

High-speed Networks, Cybersecurity, and Software-defined Networking Workshop

Jorge Crichigno, Jose Gomez, Elie Kfoury
University of South Carolina

Western Academy Support and Training Center (WASTC)
2020 Summer Conference
June 15 – June 19



National Science Foundation (NSF), Office of Advanced Cyberinfrastructure (OAC) and
Advanced Technological Education (ATE)

Lab 8: Bandwidth-delay Product and TCP Buffer Size

Content

- Introduction to TCP buffers, BDP, and TCP window
- BDP and buffer size experiments
- Modifying buffer size and throughput test

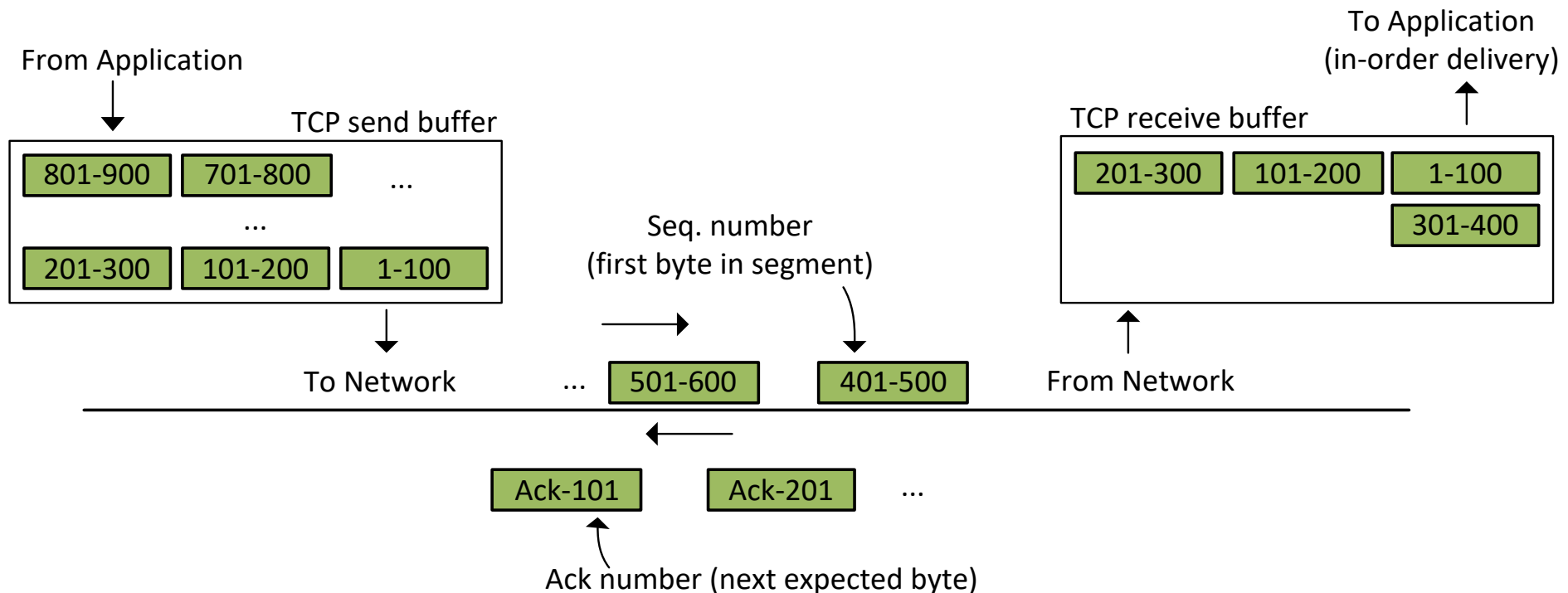
Section 1: Introduction to TCP buffers, BDP, and TCP window

TCP Buffers

- The TCP send and receive buffers may impact the performance of Wide Area Networks (WAN) data transfers
- At the sender side, TCP receives data from the application layer and places it in the TCP send buffer

TCP buffers

- Typically, TCP fragments the data in the buffer into maximum segment size (MSS) units
- At any given time, the TCP receiver indicates the TCP sender how many bytes the latter can send, based on how much free buffer space is available at the receiver



