

HTTP/3, ce n'est pas seulement HTTP/2 avec un numéro plus grand

Stéphane Bortzmeyer, Afnic

bortzmeyer@afnic.fr

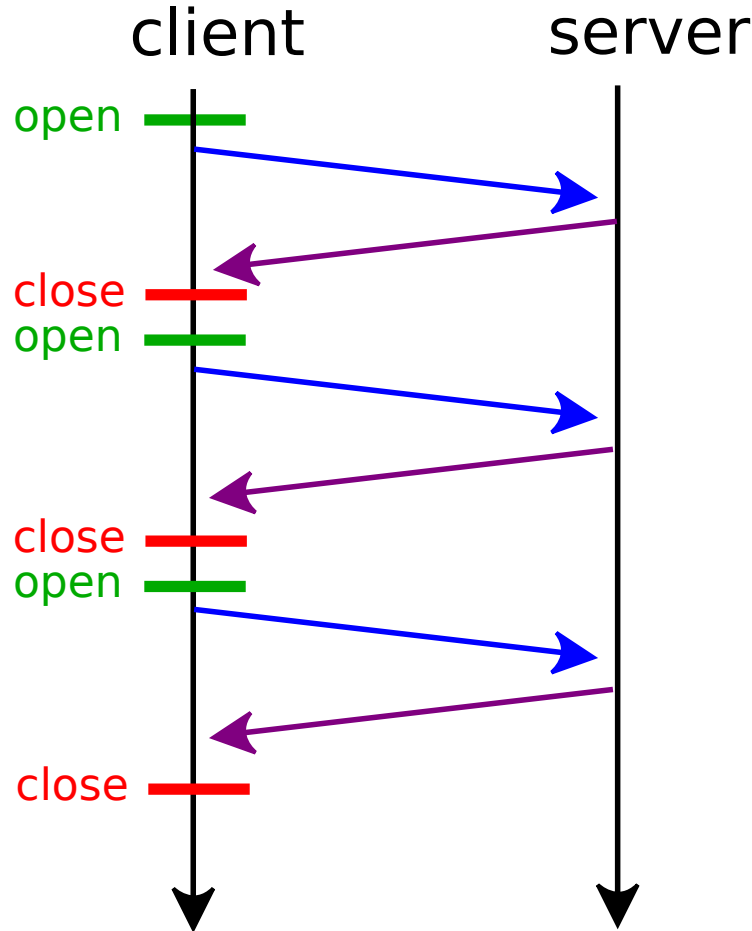
Paris Web 2021

L'histoire de HTTP

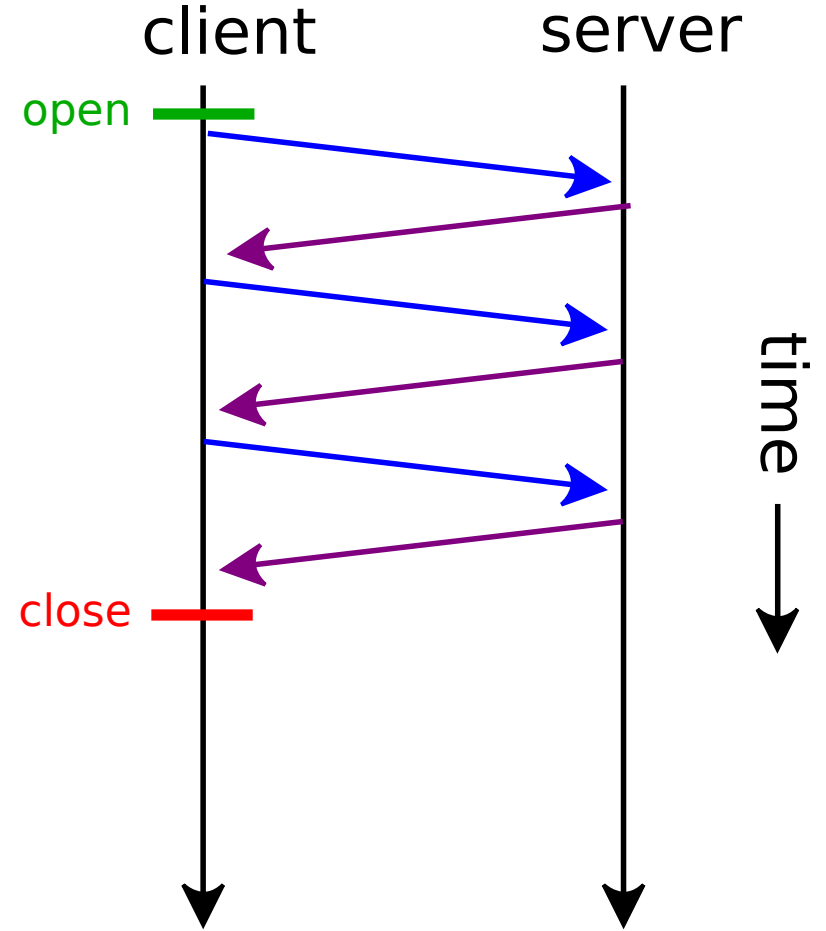
- HTTP 0.9 : pas de méthode, pas d'en-têtes
- HTTP/1 (1996) : premier à être normalisé ; pas de Host : (donc pas de virtual hosting)
- HTTP/1.1 (1997) : plein d'en-têtes, et les connexions persistentes
- Jusqu'à 1.1 inclus, encodage texte et transport sur TCP

