perf

bandwidth / throughput

bandwidth / data rate: maximum rate we can send per unit time

- most commonly measuring the speed of a link

1 gigabit/second = transmit 1 bit / nanosecond

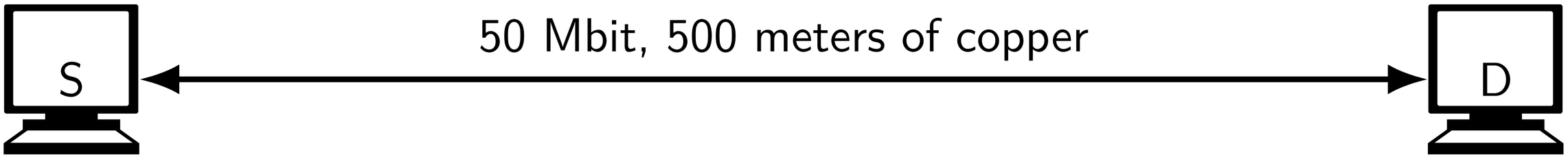
throughput: acheived rate per unit time

- often lower than total bandwidth because of losses
- (we'll give several examples throughout the semester)

latency (1)

latency: time for message: SOURCE $\rightarrow$ DEST

example: 1000 bit message from S to D:



one bit sent each 1/50M second = 0.02 μ s

1000 bits take $0.02 \times 1000 = 20$ μ s to sent

"transmission delay"

2.2 microseconds for bit to go down cable (2.3×10^8 m/s)

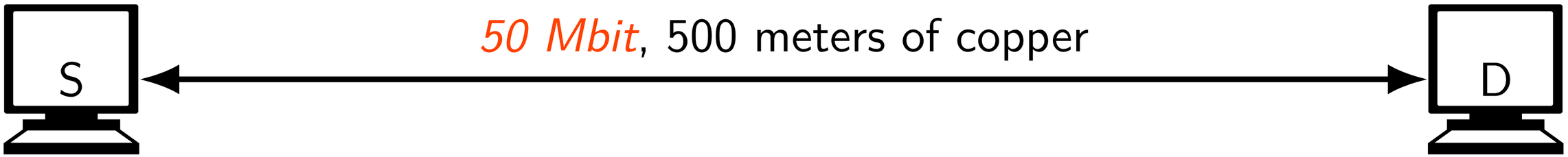
"propogation delay"

total latency of about 22.2 μ s

latency (1)

latency: time for message: SOURCE $\rightarrow$ DEST

example: 1000 bit message from S to D:



one bit sent each *1/50M second = 0.02 μ s*

1000 bits take $0.02 \times 1000 = 20$ μ s to sent

"transmission delay"

2.2 microseconds for bit to go down cable (2.3×10^8 m/s)

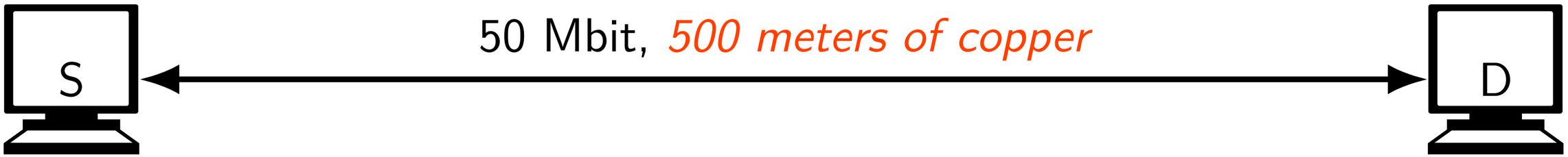
"propogation delay"

total latency of about 22.2 μ s

latency (1)

latency: time for message: SOURCE $\rightarrow$ DEST

example: *1000* bit message from S to D:



