**ISE 316 Homework #1 – Due Feb 10 at 11:59 PM**

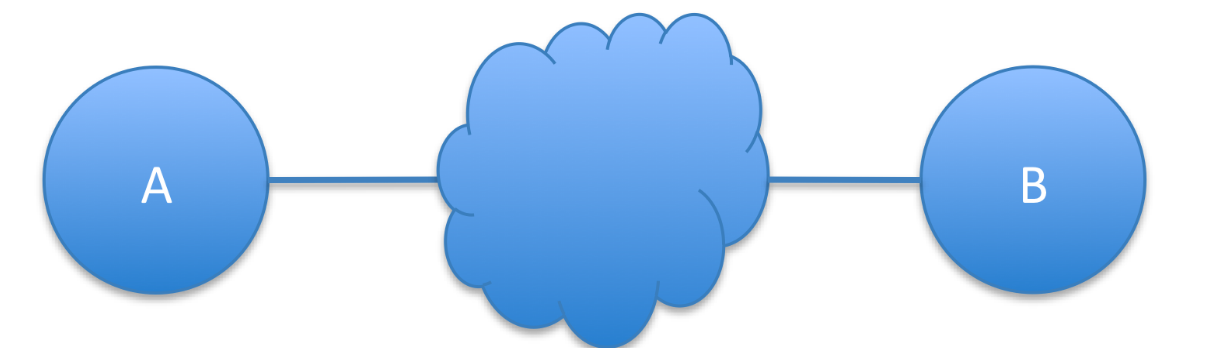
**Introduction to Networking**

**Spring 2023**

**Computer Science Department**

**Stony Brook University**

1. (3 pts) Suppose that server A is sending client B packets of size 2000 bits through router R. Also assume that the link is 1Gb/s and the rate at which packets arrive at router R is 450,000 packets per second. What is the average queueing delay at this link?
2. (3 pts) Suppose we need to send a file from A to B. The file is segmented into K packets each of B bits. The route from A to B crosses over N links, where each link has a bandwidth of R bps. How long does it take to send the file from source to destination?
3. (6 pts) Consider the network shown below



1. If most network applications transmitted data at continuous rates and for long time periods, would circuit or packet switching be preferred? Why
2. Suppose all links between A and B have a transmission rate of 10 Mbps and can handle up to 10 simultaneous connections. If A and B have established a connection, what will be their maximum transmission rate?
3. Now assume that packet switching is being used instead of circuit switching. If A and B are sending data on the network and the network is idle, what will be their transmission rate?
4. (8 pts) Assume we have the following network where each link is 100 m long and has a transmission rate of 1Gbps. Assume that the propagation speed is 2 \* 10^8 m/s and all packets size is 1000 B.

Diagram

Description automatically generated with medium confidence

1. How long will a packet sent from A to get to B take?
2. Assume host A generates 50 packets instantly. How long will the first packet wait in A’s queue before being transmitted? How long will the 50th packet wait in A’s queue before being transmitted? You can assume that no other packets are sent.
3. Suppose now that host A closed the previous connection and starts sending 300 packets every 0.1s. What is the throughput it achieves, assuming no packet losses?
4. If the transmission rate of the link between Y and Z changed to 10 Mbps. What is the maximum throughput that host A can achieve? If A tries to transmit packets at a higher rate, what will happen? Explain.
5. **(10 pts) Wireshark Exercise**:

The basic tool for observing the messages exchanged between executing protocol entities is called a **packet sniffer**. As the name suggests, a packet sniffer captures (“sniffs”) messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and

protocols running on your computer, but never sends packets itself. Similarly, received

packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer

receives a *copy* of packets that are sent/received from/by application and protocols

executing on your machine.

Figure 1 shows the structure of a packet sniffer. At the right of Figure 1 are the protocols

(in this case, Internet protocols) and applications (such as a web browser or ftp client)

that normally run on your computer. The packet sniffer, shown within the dashed

rectangle in Figure 1 is an addition to the usual software in your computer, and consists

of two parts. The **packet capture library** receives a copy of every link-layer frame that

is sent from or received by your computer. Recall from the discussion from section 1.5 in

the text that messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are

transmitted over physical media such as an Ethernet cable. In Figure 1, the assumed

physical media is an Ethernet, and so all upper layer protocols are eventually

encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all

messages sent/received from/by all protocols and applications executing in your

computer.

The second component of a packet sniffer is the **packet analyzer**, which displays the

contents of all fields within a protocol message. In order to do so, the packet analyzer

must “understand” the structure of all messages exchanged by protocols. For example,

suppose we are interested in displaying the various fields in messages exchanged by the

HTTP protocol in Figure 1. The packet analyzer understands the format of Ethernet

frames, and so can identify the IP datagram within an Ethernet frame. It also understands

the IP datagram format, so that it can extract the TCP segment within the IP datagram.

