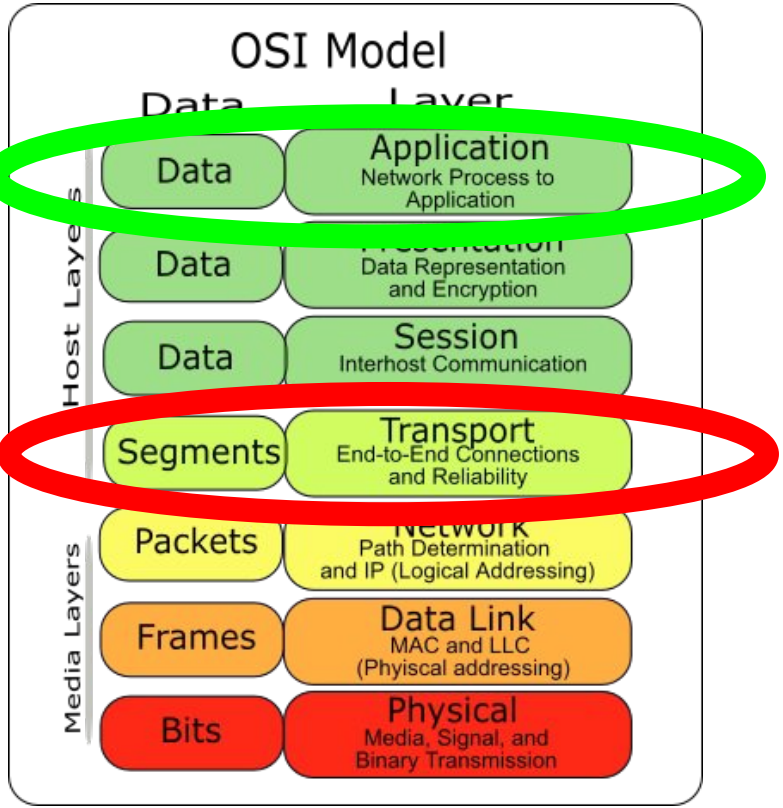
This is a simple, 5-minute presentation at the Recurse Center in NYC during my time there. All my writing and thoughts can be found by going to www.bjshin.com. Thanks!

The Selfish Protocol
Aka BBR

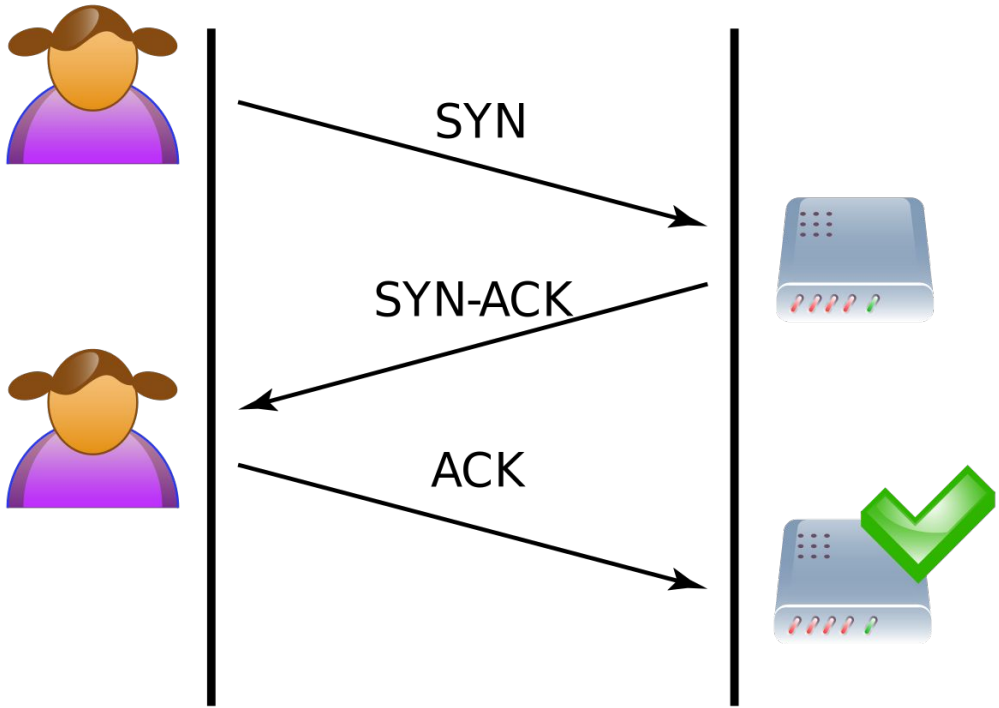
Background



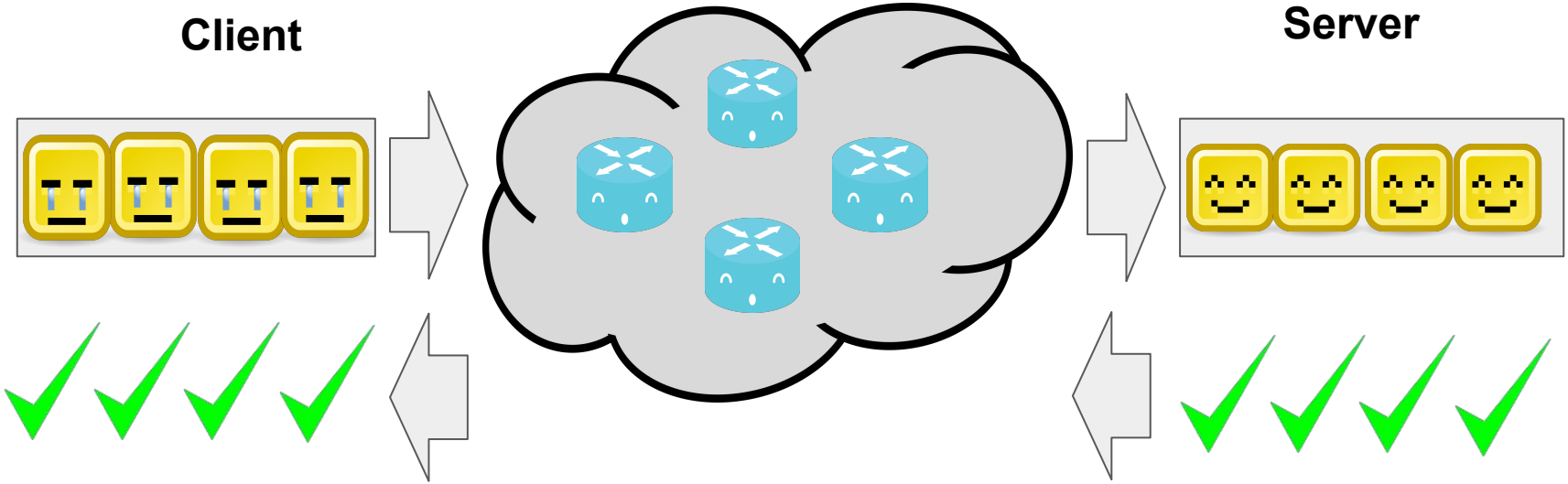
https://commons.wikimedia.org/wiki/File:Osi-model-7-layers.png