one bit sent each 1/50M second = 0.02 μ s

1000 bits take $0.02 \times 1000 = 20$ μ *s to sent*

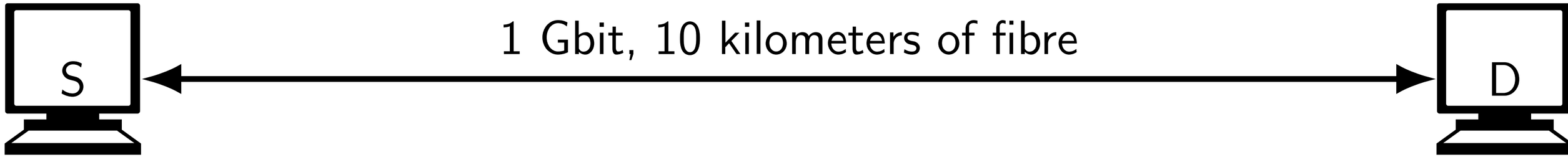
"transmission delay"

2.2 microseconds for bit to go down cable (2.3×10^8 m/s)

"propogation delay"

total latency of about 22.2 μ s

latency (1, ex)



exercise: latency for 20000 bit message from S to D

assume speed of signal through fiber of 2.0×10^8 m/s

latency (1, ex)

exercise: latency for 20000 bit message from S to D

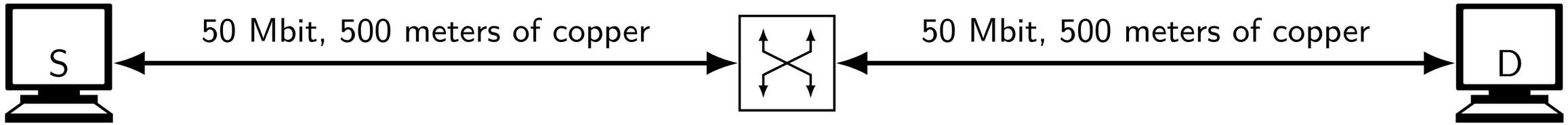
assume speed of signal through fiber of 2.0×10^8 m/s

latency (2)

example: 1000 bit packet from S to D

assume when message is received:

- 5 other 1000-bit packets in queue; no extra bits between packets
- no other switch processing time



S to switch, switch to D: 22.2 μ s (transmit+propogate delay)

within switch: wait $20 \times 5 = 100$ mu s for 5 other packets ($20\mu s$ = 1 packet transmit delay)

- "queueing delay"

total latency: $22.2 + 100 + 22.2 = 144.4$ microseconds

round trip time

round-trip-time (RTT): time for message:
SOURCE $\rightarrow$ DEST $\rightarrow$ SOURCE

much easier to measure than one-way latency

typically how we'll set latency

jitter

variation in latency

most commonly from changing queuing delays

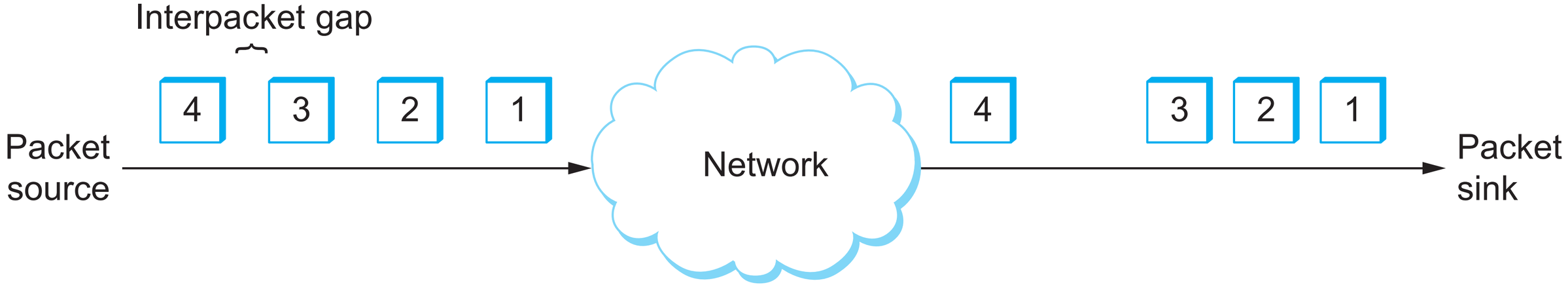


Figure 20 from Section 1.5 of *Computer Networks: A Systems Approach* (6th ed) (Peterson and Davie