Finally, it understands the TCP segment structure, so it can extract the HTTP message

contained in the TCP segment. Finally, it understands the HTTP protocol and so, for

example, knows that the first bytes of an HTTP message will contain the string “GET,”

“POST,” or “HEAD”.

**Diagram

Description automatically generated**

We will be using the Wireshark packet sniffer [ <http://www.wireshark.org/> ] for our homework, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack.  **Download the portable version of Wireshark to your machine.**

Running Wireshark

When you run the Wireshark program, the Wireshark graphical user interface shown in

Figure 2 will be displayed. Initially, no data will be displayed in the various windows.

The Wireshark interface has five major components:

* The **command menus** are standard pulldown menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data and exit the Wireshark application. The Capture menu allows you to begin packet capture.

* The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is *not* a packet number contained in any protocol’s header), the time at which the packet was captured, the packet’s source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

* The **packet-header details window** provides details about the packet selected (highlighted) in the packet listing window. (To select a packet in the packet listing window, place the cursor over the packet’s one-line summary in the packet listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/received over an Ethernet interface) and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the plus-or-minus boxes to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.
* The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
* Towards the top of the Wireshark graphical user interface, is the **packet display filter field,** into which a protocol name or other information can be entered in order to *filter* the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we’ll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

A picture containing graphical user interface

Description automatically generated

**Answer the following questions:**

Open the provided simulated trace file (introduction.pcap) with Wireshark, and answer the following questions based on your Wireshark experimentation:

* 1. List 10 different protocols that appear in the protocol column in the unfiltered packet-listing window.

To make it convenient, I just picked the first 10 :)

* No. Time Source Destination Protocol Length Info
* 1 0.000000 128.118.102.62 146.186.52.244 SMB 208 NT Create AndX Request, Path: \XXF2\Private\My Documents\icons
* 2 0.000273 146.186.52.244 128.118.102.62 SMB 93 NT Create AndX Response, FID: 0x0000, Error: STATUS\_OBJECT\_NAME\_NOT\_FOUND
* 3 0.001023 128.118.102.62 146.186.52.244 SMB 196 NT Create AndX Request, FID: 0x40aa, Path: \XXF2\Private\My Documents
* 4 0.001274 146.186.52.244 128.118.102.62 SMB 193 NT Create AndX Response, FID: 0x40aa
* 5 0.001736 128.118.102.62 146.186.52.244 SMB 204 Trans2 Request, FIND\_FIRST2, Pattern: \XXF2\Private\My Documents\icons
* 6 0.002035 146.186.52.244 128.118.102.62 SMB 126 Trans2 Response, FIND\_FIRST2, Error: STATUS\_NO\_SUCH\_FILE
* 7 0.002206 128.118.102.62 146.186.52.244 SMB 99 Close Request, FID: 0x40aa
* 8 0.002373 146.186.52.244 128.118.102.62 SMB 93 Close Response, FID: 0x40aa
* 9 0.002450 128.118.102.62 146.186.52.244 SMB 202 Trans2 Request, FIND\_FIRST2, Pattern: \XXF2\Private\My Documents
* 10 0.002721 146.186.52.244 128.118.102.62 SMB 246 Trans2 Response, FIND\_FIRST2, Files: My Documents
  1. Try the filter function: type “http”, then press “Apply” button. How long did it take from when the first HTTP GET message was sent until the first HTTP OK reply was received? (By default, the value of the Time column in the packet listing window is the amount of time in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark *View* pull down menu, then select Time *Display Format*, then select *Time-of-day*.)

No. Time Source Destination Protocol Length Info

141 10:24:15.093032 128.118.102.62 128.119.245.12 HTTP 692 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1

145 10:24:15.149546 128.119.245.12 128.118.102.62 HTTP 440 HTTP/1.1 200 OK (text/html)

As shown above, the difference delta is 149546 – 93032 = 56514 ms.

* 1. The HTTP server being used is named gaia.cs.umass.edu. What was the Internet address (IP) of the HTTP server? What was the port number of the server? What was the IP address of the client? What was the port number for the client? You may need to click the + sign to expand the information in the detailed packet content window.
* IP v4, (Client IP)Src: 128.118.102.62, (Server IP)Dst: 128.119.245.12
* (Client Port) Source Port: 53826 , (Server Port) Destination Port: 80
  1. There are two HTTP GET request messages. The first one was successful in getting some text data back from the server. What was the line of text data that was received? Congratulations! You've downloaded the first Wireshark lab file!\n
  2. Refer to the five layers of the Internet protocol stack: application, transport, network, link, and physical. Write down which protocol was used, respectively, in the first four layers (ignoring the physical layer) with the HTTP server?

-For application (presentation included as well), it was hyper-text transfer protocol

-For transport, it was transmission control protocol

-For network, it was IPv4

-For link, it was Ethernet II