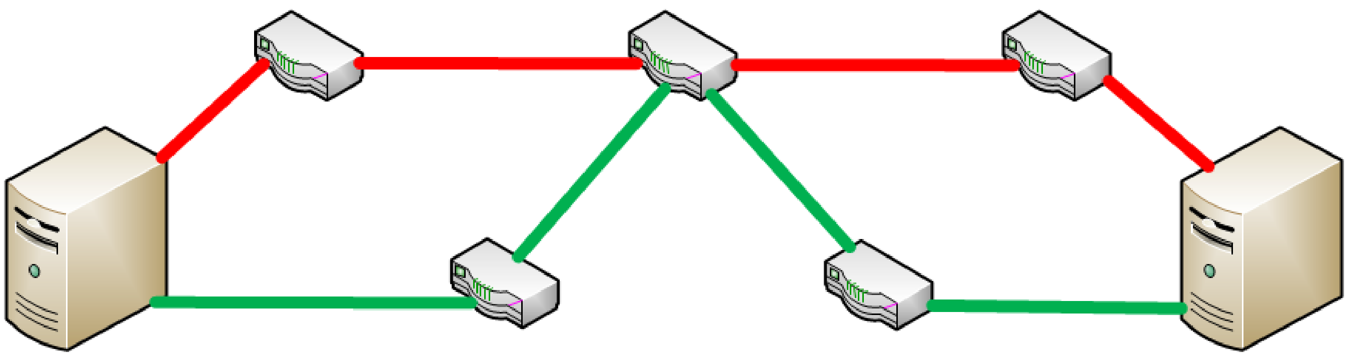
# MultiPath TCP: Hands-On

**Gerrie Veerman** - ***UvA***
**Supervisor: Ronald van der Pol** - ***SARA***
05-07-2012

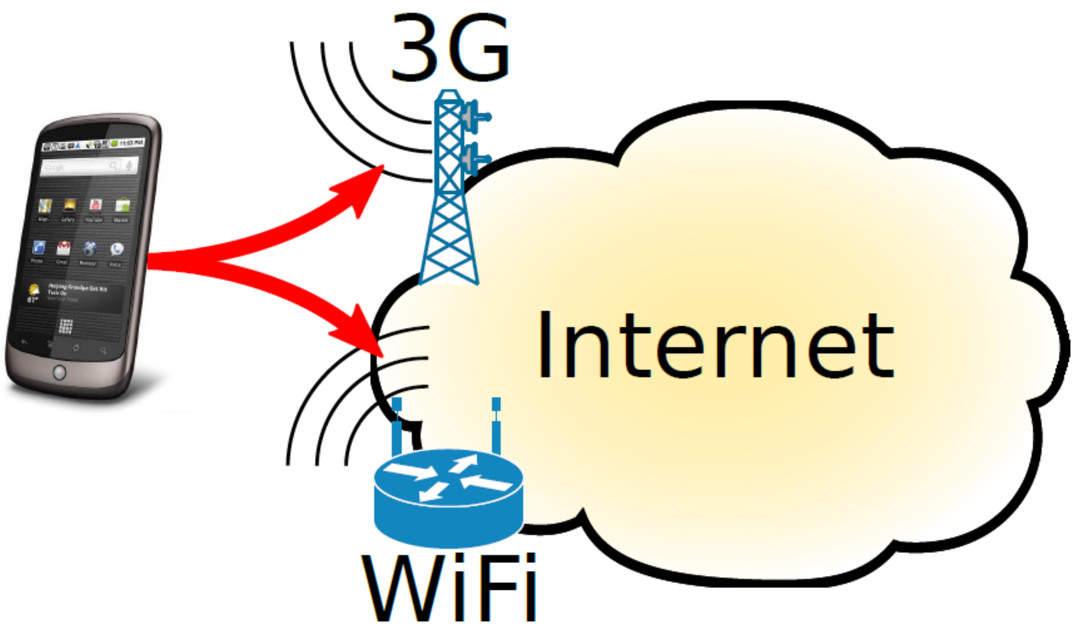
# What is MultiPath TCP?

- The ability to use multiple paths with the same connection.



# Making use of Multi-homing

- One could make use of multiple interfaces simultaneously and roam between 3G and WiFi instantly.



# The Project

**Research Question**:

*Is the current MPTCP implementation a useful technology for e-science data transfers in the GLIF environment?*

**Why are we doing this?**

- Demand for bandwidth keeps increasing
- MPTCP is still relatively new
- Can MPTCP really make efficient use of multiple paths
- How stable is the current implementation
- First hands-on experience for SARA

# History and Present

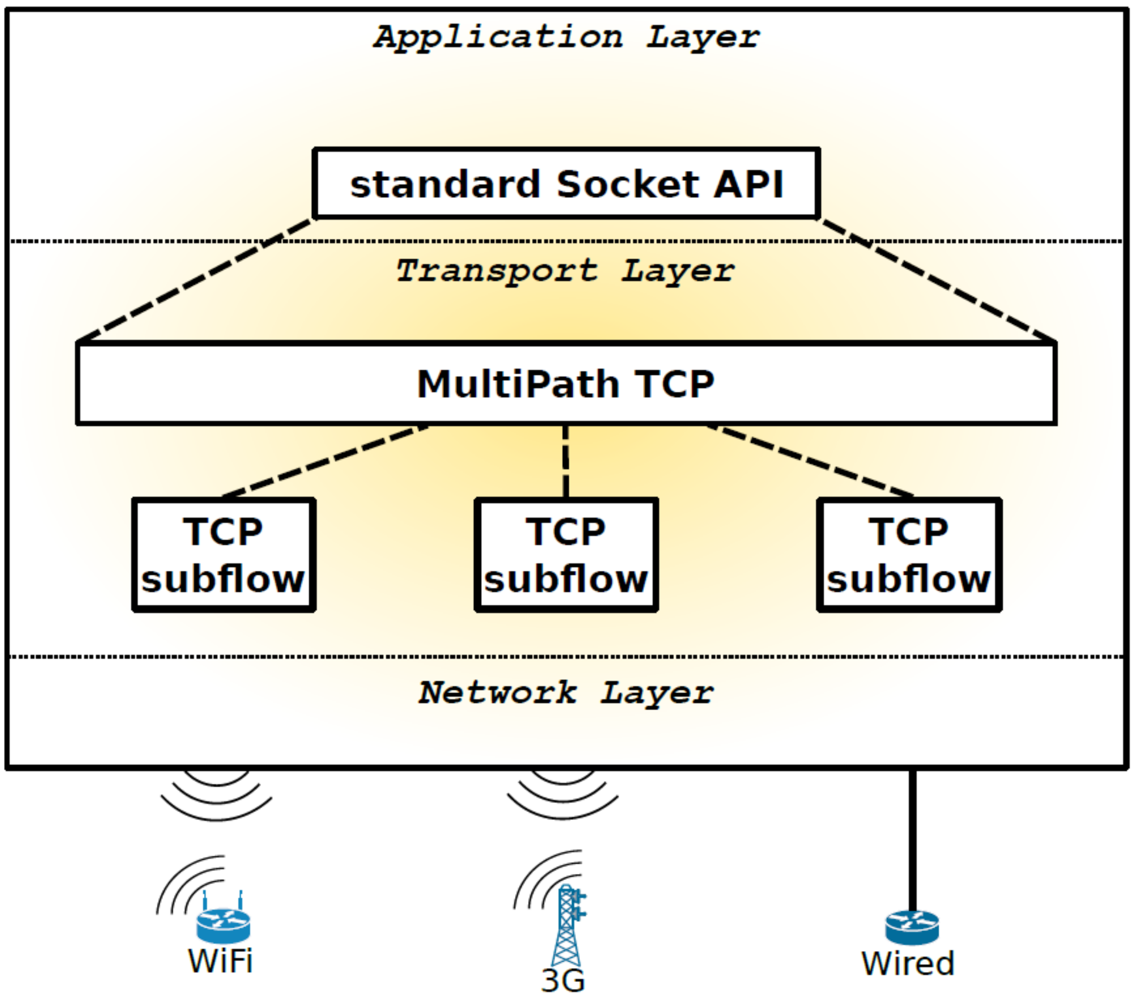
**History:**

- Christian Huitema suggested the idea in 1995
- The idea turned into MPTCP around 2006

**Present:**

- In 2011 the first RFCs appeared
- 1e implementation in the 2.6 Linux kernel in 2011 (higher versions should support it, we used 3.2)
- Currently three RFCs written and four still in draft
- MPTCP is still being developed, discussed and extensively tested

# How does MPTCP work?



# Properties of MPTCP

- MPTCP is actually implemented in TCP option fields
- For middle-boxes MPTCP looks like regular TCP packets
- Applications can use MPTCP as in a regular TCP socket API
- End-hosts need multiple routing tables, one for each path (default gateways)
- One needs higher buffers than with TCP

# Path Management

- Routes and paths are created by the network not the MPTCP protocol
- After a handshake the first initial subflow is created
- MPTCP shares all available IP addresses with each other and tries to create a full-mesh out of them
  - The connections which do not work get dropped
- MPTCP has the ability to add and remove subflows
- Every subflow has its unique subflow ID and keys (SHA-1 is used).

# The Goals of MPTCP

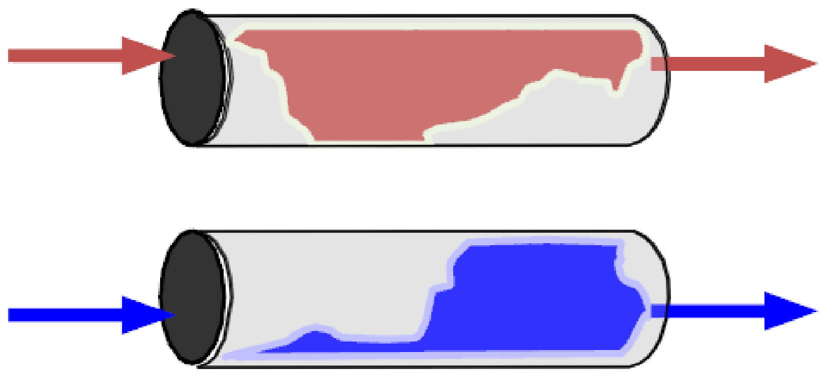
1. **Improve throughput:** Perform at least as well as a single path flow would on the best of the paths available to it.