Transmission Control Protocol

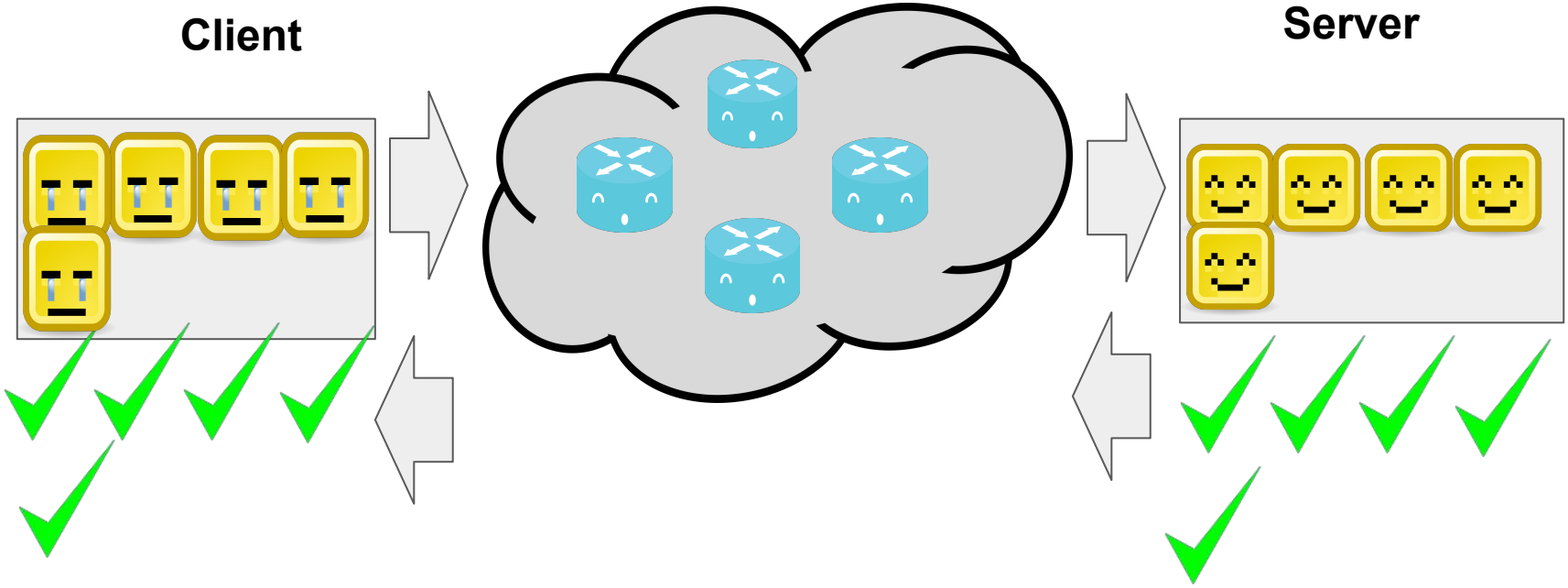
- Reliable stream
- Hold connections
- Guaranteed delivery mechanism
- Establish connection with 3 way handshake
- Afterwards start pushing data in segments.



