Lecture 2 : Performance Metrics

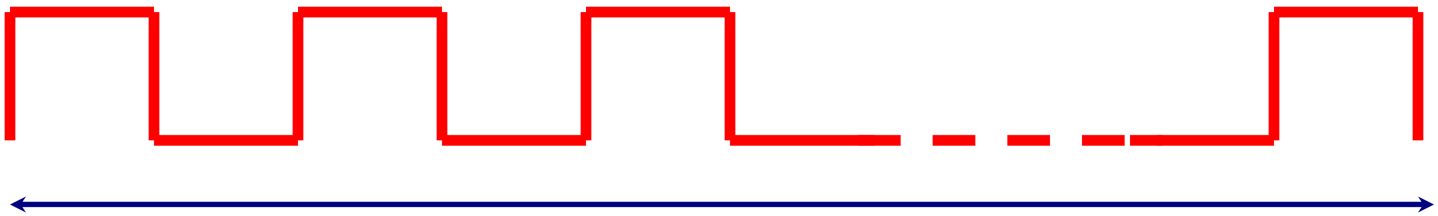
Hema Murthy
Professor
Dept. of CSE, IIT Madras

Short Term Course on “Teaching Computer Networks Effectively”. Sponsored by AICTE.

Measures of Performance

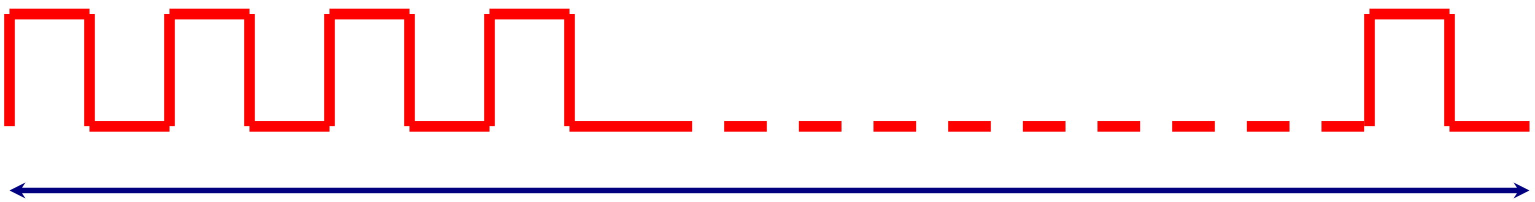
- Bandwidth
  - Aka throughput
  - Number of bits that can be transmitted
    - In a certain period
  - eg 100 Mbps, Gigabit ethernet
  - Sometimes, think of how long to transmit data
    - eg Gigabit ethernet takes 1 ns for each bit
- Latency
  - Time taken by a bit from end to another
  - Aka delay
  - Sometimes RTT is required
  - eg ping times

Width of Bit and Bandwidth



1 sec

Each bit – 1 μs wide



1 sec

Each bit – 0.5 μs wide

$\Rightarrow$ **Large Bandwidth**

Bandwidth and Throughput

- Confusing and (ab)used interchangeably
  - Number of bits/sec that can be sent = Bandwidth
  - Number of bits that are sent = Throughput
  - What’s possible vs what’s measured
- Bandwidth Requirements
  - Whatever network can give…
  - Fixed Number

Latency

- Three components
  - Speed of light propagation delay
    - Can be different in different media
    - eg only 2*10^8 m/s in fiber
  - Time taken to transmit a unit of data
    - f(network bandwidth, packet size)
  - Queue delays
- Propagation + Transmit + Queue delay = Latency
  - Propagation = Distance / Speed of light
  - Transmit = Size of packet/bandwidth

Performance Characteristics

- channel could be 1 Mbps / 100 Mbps
- Applications behave differently
  - across the continent
  - across the room
- Round trip time:
  - 1 Mbps - 100ms
  - 100 Mbps - 1ms

Performance Characteristics

- Example:
  - Channel Capacity: 1x10 Mbps
  - Datalength: 10 bits
  - Transmit time = 10 *microseconds*
  - Channel = 100 Mbps bits / sec
  - Transmit time = 0.010 *microseconds*

Performance Characteristics

- RTT = 100 *ms*, 1 *ms*
- Latency = $100+10 \times 10^{-3}$
- = 100.010 *ms*
- Latency = $1+10\times 10^{-3}$
- = 1.001 *ms*
  - Latency dominated by RTT.