2. **Do no harm:** multipath flow should not take up more capacity from any of the resources shared by its different paths

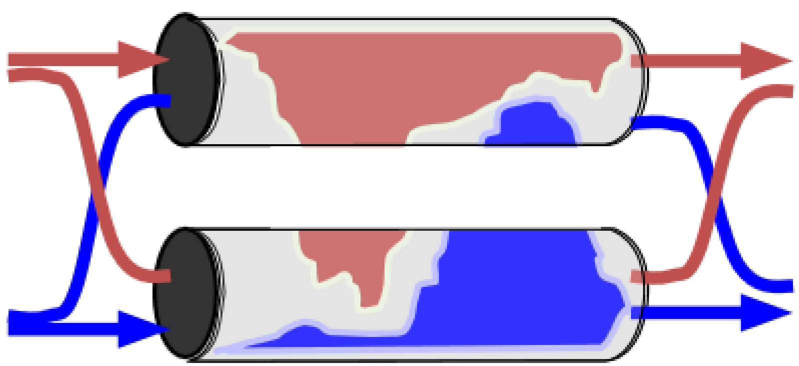
3. **Balance congestion:** A multipath flow should move as much traffic as possible off its most congested paths, subject to meeting the first two goals.

# Congestion

- With TCP:

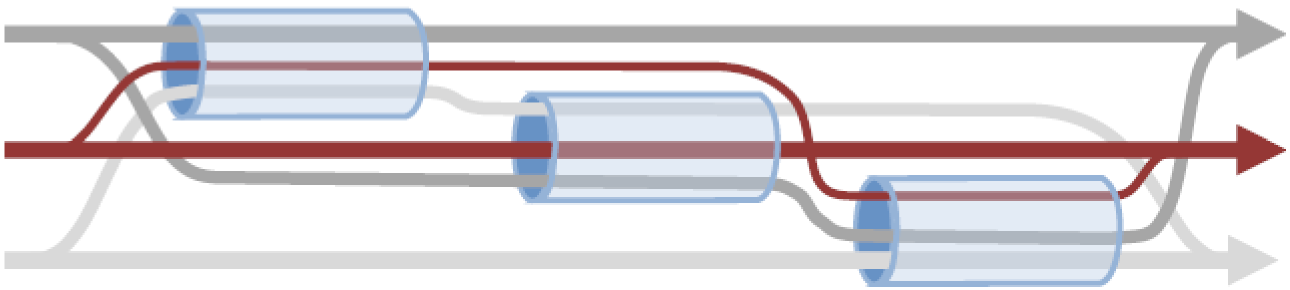


- With MPTCP:



# Congestion Algorithm

Should make sure the most efficient paths are taken and meet the design goals of MPTCP

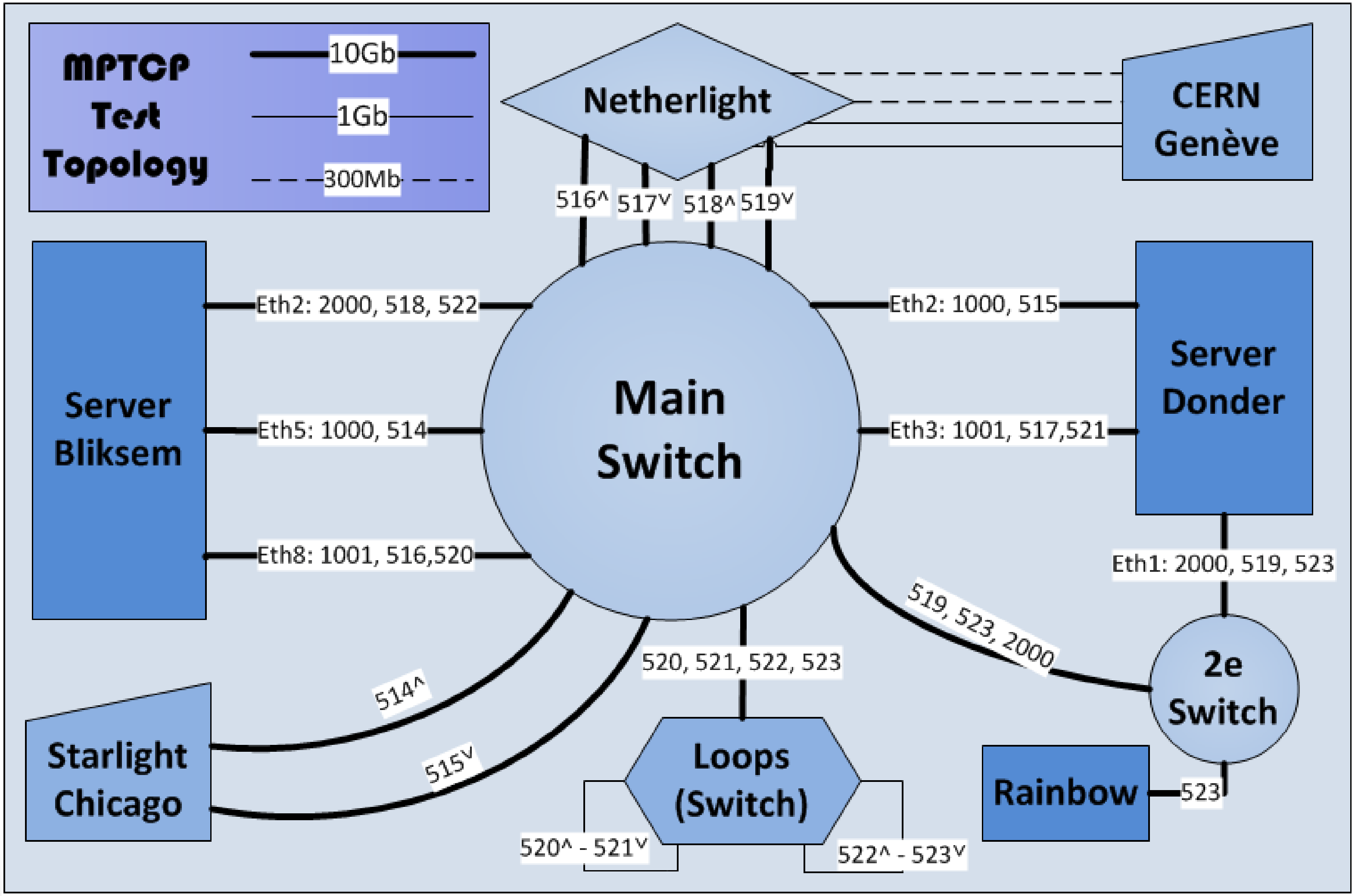


# Questions we had?

- **How is everything configured/addressed/routed?**

- **How well does the current implementation work?**
  - Can it handle a LAN and WAN environment?
  - How robust is the protocol?
  - Can it handle differences in bandwidth?
  - How well does MPTCP handle congestion?

# Created Topology



# Experiments

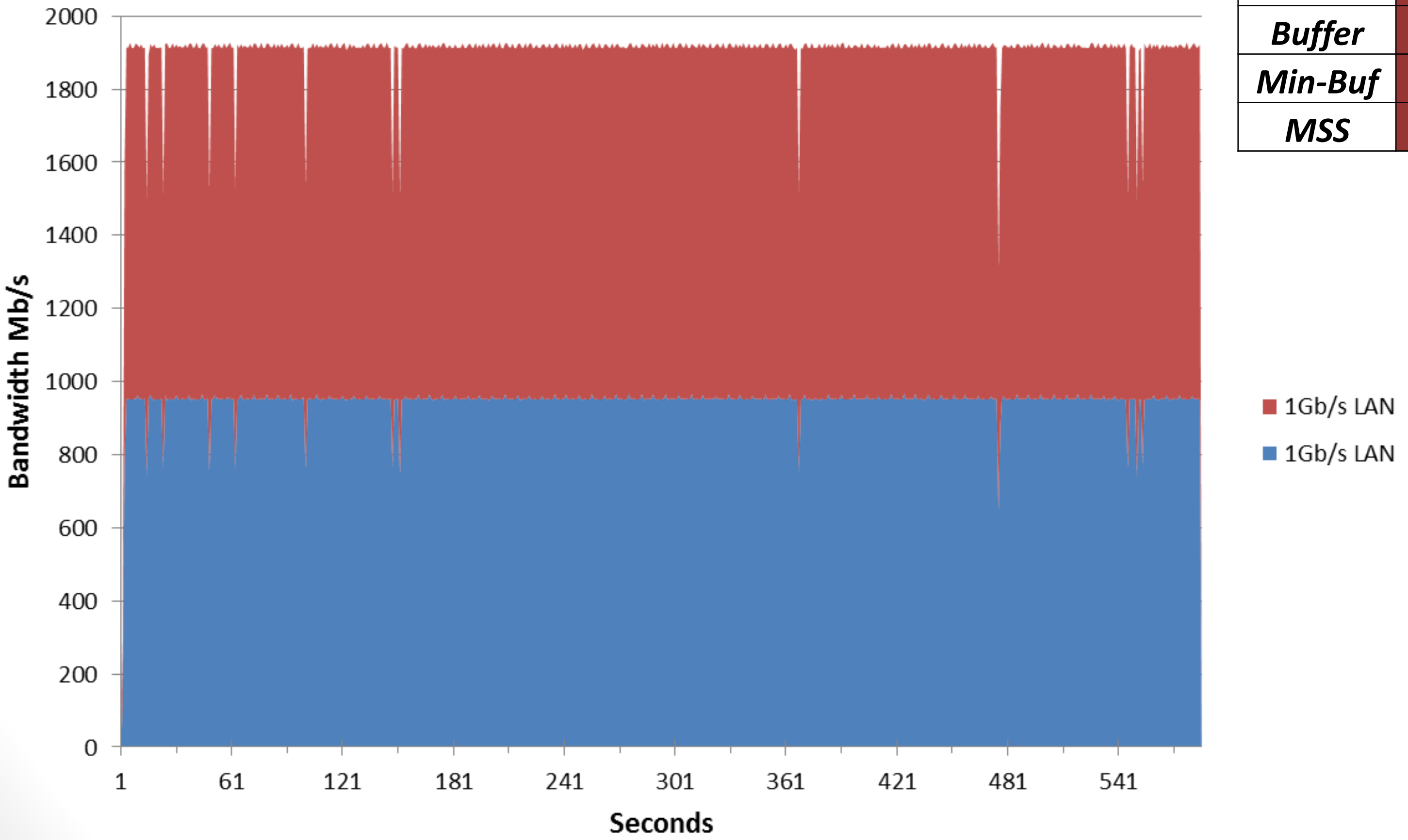
**Experiment Topics:**

- Improved throughput
- Robustness
- Congestion and Fairness
- LAN vs WAN environment

**What we used:**

- Small and large packets *(MSS)*
- For all our tests we used iperf
- Different sizes for socket buffers
- Increased the maximum buffer size for the kernel *(rmem_max, wmem_max, tcp_rmem and tcp_wmem)*.

