

QUIC Version Negotiation

[draft-ietf-quic-version-negotiation](#)

IETF 113 – Vienna – 2022-03-22

David Schinazi – dschinazi.ietf@gmail.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

2021-04: QUIC WG interim dedicated to version negotiation

Consensus to keep compatible and incompatible but try to simplify

2021-05: draft-04 published with simplified design

2021-10: draft-05 published with editorial changes

2022-03: draft-06 published with (mostly) editorial changes

#55: Backwards compatibility

Scenario: client supports this draft but server doesn't

Client sends QUICv2 INITIAL

Server responds with Version Negotiation packet containing only QUICv1

Client performs incompatible VN and sends QUICv1 INITIAL

Server completes the handshake without version_information

Client aborts

Proposal #66: special-case QUICv1?

#90: What does "compatible VN" mean

When the server transparently "upgrades" from the client's original version to a different version, that's definitely "compatible VN"

But when the server negotiates the client's original version, is that "compatible VN"?

Next Steps

This draft is close to done

Do we tie timeline to QUICv2 draft?

QUIC Version Negotiation

[draft-ietf-quic-version-negotiation](#)

IETF 113 – Vienna – 2022-03-22

David Schinazi – dschinazi.ietf@gmail.com

Eric Rescorla – ekr@rtfm.com