# 50.012 Networks (2020 Term 6)

# Homework 1

Hand-out: 22 Sep

Due: 1 Oct 23:59

Name: Poh Shi Hui Student ID: 1002921

**1.** Why two ISPs peer with each other? (Hint: consider the different types of peering: Regional ISP with Regional ISP, Tier 1 with Tier 1, and Regional with content provider) How does an IXP fit into this ecosystem? (Hint: study some IXP, e.g., https://www.sgix.sg/)

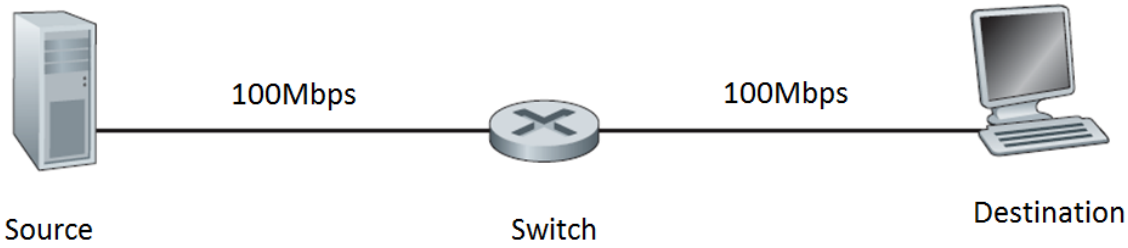
To allow interconnectability, basically to allow one ISP to exchange traffic with another ISP. (Peering) Example, Regional ISP provides internet to content provider which then provides access to content to those connected to Regional ISP - which can be another Regional ISP or a Tier 1.

IXP is an actual physical infrastructure that allows direct peering between ISPs of the same level to basically reduce costs for end users. Cheaper because direct peering uses shorter path by avoiding intermediary between them.

**2.** (Last year's mid-term exam question): Calculate the end-to-end delay (i.e., the duration from the first bit sent by the source to the last bit received by the destination) for a packet with size of 1500 bytes (12,000 bits) for the following:

12,000/100e6 = 0.00012 = 120µs

**2.1** 100 Mbps Ethernet (assume 1M = $10^6$) with a single store-and-forward switch (that is, with two links, one incoming and one outgoing, as shown in the Figure below) in the path. Assume that each link has a propagation delay of 12 µs (1 µs = $10^{-6}$s) and that the switch begins forwarding immediately after it has finished receiving the whole packet. Assume zero processing and queueing delay. 120 x 2 + 12 x 2 = **264µs**