# LAN: Throughput

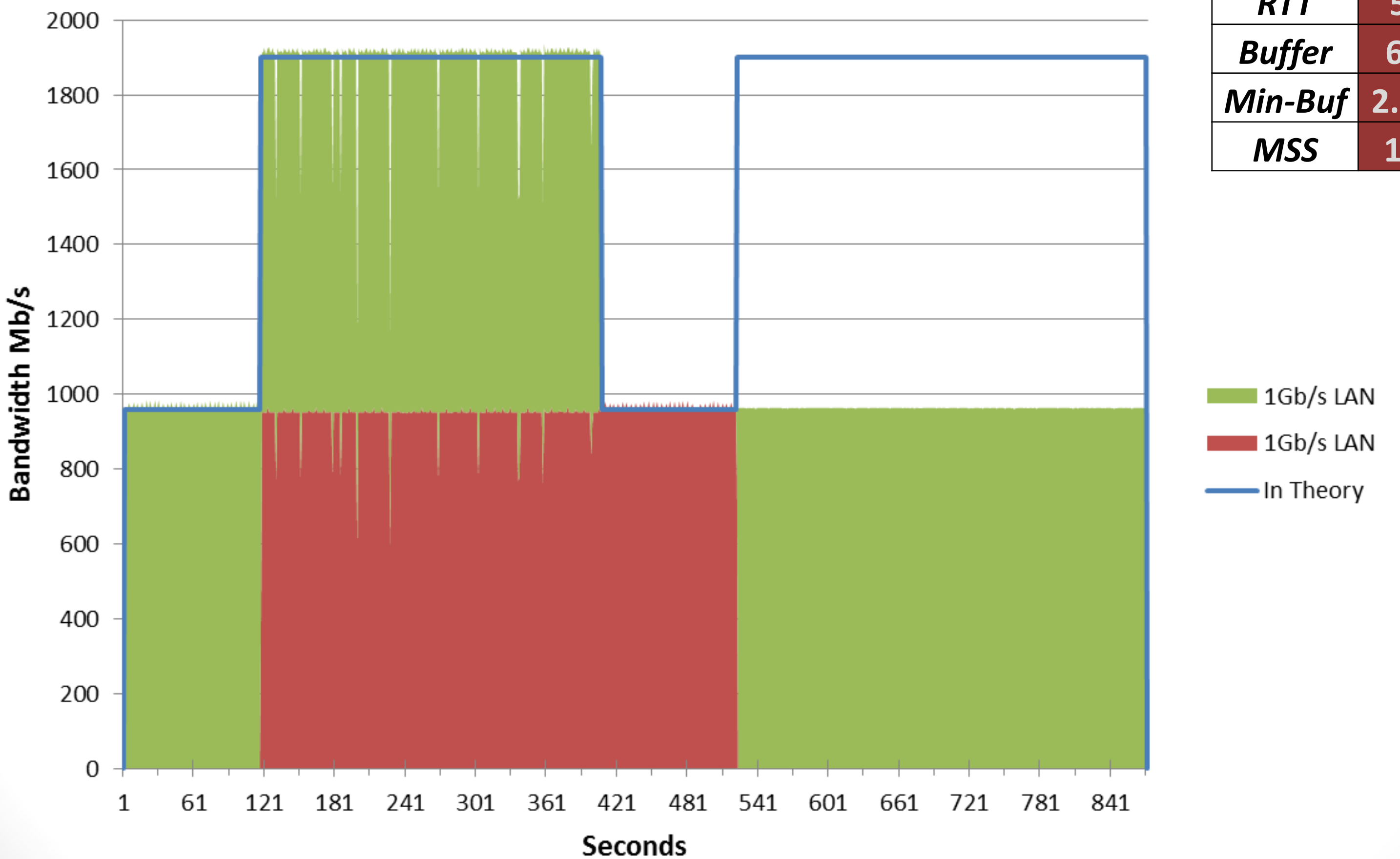



| | LAN | LAN |
|---|---|---|
| ***Speed*** | 1Gb/s | 1Gb/s |
| ***RTT*** | 5ms | 5ms |
| ***Buffer*** | 6MB | 6MB |
| ***Min-Buf*** | 2.5MB | 2.5MB |
| ***MSS*** | 1400 | 1400 |

# LAN: Robustness

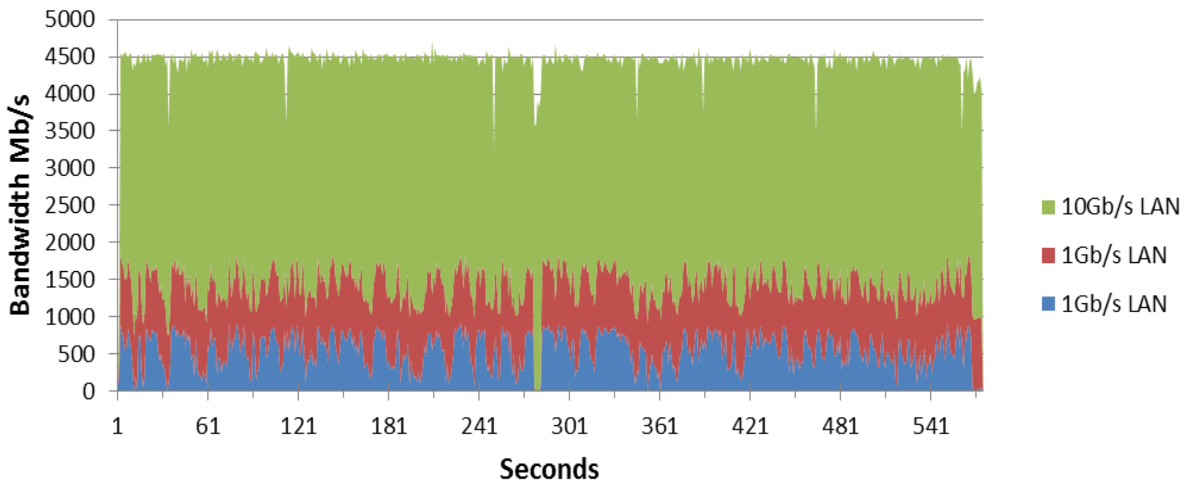
- Interfaces go UP and DOWN

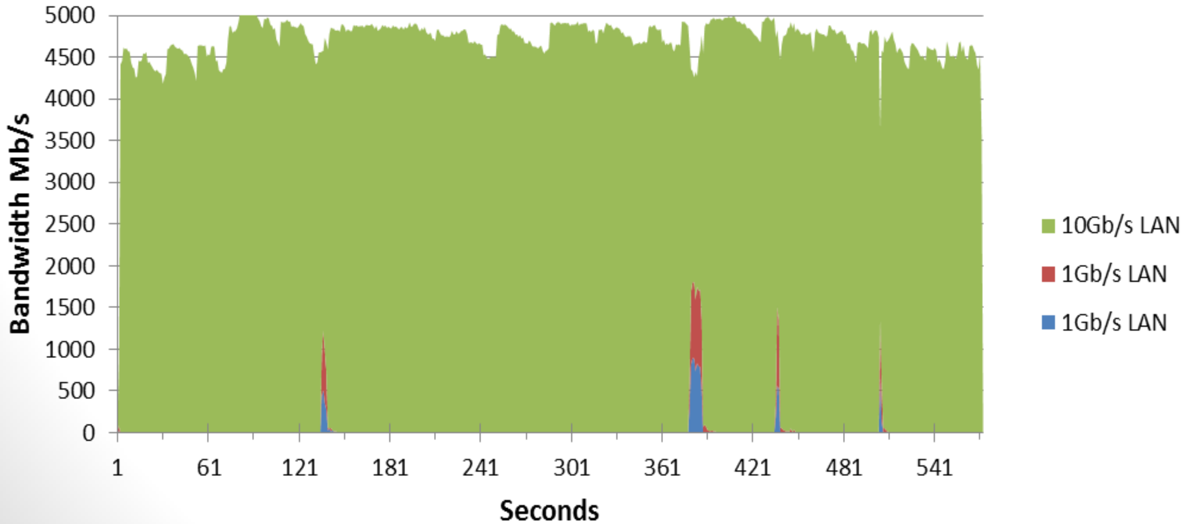
| | LAN | LAN |
|---|---|---|
| Speed | 1Gb/s | 1Gb/s |
| RTT | 5ms | 5ms |
| Buffer | 6MB | 6MB |
| Min-Buf | 2.5MB | 2.5MB |
| MSS | 1400 | 1400 |



# LAN: Balancing

- We got both graphs with the exact same experiment



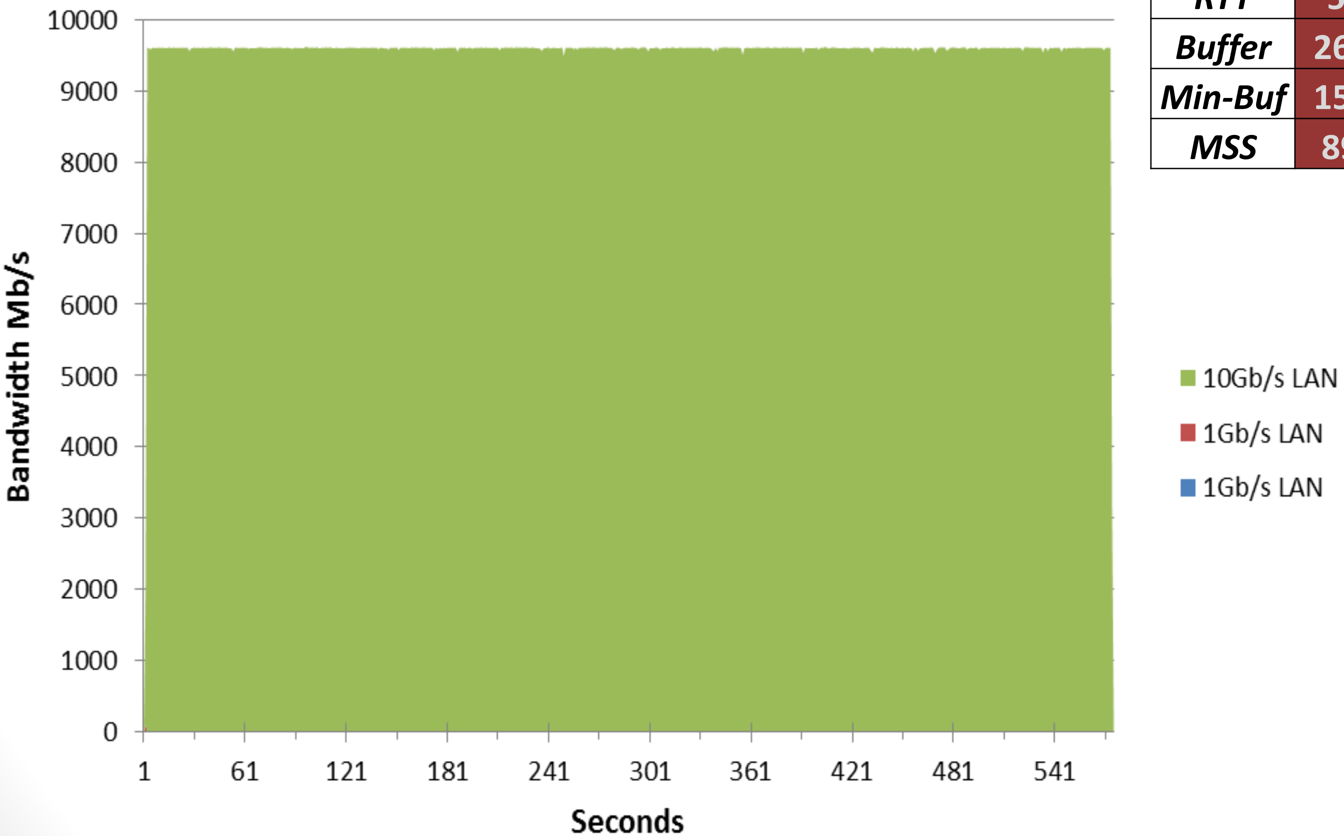




| | *LAN* | *LAN* | *LAN* |
|---|---|---|---|
| ***Speed*** | 1Gb/s | 1Gb/s | 10Gb/s |
| ***RTT*** | 5ms | 5ms | 5ms |
| ***Buffer*** | 16MB | 16MB | 16MB |
| ***Min-Buf*** | 15MB | 15MB | 15MB |
| ***MSS*** | 1400 | 1400 | 1400 |

