

# TCP: Understanding Congestion Control

16/05/2017

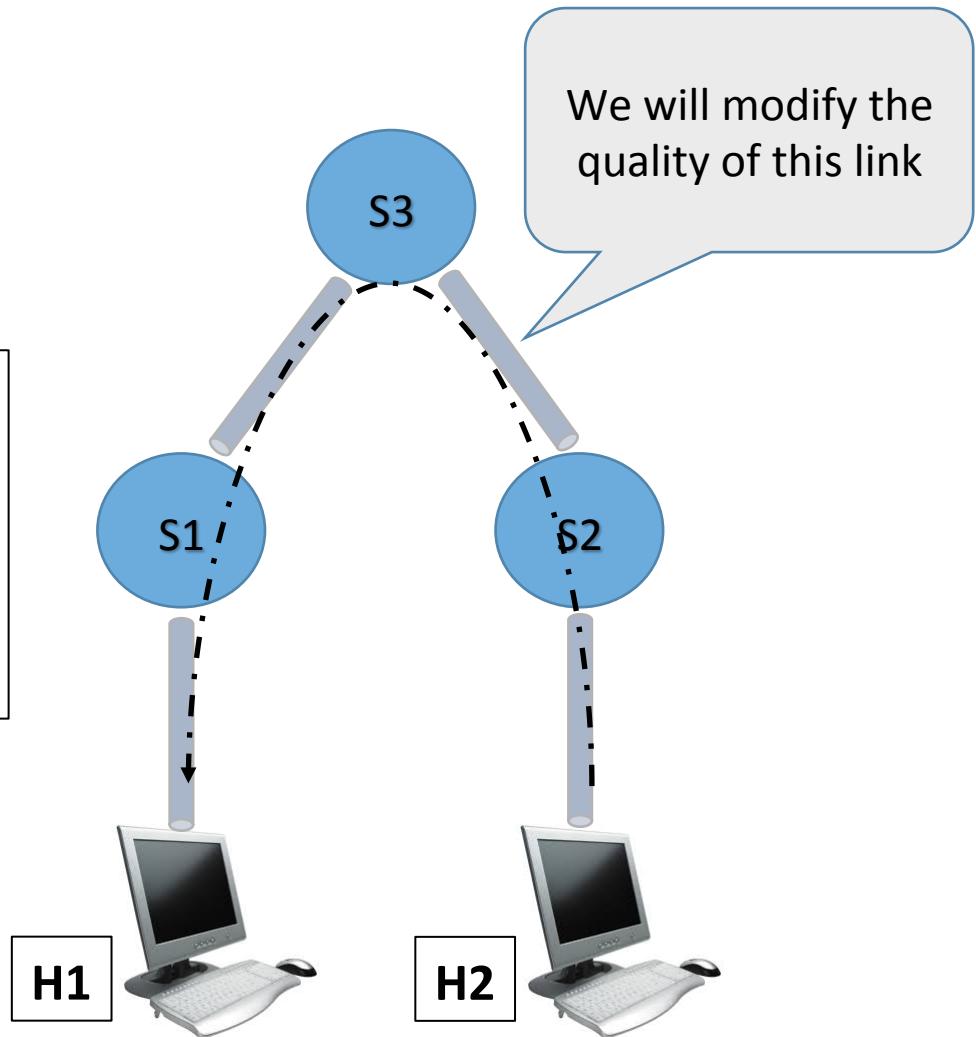
# Congestion control in Linux

- The default algorithm is Cubic, you can check with  
`cat /proc/sys/net/ipv4/tcp_congestion_control`
- The available algorithms are listed in the file:  
`/proc/sys/net/ipv4/tcp_available_congestion_control`
- You can change the TCP congestion control writing the file:  
`/proc/sys/net/ipv4/tcp_congestion_control`
- You have to modify this file on the Mininet VM, NOT on the hosts!
- Ex:

```
echo reno | sudo tee /proc/sys/net/ipv4/tcp_congestion_control
```

# Topology

- We will measure the bandwidth between h1 and h2 using iperf
- Server h1:  
`iperf -s`
- Client h2:  
`iperf -c <ip_server> -t 60`