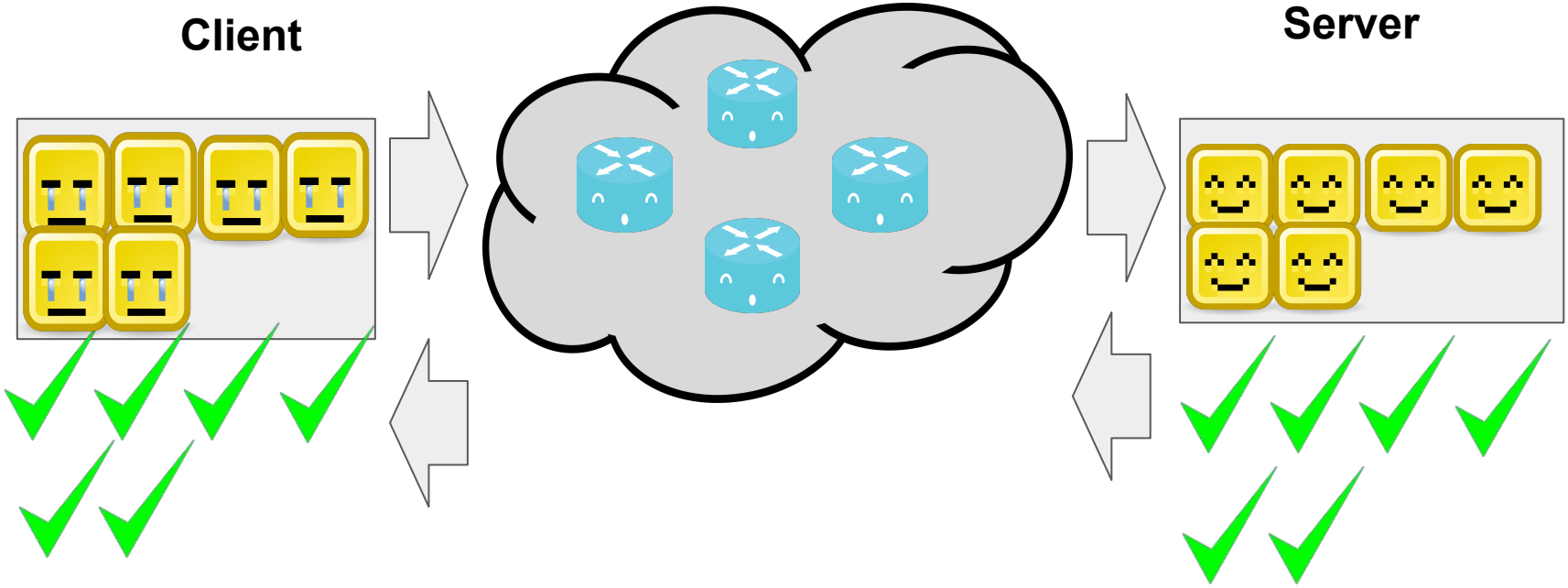
Transmission Congestion Control



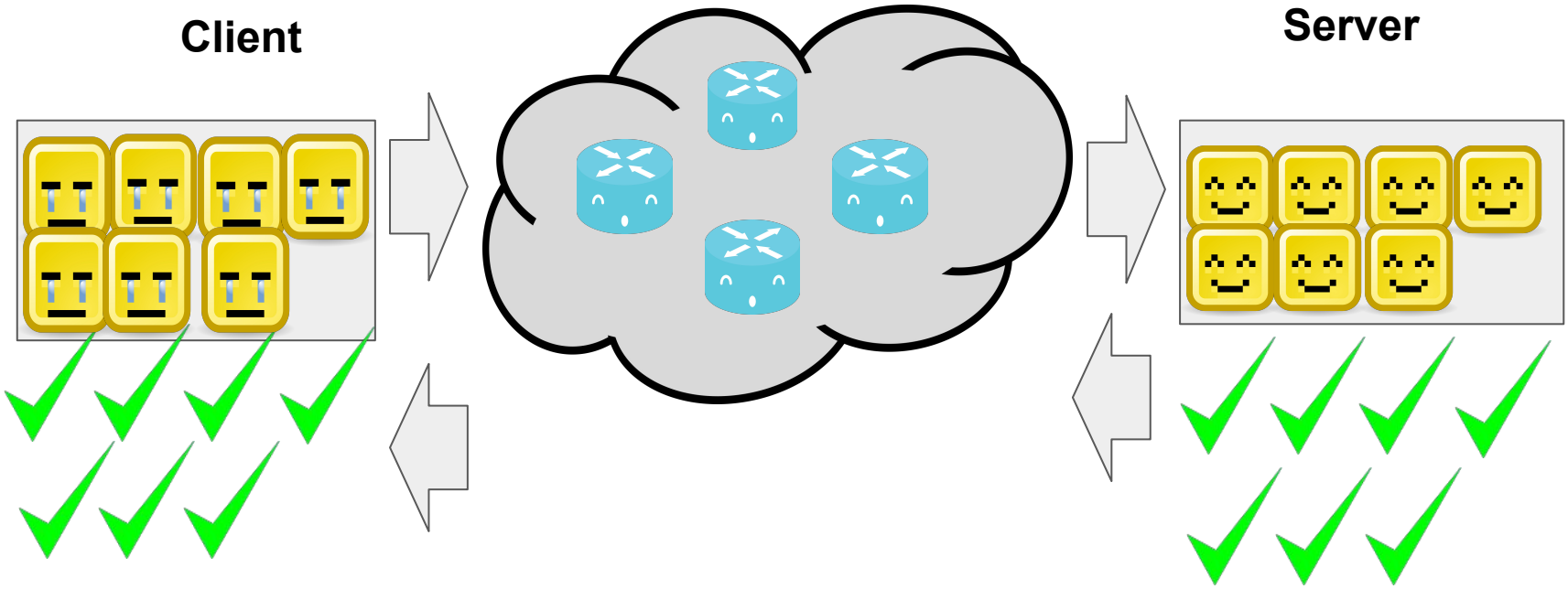
Transmission Congestion Control



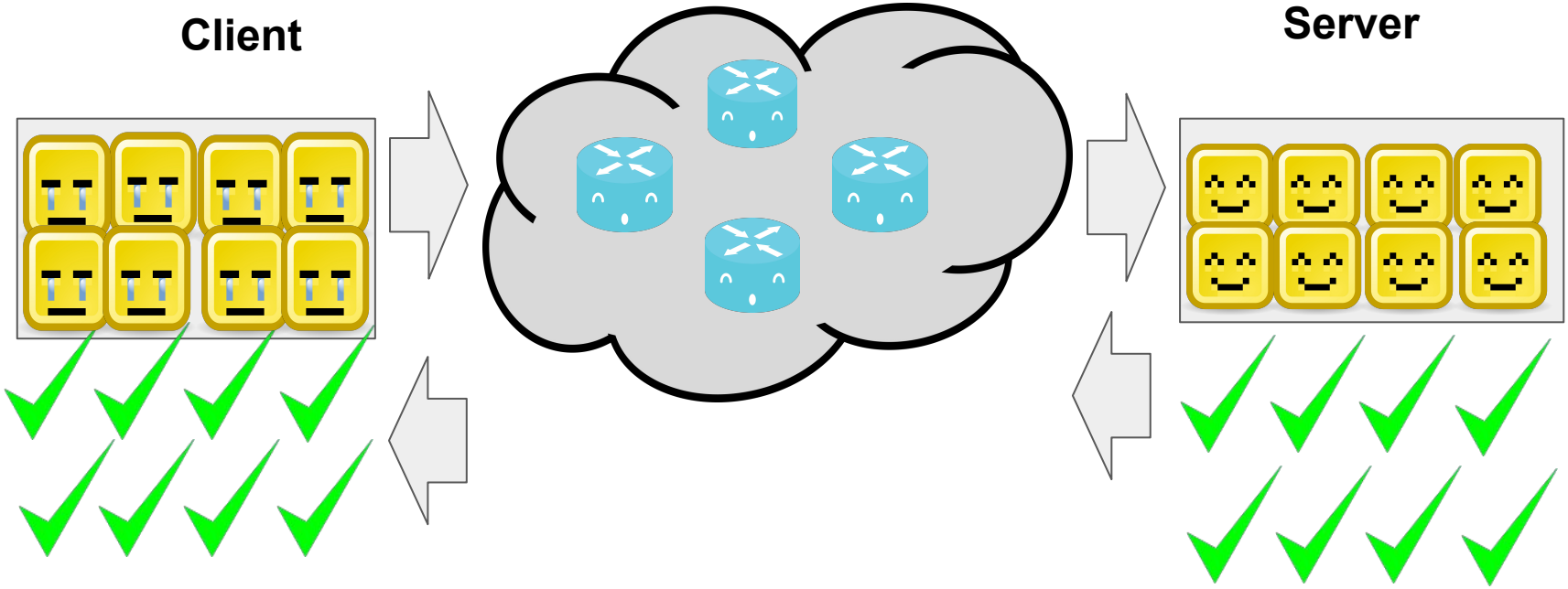
Transmission Congestion Control



Transmission Congestion Control

