TCP_REPLENISH_TIME

HotRFC, IETF 121 Dublin, November 2024

Stuart Cheshire, Apple

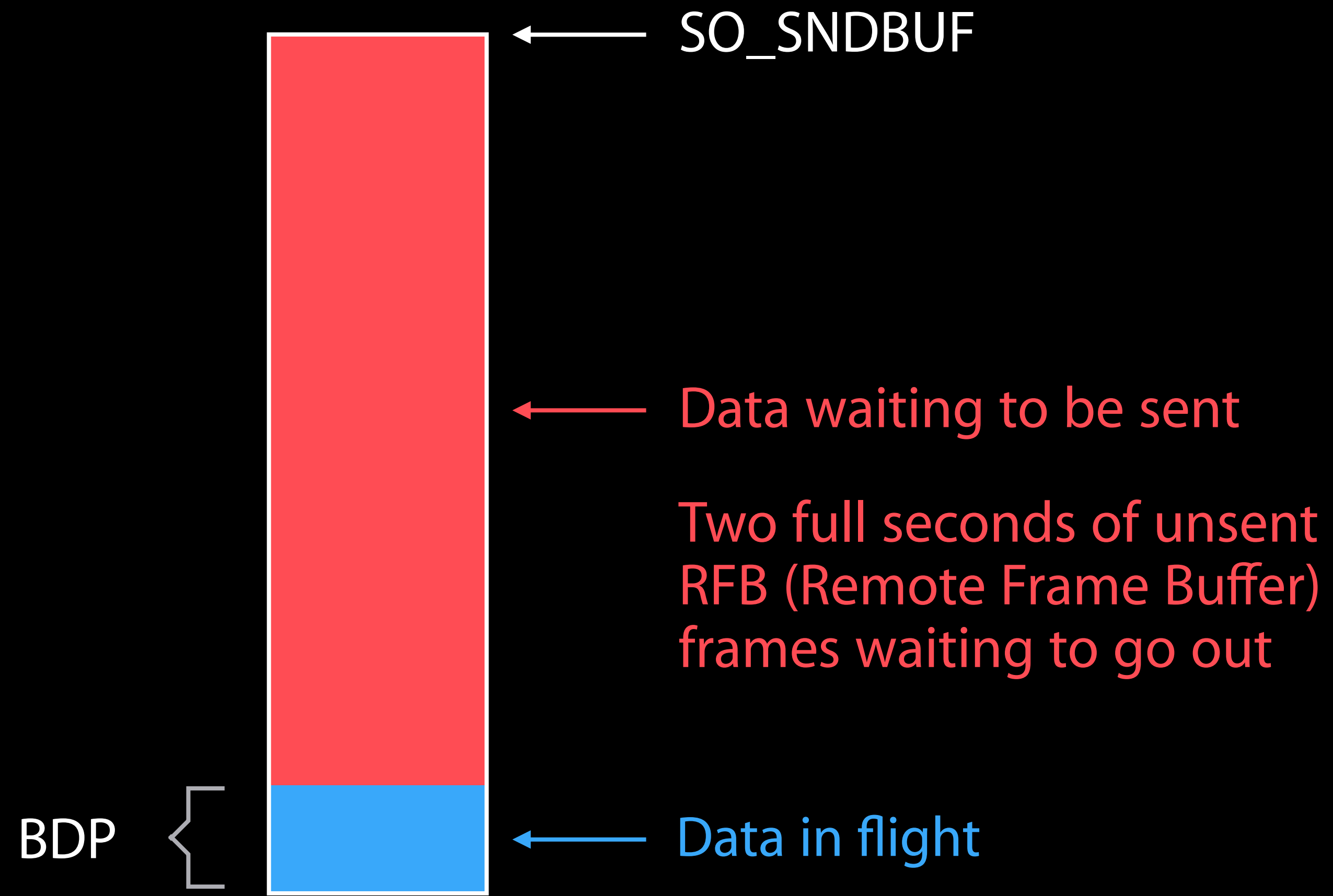# Source-Device Bufferbloat
April 2011

Mac OS Screen Sharing sluggish on slow networks

Network Bufferbloat suspected

Real cause was excessive buffering by the sender

Sluggish Screen Sharing
Source-device Bufferbloat

SO_SNDBUF

← Data waiting to be sent

Two full seconds of unsent RFB (Remote Frame Buffer) frames waiting to go out

BDP

← Data in flight

TCP_NOTSENT_LOWAT
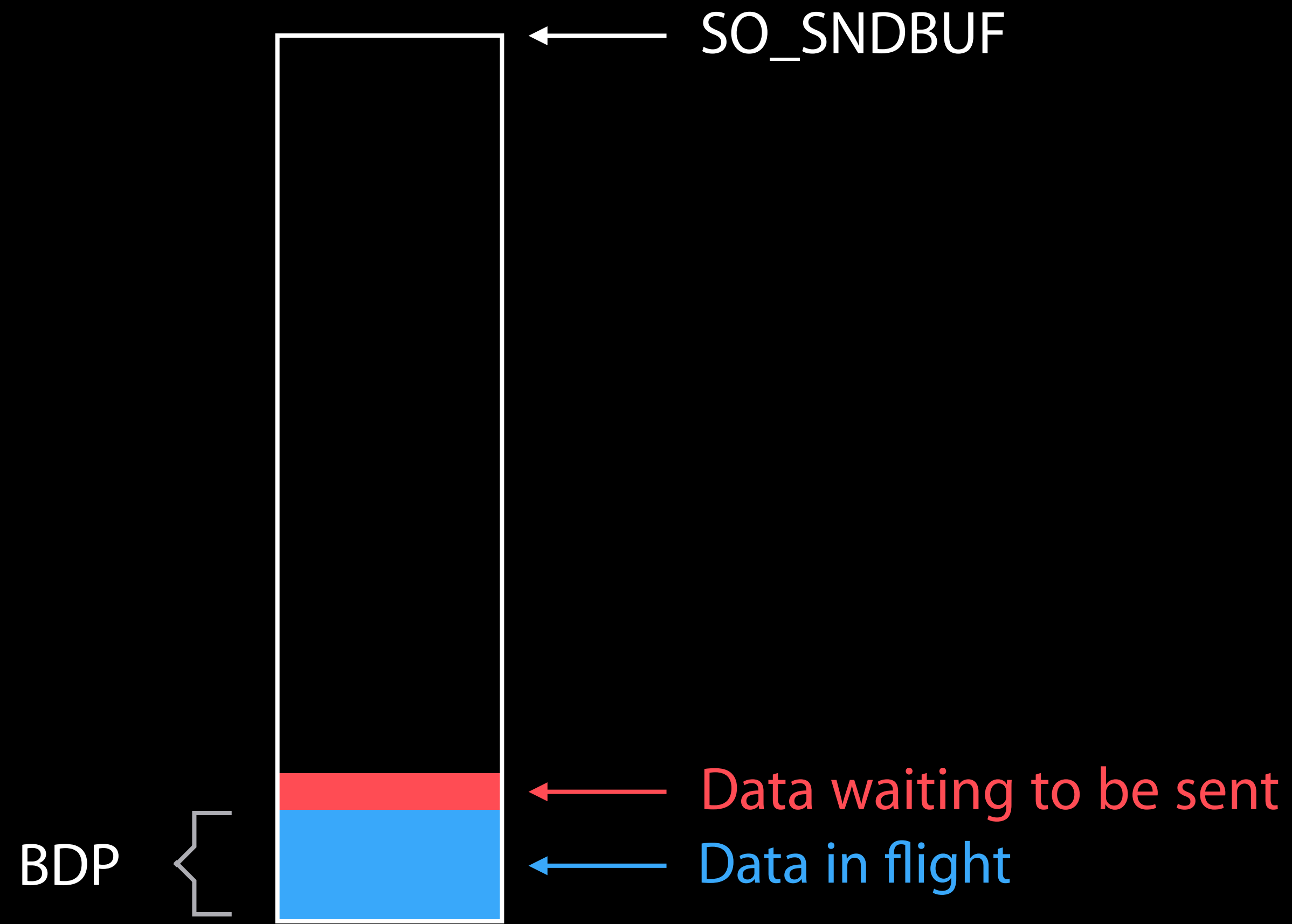## TCP Not-Sent Low-Water Mark

kevent() doesn't signal application to generate a new compressed frame
until TCP is almost ready to need more data

Fixes excessive sender-side buffering for real-time delay-sensitive applications

See Apple WWDC 2015 video "Your App and Next Generation Networks"

Snappy Screen Sharing
Using TCP_NOTSENT_LOWAT

SO_SNDBUF

Data waiting to be sent

BDP

Data in flight

TCP_NOTSENT_LOWAT problems
Low-Water Mark specified in bytes

16 kilobytes (about ten Ethernet frames) works pretty well, but…

- Can be too much on low-rate networks (e.g., 250 kb/s and less)

- Can be too little on high-rate networks (e.g., Gb/s and above)

- Would be better if specified in time (milliseconds, or microseconds) indicating how much notice the application needs to generate next chunk of data

TCP_NOTSENT_LOWAT problems
Inconsistent across different platforms

On Mac OS and iOS, socket option determines low-water mark

- When unsent backlog falls below low-water mark,
  application is *signaled* (e.g., via kqueue) to generate more data

- Application can then atomically write as much as makes sense
  for that application, up to SO_SNDBUF

On Linux, socket option determines high-water mark

- Application is *prevented* from writing more than high-water mark

- Can severely reduce throughput if TCP_NOTSENT_LOWAT set to 16 kB

TCP_REPLENISH_TIME
Opportunity to fix this

New mechanism specified in terms of how much *time* an application needs
to generate its next chunk of real-time delay-sensitive data
Make it work the same for all transport protocols, on all platforms

- TCP, QUIC, etc.
- Linux, FreeBSD, Windows, MacOS, iOS, etc.

Side Meeting, 19:00-20:00 Thursday 7th November, Wicklow Meeting Room 4

- If interested, email Stuart Cheshire <cheshire@apple.com>
  with TCP_REPLENISH_TIME in subject line
  by noon on Thursday 7th November

May look to form IETF Working Group if people feel that is appropriate next step