Bandwidth-delay product

- RTT and TCP buffer size have throughput implications
- For example, assume that the TCP buffer size is 1 Mbyte and RTT is 25ms
 - 1 Mbyte = 1,048,576 bytes = 1,048,576 · 8 bits = 8,388,608 bits
- With a bandwidth (Bw) of 10 Gbps, this number of bits is approximately transmitted in:

$$T_{\text{tx}} = \frac{\# \text{ bits}}{\text{Bw}} = \frac{8,388,608}{10 \cdot 10^9} = 0.84 \text{ milliseconds.}$$

- After 0.84 milliseconds, the TCP send buffer will be empty
- TCP must wait for the corresponding acknowledgements (arriving at t = 50ms)
- This means that the sender only uses 0.84/50 or 1.68% of the available bandwidth

Bandwidth-delay product

- The solution lies in allowing the sender to continuously transmit segments until the corresponding acknowledgments arrive back
- The number of bits that can be transmitted in an RTT period is the bandwidth-delay product (BDP)
- For the previous example

$$\text{TCP buffer size} \geq \text{BDP} = (10 \cdot 10^9) (50 \cdot 10^{-3}) = 500,000,000 \text{ bits} = 62,500,000 \text{ bytes.}$$

- The first factor ($10 \cdot 10^9$) is the bandwidth; the second factor ($50 \cdot 10^{-3}$) is the RTT

$$\text{TCP buffer size} \geq 62,500,000 \text{ bytes} = 59.6 \text{ Mbytes} \approx 60 \text{ Mbytes.}$$

Practical Observations on Setting TCP Buffer Size

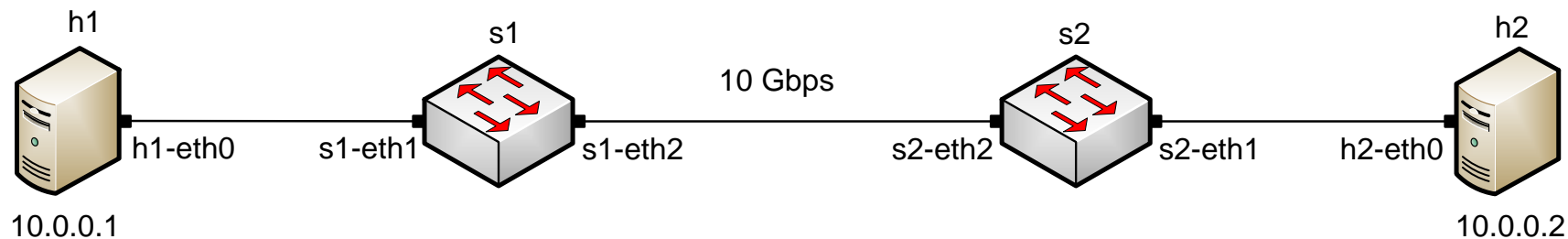
- Linux assumes that half of the send/receive TCP buffers are used for internal structures
- Thus, only half of the buffer size is used to store segments
- Considering the previous example, the TCP buffer size must be:

TCP buffer size $\geq 2 \cdot 60 \text{ Mbytes} = 120 \text{ Mbytes}$.

Section 2: BDP and buffer size experiments

Emulating a Wide Area Network

- The first figure shows the topology and the devices' interfaces
- The second and third figures show the command that sets a latency of 20ms and bandwidth to 10 Gbps



```
admin@admin-pc: ~
File Actions Edit View Help
admin@admin-pc: ~
admin@admin-pc:~$ sudo tc qdisc add dev s1-eth2 root handle 1: netem delay 20ms
[sudo] password for admin:
admin@admin-pc:~$
```

```
admin@admin-pc: ~
File Actions Edit View Help
admin@admin-pc: ~
admin@admin-pc:~$ sudo tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 10gbit
burst 5000000 limit 15000000
admin@admin-pc:~$
```

