Recitation 5

Zeinab Khalifa

September 24th, 2020

Carnegie Mellon University Qatar

Some network performance topics

Bandwidth (Capacity)

measured in bits/sec. Refers to the speed of bit transmission in a channel or a link.

Throughput

The actual data that is being transferred

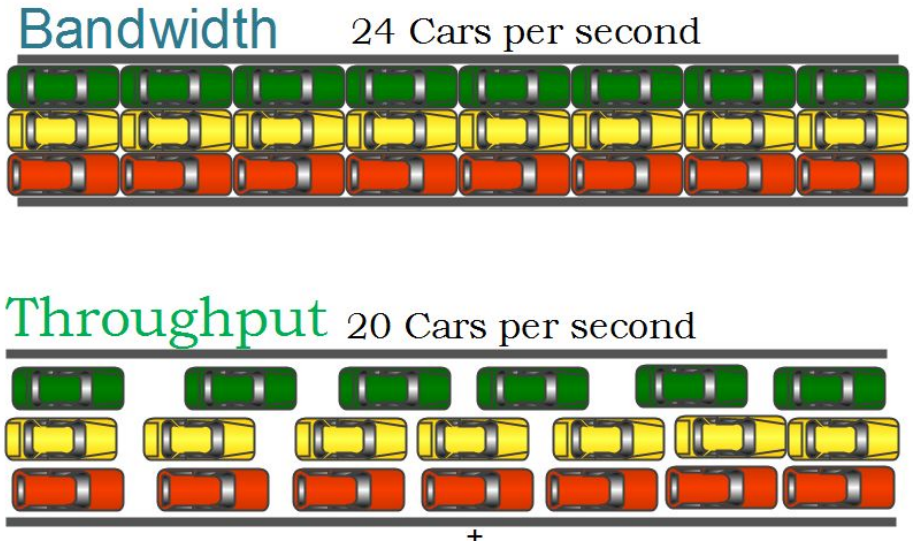


Fig1. Difference between throughput and bandwidth

Exercise (1)

A network with bandwidth of 10 Mbps can pass 12,000 frames per minute with each frame carrying 10,000 bits. What is the throughput of this network as a percentage of the bandwidth?

12,000 * 10,000 bits / 60 sec

=> ⅛ of the bandwidth

Propagation & Transmission

(LENGTH)

- **Propagation speed:**
  - The speed at which a **bit** travels through the medium from source to destination
  - Considers the medium capabilities

- **Propagation delay:**
  - Distance/ propagation speed
  - Usually is some fraction of the speed of light (optical fibre)

(WIDTH)

- **Transmission speed:**
  - The speed at which the bits of the entire message arrive from source to destination
  - Bandwidth/ Capacity considerations.

- **Transmission delay:**
  - Message size / bandwidth

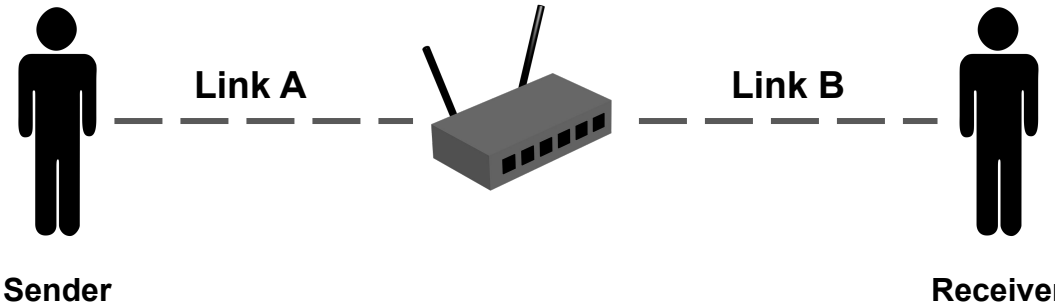
Exercise (2)

Calculate the propagation time needed for a bit to travel between two points with a connecting cable length of 12,000. Assuming that the cable's propagation speed is 2.4 * 10^8 m/s

12,000 * 1000 / (2.4 * 10^8) = 50 ms

Exercise (3.a)