Performance Metrics

- Large files
  - Image of size $25 \times 10^6 \times 8$ bits
  - Channel Capacity 10×10^6 bits/s
  - Time taken to transmit image 20 s
  - Suppose RTT = 1 ms
    - Latency = 20.001 sec
  - Suppose RTT = 100 ms
    - Latency = 20.1 sec
  - Bandwidth dominates latency

Bandwidth vs Latency

- What is more important?
  - Latency is more important than bandwidth at times
    - eg: Downloading files, course material
  - Bandwidth is also important at times
    - More bandwidth required to get images

Another Metric

- Computation is getting faster and faster
- Should computers wait for data?
- *Instructions per mile*

Performance Metrics

- Large Latency
  - Example: CPU = 200×10^6 instructions/s
  - Latency 100ms, for 5000 miles

$$200 \times 10^6 \quad - 1$$

$$? \quad - 0.1$$

$$\frac{200 \times 10^6 \times 0.1}{1} = 20 \times 10^6 \text{ instr / sec}$$

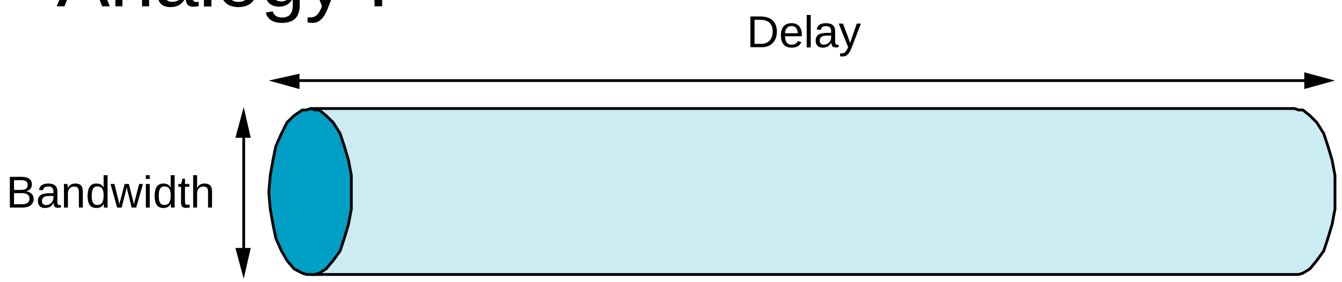
$$\Rightarrow \frac{20 \times 10^6}{5 \times 10^3} = 4000 \quad \text{instr /mile}$$

Performance Metrics

- 4000 instr / mile is lost
  - Is it worth going across network?
  - Ramification
    - Data centers, geographical spread, redundancy
- Bandwidth wasted
  - Solution
    - Treat the channel as pipe

Delay x Bandwidth Product

- Analogy :



- Product = Volume of the pipe
- Useful for :
  - Given the latency and the bit width (units of time), how many bits can the network hold?

Network as a Pipe

- Example
  - Latency - 50 ms
  - BW - 50 Mbps
- Pipe can hold
  - $50 \times 10^{-3} \times 50 \times 10^{6}$ bits of data
  - Bandwidth wasted if sender does not fill the pipe

