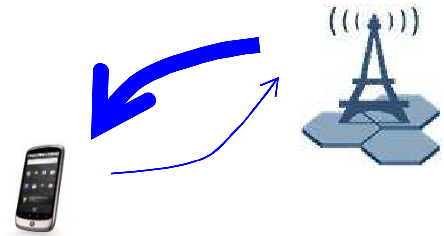
# One Way Latency Considerations for MPTCP

draft-song-mtcp-owl-03

F. Song, H. Zhang, H. Chan, and A. Wei

# Asymmetric data rate in cellular networks

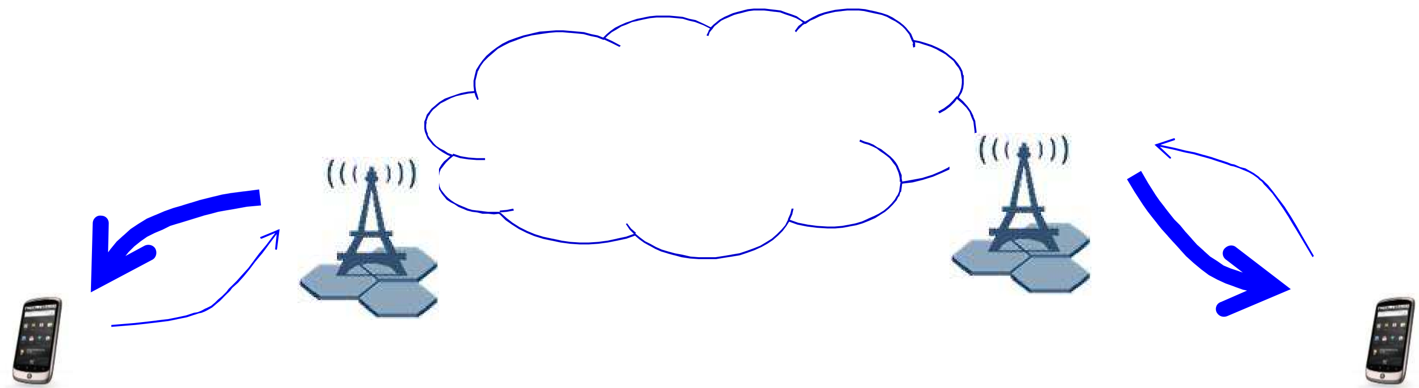
- Cellular networks provide lower speed in uplink than downlink
  - The global average 4G download: 16.6Mbps in Nov 2017 [1]. Yet the upload speed is much lower
  - E.g. LTE: 5-12Mbps download, 2-5Mbps upload [2]



- [1] https://opensignal.com/reports/2017/11/state-of-lte
- [2] https://www.verizonwireless.com/articles/4g-lte-speeds-vs-your-home-network/

# Asymmetric data rate in cellular networks

- Asymmetric for client side
- May also be asymmetric at server side



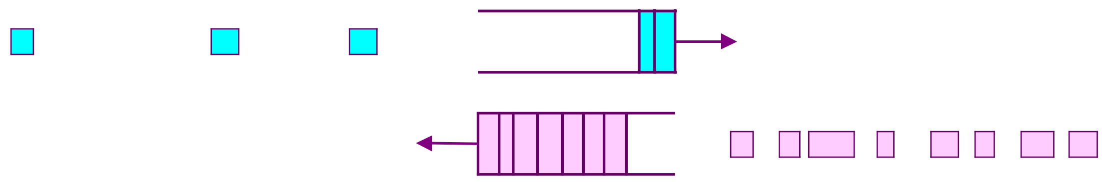
# Cellular radio access latency
# 3GPP TS36.881

| Uplink | Time (ms) |
|---|---|
| Wait for PUCCH (10ms/1ms SR) | 5/0.5 |
| UE sends scheduling request | 1 |
| eNB decodes request and generates scheduling grant | 3 |
| Transmission of scheduling grant | 1 |
| UE processing delay | 3 |
| | |
| Transmit UL data | 1 |
| Data decoding in eNB | 3 |
| Total delay | 17/12.5 |

| Downlink | Time (ms) |
|---|---|
| | |
| | |
| | |
| | |
| Process incoming data | 3 |
| TTI alignment | 0.5 |
| Transmit DL data | 1 |
| Data decoding in UE | 3 |
| Total delay | 7.5 |

