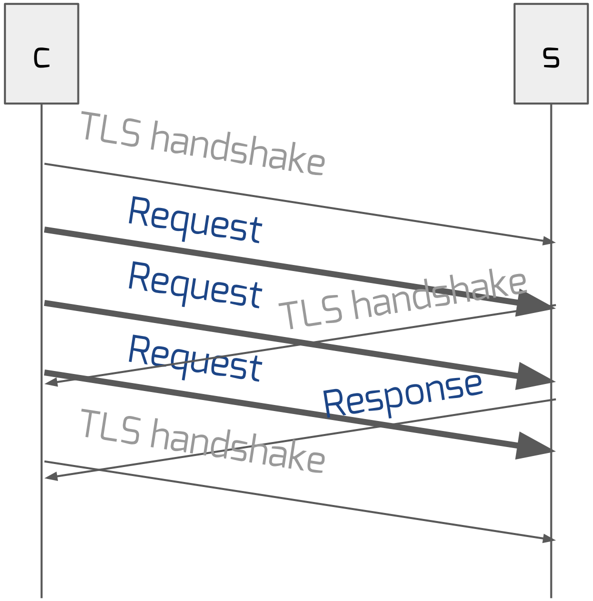
Replays in HTTP

draft-thomson-http-replay-00
IETF 99

TLS 1.3 0-RTT in HTTP



0-RTT lets the client send requests before the TLS handshake is done

These requests can be replayed

TLS mandates anti-replay

... including rejection of 0-RTT

... but measures are imperfect

Summary

The TLS connection is a single stream

- If accepted, 0-RTT is concatenated with 1-RTT

Advice for what to send in 0-RTT and what to accept

Two mechanisms for use in intermediaries

Advice for Clients

Recommend only sending requests with **safe** methods

Automatically retry requests if 0-RTT is rejected

… important: this enables an attack

Advice for Servers

Consider whether to even enable 0-RTT

Be careful about what you start processing before the handshake completes

Assertion:

> If servers always defer processing of requests until after the handshake completes, replays aren't fatal

Recommendation:

> Defer processing if you aren't sure

Intermediaries

Intermediaries can’t decide what is safe on their own

An intermediary marks requests that arrive in early data with the Early-Data header field

An origin server can reject requests with 4XX (Too Early)

… or if the request arrives in 0-RTT

Clients retry requests if they see 4XX (Too Early)

Important: Early-Data/4XX (Too Early) require coordination between gateway and origin server

4XX (Too Early)

4XX (Too Early) == Permission to retry

… even for POST

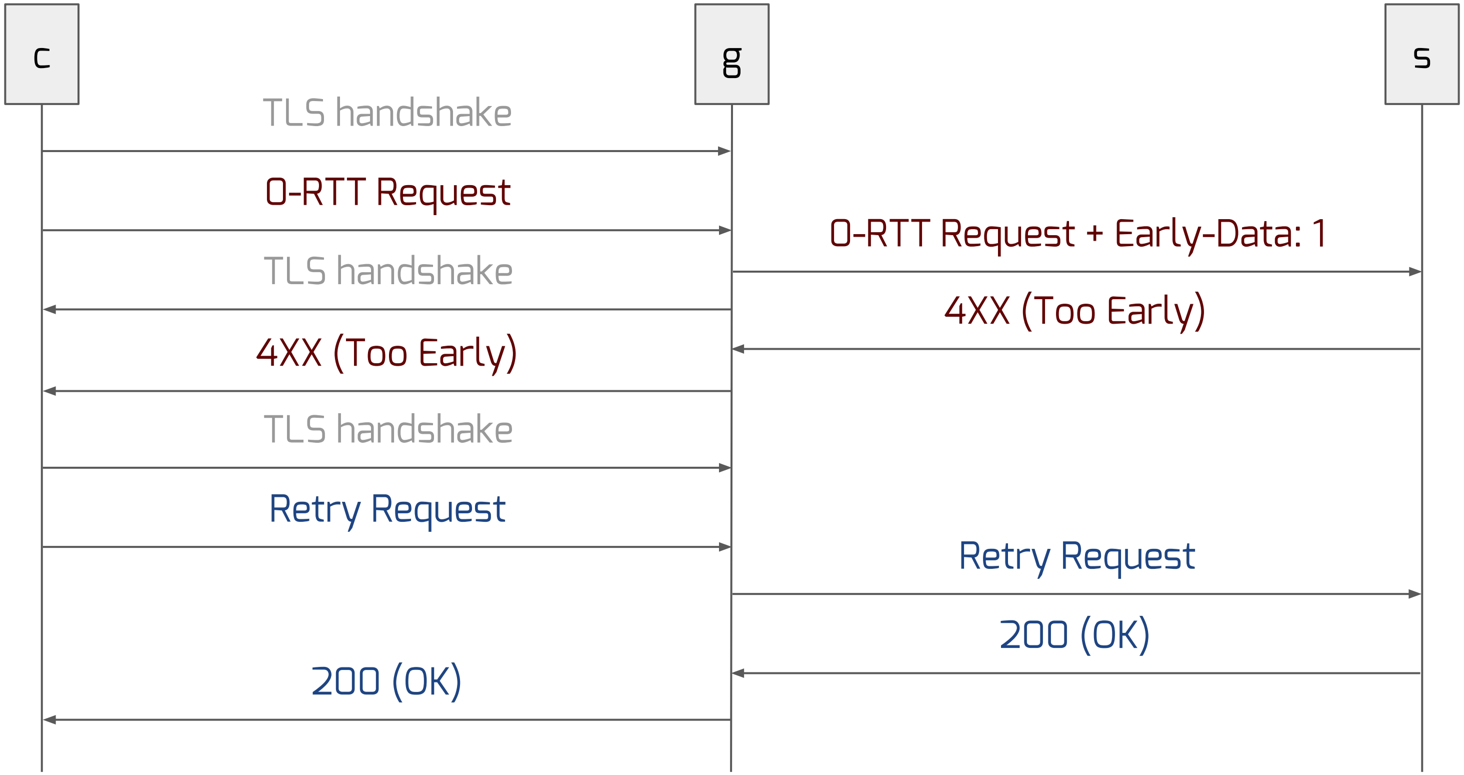
Clients can be more aggressive about attempting 0-RTT

… if they think it is safe, they can attempt it

Server decides if the request is safe to process

… they use 4XX (Too Early) if it isn’t safe

Example


c
g
s
TLS handshake
0-RTT Request
0-RTT Request + Early-Data: 1
TLS handshake
4XX (Too Early)
4XX (Too Early)
TLS handshake
Retry Request
Retry Request
200 (OK)
200 (OK)