# LAN: Balancing

- MSS and buffers increased

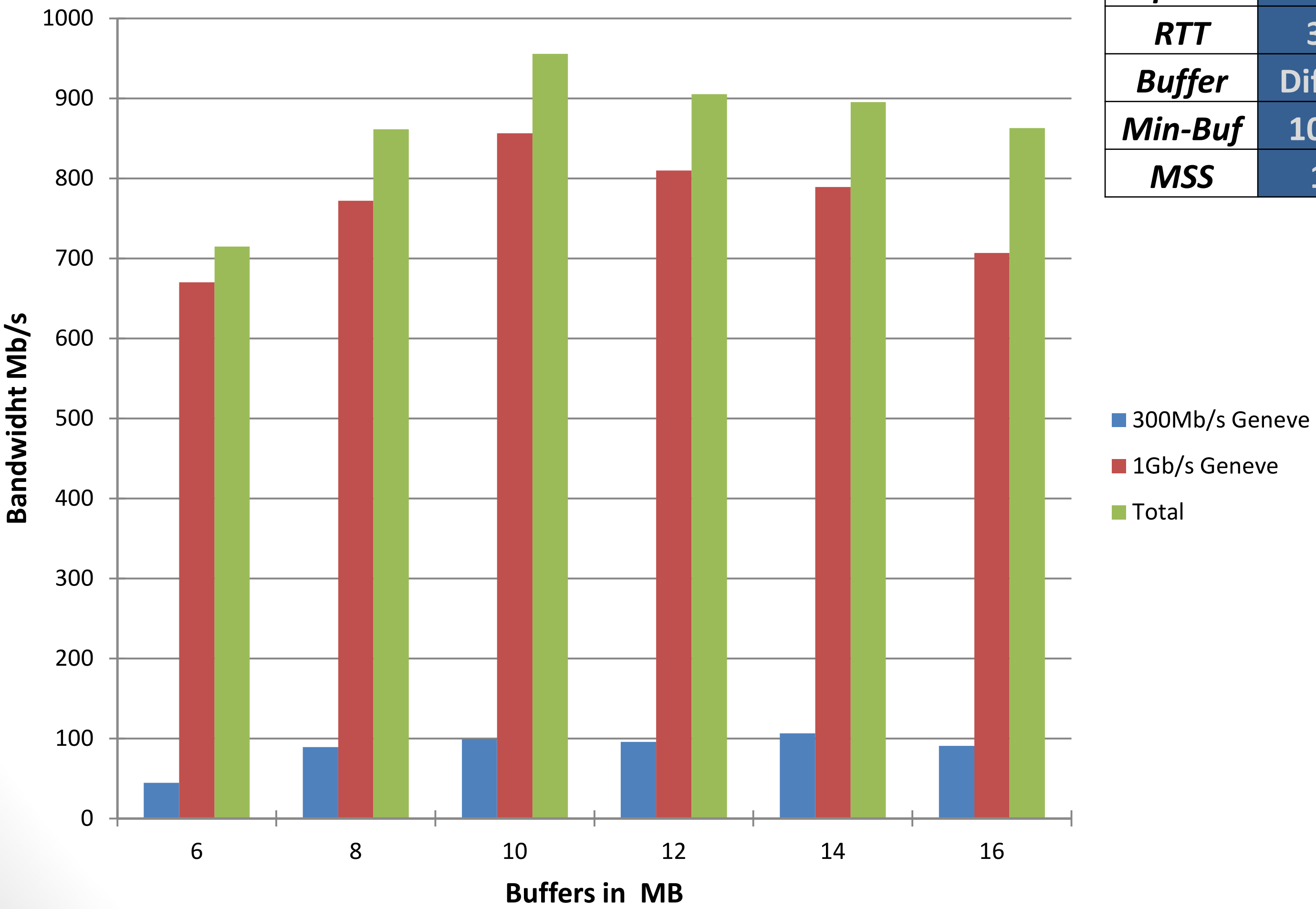
| | LAN | LAN | LAN |
|---|---|---|---|
| ***Speed*** | 1Gb/s | 1Gb/s | 10Gb/s |
| ***RTT*** | 5ms | 5ms | 5ms |
| ***Buffer*** | 26MB | 26MB | 26MB |
| ***Min-Buf*** | 15MB | 15MB | 15MB |
| ***MSS*** | 8900 | 8900 | 8900 |



# WAN: Throughput

- Increased round trip times

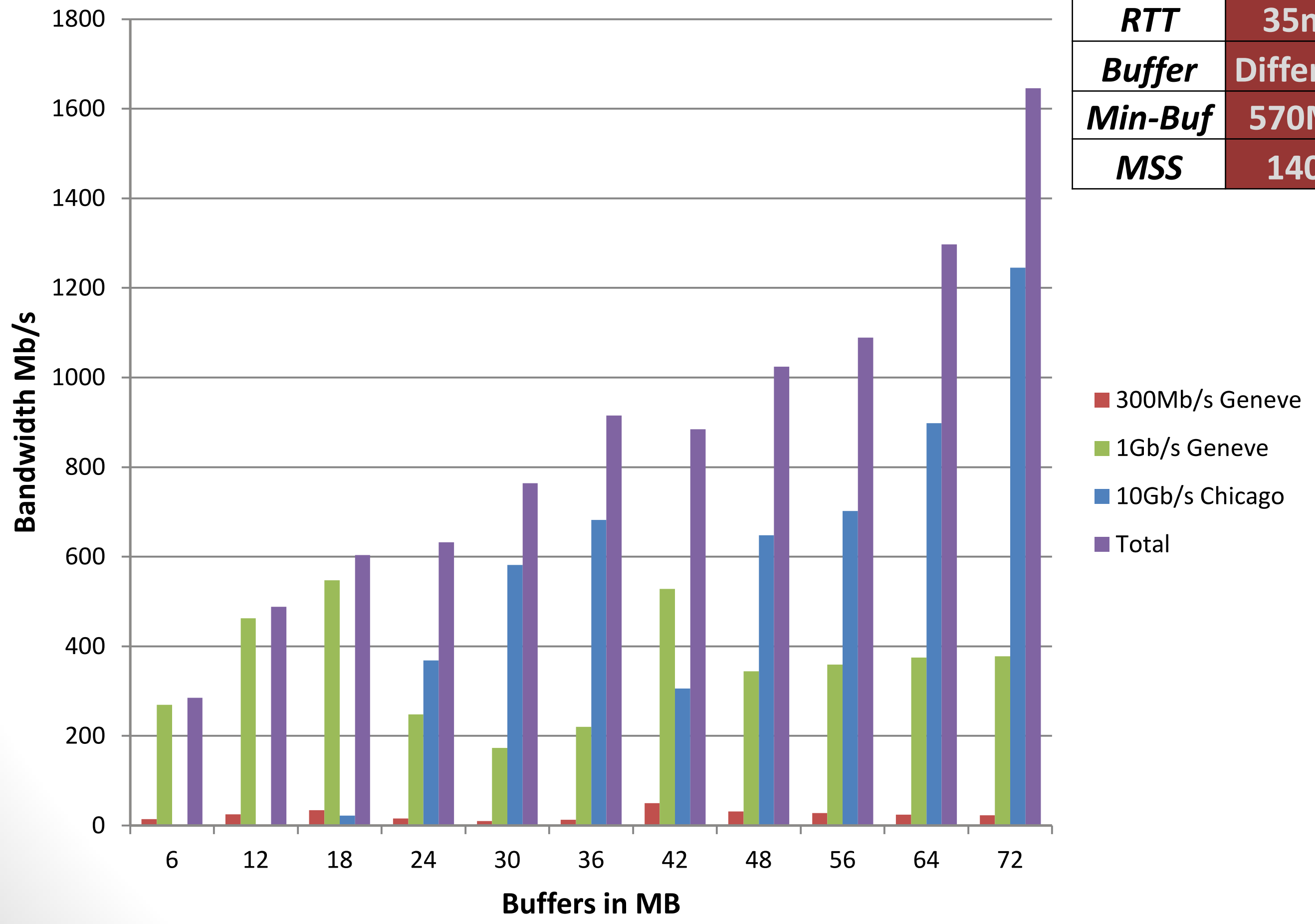
| | WAN | WAN |
|---|---|---|
| ***Speed*** | 300Mb/s | 1Gb/s |
| ***RTT*** | 35ms | 35ms |
| ***Buffer*** | Different | Different |
| ***Min-Buf*** | 10.8MB | 10.8MB |
| ***MSS*** | 1400 | 1400 |