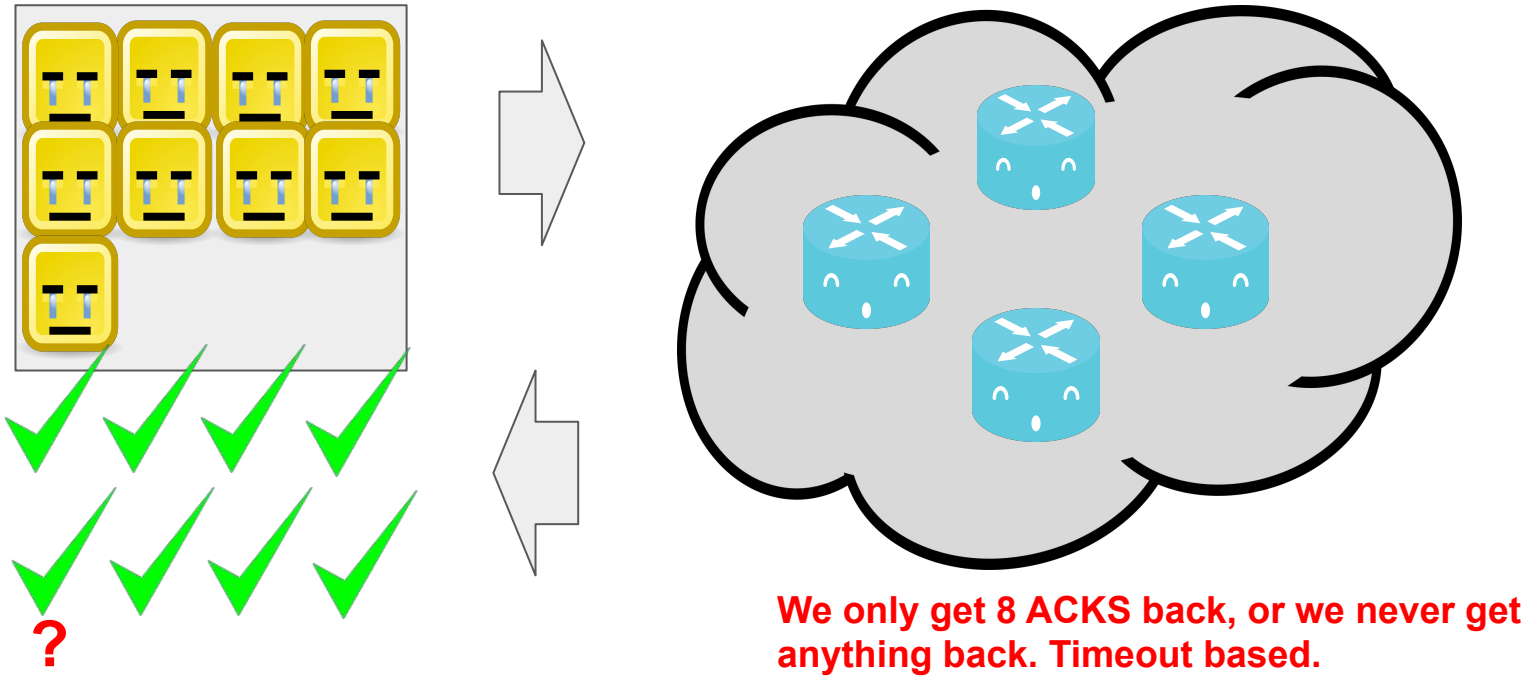


Transmission Congestion Control



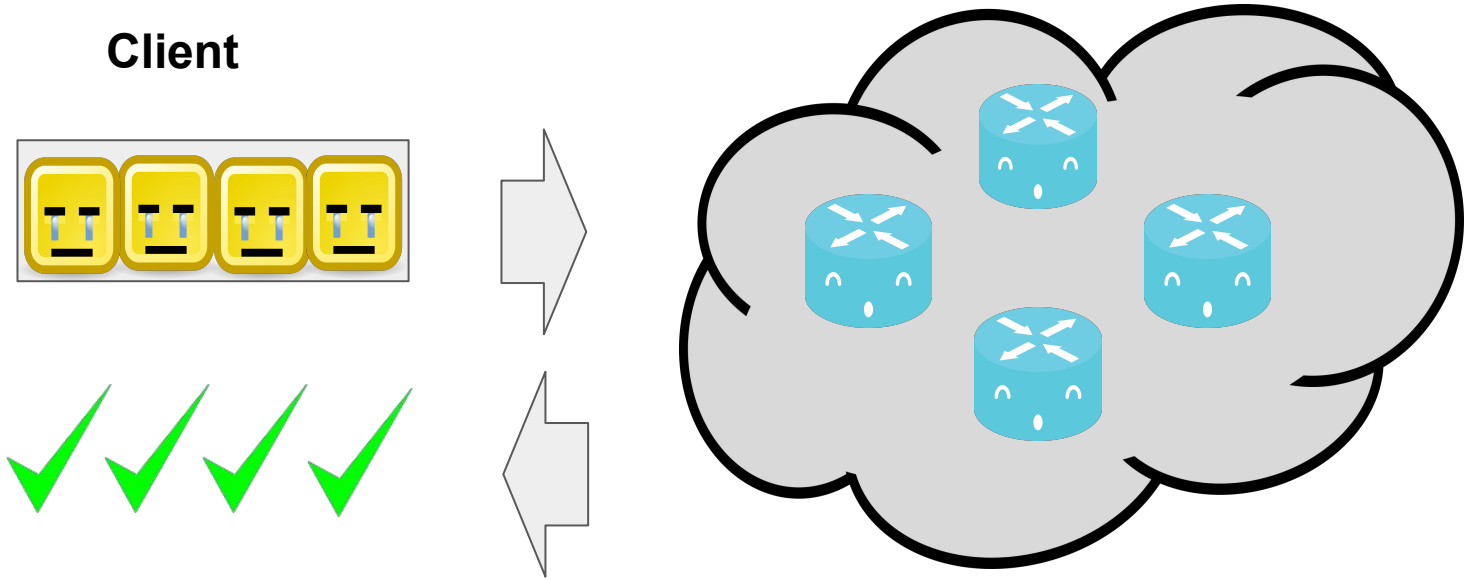
Transmission Congestion Control

Client

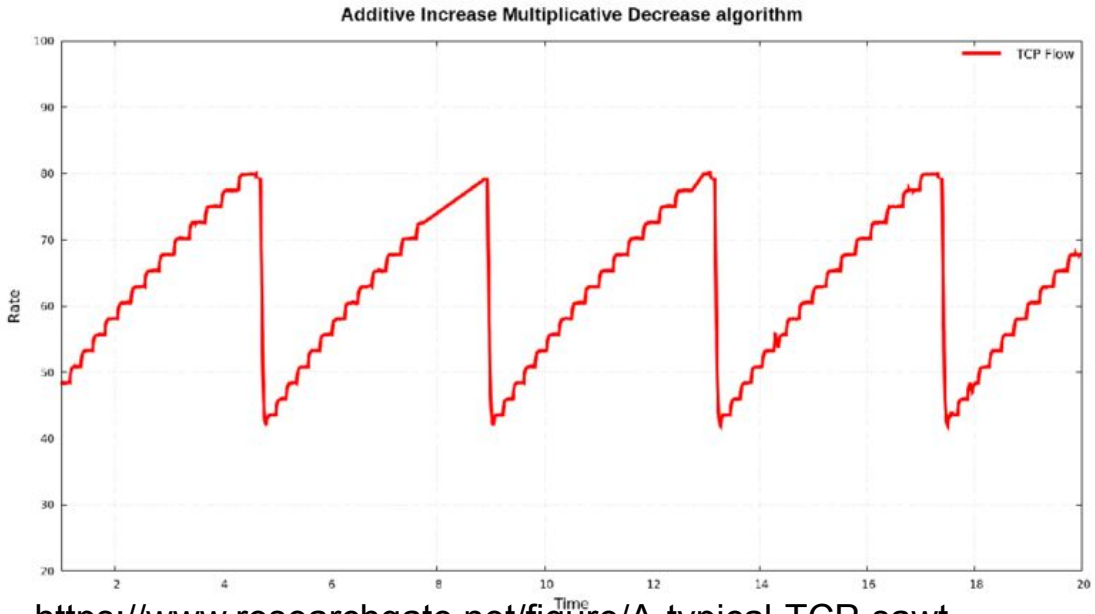


We only get 8 ACKS back, or we never get anything back. Timeout based.