# Cannot assume queue lengths are symmetric

- Long queue in one direction does not imply same long queue in reverse direction



- Increased RTT caused by near-congestion condition cannot conclude that both directions are near congestion

# Asymmetric ISP data rates

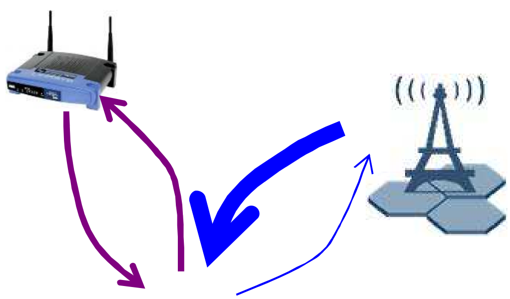
- ISP's typically also provide higher data rate for download than upload
- Example: FCC (in USA) definition of broadband [3]: minimum data rates of:

| (2015) | (2015, mobile) | (before 2015) |
| --- | --- | --- |
| 25Mbps ↓ ↑ 3Mbps | 10Mbps ↓ ↑ 1Mbps | 4Mbps ↓ ↑ 1Mbps |

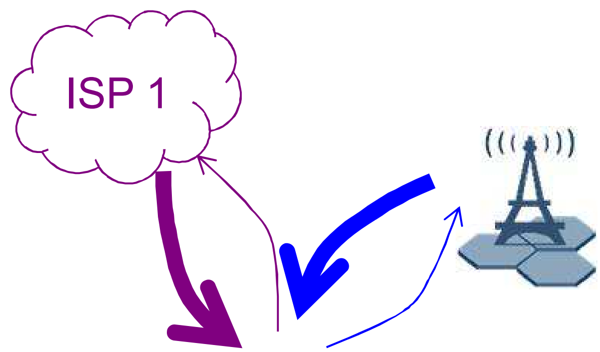
- [3] source: https://fas.org/sgp/crs/misc/R45039.pdf

# Path selection
# compare which one has smaller one-way delay

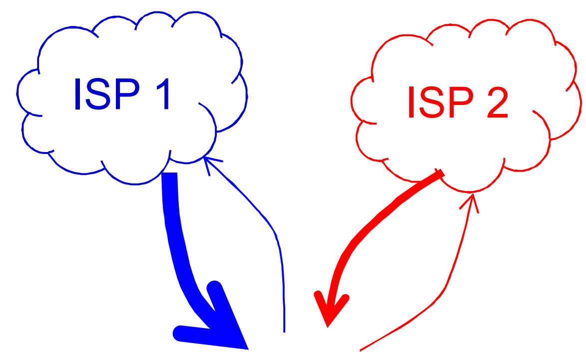
- Choosing among more than one paths:



