Communication Protocols and Internet Architectures
Harvard University
CSCI S-40, Summer 2018

Homework Assignment #3 Solutions

27 Points Total

Question 1) 6 points total

Consider an IP network composed of two networks …. ….. see assignment for complete problem.

PART A. For simplicity, this answer assumes that the layer-5 application's data fits nicely inside of one UDP datagram, which in turn fits into one IP datagram, which in turn fits into one Ethernet frame.

(1) Host A's application passes its data off to the layer-4 UDP module where it is encapsulated into a UDP packet with a UDP header. That UDP header includes destination and source ports for the application's data, the total length of the UDP datagram, and, potentially, a checksum (which covers the datagram as well as a pseudo-header).

(2) The UDP module then passes the UDP datagram off to the layer-3 IP module, where it is again encapsulated, this time into an IP datagram with an IP header. (There is now an IP datagram that contains a UDP datagram as its payload.) The IP header has a number of fields, including: the version of IP being used, the length of the IP header, a TOS setting, the total length of the IP datagram, identification and fragmentation information, a TTL, protocol information, a checksum, source and destination IP addresses for the application's data, and possibly some options. Note that the destination IP address is whatever was specified by the layer-5 application, in this case 13.0.0.4.

(3) The IP layer sees that the destination IP address is on the same network as itself. It knows this because it is on network 13.0.0.0/8, and the 8 most significant bits of the destination IP address (13.0.0.4) match the 8 most significant bits of the local host address (13.0.0.3). Therefore, the datagram can be delivered directly without the use of an intermediate router. But first a destination MAC address must be determined to correctly build the layer 2 Ethernet frame, so the destination IP address (13.0.0.4) is passed to the ARP module in a request for the matching MAC address.

(4) The ARP module looks in its cache of recently used IP/MAC address mappings (the ARP cache), and since this is the first packet ever sent from Host A to Host B, there is no entry in the cache for Host B. So the ARP module constructs an ARP request frame, requesting the MAC address corresponding to Host B's IP address (13.0.0.4) and passes this frame to the physical layer. (Note, the ARP module does not fit neatly into either layer 3 or layer 2. Consider it as "layer 2.5" or as a small module parallel to layers 2 and 3.)

(5) The physical layer broadcasts the ARP request to the local network, and Host B replies with an ARP reply message which provides its own MAC address (MAC_B). The router also receives the ARP request (because it was sent to the broadcast address), but since it does not have an IP address of 13.0.0.4 it does not reply.

(6) The ARP module makes a new entry in its ARP cache, mapping the IP address of Host B to the MAC address of Host B, and replies to the request in step 3 above by providing the MAC address MAC_B.

(7) The IP module then passes the IP datagram off to the layer-2 Ethernet module, where it is yet again encapsulated, this time into an Ethernet frame with an Ethernet header and trailer. (So now there's an Ethernet frame whose payload is an IP datagram whose payload is the UDP packet whose

payload is the data from the application. This is the "nested envelopes" idea.) This header and trailer contain a preamble, destination (MAC_B) and source (MAC_A) Ethernet addresses, specification of the frame's type, and a CRC (in the trailer).

(8) The Ethernet frame is then put out on the wire and is heard by both host B and R1. But since the frame is addressed to host B's MAC address, only host B scoops the frame off the wire and proceeds to process it through his own protocol stack.

PART B.

(1) Same as in 1a.
(2) Same as in 1a.

(3) The IP layer sees that the destination IP address is not on the same network as itself. It knows this because it is on network 13.0.0.0/8, and the 8 most significant bits of the destination IP address (13.0.0.4) do not match the 8 most significant bits of the local host address (18.0.0.8). So the IP layer passes the destination IP address (18.0.0.8) to its routing table, and sees that it must send the packet to the router at 13.0.0.1. But first a destination MAC address must be determined to correctly build the layer 2 Ethernet frame, so the router's IP address (13.0.0.1) is passed to the ARP module in a request for the matching MAC address. [NOTE: The destination IP address in the IP datagram is still 18.0.0.8, but the Ethernet frame containing this datagram will be sent to the MAC address of the router R.]

(4) The ARP module looks in its cache of recently used IP/MAC address mappings (the ARP cache), and since this is the first packet ever sent from Host A to Router R, there is no entry in the cache for R. So the ARP module constructs an ARP request frame, requesting the MAC address corresponding to R's IP address (13.0.0.1), and passes this frame to the physical layer. (Note, the ARP module does not fit neatly into either layer 3 or layer 2. Consider it as "layer 2.5" or as a small module parallel to layers 2 and 3.)