Transmission Congestion Control

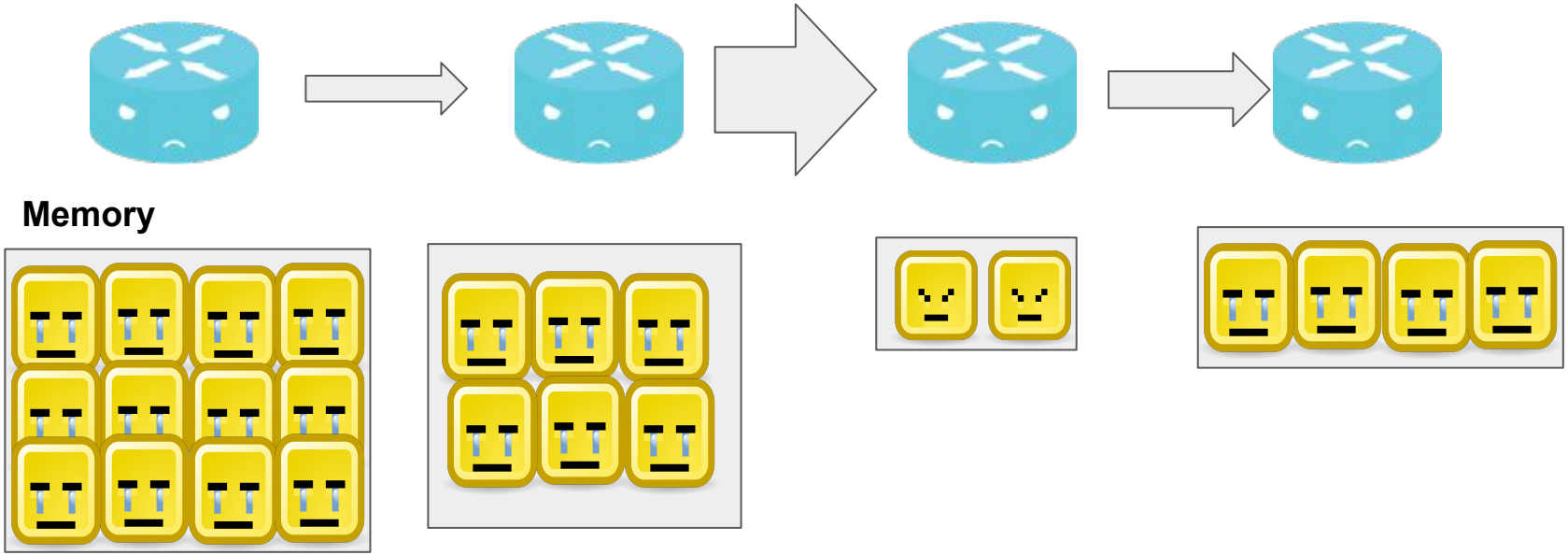


TCP Sawtooth, aka Additive Increase Multiplicative Decrease(AIMD)



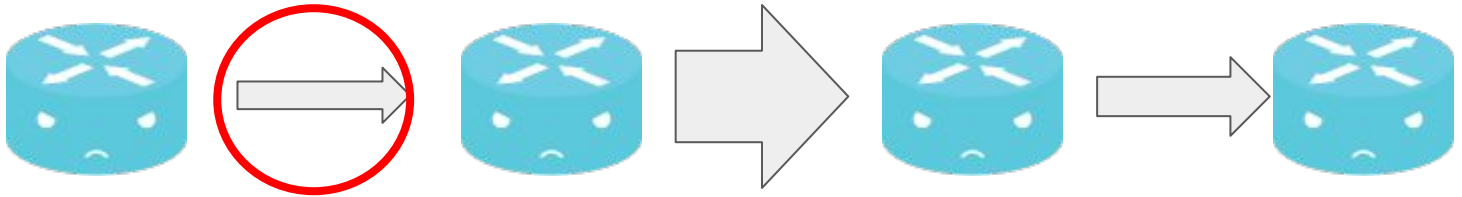
https://www.researchgate.net/figure/A-typical-TCP-sawtooth-behaviour_fig1_313851520