# WAN: Advanced Throughput

- Using only the two Geneve links is more optimal
- Big RTT difference +/-170ms

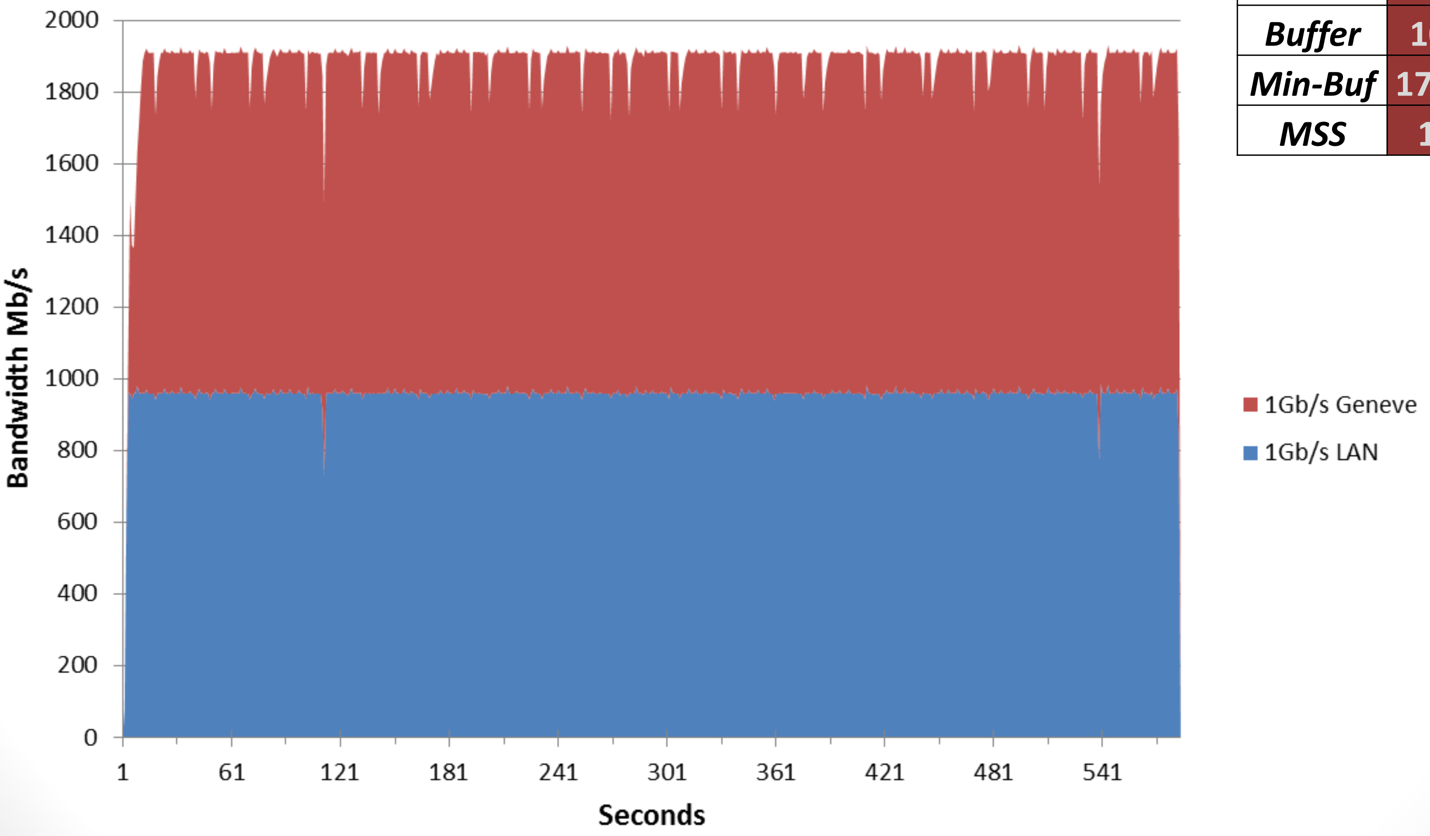
| | WAN | WAN | WAN |
|---|---|---|---|
| Speed | 300Mb/s | 1Gb/s | 10Gb/s |
| RTT | 35ms | 35ms | 202ms |
| Buffer | Different | Different | Different |
| Min-Buf | 570MB | 570MB | 570MB |
| MSS | 1400 | 1400 | 1400 |



# LAN + WAN: Throughput

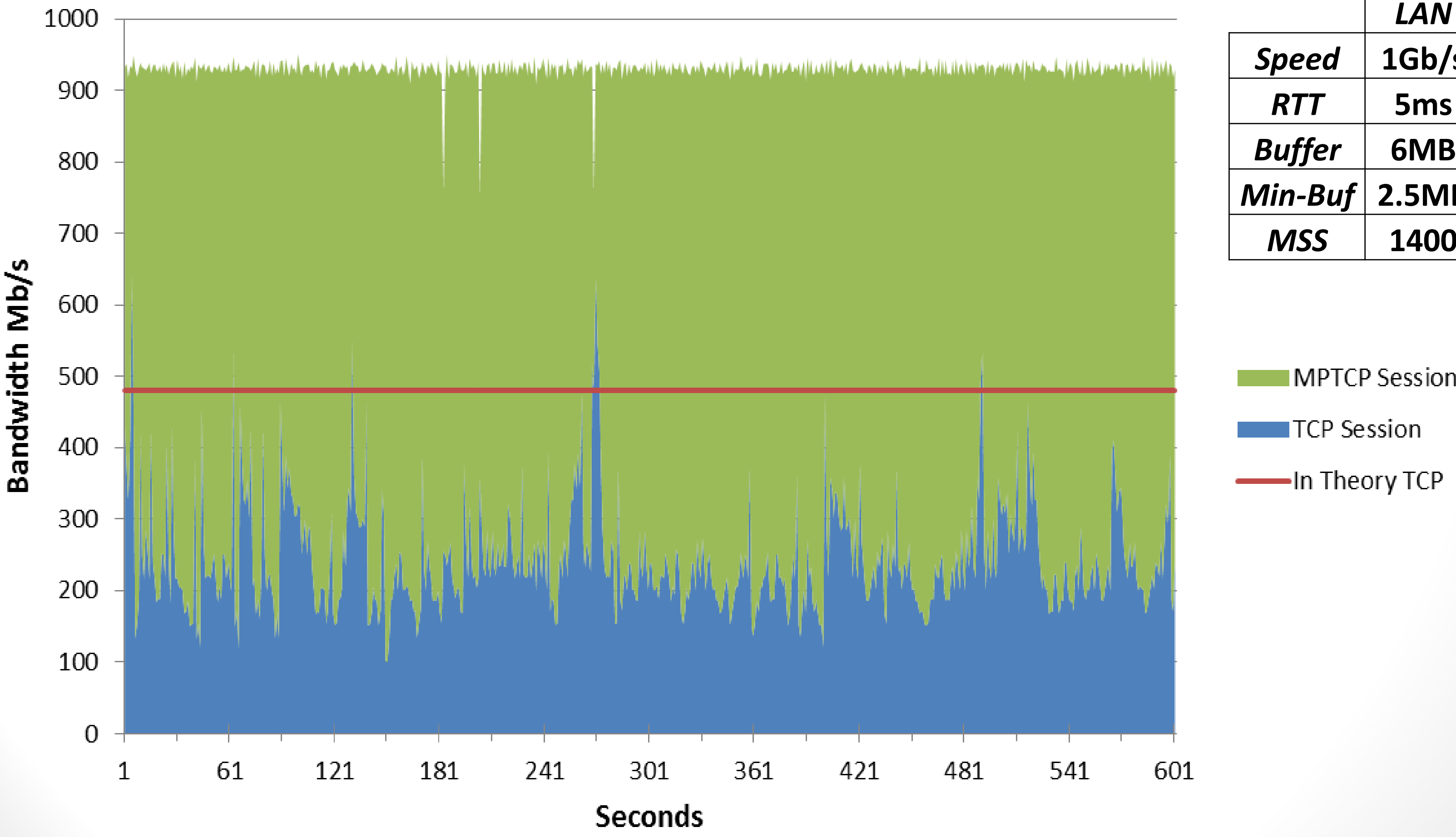
- Small difference in RTT +/- 30ms

| | WAN | LAN |
|---|---|---|
| Speed | 1Gb/s | 1Gb/s |
| RTT | 35ms | 5ms |
| Buffer | 10MB | 10MB |
| Min-Buf | 17.5MB | 17.5MB |
| MSS | 1400 | 1400 |



# LAN: Fairness

- One can see that the bandwidth TCP gets is far below what it ‘should’ get in theory




| | *LAN* |
|---|---|
| ***Speed*** | **1Gb/s** |
| ***RTT*** | **5ms** |
| ***Buffer*** | **6MB** |
| ***Min-Buf*** | **2.5MB** |
| ***MSS*** | **1400** |

# Analysis

- **Behavior of the different parameters**
- **Performance dips in graphs**
  - Window size decreases (packets are dropped)
    - Slow server?
    - Overflowing buffers?
- **Interfaces going UP and DOWN**
  - MPTCP debug option
    - Subflow count stays 1 while it should be 2, no clue why this happens
  - Tcpdump/Wireshark
    - No clear explanation yet. (indication its due to the socket buffer in combination with the window size)

# Achievements

**Experience:**

- Kernel froze sometimes, especially when interfaces went up and down
- Can work with both IPv4 and IPv6 simultaneously
- MPTCP seems quite stable overall

**Research**

- MPTCP meets its goals: *improve throughput* and *balance congestion*
- The goal: *do no harm* is not met perfectly. In our experiments MPTCP is a bit unfair to TCP
- The behavior of MPTCP in different environments with different parameters

# Conclusion

**Research Question**:

*Is the current MPTCP implementation a useful technology for e-science data transfers in the GLIF environment?*

- When the e-science environment is stable, uses the same link speeds, has high enough buffers and same RTTs
  - MPTCP seems to behave well and gets maximum throughput
- However, when you have a lot of differences in link speeds, buffer sizes and RTTs
  - MPTCP may behave less optimal and becomes as good as TCP would get. One should consider if using MPTCP gives any real benefit. However, when robustness is a key factor you can of course make use of MPTCP
- With higher speeds, one would need fast servers and one should put a lot of attention in tweaking all parameters

# Future work

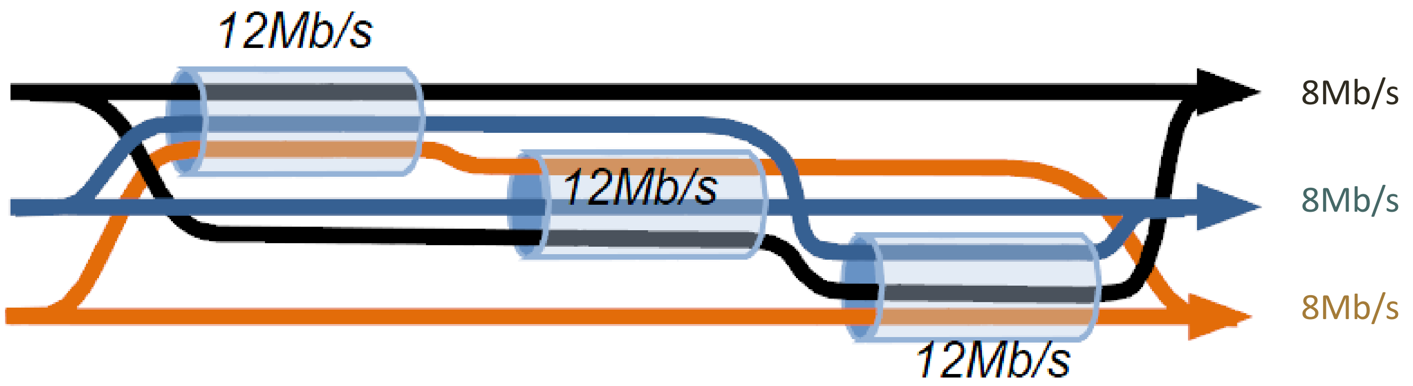
- More advanced analyzing and testing of the protocol
- Testing against other projects like GridFTP
- The GLIF test-bed topology within SARA
- Run experiments again to verify our results
- Investigate the tuning further
- Try it yourself