# Reno VS Cubic

Link S2->S3: Bandwidth 1Gbit/s – Delay 15ms

Reno

Cubic

# Reno VS Cubic

Link S2->S3: Bandwidth 1Gbit/s – Delay 15ms

## Reno

[ 19]	57.00-58.00	sec	108 MBytes	902 Mbits/sec		
[ 19]	58.00-59.00	sec	107 MBytes	900 Mbits/sec		
[ 19]	59.00-60.00	sec	107 MBytes	900 Mbits/sec		
[ 19]	60.00-60.73	sec	9.45 MBytes	109 Mbits/sec		
-----						
[ ID]	Interval		Transfer	Bandwidth	Retr	
[ 19]	0.00-60.73	sec	6.39 GBytes	903 Mbits/sec	0	sender
[ 19]	0.00-60.73	sec	6.38 GBytes	903 Mbits/sec		receiver

## Cubic

[ 19]	57.00-58.00	sec	109 MBytes	913 Mbits/sec		
[ 19]	58.00-59.00	sec	110 MBytes	924 Mbits/sec		
[ 19]	59.00-60.00	sec	108 MBytes	907 Mbits/sec		
[ 19]	60.00-60.84	sec	10.6 MBytes	106 Mbits/sec		
-----						
[ ID]	Interval		Transfer	Bandwidth	Retr	
[ 19]	0.00-60.84	sec	6.42 GBytes	906 Mbits/sec	0	sender
[ 19]	0.00-60.84	sec	6.41 GBytes	906 Mbits/sec		receiver

# Reno VS Cubic

Link S2->S3: Bandwidth 1Gbit/s – **Delay 150ms**

Reno

Cubic

# Reno VS Cubic

Link S2->S3: Bandwidth 1Gbit/s – Delay 150ms

## Reno

[ 19]	57.00-58.00	sec	14.7 MBytes	123 Mbits/sec			
[ 19]	58.00-59.00	sec	13.6 MBytes	114 Mbits/sec			
[ 19]	59.00-60.00	sec	12.2 MBytes	102 Mbits/sec			
[ 19]	60.00-60.32	sec	4.05 MBytes	107 Mbits/sec			
-----							
[ ID]	Interval		Transfer	Bandwidth	Retr		
[ 19]	0.00-60.32	sec	781 MBytes	109 Mbits/sec	575		sender
[ 19]	0.00-60.32	sec	774 MBytes	108 Mbits/sec			receiver

## Cubic

[ 19]	57.00-58.00	sec	26.1 MBytes	219 Mbits/sec			
[ 19]	58.00-59.00	sec	29.8 MBytes	250 Mbits/sec			
[ 19]	59.00-60.00	sec	23.9 MBytes	201 Mbits/sec			
[ 19]	60.00-60.32	sec	7.98 MBytes	210 Mbits/sec			
-----							
[ ID]	Interval		Transfer	Bandwidth	I Retr		
[ 19]	0.00-60.32	sec	1.45 GBytes	206 Mbits/sec	575		sender
[ 19]	0.00-60.32	sec	1.44 GBytes	206 Mbits/sec			receiver