Bufferbloat Problem

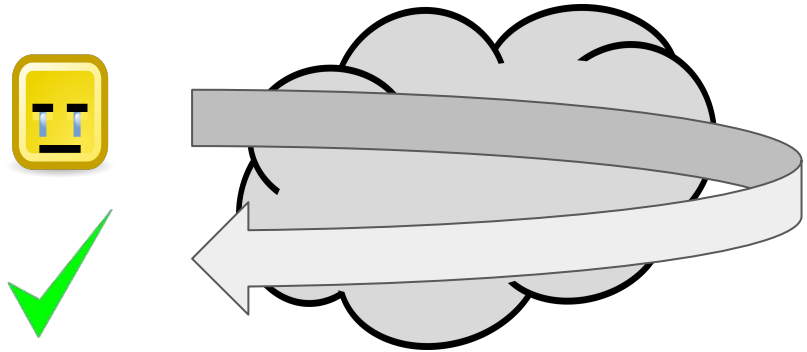


BBR (Bottleneck Bandwidth and Round Trip Time)

- **Bottleneck Bandwidth(BtlBw)**



- **Round Trip Time(RTT)**



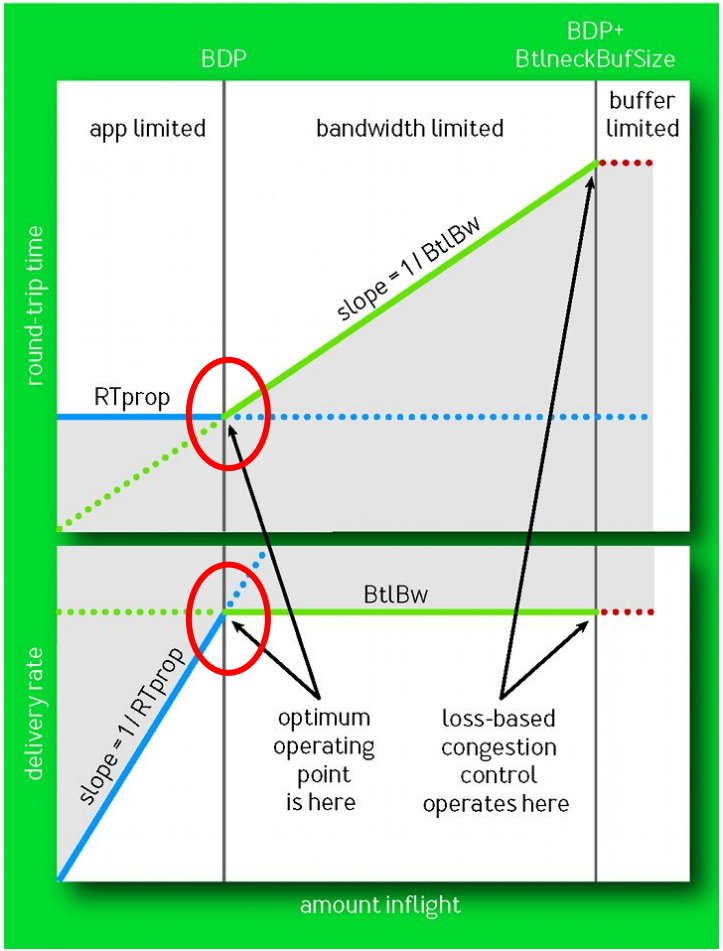
TCP BBR

- Bandwidth Delay Product = Bottleneck Bandwidth * Round Trip Time.
- This is an estimate of how much data you can push at once.

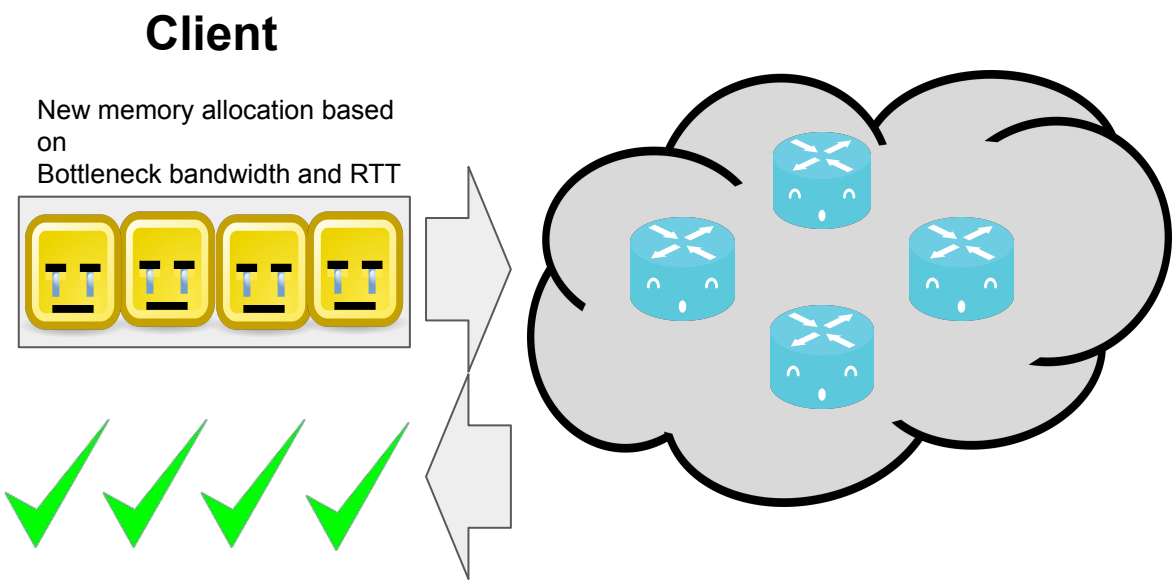
Optimal Operating Point

- Estimate of RTT: Time how long it takes to get a checkmark back
- Estimate of Bottleneck bandwidth: *deliveryRate = Δdelivered/Δt*.

