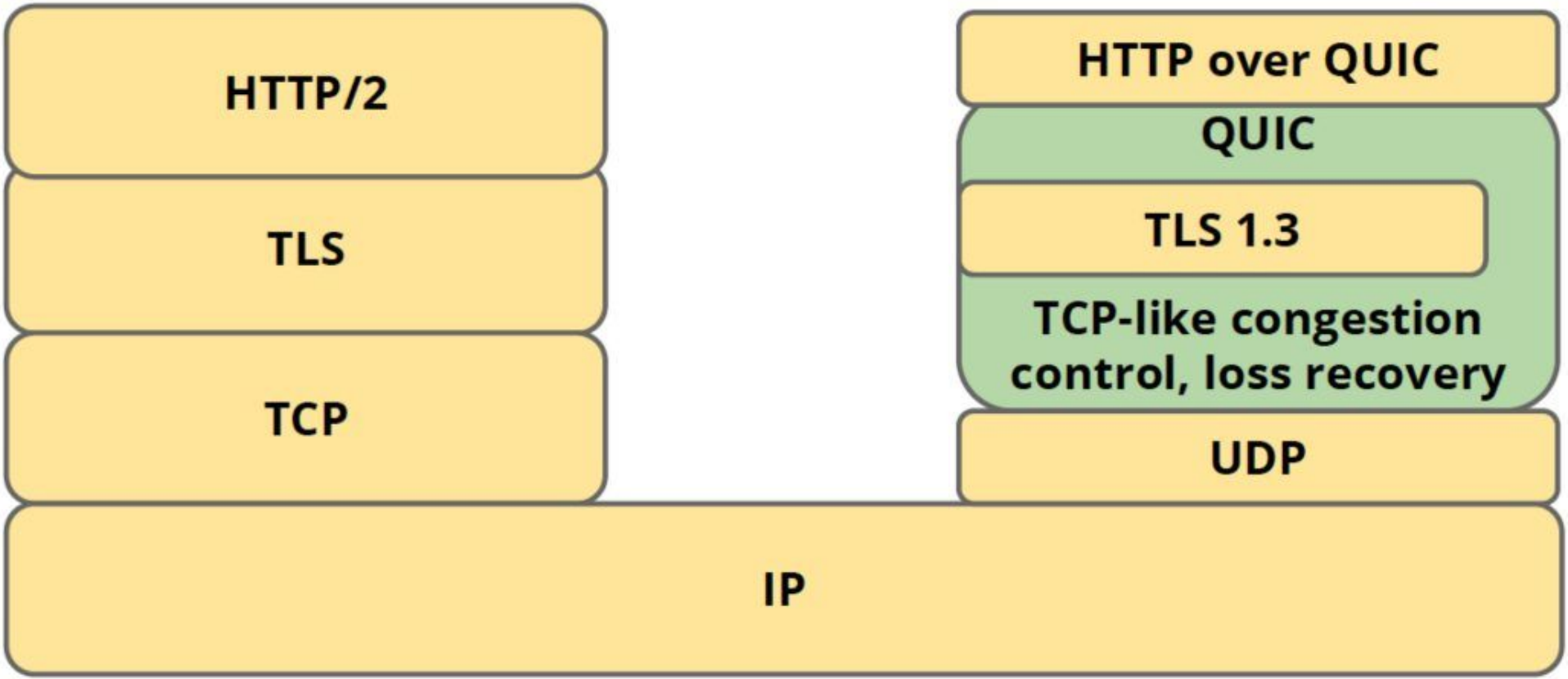
IETF 100 QUIC/HTTP 関連

後藤浩之 (グリー)

自己紹介

- 後藤 浩行 (グリー)
- ISOC-JP インターネット標準化推進委員会
- 興味：Web, HTTP・QUIC関連
  - (Artなので, Transportは...)
- IETF
  - IETF94より、オフライン3回目