# TCP Probe

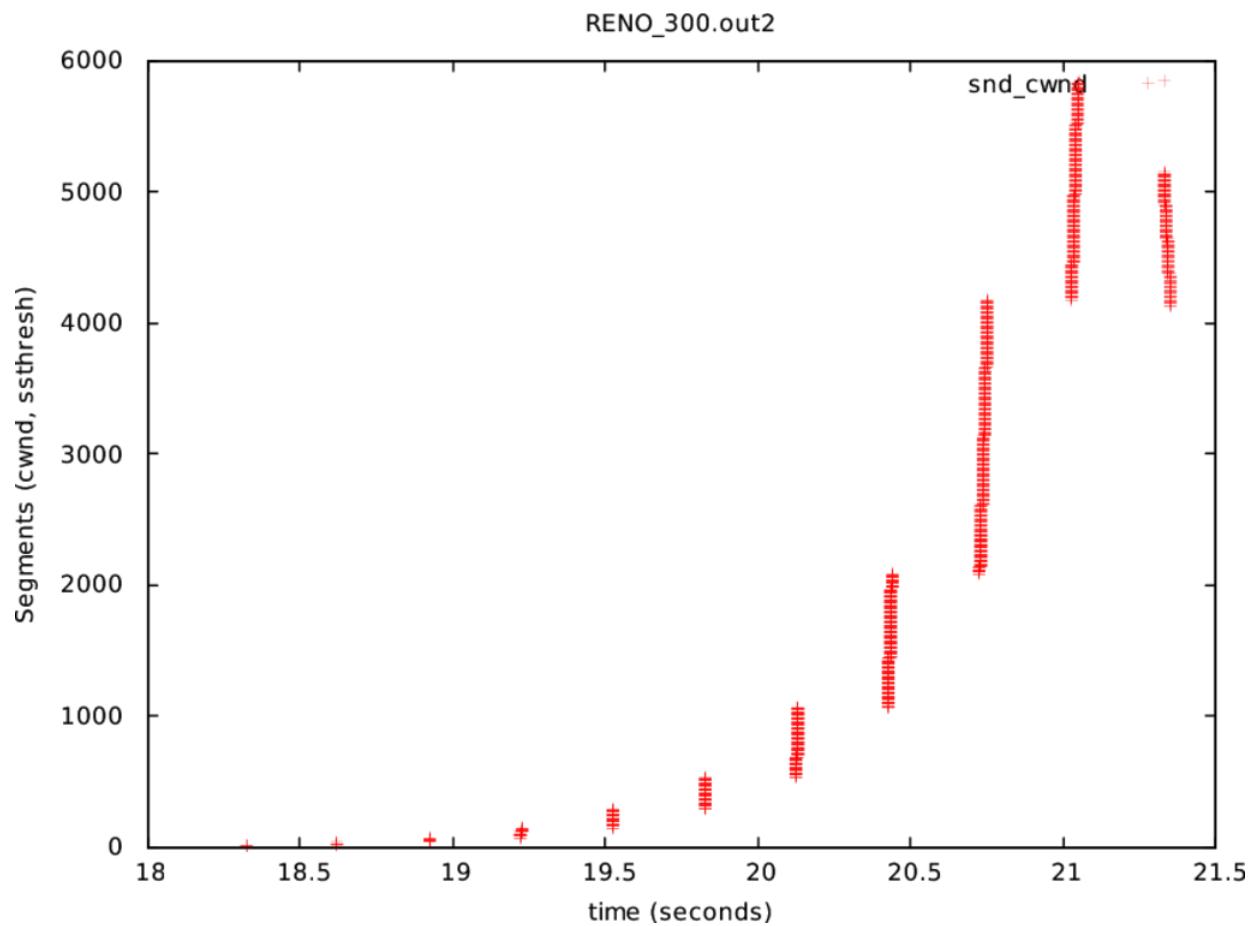
- Linux kernel module that records the state of a TCP connection in response to incoming packets
- Useful to monitor the congestion window (cwnd) and sequence number
- Example: monitor the cwnd on an iperf client

```
MN:~# modprobe tcp_probe port=5001
MN:~# cat /proc/net/tcpprobe >/tmp/data.out &
MN:~# pid=$!
HOST:~# iperf -c <ip_otherhost>
MN:~# kill $pid
```

More info here: [https://wiki.linuxfoundation.org/networking/tcp\\_testing](https://wiki.linuxfoundation.org/networking/tcp_testing)