C'est mieux quand ça dure longtemps (Source: helix84, Wikimedia Commons)

Multiple Connections



Persistent Connection



Histoire, suite

- HTTP/2 (2015) : passage à un encodage binaire. Multiplexage des requêtes sur plusieurs ruisseaux.
- HTTP/3 (2021 ?)

Les limites de HTTP/1

1. Encodage texte, plus lent à analyser
2. Le Web nécessite plusieurs ressources par page ; une connexion TCP ou plusieurs ?
3. Solution : HTTP/2 ; multiplexage des ressources sur plusieurs **ruisseaux**, dans une seule connexion TCP

Les limites de HTTP/2

1. Latence : il faut démarrer TCP **puis** TLS
2. Multiplexage limité par TCP ; un paquet perdu et tous les ruisseaux stoppent
3. Solution : HTTP/3 ; utilisation de QUIC à la place de TCP

Mais au fait à quoi sert TCP ?

1. HTTP veut un canal fiable : que les octets arrivent tous, et dans l'ordre
2. L'Internet ne fait pas cela tout seul, client et serveur doivent utiliser TCP
3. TCP = protocole de Transport (couche 4)



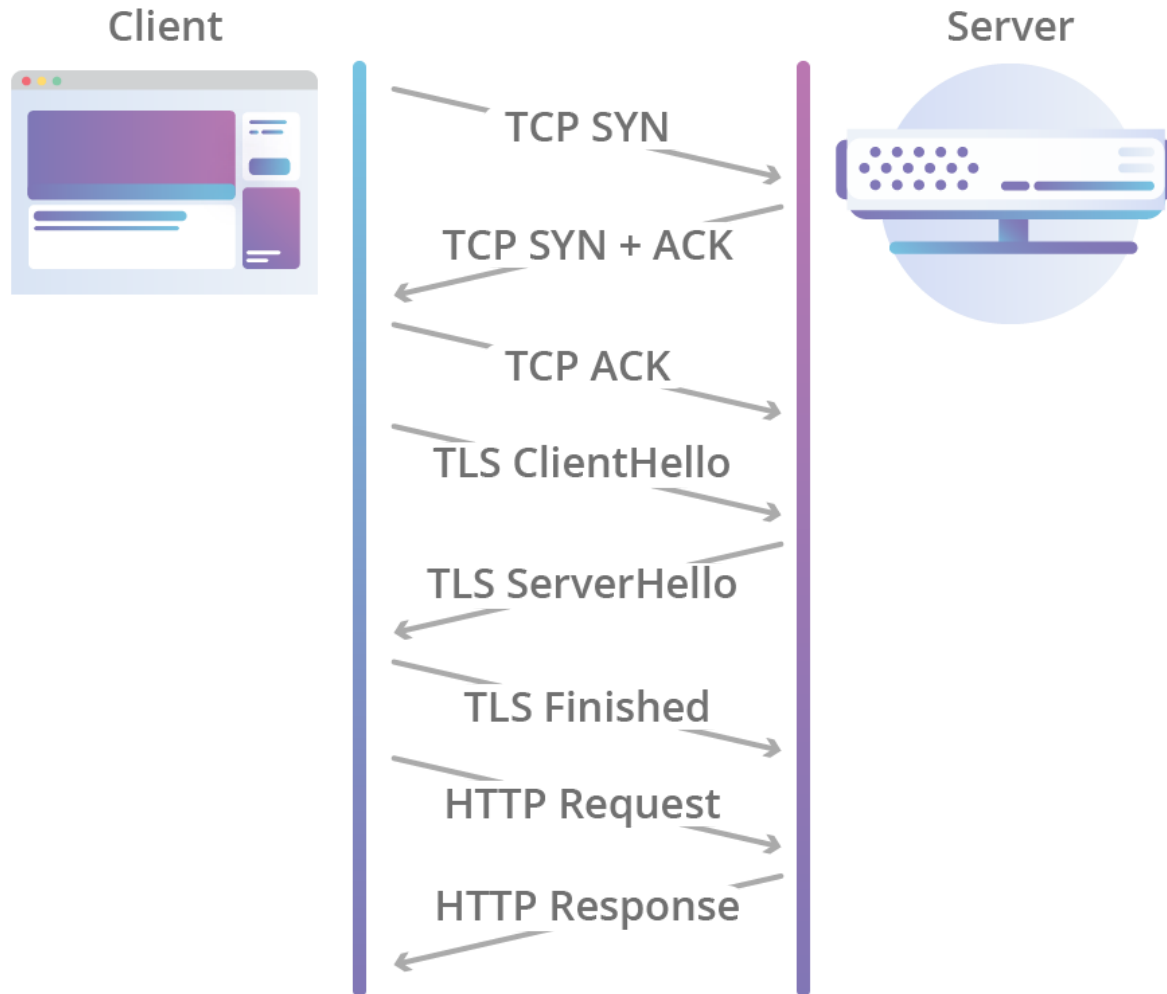
1. Fusionne transport et chiffrement ; meilleure latence
2. Contrôle de flux par ruisseau ; un paquet perdu ne stoppe plus tous les ruisseaux