QUIC関連


HTTP/2
TLS
TCP
HTTP over QUIC
QUIC
TLS 1.3
TCP-like congestion control, loss recovery
UDP
IP

QUIC WG

- 2セッション開催
- つらつら

ハッカソン

- [Second Implementation Draft]

H: handshake
D: Stream Data
C: Connection Close

client ↓ | server →	Minq	Mozquic	Quicly	Quant	ngtcp2	ngx_quic	mvfst	picoQUIC	quic-go	winquic	!gquic	f5
Minq	HD	HD	HD	**HDC**	HD		HD	HD		HD		
Mozquic	HD	HDC	HD	**HDC**	HD	HD	HD	H		HD		
Quicly	HD	HD	HD	**HDC**	HD	HD	HD			HD		
Quant	H	**HDC**	**HDC**	HDC	**HDC**		**HDC**	**HDC**	HD	**HDC**		
ngtcp2	H	HD	HD	**HDC**	HD	HD	HD	HD		HD		
ngx_quic												
mvfst	H	HD	HD	HD	HD		HDC					
picoQUIC	H	**HDC**	**HDC**	**HDC**	**HDC**			HDC		**HDC**		
quic-go	H		HD	HD			H		HDC			
winquic	HD	HD	HD	HD	HD		HD	HD		HDC		
!gquic	HD			HD							HD	
f5												

スコープ・マイルストーン

- 10月頃、チェアのmnotよりマイルストーンについて懸念が示される。
  - 2018年3月 に十分な品質の4つのコアドキュメントをIESGに提出するマイルストーン
- 議論
  - v1ではアプリケーションとしてHTTPをターゲットにする
  - 以降のバージョンのデプロイを可能にするため、フォーマットレイアウトの不変な箇所を整理する(Invariants 後述)
  - Interim / interoperability testの頻度を上げる
    - January 2018 Melbourne
  - マイルストーンの期日はIETF100では決まらず

＝＞ (12/14) 6ヶ月後ろだ押しのマイルストーンに変更された

invariants

- 将来のバージョンをデプロイしやすく(Middleboxの影響を緩和)するために、フォーマットの特定のフィールドを将来的に普遍とする
- avtcore wg側で rfc7983 (STUN, DTLS, TURN Channel… をフラグで見分ける)とのコリジョンに付いても話題に

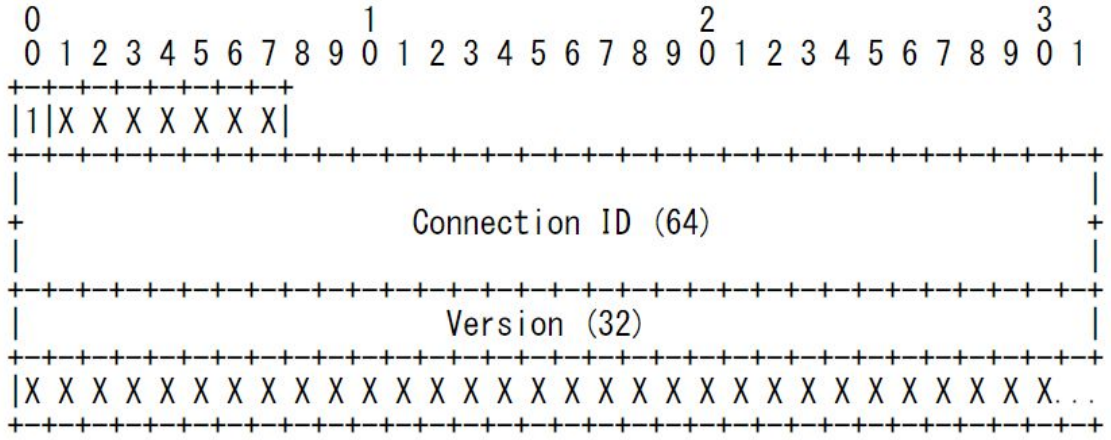


Figure 1: QUIC Long Header

draft-thomson-quic-invariants-00

Spin Bit (passive RTT measurement)