(5) The physical layer broadcasts the ARP request to the local network, and Host R replies with an ARP reply message which provides the MAC address (MAC_R1) of Host R's connection to network 13.0.0.0/8.). Host B also receives the ARP request (because it was sent to the broadcast address), but since it does not have an IP address of 13.0.0.1 it does not reply.

(6) The ARP module makes a new entry in its ARP cache, mapping the IP address of Host R (13.0.0.1) to the MAC address of Host R (MAC_R1), and replies to the request in step 3 above by providing the MAC address MAC_R1.

(7) The IP module then passes the IP datagram off to the layer 2 Ethernet module, where it is yet again encapsulated, this time into an Ethernet frame with an Ethernet header and trailer. This header and trailer contain a preamble, destination (MAC_R1) and source (MAC_A) Ethernet addresses, specification of the frame's type, and a CRC (in the trailer). [Remember, the destination IP address in the IP datagram is still 18.0.0.8 (Host D), but the Ethernet frame is addressed to MAC_R1 (the router), not directly to Host B.]

(8) The Ethernet frame is then put out on the wire and is heard by both host B and R1. But since the frame is addressed to R1's MAC address, only the router scoops the frame off the wire and proceeds to process it through his own protocol stack.

(9) After stripping off the Ethernet frame's header and trailer, the router passes the IP payload off to its layer-3 IP module.

(10) The router then examines the destination IP address (18.0.0.8) specified in the IP datagram and does a lookup in its routing table for that address. It determines that hosts having IP

addresses that start with 18 are directly connected to interface R2. It knows this because its interface R2 is on network 18.0.0.0/8, and the 8 most significant bits of the destination IP address (18.0.0.8) match the 8 most significant bits of the R2 address (18.0.0.1). But first a destination MAC address must be determined to correctly build the layer 2 Ethernet frame, so the destination IP address (18.0.0.8) is passed to the ARP module in a request for the matching MAC address.

(11) The ARP module looks in its cache of recently used IP/MAC address mappings (the ARP cache), and since this is the first packet ever sent from Host R to Host D, there is no entry in the cache for Host D. So the ARP module constructs an ARP request frame, requesting the MAC address corresponding to Host D's IP address (18.0.0.8) and passes this frame to the physical layer. (Note, the ARP module does not fit neatly into either layer 3 or layer 2. Consider it as "layer 2.5" or as a small module parallel to layers 2 and 3.)

(12) The physical layer broadcasts the ARP request to the local network, and Host D replies with an ARP reply message which provides the MAC address (MAC_D) of Host D. Host C also receives the ARP request (because it was sent to the broadcast address), but since it does not have an IP address of 18.0.0.8 it does not reply.

(13) The ARP module makes a new entry in its ARP cache, mapping the IP address of Host D (18.0.0.8) to the MAC address of Host D (MAC_D), and replies to the request in step 10 above by providing the MAC address MAC_D.

(14) And so the router passes the datagram off to its R2 Ethernet module, where it is re-encapsulated into an Ethernet frame with an Ethernet header and trailer. This header and trailer contain a preamble, destination (MAC_D) and source (MAC_R2) Ethernet addresses, specification of the frame's type, and a CRC (in the trailer). Note that the source IP address in the IP datagram is still Host A (13.0.0.3), but the source MAC address in the Ethernet frame is MAC_R2.

(15) The Ethernet frame is then put out on the wire and is heard by both Hosts C and D. But since the frame is addressed to host D's MAC address, only Host D scoops the frame off the wire and proceeds to process it through his own protocol stack.

<u>Question 2) 4 points total</u>

2.) A "tuple" is a term, which means "an ordered set of values" and …… (Please see the homework for the complete question.)

ANSWER to part A)

The following 5-tuple uniquely identifies each Internet connection:

Protocol (TCP or UDP)
Local (also incorrectly called Source) IP address
Remote (also called Foreign) IP address
Local (also incorrectly called Source) Port
Remote (also called Foreign) Port

This 5-tuple is typically maintained as an operating system table in both the server and the client. The Protocol identifies either TCP or UDP and the IP addresses specify the two ends of the connection. When the connection is established from a client to a server, the Local Port number is chosen by the client's OS and it is a unique value at that point in time for all of the connections between that client and server. The Destination Port number identifies which application layer protocol at the server is being requested. These port numbers are within the UDP or TCP header and the IP addresses are in the IP headers.