Verification

- The user can now verify the previous configuration by using the iperf3 tool to measure throughput

```
root@admin-pc:~# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 15] local 10.0.0.1 port 59976 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate    Retr  Cwnd
[ 15]  0.00-1.00    sec    328 MBytes  2.75 Gbits/sec  90  16.1 MBytes
[ 15]  1.00-2.00    sec    394 MBytes  3.30 Gbits/sec   0  16.1 MBytes
[ 15]  2.00-3.00    sec    391 MBytes  3.28 Gbits/sec   0  16.1 MBytes
[ 15]  3.00-4.00    sec    394 MBytes  3.30 Gbits/sec   0  16.1 MBytes
[ 15]  4.00-5.00    sec    394 MBytes  3.30 Gbits/sec   0  16.1 MBytes
[ 15]  5.00-6.00    sec    390 MBytes  3.27 Gbits/sec   0  16.1 MBytes
[ 15]  6.00-7.00    sec    394 MBytes  3.30 Gbits/sec   0  16.1 MBytes
[ 15]  7.00-8.00    sec    396 MBytes  3.32 Gbits/sec   0  16.1 MBytes
[ 15]  8.00-9.00    sec    396 MBytes  3.32 Gbits/sec   0  16.1 MBytes
[ 15]  9.00-10.00   sec    394 MBytes  3.30 Gbits/sec   0  16.1 MBytes
-----
[ ID] Interval      Transfer    Bitrate    Retr
[ 15]  0.00-10.00   sec    3.78 GBytes  3.25 Gbits/sec  90
[ 15]  0.00-10.04   sec    3.78 GBytes  3.23 Gbits/sec
iperf Done.
root@admin-pc:~#
```

Client (h1)

```
root@admin-pc:~# iperf3 -s
-----
Server listening on 5201
-----
```

Server (h2)

Section 3: Modifying buffer size and throughput test

BDP and buffer size

- To achieve the full throughput, the user has to modify the send and receive windows in host h1 and host h2

```
X "Host: h1"
root@admin-pc:~# sysctl -w net.ipv4.tcp_rmem='10240 87380 52428800'
net.ipv4.tcp_rmem = 10240 87380 52428800
root@admin-pc:~#
```

```
X "Host: h2"
root@admin-pc:~# sysctl -w net.ipv4.tcp_rmem='10240 87380 52428800'
net.ipv4.tcp_rmem = 10240 87380 52428800
root@admin-pc:~#
```

```
X "Host: h1"
root@admin-pc:~# sysctl -w net.ipv4.tcp_wmem='10240 87380 52428800'
net.ipv4.tcp_wmem = 10240 87380 52428800
root@admin-pc:~#
```

```
X "Host: h2"
root@admin-pc:~# sysctl -w net.ipv4.tcp_wmem='10240 87380 52428800'
net.ipv4.tcp_wmem = 10240 87380 52428800
root@admin-pc:~#
```

Verification

- The user can now verify the previous configuration by using the iperf3 tool to measure throughput

```
root@admin-pc:~# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 15] local 10.0.0.1 port 47094 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate    Retr  Cwnd
[ 15] 0.00-1.00    sec    925 MBytes  7.76 Gbits/sec  45  39.8 MBytes
[ 15] 1.00-2.00    sec    1.11 GBytes  9.57 Gbits/sec   0  39.8 MBytes
[ 15] 2.00-3.00    sec    1.11 GBytes  9.56 Gbits/sec   0  39.8 MBytes
[ 15] 3.00-4.00    sec    1.11 GBytes  9.56 Gbits/sec   0  39.8 MBytes
[ 15] 4.00-5.00    sec    1.11 GBytes  9.56 Gbits/sec   0  39.8 MBytes
[ 15] 5.00-6.00    sec    1.11 GBytes  9.55 Gbits/sec   0  39.8 MBytes
[ 15] 6.00-7.00    sec    1.11 GBytes  9.56 Gbits/sec   0  39.8 MBytes
[ 15] 7.00-8.00    sec    1.11 GBytes  9.56 Gbits/sec   0  39.8 MBytes
[ 15] 8.00-9.00    sec    1.11 GBytes  9.56 Gbits/sec   0  39.8 MBytes
[ 15] 9.00-10.00   sec    1.11 GBytes  9.56 Gbits/sec   0  39.8 MBytes
-----
[ ID] Interval      Transfer    Bitrate    Retr
[ 15] 0.00-10.00   sec    10.9 GBytes  9.38 Gbits/sec  45
[ 15] 0.00-10.04   sec    10.9 GBytes  9.34 Gbits/sec
iperf Done.
root@admin-pc:~#
```

Client (h1)

```
root@admin-pc:~# iperf3 -s
-----
Server listening on 5201
-----
```

Server (h2)