**Questions**

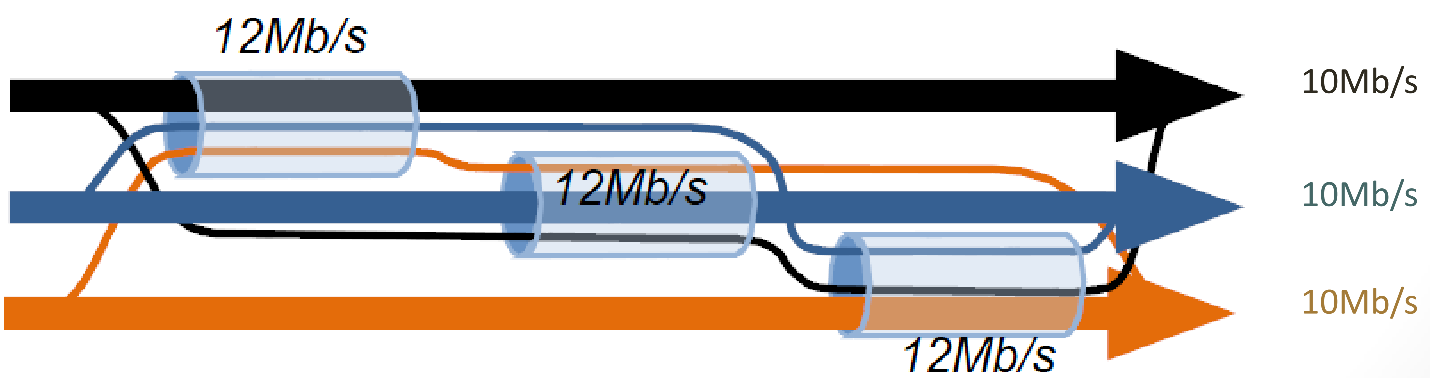
# Backup Slides

# Congestion Algorithm

- Should make sure the most efficient paths are taken and meet the design goals of MPTCP

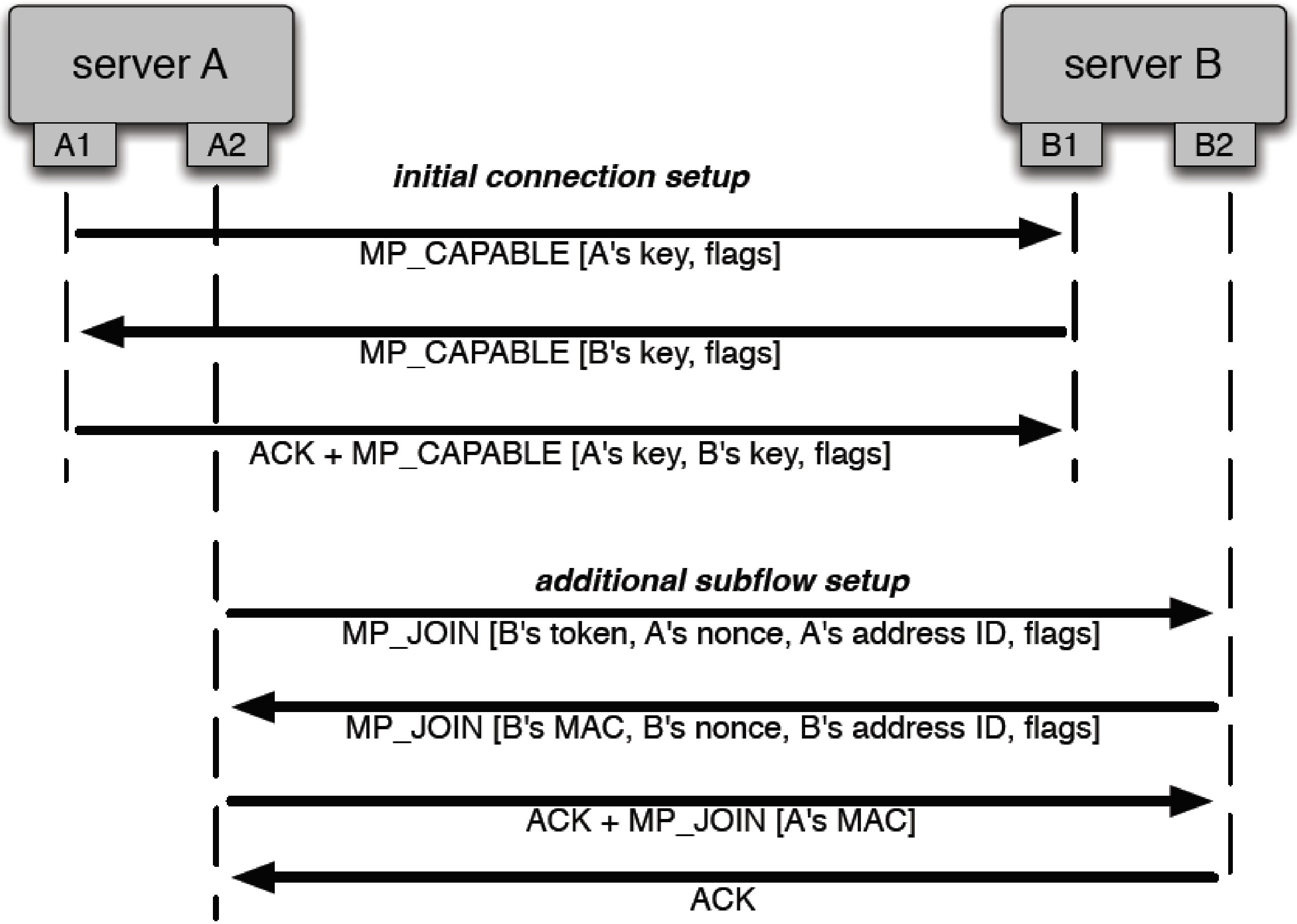



Flow 1:1




Flow 4:1

# MPTCP Handshake



# Buffer calculation

- TCP:

$$Buffersize_{TCP} = RTT_{max} * LinkMax_{bits}$$

- MPTCP:

$$Buffersize_{MPTCP} = RTT_{max} * AllLinksMax_{bits} * 2$$

- Example: RTT=36ms, 2x 1Gb/s

$$0.036 * 2000000000 * 2 = 144000000bit = 18\text{MB}$$

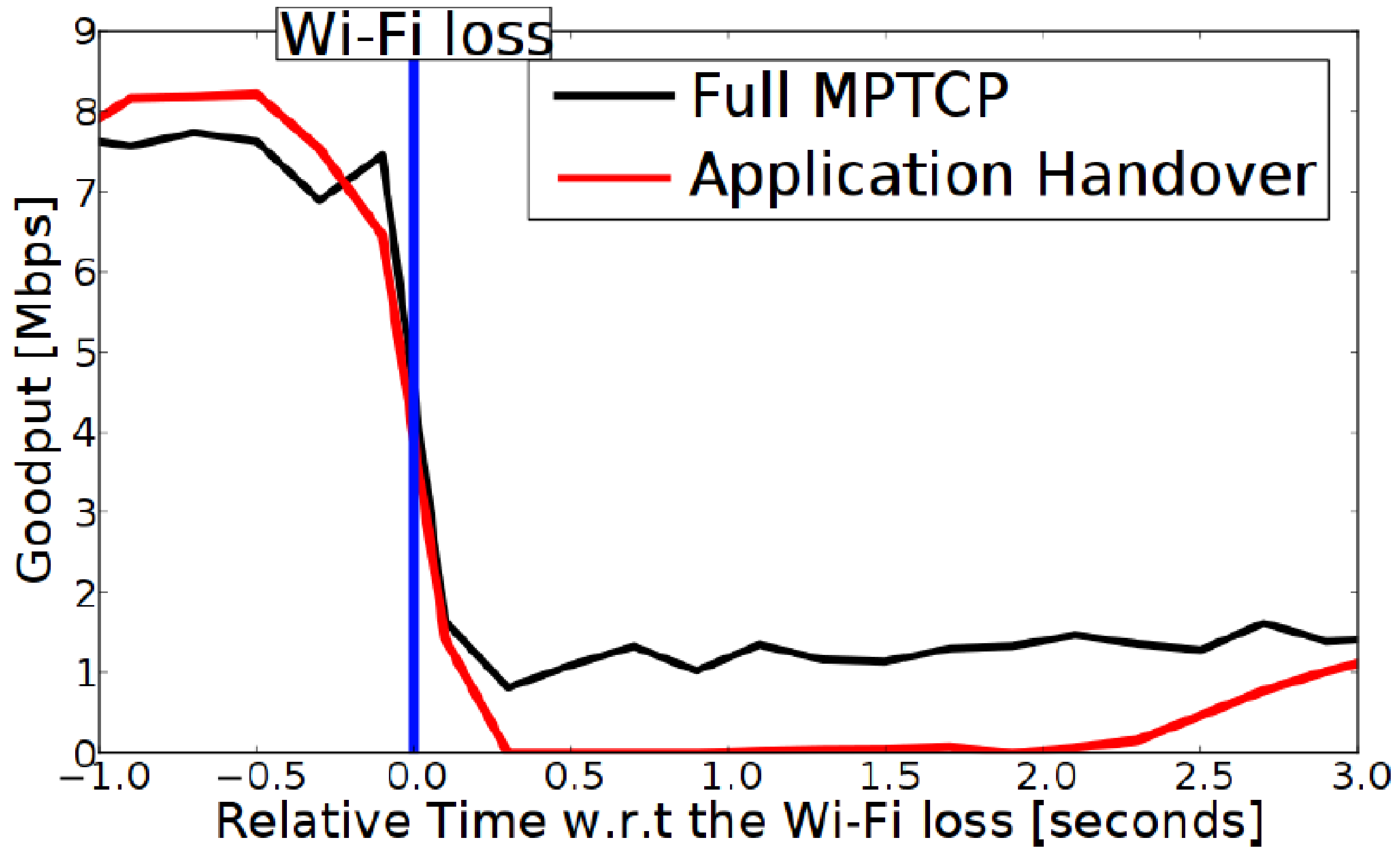
# MPTCP Algorithm

- Window size increase rule is only changed

$$cwnd_i = cwnd_i + min\left(\frac{alpha}{cwnd_{tot}}, \frac{1}{cwnd_i}\right)$$

$$alpha = cwnd_{tot}\frac{max_i(\frac{cwnd_i * mss_i^2}{RTT_i^2})}{(\sum_i \frac{cwnd_i * mss_i}{RTT_i})^2}$$