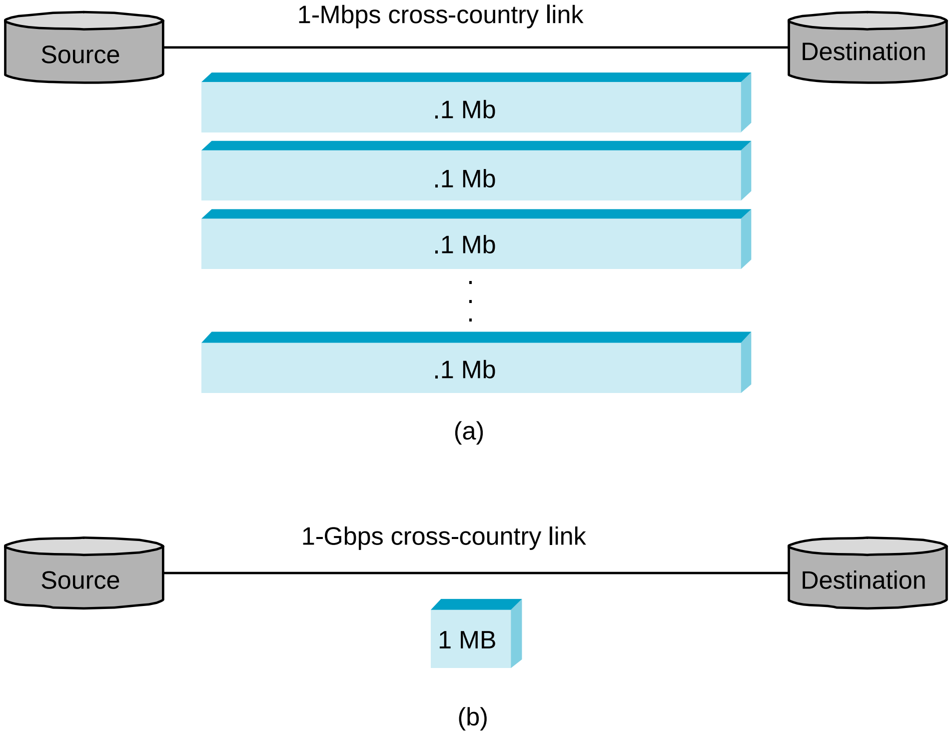
Inflight Data

- Useful for high speed networks
- Sender can send Bandwidth x Latency data before the first bit arrives at receiver
- Sender can send another Bandwidth x Latency data for the receiver to respond back all is well
- Total data *in-flight* = 2 * Bandwidth * Latency
- If the receiver says stop
  - expect latency * bandwidth data anyway to come in

High Speed Networks

- Eternally connected world
- Bandwidth is increasing continuously
  - Lay more cable !
- Latency is harder to improve though
  - Laws of physics

Fixed Latency Scenario



Eg : Let RTT = 100ms

Bandwidth vs Latency (contd.)

- Infinite bandwidth
  - In a single RTT, more data can be pushed across
- RTT itself becomes a problem now
  - 100 vs 101 RTTs
  - 1 vs 2 RTTs

Throughput

- Throughput:
  - Transfer Size / Transfer Time
- Transfer Time
  - RTT + (Transfer Size / BW)
- If RTT large, increase in BW does not reduce transfer time

Throughput

- Example:
  - Latency: 100ms
  - Channel Capacity: 1 Mbps, 1 Gbps
  - Data: 10 MB
  - On 1 Mbps channel
    - Time taken = 80.1s
  - On 1 Gbps channel
    - Time taken = 0.180s

Throughput

- Throughput for 1 Mbps channel
  - 80/80.1 Mbps = 99.87 Mbps ➔ very efficient → reaches channel capacity
- Throughput for 1 Gbps channel
  - 80/0.180 Mbps =444.4 Mbps ➔ very inefficient → very low compared to channel capacity

Application Performance Needs

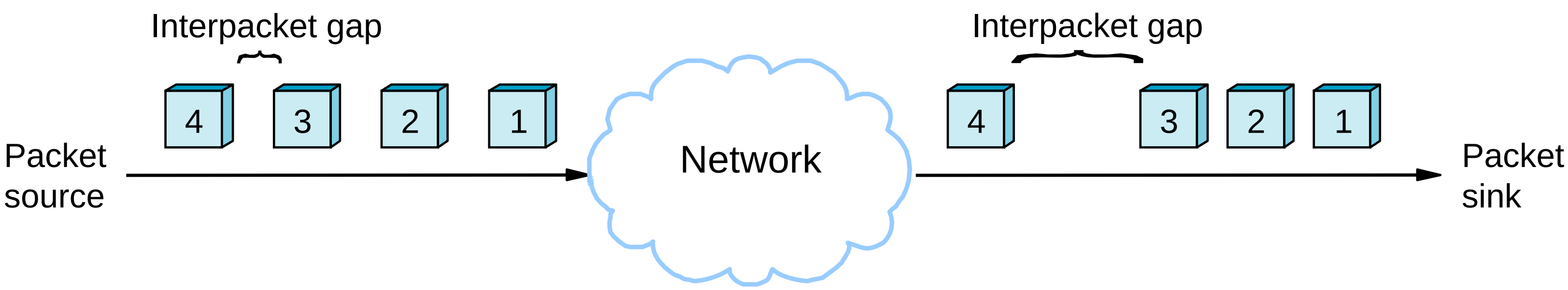
- Whatever you can give vs. fixed bandwidth requirement
- Average requirement is not a good metric
  - Bursts of requirements from the application
- Characterizing burst
  - Can I sustain peak rate for a fixed time
  - Number of bytes I can send at peak requirement before reverting back to average rate

Application Performance Needs (contd.)

- Delay can also be similar to bandwidth
  - As little delay as possible vs a fixed delay requirement
- Sometimes the one-way latency does not matter
  - Latency variation from packet to packet is a concern though
- Variation is called *jitter*
  - Can be concern to certain kinds of applications

Jitter

- Packets are generated every so many time units
  - Difference is called the inter-packet gap



- Fixed inter-packet gaps are good
- Variations can be a problem
  - Usually single links do not create considerable jitter
  - Multi-hop networks are the worst affected

Jitter Case Study

- Video – 30 frames a sec.
  - => a packet every 33 ms
- Receiver’s side : You need atleast one packet every 33 ms
- Two scenarios
  - Frame arrives early : Buffer it and play
  - Frame arrives late : Bad Quality video
- Should you remove jitter?
  - No
  - If jitter is known, you can work around it
  - Unknown jitter is the problem