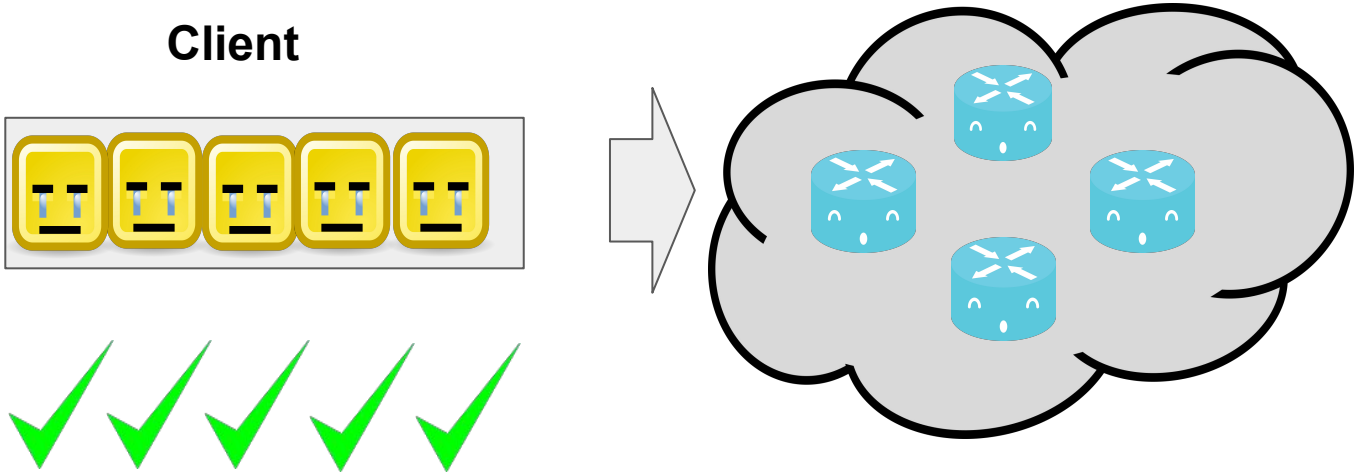
FIGURE 1: **DELIVERY RATE AND ROUND-TRIP TIME VS. INFLIGHT**



BBR - Probing


Client
New memory allocation based on
Bottleneck bandwidth and RTT

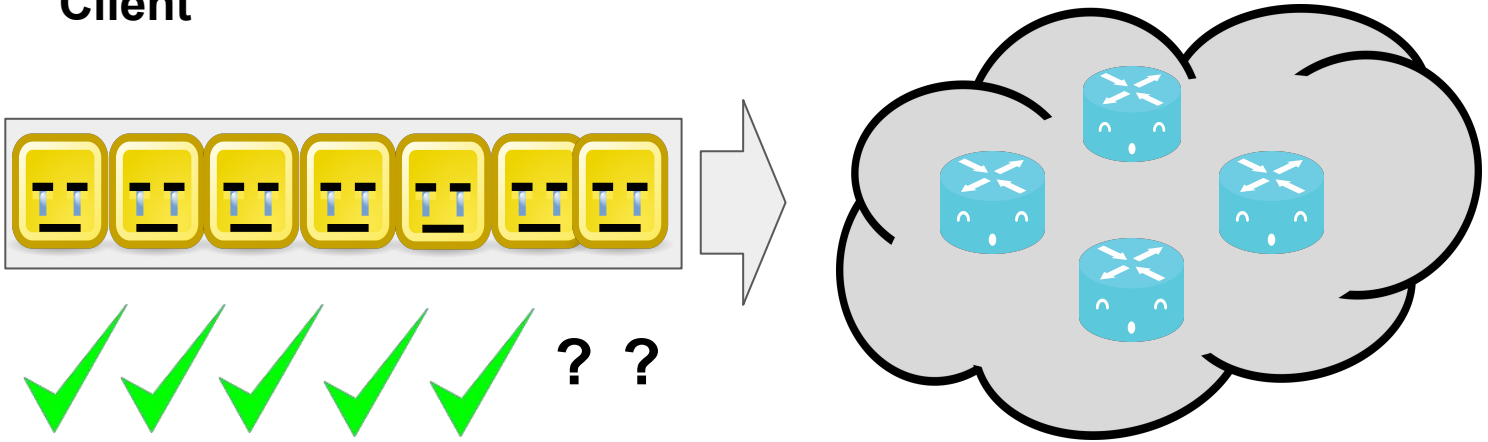
BBR - Increase window by gain factor (i.e. 1.25)



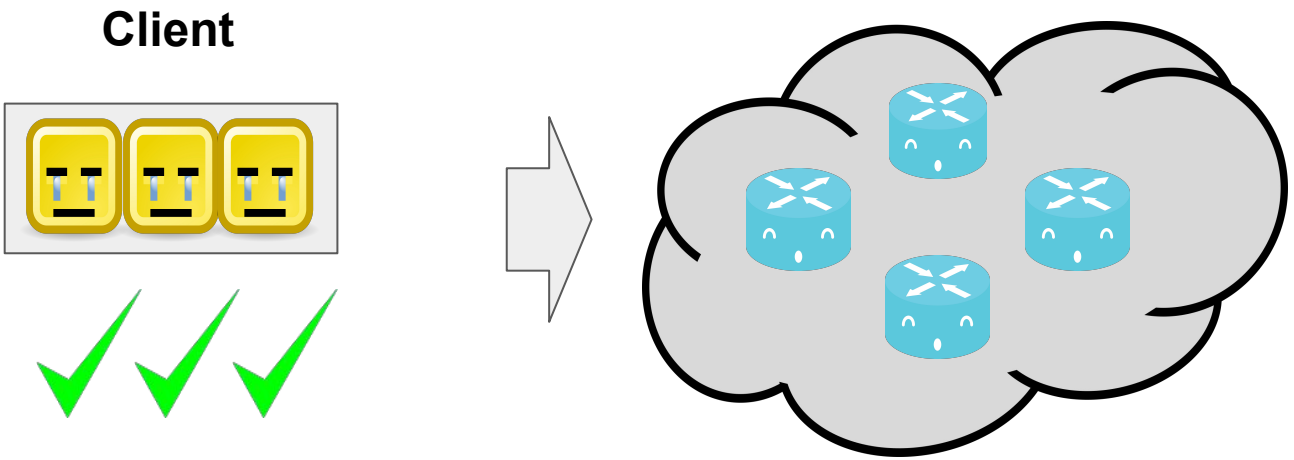
Continuously update BtlBw, and RTT, allocate send memory, cruise at this estimate for a bit.