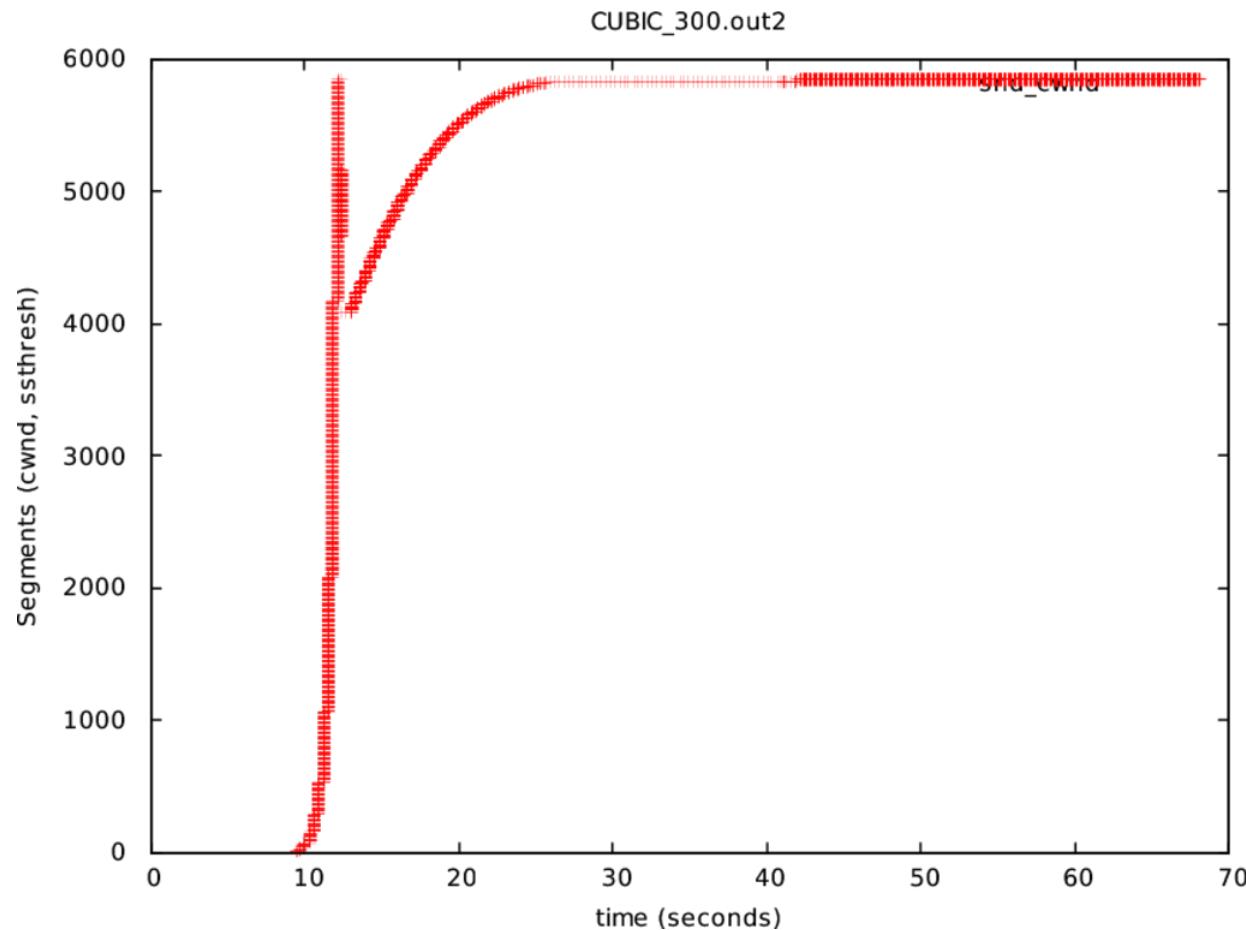
# Reno VS Cubic - cwnd

- Reno



# Reno VS Cubic - cwnd

- Cubic



# Cubic VS Packet Loss

Link S2->S3: Bandwidth 1Gbit/s – Delay 15ms – **Loss 1%**

# Cubic VS Packet Loss

Link S2->S3: Bandwidth 1Gbit/s – Delay 15ms – **Loss 1%**

[ 19]	48.00-49.00	sec	625 KBytes	5.12 Mbits/sec			
[ 19]	49.00-50.00	sec	940 KBytes	7.70 Mbits/sec			
[ 19]	50.00-51.00	sec	583 KBytes	4.77 Mbits/sec			
[ 19]	51.00-52.00	sec	301 KBytes	2.47 Mbits/sec			
[ 19]	52.00-53.00	sec	484 KBytes	3.96 Mbits/sec			
[ 19]	53.00-54.00	sec	402 KBytes	3.29 Mbits/sec			
[ 19]	54.00-55.00	sec	410 KBytes	3.36 Mbits/sec			
[ 19]	55.00-56.00	sec	631 KBytes	5.17 Mbits/sec			
[ 19]	56.00-57.00	sec	359 KBytes	2.94 Mbits/sec			
[ 19]	57.00-58.00	sec	481 KBytes	3.94 Mbits/sec			
[ 19]	58.00-59.00	sec	1.11 MBytes	9.34 Mbits/sec			
[ 19]	59.00-60.00	sec	930 KBytes	7.62 Mbits/sec			
[ 19]	60.00-60.04	sec	25.5 KBytes	4.74 Mbits/sec			
-----							
[ ID]	Interval		Transfer	Bandwidth	Retr		
[ 19]	0.00-60.04	sec	32.5 MBytes	4.55 Mbits/sec	234		sender
[ 19]	0.00-60.04	sec	32.3 MBytes	4.52 Mbits/sec			receiver