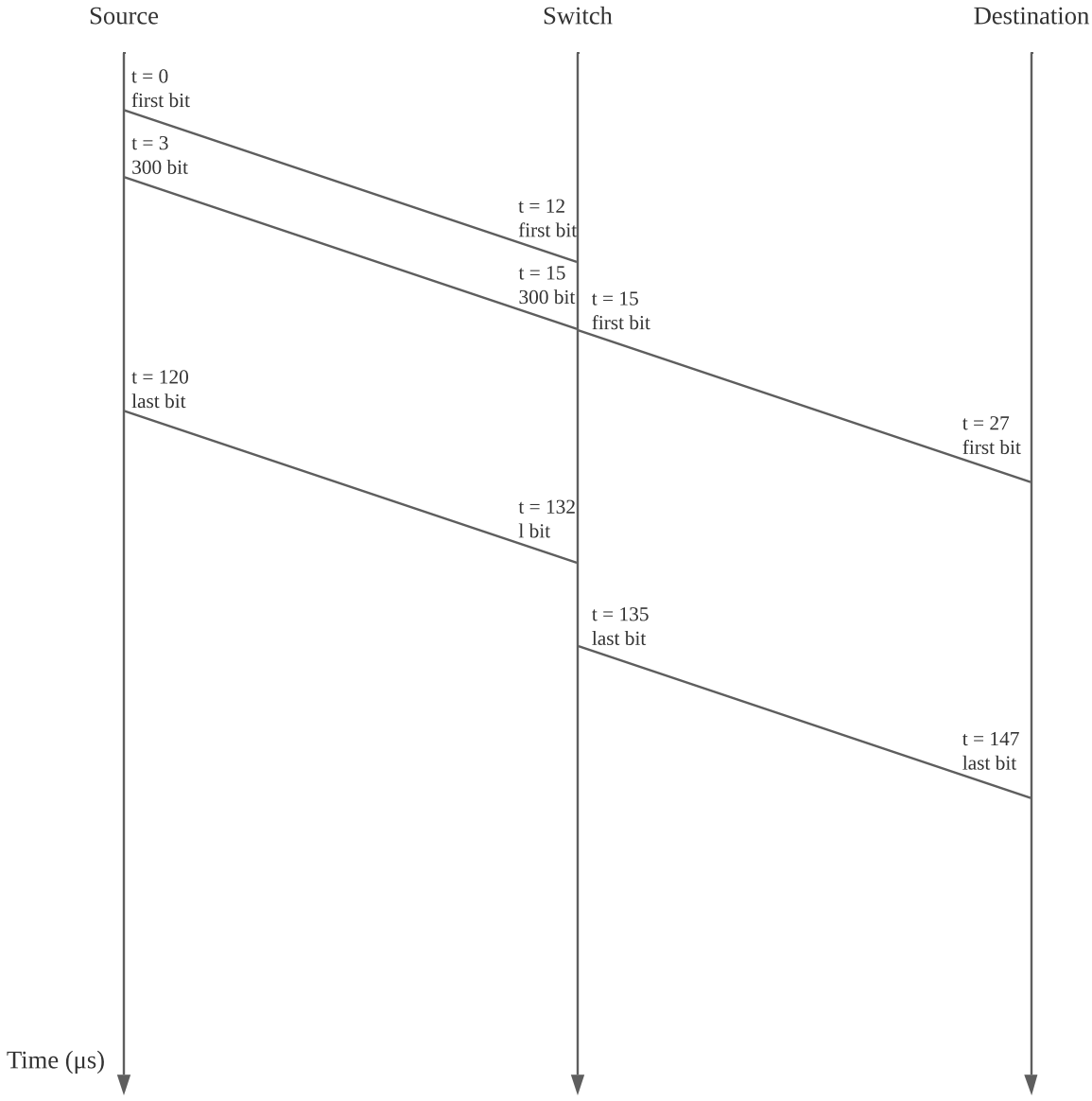



**Single switch case**

**2.2** Same scenario as **2.1** above, calculate the end-to-end delay when there are four switches (each switch and each link has the same setting as in scenario **2.1**) in the path. 120 x 5 + 12 x 5 = **660µs**

**2.3** Same scenario as **2.1** above, i.e., only a single switch, but assume the switch implements "cut-through" switching, i.e., the switch begins to forward the packet after the first 300 bits of the packet have been received. Calculate the end-to-end delay. 300/100e6 = 3µs 120 + 12 + 3 + 12 = **147µs**

**2.4** Draw a space-time diagram to depict the transfer in scenario **2.3** above.




Poh Shi Hui 1002921
Done using Lucid Chart
Insanely not drawn to scale...

**3.** (textbook chapter 2, review problem R26): In Section 2.7 of the textbook, the UDP server described needed only one socket, whereas the TCP server needed two sockets. Why? If the TCP server were to support n simultaneous connections, each from a different client host, how many sockets would the TCP server need?

UDP server does not need a welcoming socket, hence it only needs the one socket that handles incoming data from anywhere.

TCP needs a welcoming socket that waits for connection establishment requests and then creates another new socket that handles the incoming data, hence it needs 2 sockets.

For n simultaneous connections, needs **n + 1 sockets**.

**4.** (textbook problem chapter 2, problem 4) Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message (i.e., this is the actual content of an HTTP GET message). The characters <cr><lf> are carriage return and line-feed characters (that is, the italized character string <cr> in the text below represents the single carriage-return character that was contained at that point in the HTTP header). Answer the following questions, indicating where in the HTTP GET message below you find the answer.

GET /cs453/index.html HTTP/1.1<cr><lf>Host: gaia.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gecko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ext/xml, application/xml, application/xhtml+xml, text/html;q=0.9, text/plain;q=0.8, image/png,*/*;q=0.5<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-Encoding: zip,deflate<cr><lf>Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr><lf>Connection:keep-alive<cr><lf><cr><lf>

a. What is the URL of the document requested by the browser?

b. What version of HTTP is the browser running?

c. Does the browser request a non-persistent or a persistent connection?

d. What is the IP address of the host on which the browser is running?

e. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

**a)** http://gaia.cs.umass.edu/cs453/index.html
After GET and HOST

**b)** 1.1
Indicated in HTTP/1.1

**c)** Persistent
Requested connection is keep-alive

**d)** ? It's not in the message ?

**e)** Mozilla 5.0
In Agent
Different browsers handle data differently, so this helps the server send a version that the browser can handle.

**5.** (textbook problem chapter 2, problem 5)The text below shows the reply sent from the server in response to the HTTP GET message in the question above. Answer the following questions, indicating where in the message below you find the answer.

HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008 12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46 GMT<cr><lf>ETag: “526c3-f22-a88a4c80”<cr><lf>Accept-Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>Keep-Alive: timeout=max=100<cr><lf>Connection:Keep-Alive<cr><lf>Content-Type: text/html; charset=ISO-8859-1<cr><lf><cr><lf><!doctype html public “//w3c//dtd html 4.0 transitional//en”><lf><html><lf><head><lf> <meta http-equiv=”Content-Type”content=”text/html; charset=iso-8859-1”><lf> <meta name=”GENERATOR” content=”Mozilla/4.79 [en] (Windows NT 5.0; U) Netscape]”><lf> <title>CMPSCI 453 / 591 / NTU-ST550A Spring 2005 homepage</title><lf></head><lf> <much more document text following here (not shown)>

a. Was the server able to successfully find the document or not? What time was the document reply provided?

b. When was the document last modified?

c. How many bytes are there in the document being returned?

d. What are the first 5 bytes of the document being returned? Did the server agree to a persistent connection?

**a)** Yes. Tue, 07 Mar 2008 12:39:45GMT
200 OK indicates request successful and the reply time is indicated in Date

**b)** Sat, 10 Dec2005 18:27:46 GMT
indicated in Last-Modified

**c)** 3874 bytes
based on Content-Length

**d)** First 5 bytes are "<!doc". Yes.