**2 paths of wireless access**

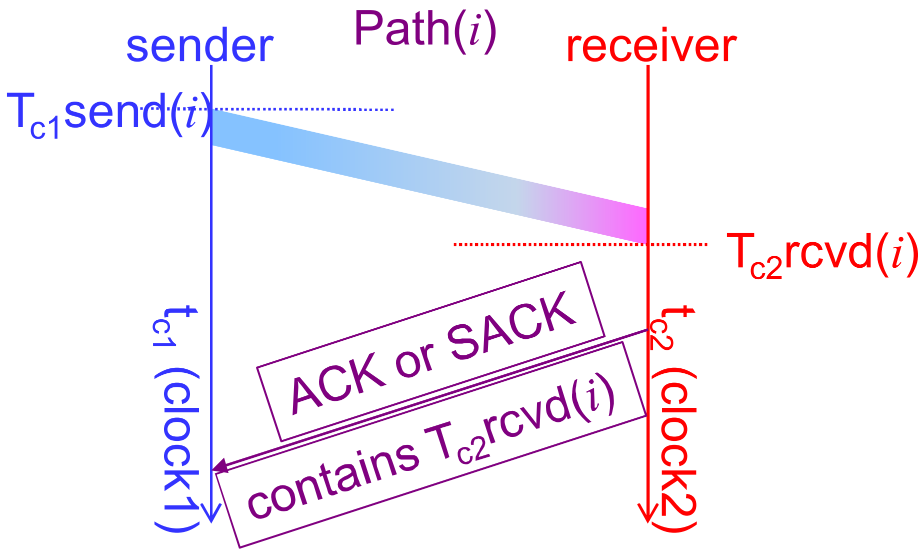


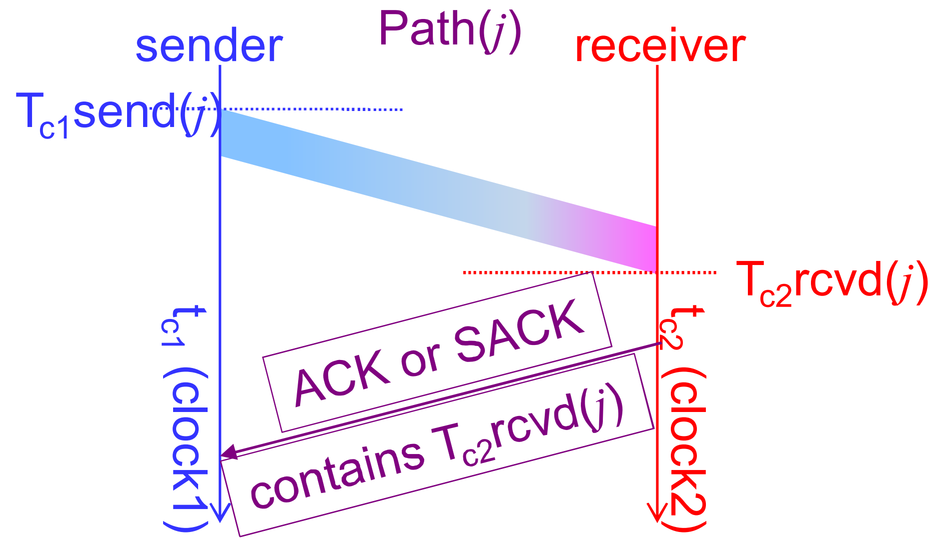

**1 path with wireless access and 1 with wired connection**




**Dual homed to 2 ISPs**

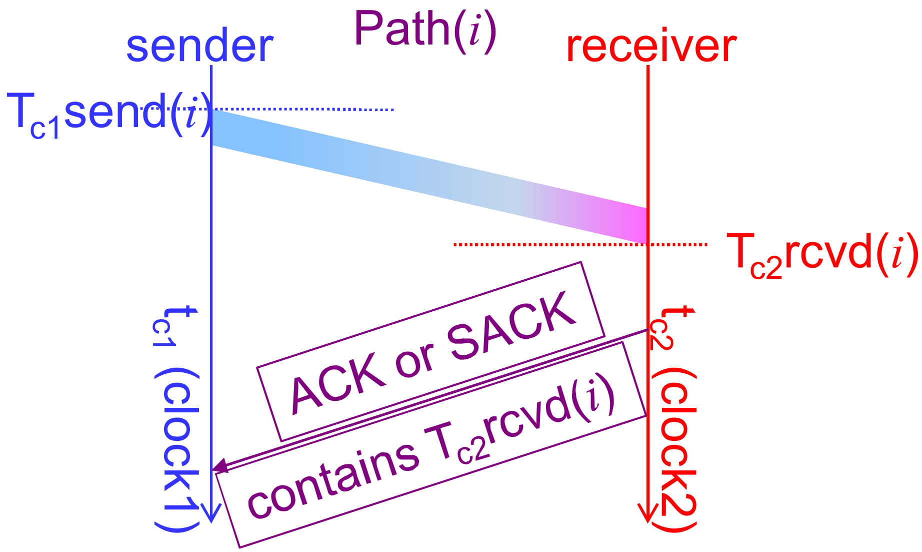
# Exact one-way delay with clock synchronization ($t_{c1}=t_{c2}$)