HTTP/3

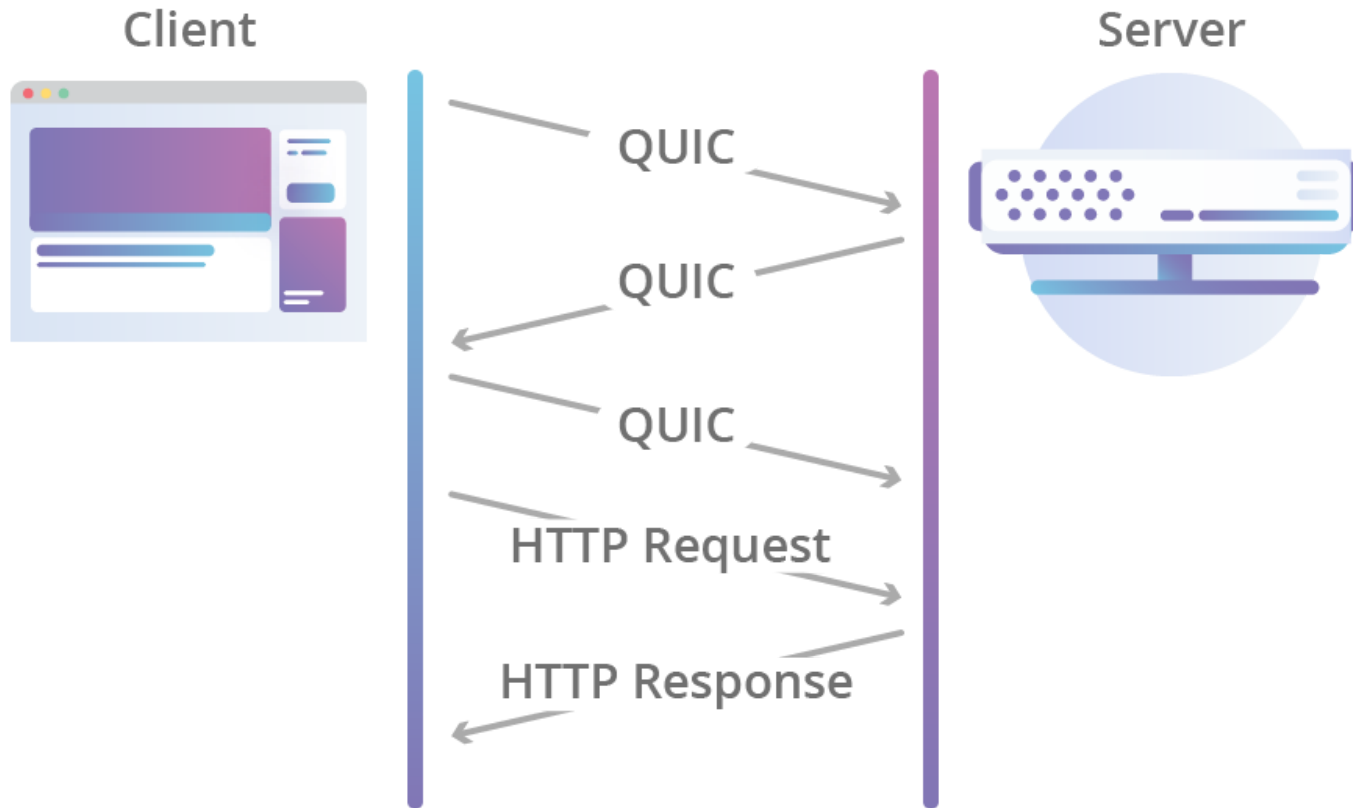
1. HTTP sur QUIC et plus sur TCP
2. TLS systématique
3. Encodage binaire, très proche de celui de HTTP/2
4. Création d'un ruisseau pour chaque couple requête/réponse : parallélisme maximal
5. Pas de changement pour l'utilisateur : c'est du Web comme avant, avec ses pubs et ses contenus sans intérêt