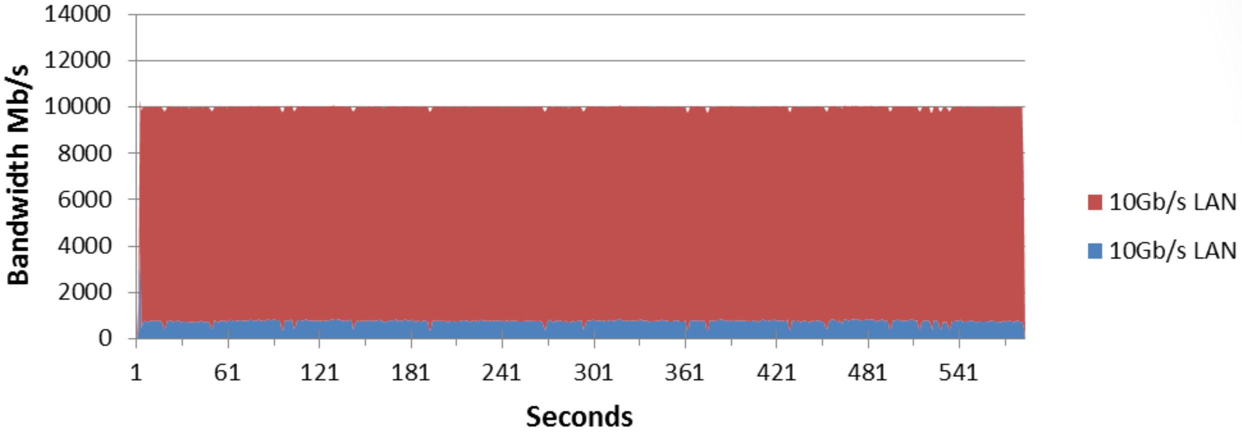
# MPTCP Handover



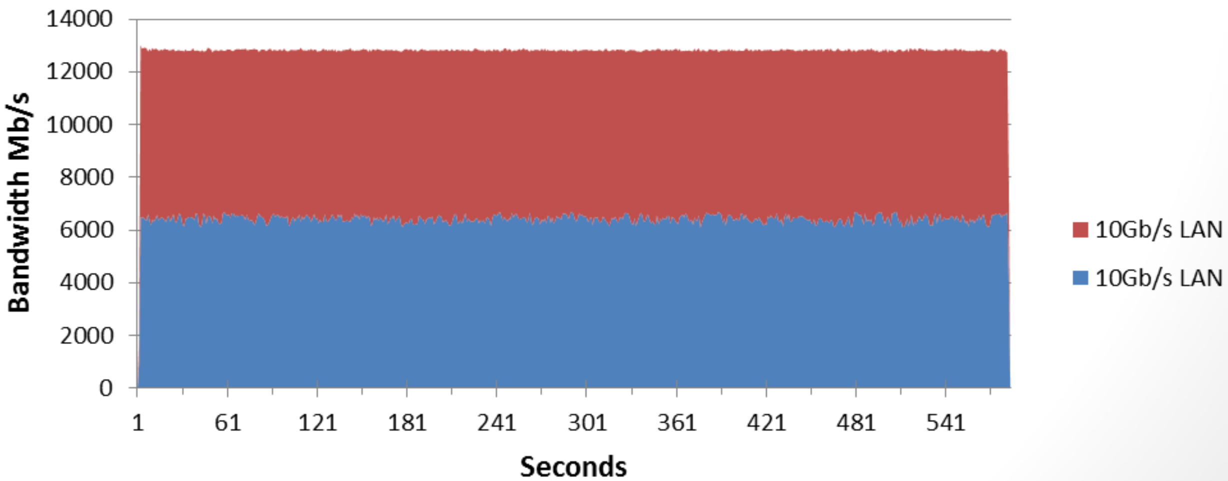
# LAN: 2x 10Gb/s Link

| | LAN | LAN |
|---|---|---|
| Speed | 10Gb/s | 10Gb/s |
| RTT | 5ms | 5ms |
| Buffer | 20MB | 20MB |
| Min-Buf | 25MB | 25MB |
| MSS | 8900 | 8900 |