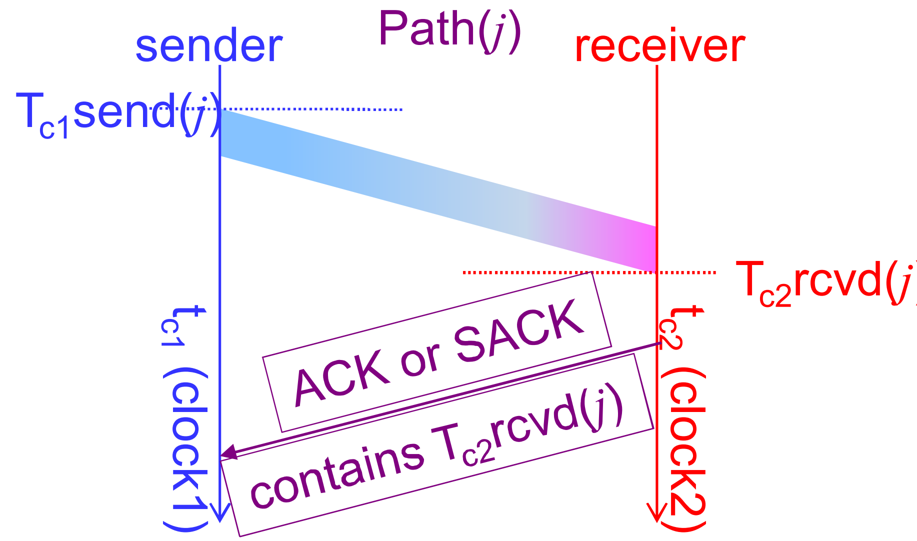



- $OWL(i) = T_{c2}rcvd(i) - T_{c1}send(i)$
- $OWL(j) = T_{c2}rcvd(j) - T_{c1}send(j)$
- TS in reply $\neq$ Trcvd especially with SACK
- $\Delta OWL(i,j) = OWL(j) - OWL(i)$

# Relative one-way delay does not need synchronization




- $\mathrm{OWL}(i) = T_{c2}\mathrm{rcvd}(i) - T_{c1}\mathrm{send}(i) +$ clock synchronization error
- $\mathrm{OWL}(j) = T_{c2}\mathrm{rcvd}(j) - T_{c1}\mathrm{send}(j) +$ clock synchronization error
- $\Delta\mathrm{OWL}(i,j) = \mathrm{OWL}(j) - \mathrm{OWL}(i)$ (same result)

# One way latency measurement

- Capability negotiation – to be defined
- Sender

# Many approaches for OWL

- Client remembers Tsend
  - client stores Tsend when sending data to server
  - server ACK/SACK with Trcvd
  - client calculates OWL, and compares OWL among different paths
- Client sends Timestamp
  - client sends Tsend
  - server echoes Tsend and includes Trcvd
  - client calculates OWL, and compares OWL among different paths
- Server calculates OWL
  - client sends timestamp when sending data
  - server notes Trcvd, calculates OWL and sends OWL back to client
  - client calculates OWL, and compares OWL among different paths

# Download

- Client calculates OWL
  - server sends timestamp (Tsend) information
  - client checks Trcvd to calculate OWL for each path
  - client compares OWL for different paths

# Questions/Comments please