BBR - Excessive Increase

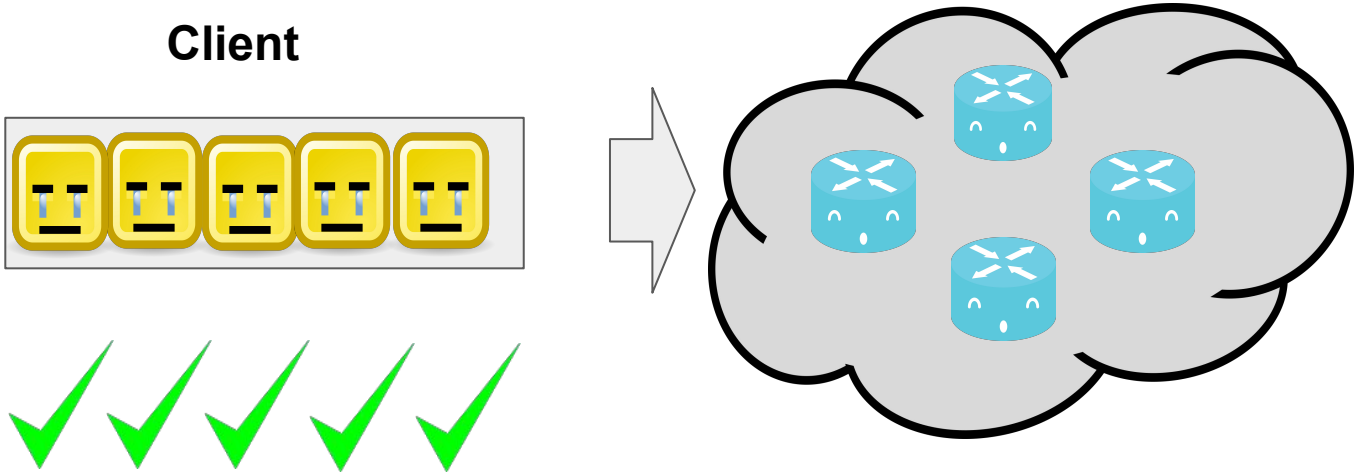
Client

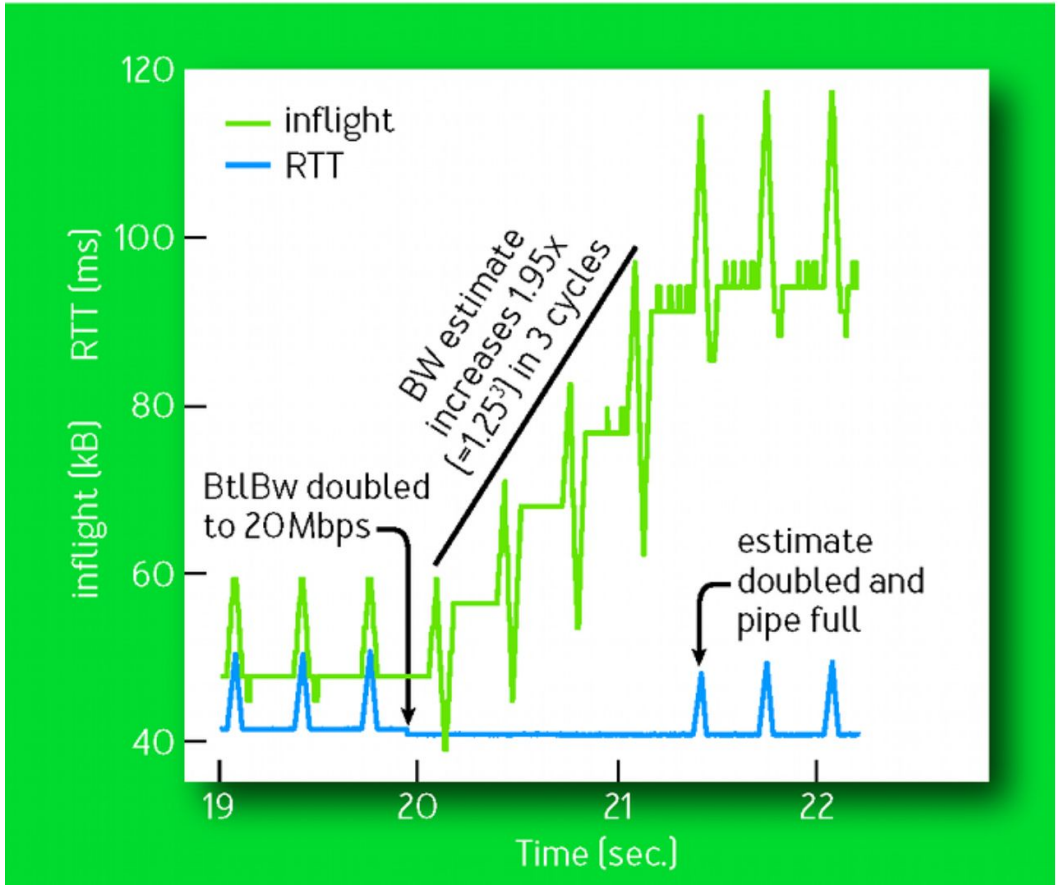


BBR - Decrease send size to relieve stress on previous send



BBR - Bring it back to the original state (i.e. 1.25)




120
100
80
60
40
inflight (kB)
RTT (ms)
inflight
RTT
BW estimate
increases 1.95x
(=1.25³) in 3 cycles
BtlBw doubled
to 20Mbps
estimate
doubled and
pipe full
19
20
21
22
Time (sec.)

OK - So should you BBR?

1. **You like the numbers below.**

Total Processing Time(s)		
	CUBIC	**BBR**
Google	0.05520932847	0.0574670073
Baidu(China)	0.4811333796	0.3861766131
Reddit	0.006109985401	0.006061211679
Naver	0.4096814706	0.2559053796
Yandex	0.2846899781	0.2837229781

2. You are selfish.

3. You are not on Windows or Mac.

References

- **Dhananjay**
- https://www.cyberciti.biz/cloud-computing/increase-your-linux-server-internet-speed-with-tcp-bbr-congestion-control/
- https://www.tecmint.com/test-website-loading-speed-in-linux-terminal/
- https://www.techrepublic.com/article/how-to-enable-tcp-bbr-to-improve-network-speed-on-linux/
- https://news.ycombinator.com/item?id=17063582
- https://thesquareplanet.com/blog/how-the-internet-works/