- One MPTCP session

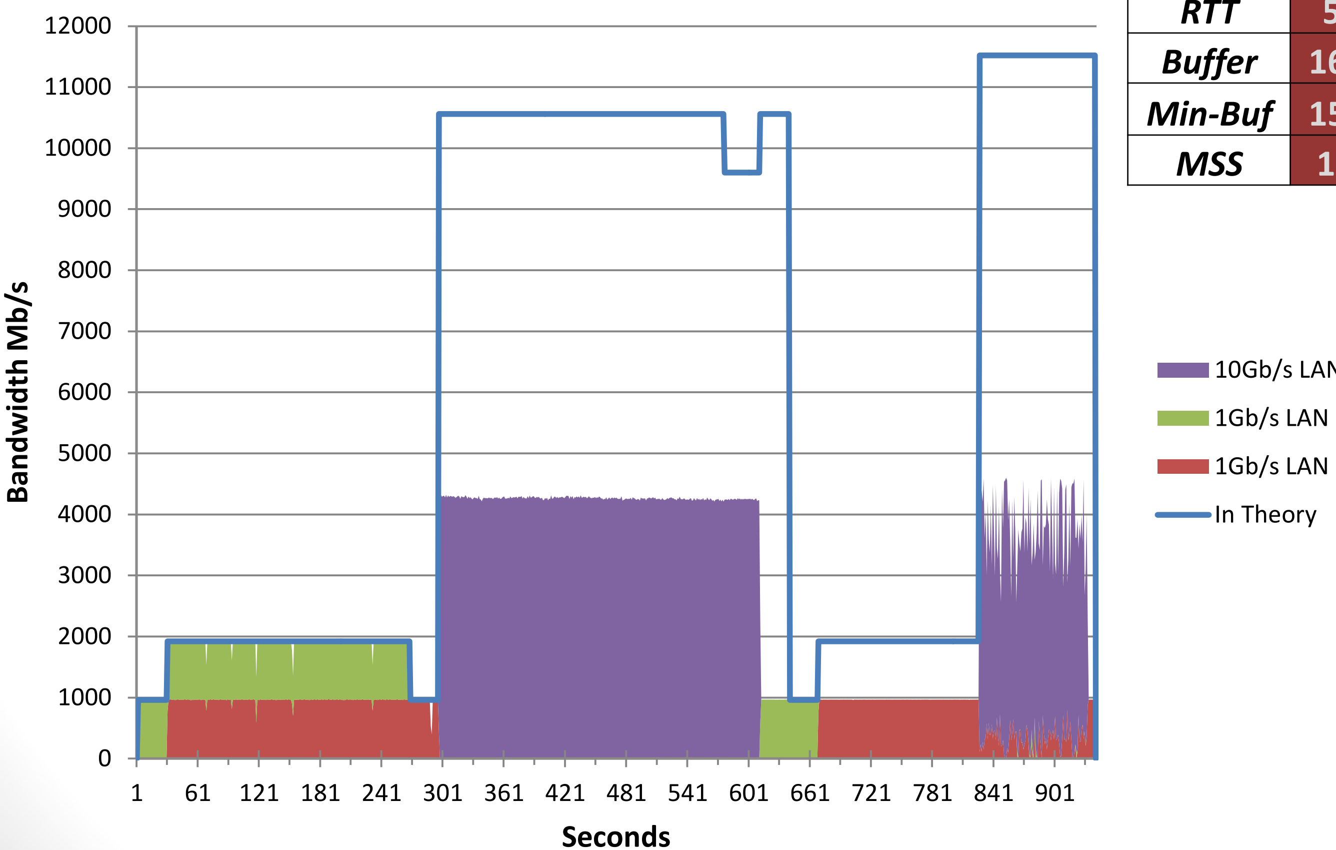



- Two MPTCP sessions

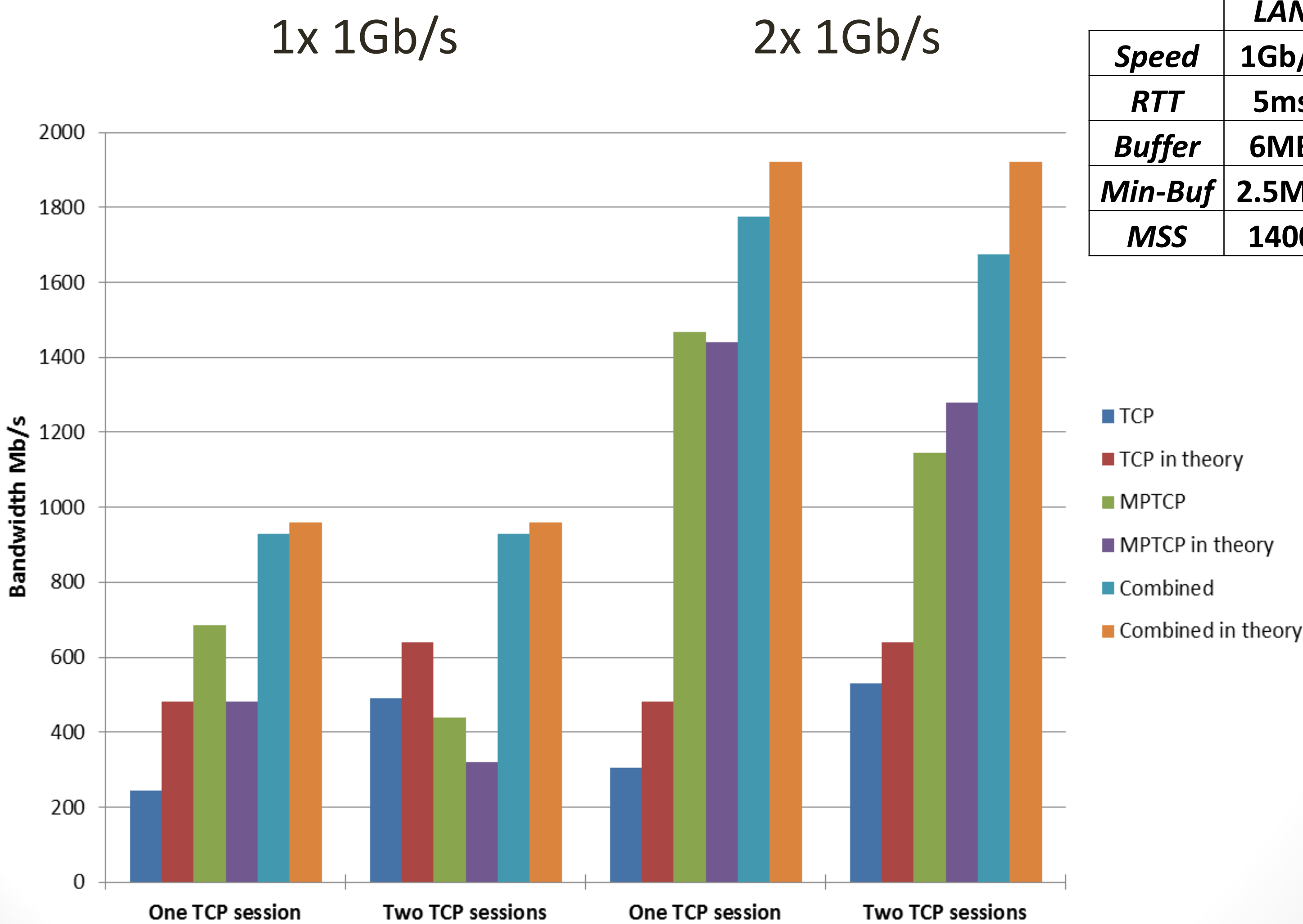


# LAN: Advanced changes

| | LAN | LAN | LAN |
|---|---|---|---|
| Speed | 1Gb/s | 1Gb/s | 10Gb/s |
| RTT | 5ms | 5ms | 5ms |
| Buffer | 16MB | 16MB | 16MB |
| Min-Buf | 15MB | 15MB | 15MB |
| MSS | 1400 | 1400 | 1400 |

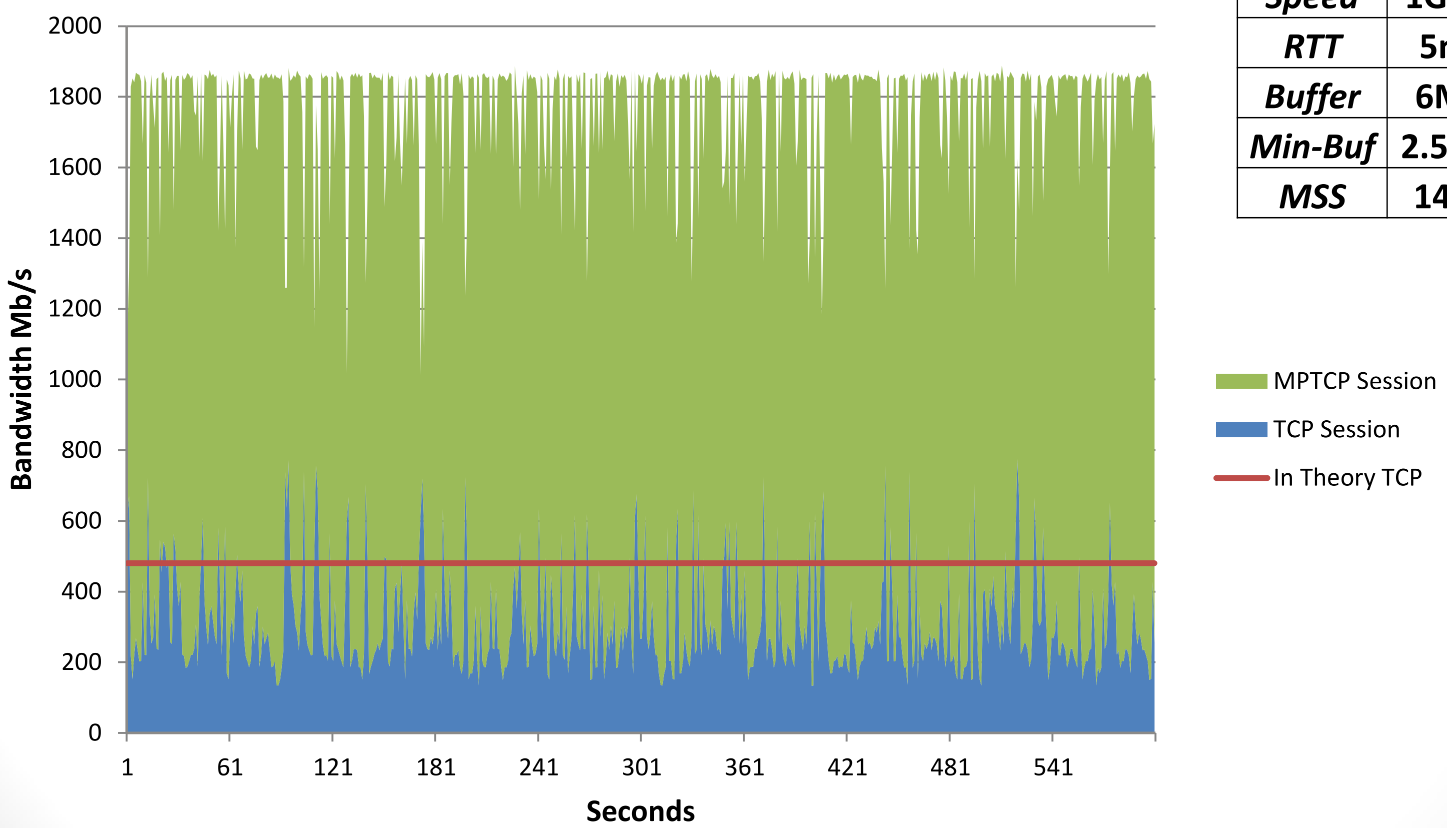


# LAN: Fairness with a TCP session



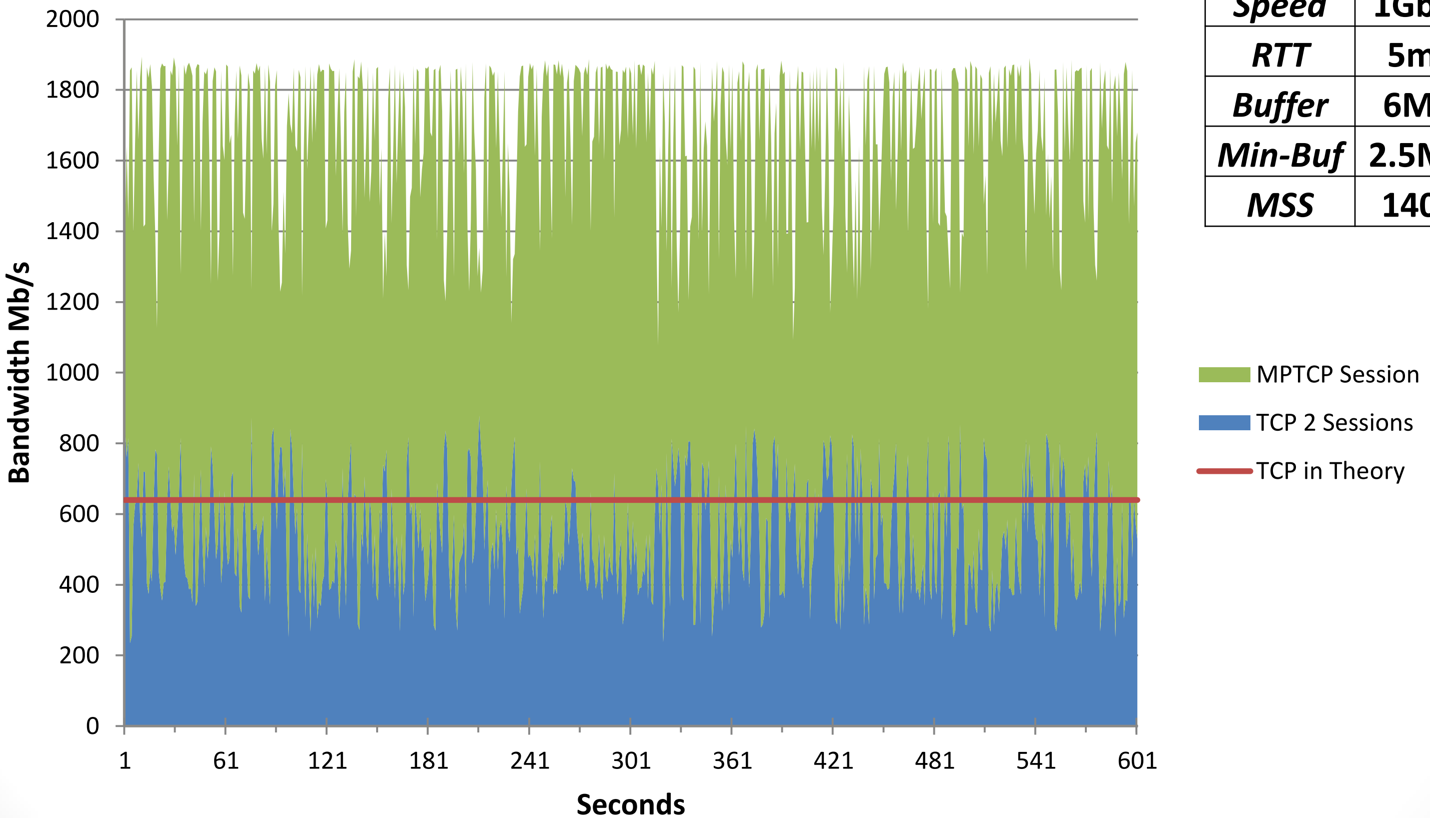

| | *LAN* | *LAN* |
|---|---|---|
| ***Speed*** | **1Gb/s** | **1Gb/s** |
| ***RTT*** | **5ms** | **5ms** |
| ***Buffer*** | **6MB** | **6MB** |
| ***Min-Buf*** | **2.5MB** | **2.5MB** |
| ***MSS*** | **1400** | **1400** |

# LAN: Fairness on 2x 1Gb/s links




| | *LAN* | *LAN* |
|---|---|---|
| ***Speed*** | **1Gb/s** | **1Gb/s** |
| ***RTT*** | **5ms** | **5ms** |
| ***Buffer*** | **6MB** | **6MB** |
| ***Min-Buf*** | **2.5MB** | **2.5MB** |
| ***MSS*** | **1400** | **1400** |

# LAN: Fairness on 2x 1Gb/s links




| | ***LAN*** | ***LAN*** |
|---|---|---|
| ***Speed*** | **1Gb/s** | **1Gb/s** |
| ***RTT*** | **5ms** | **5ms** |
| ***Buffer*** | **6MB** | **6MB** |
| ***Min-Buf*** | **2.5MB** | **2.5MB** |
| ***MSS*** | **1400** | **1400** |