measuring round-trip time (1a)

```
charles@reisst14$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=52 time=13.8 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=52 time=15.0 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=52 time=12.5 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=52 time=12.3 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=52 time=13.5 ms
64 bytes from 1.1.1.1: icmp_seq=6 ttl=52 time=12.5 ms
64 bytes from 1.1.1.1: icmp_seq=7 ttl=52 time=13.3 ms
64 bytes from 1.1.1.1: icmp_seq=8 ttl=52 time=13.2 ms
64 bytes from 1.1.1.1: icmp_seq=9 ttl=52 time=13.3 ms
64 bytes from 1.1.1.1: icmp_seq=10 ttl=52 time=14.1 ms
^C
--- 1.1.1.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 12.273/13.343/15.024/0.786 ms
```

measuring round-trip-time (1b)

No.	Time	Source	Destination	Protocol	Length	Info
17	2.766137597	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=1/256, ttl=
18	2.779916793	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=1/256, ttl=
28	3.768363948	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=2/512, ttl=
29	3.783346606	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=2/512, ttl=
59	4.769791044	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=3/768, ttl=
60	4.782213735	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=3/768, ttl=
73	5.771827936	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=4/1024, tt
74	5.784055865	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=4/1024, tt
79	6.773358205	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=5/1280, tt
80	6.786831460	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=5/1280, tt
81	7.775177274	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=6/1536, tt
82	7.787654160	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=6/1536, tt
85	8.776273952	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=7/1792, tt
86	8.789562086	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=7/1792, tt
93	9.777262659	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=8/2048, tt
94	9.790425188	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=8/2048, tt
110	10.778251280	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=9/2304, tt
119	10.791477471	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=9/2304, tt
120	11.779834642	172.25.188.87	1.1.1.1	ICMP	98	Echo (ping) request id=0x0002, seq=10/2560, t
121	11.793858285	1.1.1.1	172.25.188.87	ICMP	98	Echo (ping) reply id=0x0002, seq=10/2560, t

measuring round-trip-time (1c)

```
▸ Frame 17: 98 bytes on wire (784 bits), 98 bytes captured (784
▸ Ethernet II, Src: f4:6d:3f:d3:64:59 (f4:6d:3f:d3:64:59), Dst:
▸ Internet Protocol Version 4, Src: 172.25.188.87, Dst: 1.1.1.1
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xfa68 [correct]
    [Checksum Status: Good]
    Identifier (BE): 2 (0x0002)
    Identifier (LE): 512 (0x0200)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
    [Response frame: 18]
    Timestamp from icmp data: Sep  2, 2024 12:59:20.000000000 El
    [Timestamp from icmp data (relative): 0.093073545 seconds]
  ▾ Data (48 bytes)
      Data: 7f6b0100000000010111213141516171819 1a1b1c1d1e1f20
      [Length: 48]
```

non-ICMP pings (1)

```
HPING www (enp0s31f6 128.143.67.8): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=3.5 ms
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=3.2 ms
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=7.1 ms
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=6.8 ms
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=6.5 ms
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=5 win=0 rtt=6.2 ms
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=6 win=0 rtt=5.8 ms
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=7 win=0 rtt=5.4 ms
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=8 win=0 rtt=5.0 ms
len=46 ip=128.143.67.8 ttl=63 DF id=0 sport=0 flags=RA seq=9 win=0 rtt=4.7 ms
^C
--- www hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 3.2/5.4/7.1 ms
```