- IETF99 より、経路上でRTTを測定し最適化を行うケースがあり、QUICでどのようにするか議論があった。
  - QUICではAckも暗号化され、ハンドシェイク時しかRTTが測定できない
- 1 bitを経路上に露出し、RTTを測定可能にする提案
  - クライアントはパケットが往復するたびにbitを反転する。サーバはクライアントのbitをそのまま返す
- デザインチームにより、有用性およびプライバシーへの影響が調査される
- IETF100 後にi-dが出た (draft-trammell-quic-spin)

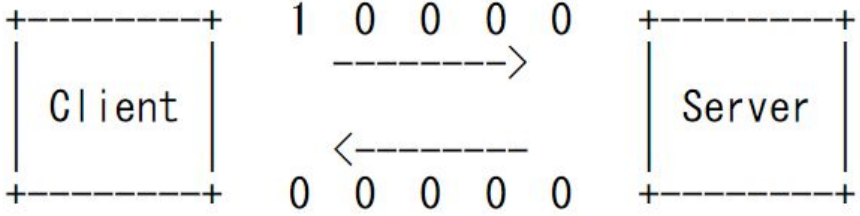


Figure 4: The bit begins spinning

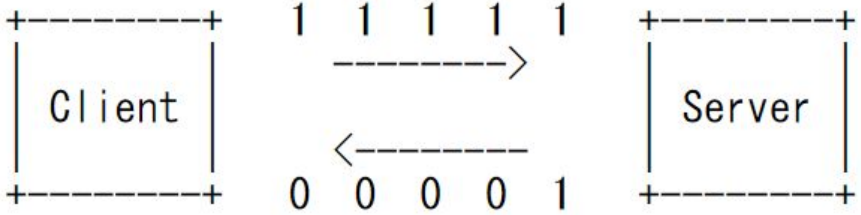


Figure 5: Server reflects the spin edge

その他のトピック

- Loss Recovery
  - Loss Ditection, Timeout, Ack等の話
- Connection Migration
  - マイグレーションの手順(Ping, Pong)
- Third Impremataion Draft
  - 0-RTT, loss recovery などのinterop
  - HTTPを試すためにヘッダ圧縮の議論をすすめる必要がある

HTTP関連

HTTP WG

- 2セッション開催
  - 2セッションともHTTP関連(QUICは無し)
- 扱っているItemは多い (次ページ)
  - パフォーマンスの改善
  - セキュリティの改善
  - メンテナンス系

Item

- **Bootstrapping WebSockets with HTTP/2 (new)**
- **Variants (new)**
- **Structured Headers for HTTP (new)**
- Secondary Certificates (**wg draftへ**)
- HTTPter
- Using Early Data in HTTP
- Random Access and Live Content (**wg draftへ**)
- Expect-CT
- Cache Digests for HTTP/2 - presentation
- Client Hints
- RFC6265bis: Cookies
- BCP56bis (**進めるコンセンサス**)

Bootstrapping WebSockets with HTTP/2

- 現状 HTTP/2 上でWebSocketsの通信することは出来ない
  - upgradeヘッダや、ConnectメソッドがHTTP/2では意味が異なるため
- Connectメソッドを使用できるように変更し、ストリームをWebSocket通信用に変更する
- SETTINGSパラメータが追加された
- WG Adopted

5.1. Example

[[From Client]]

HEADERS + END_HEADERS
:method = CONNECT
:protocol = websocket
:scheme = https
:path = /chat
:authority = server.example.com:443
sec-websocket-protocol = chat, superchat
sec-websocket-extensions = permessage-deflate
sec-websocket-version = 13
origin = http://www.example.com

