50.012 Networks (2020 Term 6)

Homework 1

Hand-out: 21 Sep

Due: 2 Oct 23:59

Name: Victoria Yong Student ID: 1004455

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 ISP with content provider) What is the role of an IXP? How does an IXP generate revenue? (Hint: study some IXP, e.g., https://www.sgix.sg/)

ISPs peer with each other for business and technical reasons. ISPs are located in different geographical reasons. Peering allows for greater network availability to customers. Peering also increases network redundancy as it creates a better network topology, making their network service more resilient and reliable. Peering also affords ISPs greater routing control, which helps them manage network traffic better and reduce congestion along internet paths which would cause greater latency and packet loss due to insufficient bandwidth for customers. Most Tier 1 ISPs engage in settlement-free peering, in which both networks agree to transmit each other's data for free, with the underlying assumption that both networks are sending and receiving similar amounts of data.

IXPs are data centers housing many ethernet switches, interconnected across one or more buildings. They route traffic between the network operators who maintain the IXP. Internet infrastructure companies (i.e. ISPs, CDNs, cloud services etc.) connect to IXPs to exchange internet traffic. A

There are 5 types of business models for IXPs: Nonprofit organization, Association of ISPs, Operator-neutral for-profit company, University/Government Agency, Informal association of networks. Companies pay a monthly or annual fee to participate in the IXP services. Fees are usually

based on speed of port(s) the company is using. Some IXPs also charge a setup fee to offset hardware costs for the new participant. Since IXPs operate based on peering, when an IXP incurs operating costs, they will be shared among all its participants.

2. (2019'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 settings:

2.1 The source and the destination are connected via a store-and-forward switch (as in the figure below). Assume that each link in the figure has a propagation delay of 12 μs (1 μs = 10^{-6}s) and operates at 100Mbps (consider 1M = 10^6). The switch begins forwarding immediately after it finishes receiving the whole packet. Assume zero processing and queueing delay.
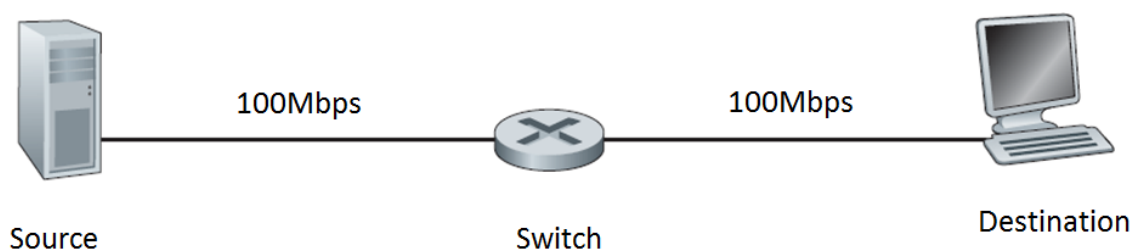

Single switch case

delay = d_{prop} + d_{trans} = 2(12x10^{-6}) + 2(12x10^3/100x10^6) = 0.000264s

2.2 Same scenario as **2.1** above, calculate the end-to-end delay when there are four switches chained together (each switch and each link has the same setting as in scenario **2.1**) in the path between the source and the destination.

Source → switch 1 → switch 2→ switch 3 → switch 4 → Destination

delay = d_{prop} + d_{trans} = 5(12x10^{-6}) + 5(12x10^3/100x10^6) = 0.00066s

2.3 Same scenario as **2.1** above, i.e., there is only a single switch between the source and the destination, 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. (Hint: Draw a space-time diagram.)
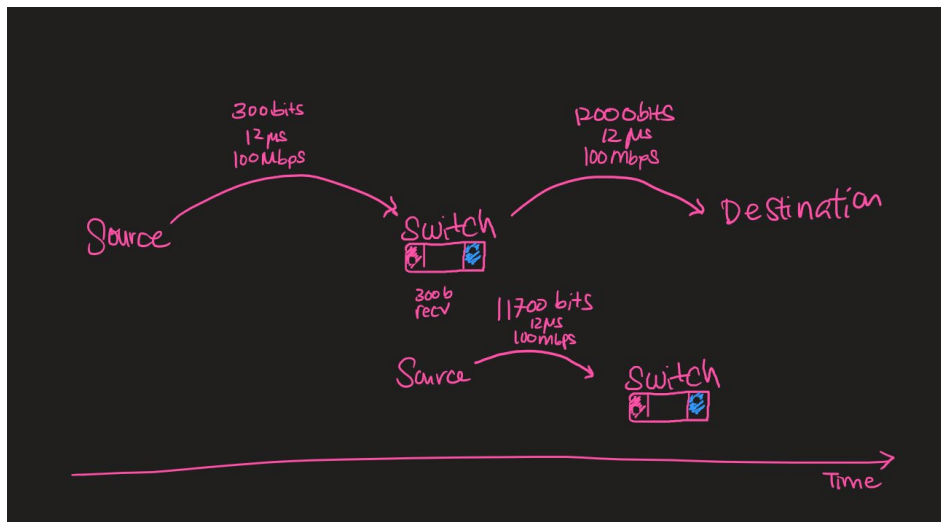
delay = d_{prop} + d_{trans}

= $2(12 \times 10^{-6}) + (300/100 \times 10^{6}) + (12 \times 10^{3}/100 \times 10^{6})$

= 0.000147s

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?

