Review from socket programming chapters:

UDP Socket Functions:

- socket(), bind(), recvfrom(), sendto(), close()

TCP Socket Functions:

- socket(), bind(), listen(), accept(), connect(), recv(), send(), read(), write(), close()

Other commonly used ones:

- setsockopt(), gethostbyname(), htons(), htonl(), ntohs(), ntohl(), bzero(), etc.

Debugging:

- return value, localhost, nc, tcpdump

Network Performance

- Bandwidth (aka throughput)

* A **bandwidth** of a network is the # of bits that can be transmitted over the network in a period of time
Throughput = TransferSize / TransferTime
* 100 Mbps => transfer 100 M bits per second
* can be defined for a single link or end-end channel

- Bandwith vs Throughput

* Literally, bandwidth is a measure of **width of frequency band**. It is measured in *hertz*
* When we talk about bandwidth of a communication link, it measured in ***bits per second*** (data rate)
* **Throughput** is often used to refer to the measured performance of a system
* Due to efficiency problems, throughput may be smaller than the bandwidth of a link (e.g., a link with 10 Mbps may achieve a throughput of 2 Mbps)

- Latency (aka delay)

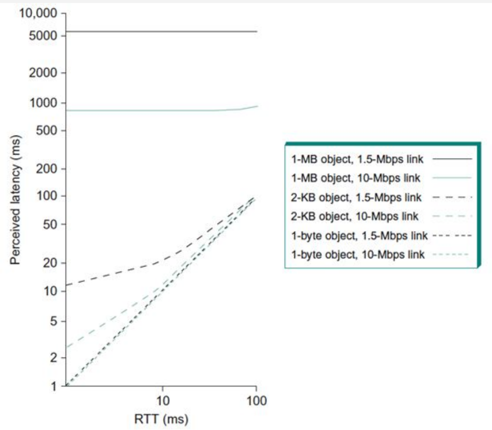
* Describes how long it takes a message to travel across the network; measured strictly in terms of time
* can be defined for a single link or end-end channel
* **Round Trip Time (RTT)** is more important sometimes

* Components of Network Latency:

Latency = Propagation + Transmit + Queue

Propagation = Distance / Speed of Light

Transmit = Size / Bandwidth



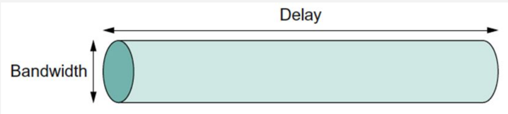
<= when object size is large, tramission delay dominates

<= when object size is medium, both transmission & propagation matter

<= when object size is small, propagration delay dominates

- Bandwith & Latency define the performace of a link/channel
* Latency dominates some apps (e.g., a client sends 1 byte to the server)
* Bandwidth dominates other apps (e.g., fetch a 1 GB file from a FTP server)
* Who dominates depends on the size of the data, the bandwidth of the link/channel, the latency of the link/channel

- Delay * Bandwidth…



* Think of channel as a hollow pipe; latency = length, and bandwidth = diameter of pipe
* *The product of the 2 gives the volume of the pipe*:
max # of bits that could be in transit through the pipe at a given instant

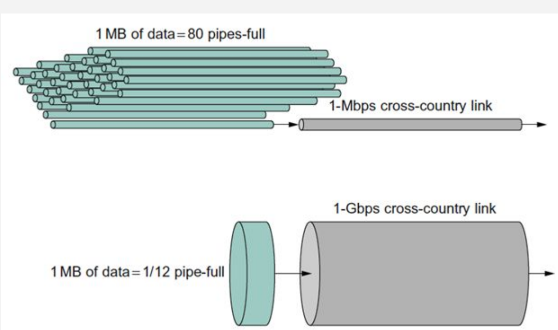
* For example, a channel with one-way latency of 100 ms and a bandwidth of 5 Mbps is able to hold:
100*10^(-3) s * 5*10^(6) bits/s = 500 K bits = 62.5 KB
* If the app considers the round-trip delay, the product is RTT*Bandwidth
200*10^(-3) s * 5*10^(6) bits/s = 1 M bits = 125 KB
* It corresponds to the number of bits a sender can transits before hearing something back from receiver
(most of the time, we use RTT in the delay * Bandwidth product)
(some rough RTT values for quick-dirty calculation: 100 ms for cross-country RTT; 1 ms for a local network)

Example: Transfer a 1 MB file over a 1 Mbps network vs a 1 Gbps network. RTT for both network is 100 ms.

Q: What will be the transfer time and throughput of the two networks?
TransferTime = RTT + TransferSize / Bandwidth
Throughput = TransferSize / TransferTime

For 1 Mbps:
Transfer time: 100 ms + 1 MB/1Mbps = 8.1 s
Throughput: 1 MB/8.1 s ~= 0.99 Mbps
For 1 Gbps:
Transfer time: 100 ms + 1 MB/1Gbps = 108 ms
Throughput: 1 MB/108 ms ~= 74.1 Mbps

Because of RRT, throughput of 1 Gbps is only 74 times (not 1000 times) higher than the throughput of 1 Mbps!

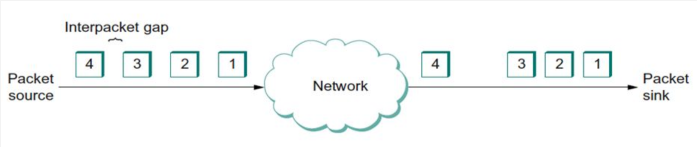


<= 1 MB of data is sent like a **stream** on the 1Mbps channel

<= 1 MB of data is sent like a **packet** on the 1Gbps channel

- Network Jitter

* Describes how much the **latency varies** from packet to packet
* Very important for time-sensitive apps such as streaming and video applications
* Understanding the lower and upper bounds of packet latency would help reduce jitter (e.g., app can delay the play of the first frame)



What's next on Computer Networks?

→ Ubiquitous networking
- Get internet access anywhere & anytime (e.g. planes and trains, etc.)
- Interconnect a great variety of devices (computers, mobile phones, sensors, etc.)

→ Scalability
- Support several orders of magnitude more devices
- Connect to large datacenters filled with tens of thousands processors and many petabytes of data (Cloud Computing)

→ Applications
- Smart Home, Smart City, Smart Planet
- Internet of Things