Link	One-way propagation delay	Capacity
Link A	10ms	1Gbit/second
Link B	20ms	1Gbit/second

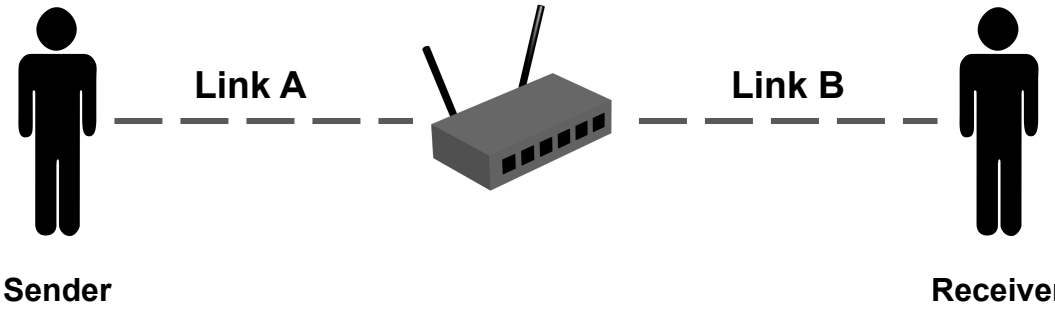


Sender **"ping"**s a receiver in a different country. Ping would send a packet of size 64 bytes. What is the minimum time that it will take the ping packet to return to the sender?

4 * 64 bytes * (8 bits / bytes) / (1073,741, 824 bits/s) + 60 ms

Exercise (3.b)

The speed of light = 299,792 km/s



Assuming that the sender is in China and the Receiver is in the U.S. the distance two parties equals 9300 km. What is the round-trip time ? 9300 km / 299792 km/s

Exercise (3.c)

Assume that you wanted to send a command to the receiver. To do so, you establish a TCP connection to port 80 and send an HTTP request with a particular format. What is the minimum latency between when you run your program locally to when the command is actually executed on the receiver's end?

Assume that the round-trip time is: 48.327ms

1.5 * 48.327 ms (1 for the TCP connection establishment and 0.5 for sending the command)

RPC Failures

Failures can be due to client or server crash or data transmission loss in the network.

Exercise (4) How would you handle the following situations?

If the client is unable to find the server?

The client should return an error message.

Exercise (4) How would you handle the following situations?

If client and server both are running but something goes wrong on the network and RPC request packet doesn't reach the server for processing or the server sent the reply back to client but client never received the reply?

Timeout mechanism. Resend the request if the client does not receive a response before the timeout

Exercise (4) How would you handle the following situations?

UDP failures, no error correction or handling of packet losses?

Timeout

Exercise (4)

What issues would occur if we implement UDP with non-idempotent calls? Explain.

If UDP is used to transfer data, before making a request to a stateful server one need to check if the RPC calls are idempotent, i.e. re-executing the calls on the server shouldn't cause errors.

RPC Server Failure Semantics

In scenarios where the server crashes while it is executing the RPC call, the client should be aware of what was the state of the server before it crashed, so that it will know what action to take when the server comes up.

RPC Server Failure Semantics *at-least-once*

- If the client received a reply from the server, it means that call has been executed at least once on the server.
- It may be possible server executed the call and then crashed before sending the reply, it then comes up and then again executes the call and sends the reply to the client.

RPC Server Failure Semantics *at-most-once*

- If the client gets a reply from the server, it means that the RPC call have been executed at most once on the server.

RPC Server Failure Semantics *exactly-once*

- This is the most desirable scenario, no matter what happened on the server side (crashes / restarts)
- If the client gets back reply exactly once, then it is confirmed that the RPC call was executed exactly once on the server (in the local-procedure call world, exactly-once is used)

Exercise (5)

Given the strength of the semantics: exactly once -> at most once -> at least once. Which RPC semantic would you choose for the below situations:

- Transferring money to your friend? *At most once (Easy to implement and can retry upon failures)*
- Posting a message on messenger? *At least once / at most once*
- Deleting a friend from social media? *At least once*