non-ICMP pings (2)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	128.143.71.27	128.143.67.8	TCP	54	1385 → 0 [<None>] Seq=1 Win=512 Len=0
2	0.000228953	128.143.67.8	128.143.71.27	TCP	60	0 → 1385 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
3	1.000254996	128.143.71.27	128.143.67.8	TCP	54	1386 → 0 [<None>] Seq=1 Win=512 Len=0
4	1.000555183	128.143.67.8	128.143.71.27	TCP	60	0 → 1386 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5	2.000547949	128.143.71.27	128.143.67.8	TCP	54	1387 → 0 [<None>] Seq=1 Win=512 Len=0
6	2.000861315	128.143.67.8	128.143.71.27	TCP	60	0 → 1387 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7	3.000765651	128.143.71.27	128.143.67.8	TCP	54	1388 → 0 [<None>] Seq=1 Win=512 Len=0
8	3.000975736	128.143.67.8	128.143.71.27	TCP	60	0 → 1388 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
9	4.000919160	128.143.71.27	128.143.67.8	TCP	54	1389 → 0 [<None>] Seq=1 Win=512 Len=0
10	4.001230544	128.143.67.8	128.143.71.27	TCP	60	0 → 1389 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
11	5.001235561	128.143.71.27	128.143.67.8	TCP	54	1390 → 0 [<None>] Seq=1 Win=512 Len=0
12	5.001548682	128.143.67.8	128.143.71.27	TCP	60	0 → 1390 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
13	6.001564520	128.143.71.27	128.143.67.8	TCP	54	1391 → 0 [<None>] Seq=1 Win=512 Len=0
14	6.001886342	128.143.67.8	128.143.71.27	TCP	60	0 → 1391 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	7.001812043	128.143.71.27	128.143.67.8	TCP	54	1392 → 0 [<None>] Seq=1 Win=512 Len=0
16	7.002151624	128.143.67.8	128.143.71.27	TCP	60	0 → 1392 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
17	8.002159545	128.143.71.27	128.143.67.8	TCP	54	1393 → 0 [<None>] Seq=1 Win=512 Len=0
18	8.002492960	128.143.67.8	128.143.71.27	TCP	60	0 → 1393 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
19	9.002512076	128.143.71.27	128.143.67.8	TCP	54	1394 → 0 [<None>] Seq=1 Win=512 Len=0
20	9.002826074	128.143.67.8	128.143.71.27	TCP	60	0 → 1394 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
21	10.002865037	128.143.71.27	128.143.67.8	TCP	54	1395 → 0 [<None>] Seq=1 Win=512 Len=0
22	10.003183918	128.143.67.8	128.143.71.27	TCP	60	0 → 1395 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

measuring throughput?

```
$ scp test.dat portal.cs.virginia.edu:test.dat
test.dat                          100%   32MB  23.0MB/s   00:01
$ scp portal.cs.virginia.edu:test.dat .
test.dat                          100%   32MB  28.2MB/s   00:01
```

(but might be measuring disk speed instead)

also more specialized tools like `iperf`

require program to run on both ends

measuring throughput

```
$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  1] local 128.143.71.87 port 5001 connected with 128.143.71.27 port 54760
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0147 sec  1.09 GBytes   934 Mbits/sec
```

```
$ iperf -c kytos02 | tee iperf.out
------------------------------------------------------------
Client connecting to kytos02, TCP port 5001
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  1] local 128.143.71.27 port 54760 connected with 128.143.71.87 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0256 sec  1.09 GBytes   933 Mbits/sec
```

measuring transmission delay?

```
$ ping -s 1000 www -i 0.05 -c 1000 -q
PING www.cs.virginia.edu (128.143.67.8) 1400(1428) bytes of data.
--- www.cs.virginia.edu ping statistics ---
1000 packets transmitted, 1000 received, 0% packet loss, time 50638ms
rtt min/avg/max/mdev = 0.319/0.461/1.222/0.039 ms
$ ping -s 16 www -i 0.05 -c 1000 -q
PING www.cs.virginia.edu (128.143.67.8) 16(44) bytes of data.
--- www.cs.virginia.edu ping statistics ---
1000 packets transmitted, 1000 received, 0% packet loss, time 50995ms
rtt min/avg/max/mdev = 0.156/0.345/1.539/0.068 ms
```

approx. $0.461 - 0.345 = 0.116$ ms delay for $1400 - 16$ extra bytes

with two links in each direction = approx $\frac{0.116}{4} = 0.029$ ms/link

$\frac{1400-16\text{byte}}{0.029\text{ms}} \approx 50$ Mbit/sec (does not match Gigabit ethernet)

probably other processing time besides sending on links, though

backup slides