In a server, the table is filled-in when the TCP message is received and the connection is opened (or the UDP packet is received.) The port number and address information is taken from the TCP header and the IP header when the packet is received and processed. Of course, when a packet is sent back to the source, the source and destination addresses and port numbers must be reversed within the actual datagram that is sent back.

ANSWER to part B)

The 5-tuples described in question 2a are used to identify individual end-to-end TCP connections or UDP transactions, and are typically maintained as an operating system table in both the client and the server. The Protocol field identifies either TCP or UDP, and the IP addresses specify the source and destination addresses of the sender and recipient. When a TCP connection (or UDP transaction) is established from a client to a server, a unique Local Port number is chosen by the client's OS. The specified Destination Port number identifies which application layer protocol is being requested at the server. These port numbers are located in the UDP or TCP headers, and the IP addresses are located in the IP headers.

In a server, a row in the connection table is created and filled-in as part of the process of initializing the TCP connection; the process begins when the TCP SYN message is received from the client. As described above, the port number and address information is taken from TCP (or UDP) and IP headers when incoming packets are received and processed. Also, each connection established between a server and different client machines, or each connection between a server and a single client with multiple windows open to the server, would create a different 5-tuple entry in the table.

Given this, and based on the IP address and application information provided, the connection table in the server would have entries similar to the following for the email connection and web connection:

Protocol	Local IP	Foreign IP	Local Port	Foreign Port
TCP	18.19.20.21	128.103.104.105	25	52095
TCP	18.19.20.21	128.103.104.105	80	52096

•

- Protocol: Options are TCP and UDP. This would be TCP for HTTP and SMTP.
- Foreign (remote) port: Port used by the laptop's web and email applications. These port numbers are unique for each connection from the laptop.
- Foreign (remote) IP Address: IP address of the laptop.
- Local IP address: IP address of the server.
- Local Port: Port used by the web application (80) and email application (25).

The connection table in the laptop would have entries similar to the following for the email connection and web connection:

Protocol	Local IP	Foreign IP	Local Port	Foreign Port
TCP	128.103.104.105	18.19.20.21	52095	25
TCP	128.103.104.105	18.19.20.21	52096	80

•

- Protocol: Options are TCP and UDP. This would be TCP for HTTP and SMTP.
- Foreign (remote) port: Port used by the server's web application (80) and email application (25).
- Foreign (remote) IP Address: IP address of the server.
- Local IP address: IP address of the laptop.
- Local Port: Port used by the web and email applications, which are selected by the laptop's OS.

Question 3) 4 points total
3a.) Describe the type of information that would be found in a distance vector routing …… (Please see the homework for the complete question.)

ANSWER PART A
The typical column headings in a distance vector routing table for RIP would be as follows:
- **Destination Network and Mask** – This field is the IP network address of the destination network and corresponding network mask.
- **Metric** -- The metric is the "distance" or "cost" of the path to the destination; in RIP, this is simply the number of hops.
- **Next hop IP address** -- The next hop indicates the IP address of the next router in the path to the destination. For those networks the router is directly connected to, there would be no next hop; in these cases, the datagram would be delivered directly to the destination host.
- **Port** -- The port (network interface) that the router uses to send the packet

The following are headings that might be included in the routing table, but were not needed for your answer.
- **Route change flag** -- The route change flag indicates whether information about the route has changed recently.
- **Timers** -- These values provide information about various timers associated with the path.

For example, the table for a two-port router in a small network that was directly connected to two of four interconnected networks would look as follows. Again, the Flags and Timers field were not needed in your answer.

Dst Net and mask	Metric	Next Hop	Port	Flags	Timers
14/8	1	Directly attached	1	n/a	t1, t2, t3, …
15/8	1	Directly attached	2	n/a	t1, t2, t3, …
16/8	2	15.0.0.2	2	n/a	t1, t2, t3, …
17/8	3	15.0.0.2	2	n/a	t1, t2, t3, …

ANSWER PART B
The actual routing protocol selection would ultimately depend on the specific network configuration and would take into consideration a number of different factors. In almost all cases though, a "large private enterprise network" would use OSPF. This is because OSPF offers many services and management features over RIP that would be beneficial for a large network. These include:
- OSPF is not limited by the number of hops between networks like RIP, which is limited to a maximum of 15 hops.
- OSPF converges faster than RIP because routing changes are quickly propagated.
- OSPF scales more efficiently than RIP.
- OSPF provides load balancing features, this allows multiple paths to be used for forwarding
- OSPF allow for networks to be partitioned and divided into areas.
- OSPF provides better security features.