# Cubic VS Packet Loss VS Delay

Link S2->S3: Bandwidth 1Gbit/s – Delay 150ms – Loss 1%

# Cubic VS Packet Loss VS Delay

Link S2->S3: Bandwidth 1Gbit/s – Delay 150ms – Loss 1%

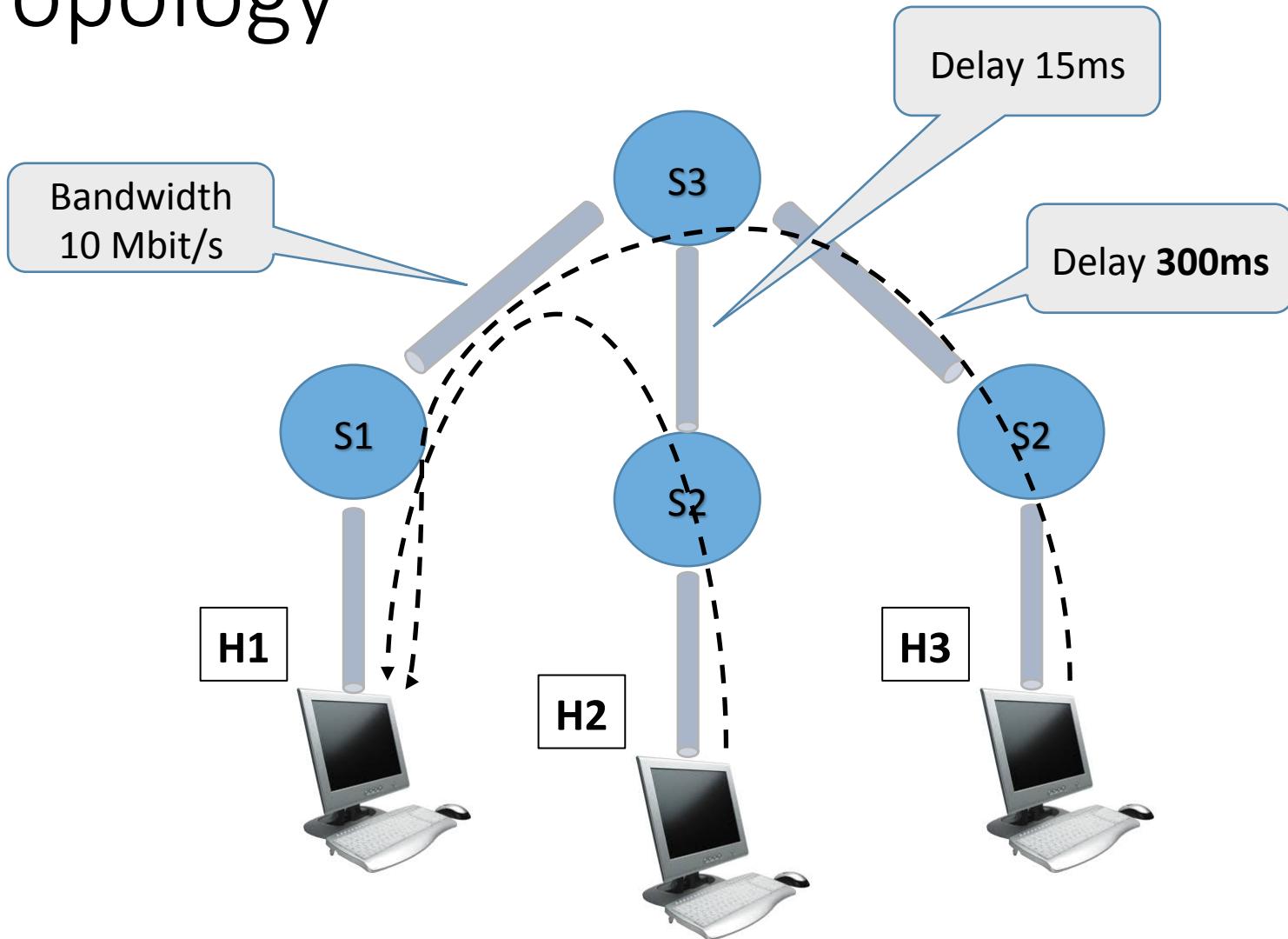
[ 19]	47.00-48.00	sec	49.5 KBytes	405 Kbits/sec			
[ 19]	48.00-49.00	sec	67.9 KBytes	556 Kbits/sec			
[ 19]	49.00-50.00	sec	53.7 KBytes	440 Kbits/sec			
[ 19]	50.00-51.00	sec	67.9 KBytes	556 Kbits/sec			
[ 19]	51.00-52.00	sec	116 KBytes	950 Kbits/sec			
[ 19]	52.00-53.00	sec	130 KBytes	1.07 Mbits/sec			
[ 19]	53.00-54.00	sec	96.2 KBytes	788 Kbits/sec			
[ 19]	54.00-55.00	sec	102 KBytes	834 Kbits/sec			
[ 19]	55.00-56.00	sec	110 KBytes	904 Kbits/sec			
[ 19]	56.00-57.00	sec	62.2 KBytes	509 Kbits/sec			
[ 19]	57.00-58.00	sec	141 KBytes	1.16 Mbits/sec			
[ 19]	58.00-59.00	sec	102 KBytes	834 Kbits/sec			
[ 19]	59.00-60.00	sec	102 KBytes	834 Kbits/sec			
[ 19]	60.00-60.31	sec	33.9 KBytes	897 Kbits/sec			
-----							
[ ID]	Interval		Transfer	Bandwidth	Retr		
[ 19]	0.00-60.31	sec	8.27 MBytes	1.15 Mbits/sec	58		sender
[ 19]	0.00-60.31	sec	7.87 MBytes	1.09 Mbits/sec			receiver