(Source: Cloudflare)

HTTP Request Over TCP + TLS



HTTP Request Over QUIC



Le problème du choix

1. Comment savoir si le serveur en face fait du QUIC ?
2. Rien ne l'indique dans l'URL
3. Essayer en HTTP/2 et être redirigé par l'en-tête Alt-Svc:
4. Et mémoriser

Essaie plutôt en HTTP/3

```
% curl -v https://www.google.com/  
...  
< HTTP/2 200  
< date: Tue, 17 Aug 2021 06:39:17 GMT  
< content-type: text/html; charset=ISO-8859-1  
< server: gws  
< alt-svc: h3=":443"; ma=2592000
```

Que sont nos adresses IP devenues ?

1. QUIC, encore plus que HTTP/1.1, encourage les connexions de longue durée
2. QUIC permet la migration d'adresse IP en cours de connexion
3. Les logs ne sont plus ce qu'ils étaient

Normalisation de HTTP/3

1. Les RFC sur QUIC ont été publiés en mai 2021
2. Le futur RFC sur HTTP/3 est terminé
3. Sa publication dépend de deux normes presque finies, génériques à toutes les versions de HTTP (sémantique et mémorisation)

HTTP/3 et TLS

1. HTTP/3 utilise TLS **mais** :
2. En fait, juste une partie
3. Les bibliothèques TLS classiques ne permettaient pas cela
4. Si vous développez, vous devrez peut-être changer de bibliothèque

Clients HTTP/3

1. `google-chrome --enable-quirks`
2. Firefox `about:config` puis `network.http.http3.enabled`
3. `curl --http3`
4. Pas encore de HTTP/3 dans les bibliothèques client typiques, il faut bricoler

Serveurs HTTP/3

1. Pour l'instant, pas évident à déployer
2. Test en <https://http3check.net/>
3. Les habituels GAFA
4. Pas encore dans nginx (fin 2021 ?) ou Apache (rien d'annoncé)
5. Si vous testez, vérifiez que le pare-feu est bien configuré (HTTP/3 n'utilise pas TCP)

Déploiement de HTTP/3

1. En pratique, ça va vraiment plus vite ?
2. Surtout utile pour les gros sites (connexions de longue durée)
3. Attention au risque de suivi de l'utilisateur

HTTP/3 vu par Wireshark

```
▶ Frame 8: 115 bytes on wire (920 bits), 115 bytes captured (920 bits)
▶ Ethernet II, Src: 4e:d7:64:90:67:4f (4e:d7:64:90:67:4f), Dst: fe:5a:84:0a:0b:37 (fe:5a:84:0a:0b:37)
▶ Internet Protocol Version 4, Src: 10.30.1.1, Dst: 45.77.96.66
▶ User Datagram Protocol, Src Port: 43260, Dst Port: 8443
- QUIC IETF
  ▶ QUIC Connection information
    [Packet Length: 73]
  ▶ QUIC Short Header DCID=3bd5658cf06b1a5020d44410ab9682bcf277610f PKN=1
  - STREAM id=0 fin=1 off=0 len=32 uni=0
    ▶ Frame Type: STREAM (0x0000000000000000b)
      Stream ID: 0
      Length: 32
      Stream Data: 011e0000d1c1d7508bed6988ba4a49ee3cd34cff5f508825b650c3abbae2e1dd
- Hypertext Transfer Protocol Version 3
  Type: HEADERS (0x00000000000000001)
  Length: 30
  Frame Payload: 0000d1c1d7508bed6988ba4a49ee3cd34cff5f508825b650c3abbae2e1dd
```

Bonne lecture

<https://http3-explained.haxx.se/>



Des questions ? Des
objections ? Des
remarques ?