In a link-state protocol such as OSPF, each router keeps a map (topology) of the entire network. Each router periodically tests its connectivity to neighboring routers and propagates (through broadcasts or message forwarding) the status of that connectivity to all routers on the network.

The OSPF protocol tends to be more stable and efficient as the network size increases, versus a distance vector protocol like RIP. Specifically, link state algorithms converge (stabilize) faster than distance vector algorithms and link-state algorithms scale more efficiently than distance vector algorithms. The size of messages in distance vector protocols grows for every network that is reachable, whereas in a link-state protocol the message size is relative to the number of direct links for each router. Link state protocols aren't limited by the number of hops between networks unlike distance vector protocols such as RIP that are limited to a maximum of 15 hops max. The OSPF protocol uses link-state information along with type of service and load balancing information to calculate the shortest route. OSPF can also use authentication so that only authorized routers can change routing information.

Question 4) 2 points total
4. UDP uses a pseudo-header when calculating the checksum …… (Please see homework for the complete question.)

ANSWER
Before a UDP datagram is transmitted, the sending host calculates the UDP checksum (using one's compliment arithmetic) from the pre-pended UDP pseudo-header and the UDP datagram.

The UDP pseudo-header is twelve octets in size, and contains the following fields:

- Source Host IP address
- Destination Host IP address
- UDP protocol number (17),
- Byte count of the UDP datagram in question (not including the pseudo-header)

Since the UDP checksum is computed using IP address information, which of course is part of the network layer (i.e., the IP layer), it is clearly in violation of the protocol layering rules we have discussed in class. This violation is necessary to provide assurances that a UDP datagram has indeed been delivered to the correct destination host and correct protocol stack (UDP versus TCP.) Recall that a UDP header itself only specifies port numbers.

The UDP checksum is end-to-end. That is, when the UDP packet is received by the destination host, the pseudo-header is re-assembled and the UDP checksum is re-calculated. If the sender detects an error in the re-calculated checksum, then the UDP datagram is silently discarded since the assumption is made that the IP address information is wrong, the protocol number is incorrect or there was another problem.

Question 5) 4 points total
5. TCP includes functionality for both flow control and congestion control …… (Please see homework for the complete question.)

ANSWER
Flow control manages the flow of data between the two ends of a connection so that a sender does not overwhelm a receiver with more date than it can process.

TCP flow control is implemented using an octet-based sliding window. The sliding window is continuously adjusted through both the use of the *Window* field in TCP Acknowledgements, which is used to advertise available buffer space, and the *Acknowledgement Number* which is used to tell the receiver which octets have been accepted. The sliding window is typically larger than a single packet (TCP stream segment) so the network can be used more efficiently by sending more than one packet before an acknowledgement is received.

In contrast to flow control, congestion control manages the sender-to-network conditions.

Since TCP begins transmitting into a network with unknown conditions, it must slowly probe the network to determine the available capacity. This is important in order to avoid congesting the network with a large burst of data that the network cannot handle (regardless of whether the far end of the connection can handle the data.) The slow start algorithm is used for this purpose at the beginning of a transfer, or after a loss of data.

When slow start begins, the window size is set to a value of M and a single full-sized segment is sent to the receiver. If acknowledged, the window size is then doubled and two segments are sent out. If this is acknowledged, the window size is doubled once again. This continues until one of a number of things occur: a packet is lost, the receiver's window is reached, or the slow start threshold is reached.

If a packet is lost, the threshold is set to half the current congestion window size and things start again. If the congestion window reaches the slow start threshold, the congestion avoidance algorithm takes over. The term "slow start" is misleading since the start is not slow at all, it is an exponential process. It is "slow" only in the sense that it is slower than immediately sending all of the data into the network as soon as the connection is opened.

Both flow control and congestion control are needed to efficiently use the network. Flow control keeps the sender from overwhelming the receiver and triggering retransmissions. Congestion control keeps the sender from overloading the network and triggering retransmission.

Question 6) 4 points total

6a. and 6b.) Imagine that you are designing a NAT box that used port-mapped …. (Please see the homework for the complete question.)

