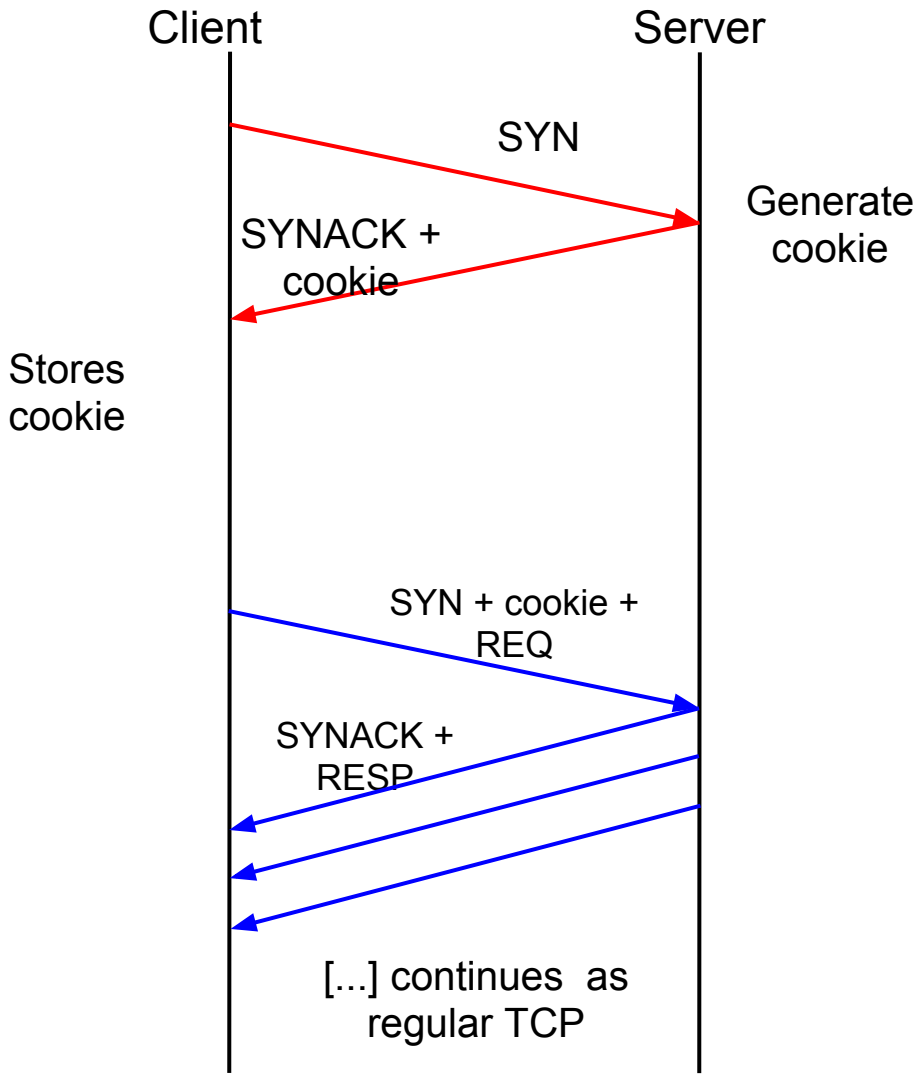
en.wikipedia.org/wiki/TCP_Fast_Open

draft-ietf-tcpm-fastopen-03

Yuchung Cheng, Jerry Chu, Siva Radhakrishnan, Arvind Jain

TCP Fast Open (TFO) recap



Server grants nonce

Clients replays nonce with SYN/data

Nonce

- AES_encrypt(cli_IP, secret)
- TCP option (32 - 64bits)

Defend simple SYN-data flood attacks

What changed bet. -02 and -03

Draft

- Extended discussions on duplicate SYN-data cases
- Impact of CC on SYNACK losses
- Simultaneous (Fast) Open
- Negative caching is SHOULD

Deployment

- Entire Google.com is TFO ready
- Chrome (27.0.1425.0) with --enable-tcp-fastopen on Linux 3.6+

Caveats of SYN-data duplication

If the server receives a (network) duplicated SYN-data after

1. server reboots after receiving original one
2. original connection is closed w/o a 2MSL wait (receiving FIN)

Possible but improbable the data is delivered twice.

Section 2: TFO MUST NOT enabled by default, and applications need to read Section 7 before using TFO