# Jain's fairness index

L'indice di fairness è una misura che indica se determinati utenti o applicazioni ricevono la stessa quantità di risorse del sistema

$$J(x_{\downarrow 1}, x_{\downarrow 2}, \dots, x_{\downarrow n}) = (\sum_{i=1}^n x_{\downarrow i})^2 / n \sum_{i=1}^n x_{\downarrow i}^2$$

# Topology



# FAIRNESS: Cubic VS RENO

Cubic: fairness **69%**

[ ID]	Interval	Transfer	Bandwidth	Retr	
[ 23]	0.00-184.06 sec	183 MBytes	8.32 Mbits/sec	6827	sender
[ 23]	0.00-184.06 sec	174 MBytes	7.93 Mbits/sec		receiver

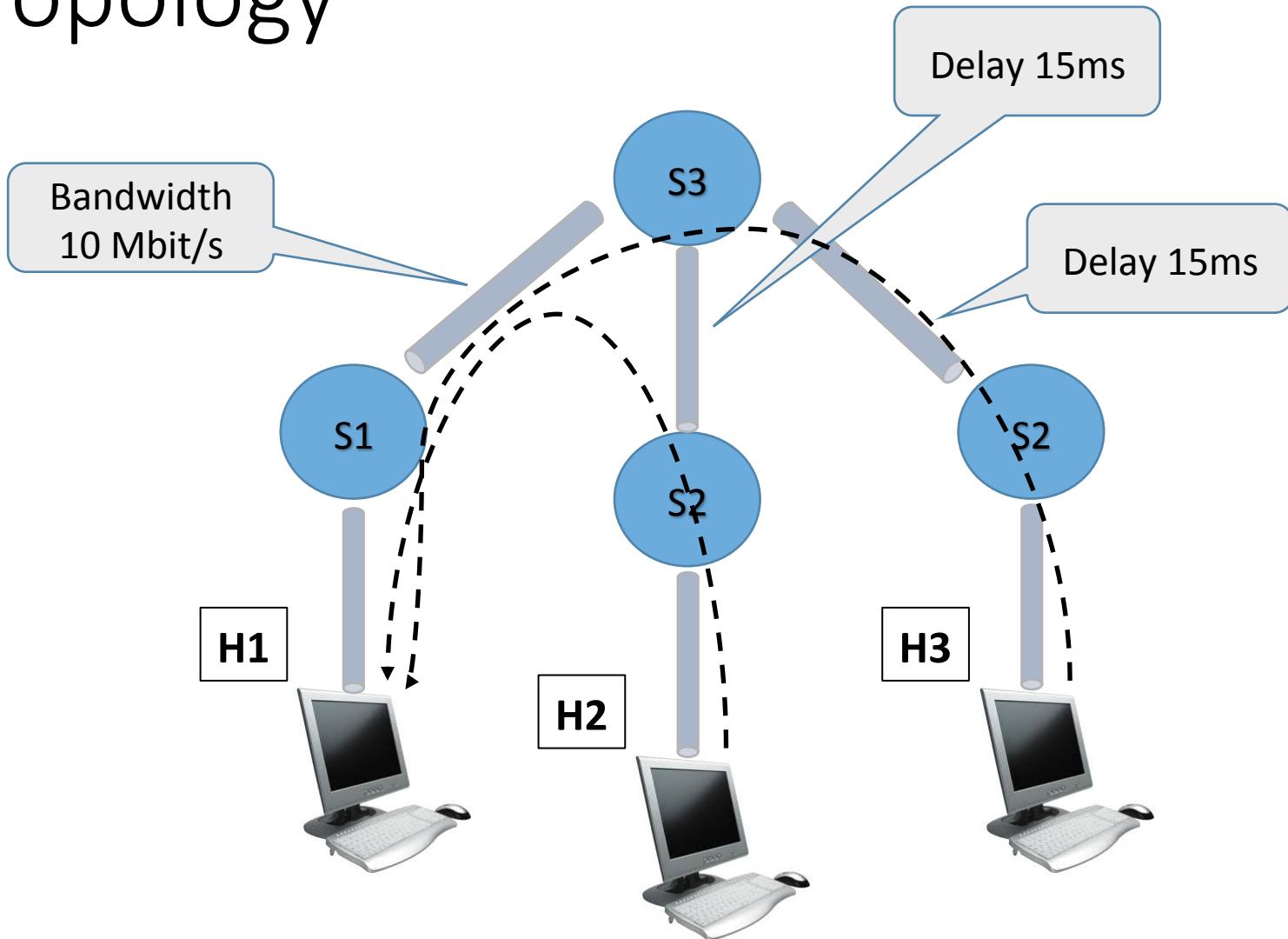
[ ID]	Interval	Transfer	Bandwidth	Retr	
[ 23]	0.00-181.87 sec	34.9 MBytes	1.61 Mbits/sec	3041	sender
[ 23]	0.00-181.87 sec	31.1 MBytes	1.43 Mbits/sec		receiver

Reno: fairness **51%**

[ ID]	Interval	Transfer	Bandwidth	Retr	
[ 23]	0.00-182.51 sec	199 MBytes	9.14 Mbits/sec	3377	sender
[ 23]	0.00-182.51 sec	191 MBytes	8.79 Mbits/sec		receiver

[ ID]	Interval	Transfer	Bandwidth	Retr	
[ 23]	0.00-180.61 sec	3.31 MBytes	154 Kbits/sec	12	sender
[ 23]	0.00-180.61 sec	3.11 MBytes	145 Kbits/sec		receiver

# Topology



# FAIRNESS: TCP VS UDP

- Server h1:  
`iperf -s`  
`iperf -s -u`
- Client h2:  
`iperf -c <ip_server> -t 60`
- Client h3:  
`iperf -c <ip_server> -t 60 -u -b 10M`

## TCP

[ ID]	Interval	Transfer	Bandwidth	Retr	sender
[ 23]	0.00-63.62 sec	4.27 MBytes	563 Kbits/sec	0	receiver
[ 23]	0.00-63.62 sec	4.07 MBytes	537 Kbits/sec		

## UDP

[ 23]	0.00-64.40 sec	71.4 MBytes	9.30 Mbytes/sec	13.029 ms	0/9141 (0%)
-------	----------------	-------------	-----------------	-----------	-------------