The UDP socket API allows one socket to send and receive data from many endpoints. Hence, multiple sockets are largely unnecessary when using UDP protocol. TCP socket API requires 2 sockets since each TCP connection is mapped to one socket and cannot receive data from multiple endpoints like UDP.

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?

   /cs453/index.html

   b. What version of HTTP is the browser running?

   HTTP 1.1

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

   persistent

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

   gaia.cs.umass.edu

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

   Mozilla Firefox 5.0

   Browser type is needed because

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?

      The server found the document successfully as indicated by 200 OK.

      The reply was given on Tue, 07 Mar 2008 at 12:39:45GMT

   b. When was the document last modified?

      Last-Modified: Sat, 10 Dec 2005 18:27:46 GMT

   c. How many bytes are there in the document being returned?
      3874

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

      charset=ISO-8859-1 indicates an 8bit encoding so the first 5 bytes of the document will contain CMPSC

      The server agreed to a persistent connection.

6. Recall the simple model for HTTP streaming shown in the Figure below. Let B denote the size of the client's application buffer, and let Q denote the number of bits that must be buffered before the client application begins playout. Also, let r denote the video consumption rate. Assume that the server sends bits at a constant rate x whenever the client buffer is not full.
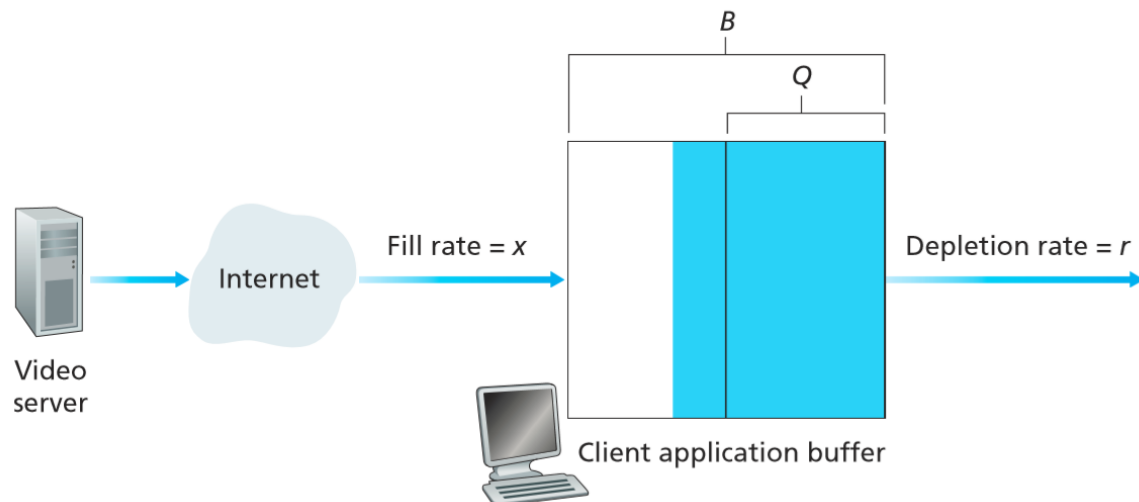
6.1 Suppose that $x < r$. In this case playout will alternate between periods of continuous playout and periods of freezing. Determine the length of each continuous playout and freezing period as a function of Q, r, and x.

$T_{playout} = Q/(r-x)$ seconds

$T_{freeze} = Q/x$ seconds

6.2 Now suppose that $x > r$. At what time does the client application buffer become full?

$T_{full} = (B-Q)/(x-r)$

7. (2019's midterm exam question) Consider distributing a file of $F = 6 \times 10^9$ bits to N=100 peers. The server has an upload rate of u_s = 30 Mbps, and each peer has a download rate of d_i = 2 Mbps and an upload rate of u_i=1 Mbps. Assume $1M = 10^6$. Calculate the minimum distribution time (i.e., to let every peer have a copy of the file) for:

7.1 the client-server distribution mode, and

7.2 the P2P distribution mode.

$T_{\text{client-server}}$ = max{ $100(6 \times 10^9/30 \times 10^6)$, $(6 \times 10^9/2 \times 10^6)$ }

$\qquad$ = max{ 20000, 3000}

$\qquad$ = 20 000 s

$\qquad$ = 5.5 hours

T_{P2P} = max{ $(6 \times 10^9/30 \times 10^6)$, $(6 \times 10^9/2 \times 10^6)$, $100(6 \times 10^9/(30 \times 10^6 + 100 \times 10^6)$ }

$\qquad$ = max{ 200, 3000, 4615 }

$\qquad$ = 4615 s

$\qquad$ = 1.28 hours

=== END ===