QUIC Version Negotiation

draft-ietf-quic-version-negotiation

IETF 110 – Virtual Prague – 2021-03

David Schinazi – dschinazi@google.com
Eric Rescorla – ekr@rtfm.com

A brief history of QUIC Version Negotiation

2013: GoogleQUIC adds version negotiation and downgrade protection

2016-07: IETF QUIC initially had it too

2018-09: issues found with incremental server deployments Issue#1810

2019-02: removed VN from the base drafts to unblock them via PR#2313

2019-03: published draft-schinazi-quic-version-negotiation-00

2020-02: adopted as draft-ietf-quic-version-negotiation-00

QUICv1 is about to ship, where does it stand on VN

Invariants define format of VN packet

QUICv1 says that client aborts the connection on receipt of VN

Main use-case of HTTP/3 can survive without VN because of Alt-Svc

But QUIC is general-purpose

Requirement: allow VN without spending a round trip for similar versions

Incompatible Version Negotiation

Client sends first flight using version A ------->

<------- Server sends VN with list of support versions

Client sends another first flight with version B ------->

Compatible Version Negotiation

Client sends first flight using version A (listing compatible versions) ------->

<------- Server sends first flight with a compatible version

But what is a "Compatible Version" anyway?

Conceptually means that you can convert a first flight from one version to another

Note: not bijective, A compatible with B doesn't imply B compatible with A

Note: not all first flights need to be compatible (e.g., new frame added but not used)
client might profile its first flight to facilitate compatibility

Handshake Version Information

Sent during handshake – in QUICv1, uses transport parameter

Prevents downgrade attacks
Server performs verification to allow gradual deployment and multi-CDN

Allows exchanging compatible and supported versions

```
Client Handshake Version Information {
   Currently Attempted Version (32),
   Previously Attempted Version (32),
   Received Negotiation Version Count (i),
   Received Negotiation Version (32) ...,
   Compatible Version Count (i),
   Compatible Version (32) ...,
}
```

```
Server Handshake Version Information {
   Negotiated Version (32),
   Supported Version Count (i),
   Supported Version (32) ...,
}
```

Design decision: where is compatibility defined? #19

Current draft states that compatibility can be defined at any time
e.g., two versions developed independently can be made compatible later

This means that client and server can disagree on compatibility

What happens if client assumed compatibility but server disagrees?
- if server understands original version, that is negotiated
- if server doesn't understand original version, it sends VN

We could drop the requirement and slightly simplify the design

QUIC Version Negotiation

draft-ietf-quic-version-negotiation

IETF 110 – Virtual Prague – 2021-03

David Schinazi – dschinazi@google.com
Eric Rescorla – ekr@rtfm.com