Answer:

The NAPT allows hosts within a private (internal) network to gain simultaneous and transparent access to hosts in an external network (typically the Internet) using a single externally-known IP address. The table information described in a prior question is not sufficient and the table data must be updated to include the following:

Private Source (local) IP address – source address of internal host that is accessing the external network
Private Source (local) Port number - source port of internal host that is accessing the external network
Destination (remote) IP address
Destination (remote) Port number
Unique NAT Source Port number – port number which replaces the source port of internal host. We referred to this as the mapped port number in lecture.
Public Source IP address – external IP address of the NAT box. This replaces the Private IP address of the internal host.
Protocol Type – TCP or UDP or ICMP, etc.

For each connection from the internal network to the external network, the NAPT box creates and uses a unique NAT port number as the source port number in the UDP or TCP datagram. The destination port number is not changed and neither is the destination IP address. The Private source IP address is of course replaced with the registered (external) IP address. The unique NAT source port number (along with the other information) allows the NAT box to match internal and external connections.

The NAT box must of course do more than just substitute IP addresses and the source port number. In the IP header, the Checksum field will have to be recalculated, and the checksum field in the TCP and UDP headers will of course also have to be changed.

The most important side effect of using a NAT box is that it takes away the end-to-end significance of the IP address and the port number. This means that the destination host does not know the true IP address of the source machine Some protocols such as FTP use this information about the source host to operate properly. (FTP opens a second connection for the exchange of data using information on the source host contained within the application protocol.) The NAT box must understand the application layer protocol and substitute source host information with NAT box specific information. This is fairly easy for a well understood application such as FTP but much more difficult for complex protocols such as H.323 and SIP (which are used for video and voice over IP.) Another protocol which must be modified by the NAT box is ICMP. Remember that ICMP is a network layer protocol; there is no UDP or TCP packet encapsulated within the IP packet and hence no port number. ICMP works when going through a NAT box because ICMP carries a query identifier that is used by the NAT box to correlate responses with requests.

Finally, the use of NAT is largely one-way. Hosts on the "private" network side may access the Internet without a problem. Hosts on the Internet cannot initiate connections to hosts on the private (internal) network (without special configuration of the NAT box) since the private IP addresses are not advertised to the outside world. Special configuration would be required if you placed your public web server behind a NAT box.

<u>Question 7.).</u> <u>*(3 points total)*</u>
a.) What does it mean when we say that a network supports IPv6 Network? ……(Please see the homework for the complete question.)

From a technically pure perspective, a network would be said to fully and completely support IPv6 if all of the network devices (such as routers, firewalls, proxies, etc.), all the hosts and websites that make up the network, the supporting and infrastructure protocols (such as DNS and the router protocols such as OSPF and BGP), and the application layer protocols were all operable with IPv6 at the same level of functionality as is currently the case with IPv4. A thought experiment would be to ask whether the systems and networks would continue to work as they had if IPv4 were suddenly turned off.

Some of the most straightforward items are whether all of the routers in the path support IPv6 addresses, and whether routing protocols such as OSPF, and BGP have been updated. The DNS protocols and systems would of course need to fully support IPv6 and while many core servers do so today, there are many local and company-specific DNS servers and DNS caches that do not support IPv6.

Such a technically pure environment will not be in place for quite a while and until that time, system and technologies will need to support both IPv6 and IPv4, as well as a transition path from IPv4 to IPv6. These mechanisms must provide the flexibility for IPv4 nodes to communicate with other IPv4 nodes, as well as to IPv6-only nodes. They must also allow IPv6-enabled nodes to communicate with each other using IPv6. Examples of such approaches include dual-stack systems and various forms of tunneling. As we demonstrated in class, the 4G cell phone network includes this dual-stack functionality.

b.) Determine whether the network you use supports IPv6.

You will receive credit for this portion of the question if your answer described how you determined whether or not your system and network supports IPv6, the analysis you did, and the tools and techniques you used to derive the needed information. There are many websites (such as www.test-ipv6.com) that you could have used to determine your IPv6 connectivity. These sites test and report on network characteristics such as: IPv6 address detection, IPv6 DNS support, browser support for IPv6, etc. Since supporting IPv6 requires that both your ISP and your local corporate or home network supports IPv6, an additional step would be to analyze your own system's IPv6 network configuration, and local LAN support for IPv6 via tools such as IPv6-enabled ping and traceroute. (OSX has ping6 and traceroute6.)