Data replay in Web/HTTP

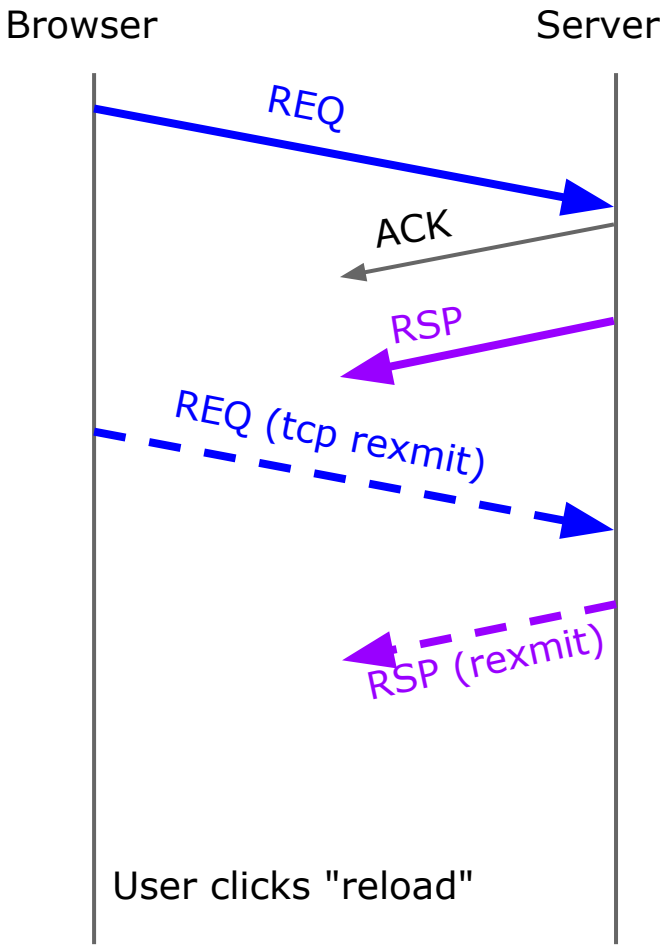
Reloading a Web page implies replaying same requests twice

Web app implements separate transaction mechanisms (e.g., uid in POST).

With TFO the replay may happen w/o user reload:

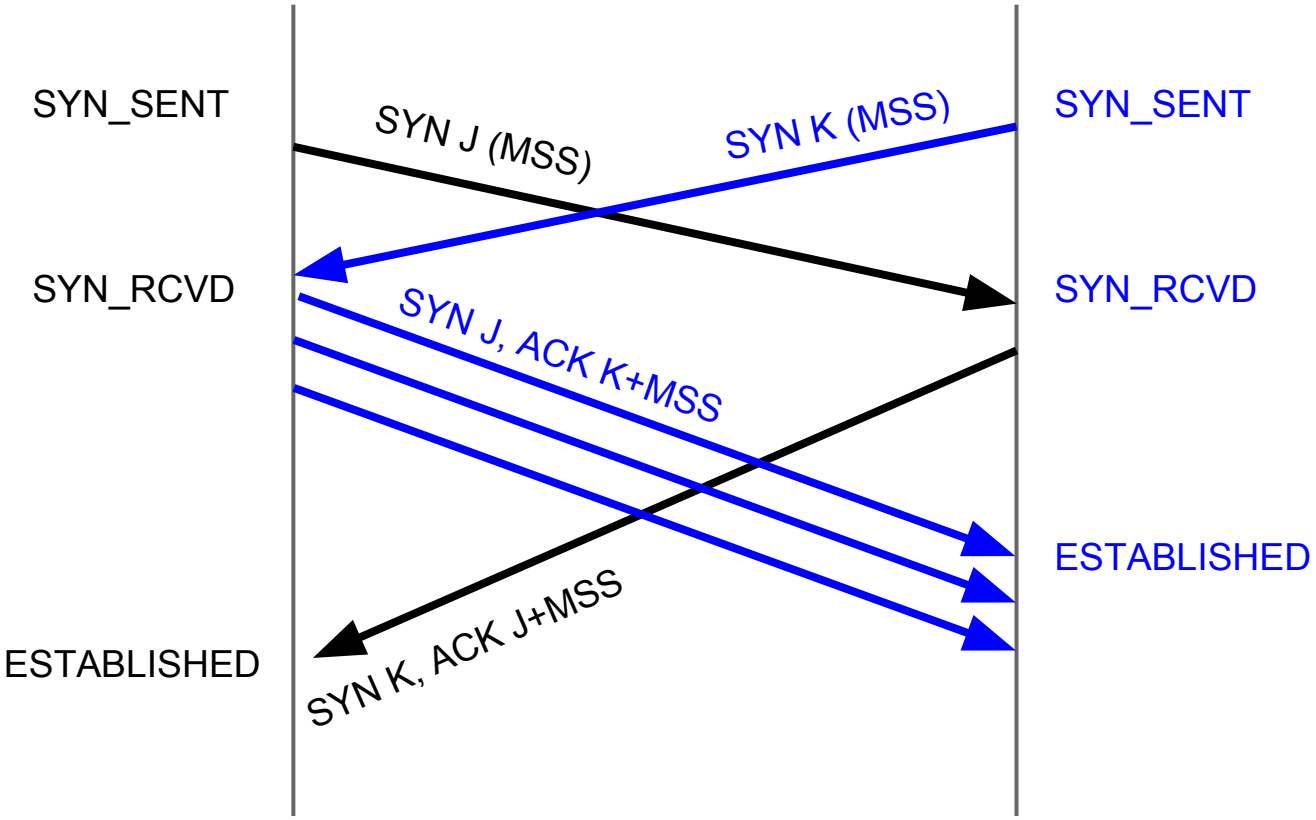
1. http: first HTTP request
2. https: SSL HELLO

Browser should only use TFO if the request is safe to replay



Simultaneous (Fast) Open

No special handling needed b/c RFC793 supports simul. open and data in SYN already



Implications to Congestion Control

TFO does NOT change congestion control but has subtle different behavior if SYN-ACK is lost.

Standard: initial window = LW = 1

TFO: initial window = IW > 1
but react to SYN-ACK loss later

Recommendation?