DATA
WebSocket Data

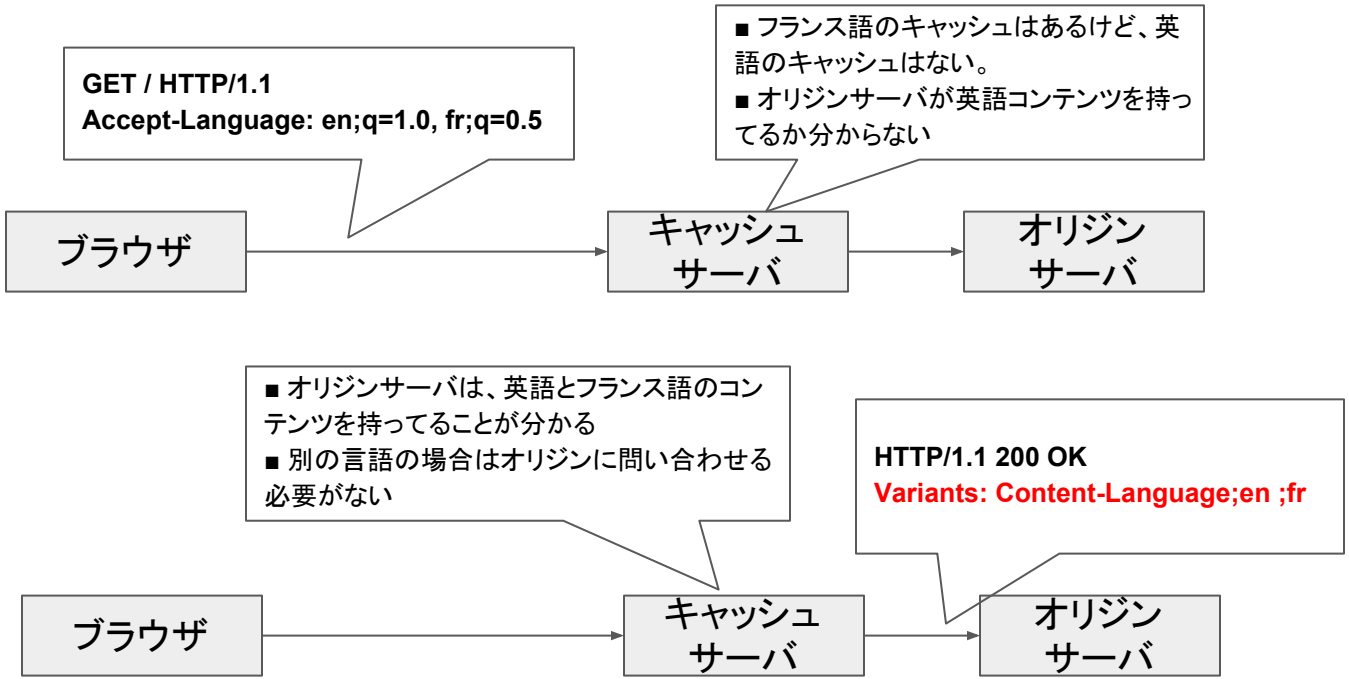
[[From Server]]

SETTINGS
ENABLE_CONNECT_PROTOCOL = 1

HEADERS + END_HEADERS
:status = 200
sec-websocket-protocol = chat

Variants

- オリジンサーバからキャッシュサーバに、自信の持ってるコンテンツの種類を通知可能にする variants HTTPレスポンスヘッダを定義する提案
- より多くの知見，フィードバックが必要



Structured Headers for HTTP

- HTTPヘッダに構造・シンタックス定義を与える提案
- 今までのHTTPヘッダは、ヘッダ毎に構造やシンタックスを定義している
  - パーサーが再利用できない
  - 新しいヘッダを定義するたびに、構造及びシンタックスを定義する必要がある
- 既存のヘッダを置き換えるものではない
- 会場からは好意的な意見

```
ExampleNumberHeader: 4.5
ExampleLabelListHeader: foo, bar, baz_45
ExampleParamHeader: abc; a=1; b=2; c
ExampleBinaryHeader: *cHJldGVuZCB0aGlzIGlzIGJpbmFyeSBjb250ZW50Lg
```

おわり