Homework 3

Scott Ouellette

Scott_Ouellette@hms.harvard.edu

1.)

Consider an IP network composed of two networks (13.0.0.0/8 and 18.0.0.0/8) connected by a router, R, having interfaces 1 and 2. The IP addresses of hosts A, B, C, and D, as well as of interfaces R1 and R2, are given in the figure below. Let the Ethernet address of host A be called MAC_A, of R1 be called MAC_R1, and so forth.

```
   A           B    13.0.0.1   18.0.0.1      C              D
   |           |          +------+           |              |
---------------------------|1 R 2|--------------------------
13.0.0.3    13.0.0.4       +------+      18.0.0.7    18.0.0.8
```

a) Suppose that host A has some application layer data (layer-5) to send to host B using UDP. Describe this process in detail. Specifically, explain what happens on host A at each of the five layers of the Internet Model. Assume for this question that Host A has never sent data before to Host B. Write your answer in the form of a detailed list, such that step 1 in your list is "(1) Host A's application passes its data off to the layer-4 UDP module where it is encapsulated and a UDP header is added." Be sure to include information on how MAC addresses, IP addresses and subnet masks are used by the host. Include information on the use of ARP and the use of a routing table (if necessary) in your answer. Your explanation can stop at the point when A's data reaches B's Ethernet interface; you needn't explain what happens in B's protocol stack.

- The application sending UDP data on Host A will leverage the Operating System's capabilities to it send the information up to the transport layer.
- At the transport layer, the payload will be encapsulated in a UDP packet where a UDP header is added.
- This UDP packet now includes information about the:
    - Source Port Number: Host A's Port that it's application is interfacing with
    - Destination Port Number: Depending on what the remote application is, this could be many different things. If we were interacting with a standard web server it would be port 80
    - UDP Length: The length of the UDP header and UDP data
    - UDP Checksum: a checksum used for error detection and not error correction
    - UDP Pseudo-header: A prefix to the regular UDP header to help prevent misdeliveries. As part of the checksum computation, it pulls in the originator's source and destination address, IP protocol field and the UDP length
- This packet is now ready to move down another layer to the Network layer (layer 3)
- Another round of encapsulation occurs, this time constructing an IP Packet:
- This packet will contain the following information:
    - Some Controls & Flags like IP Version, QoS, Length, Fragmentation Information, etc.
    - TTL: set to a reasonable initial value and decremented upon each hop the packet takes
    - Protocol: the corresponding value here for UDP since the data being encapsulated was a UDP packet (17 in this case for UDP)
    - An IP Checksum: Again, used for error detection and not error correction
    - Source IP address: Host A's IP address
    - Destination IP address: IP of the device we are to be interacting with
- This packet is now ready to move down another layer to the Ethernet layer (layer 2, MAC layer, Data link layer)
- Upon reaching this layer an Ethernet frame will be built encapsulating the "bundle of bits" we have so far (our IP Packet)
- This frame will contain the following information:
    - Source address: MAC address of Host A
    - Destination address:
        - This is determined using an ARP request to broadcast: "Hey, this is my IP & MAC and I'm trying to send data to this IP... does anyone know about it?"

        - ARP Table entries on the Hosts receiving this request will be updated to include Host A's information
        - When another device with the requesting IP receives this request, it will send an ARP reply back with its identifying information and Host A will continue with the construction of the Ethernet frame
        - Notably, in this example, the destination ip address is on the same subnet as the host IP address, therefore there is no need to interact with the default gateway and they can communicate directly!
    - EtherType
    - Checksum (Tacked on to the end of the frame!)
- This frame is now ready to be put out on the wire at the physical layer (layer 1)
- This frame will be delivered directly to Host B due to the fact that they are on the same subnet.

b) Using the same list format, explain the process by which host A sends some application layer data to host D. Again, your explanation can stop at the point when A's data reaches D's Ethernet interface. Assume for this question that Host A has never sent data before to the router, or to Host D. PLEASE NOTE: Answering this entire question will probably require two-pages.

Th information from answer 1a. is the same up until the ARP reply

- Upon reaching this layer an Ethernet frame will be built encapsulating the "bundle of bits" we have so far (our IP Packet)
- This frame will contain the following information:
    - Source address: MAC address of Host A
    - Destination address:
        - This is determined using an ARP request to broadcast: "Hey, this is my IP & MAC and I'm trying to send data to this IP... does anyone know about it?"
        - ARP Table entries on the Hosts receiving this request will be updated to include Host A's information (IP & MAC)
        - When another device with the requesting IP receives this request, it will send an ARP reply back with its identifying information and Host A will continue with the construction of the Ethernet frame.
        - In this scenario, the ARP reply is actually coming back from the default gateway Interface 1 (MAC_R1)! Why is this!? This is because the ARP propagated to the right side of the router in the diagram, found Host D, updated it's routing table with information on how to get to Host D, and then proceeded to tell Host A that: "Hey, talk to me if you want to get to Host D" since there isn't a direct connection to the receiving host as there was in the last example.
    - EtherType
    - Checksum (Tacked on to the end of the frame!)
- This frame is now ready to be put out on the wire at the physical layer (layer 1)
- The frame is received on interface 1 of the router
- The ethernet header is stripped and a new ethernet frame is reconstructed with the information available from the router's routing table about how to reach Host D (Change Destination address to MAC_R2)
    - Note that the payload remains untouched !
- After the newly update ethernet frame is ready it is sent out on the wire where Host D receives it

2.)

A "tuple" is a term which means "an ordered set of values" and in our readings and discussions we have observed that connections in the Internet can be uniquely identified using a very specific 5-tuple.

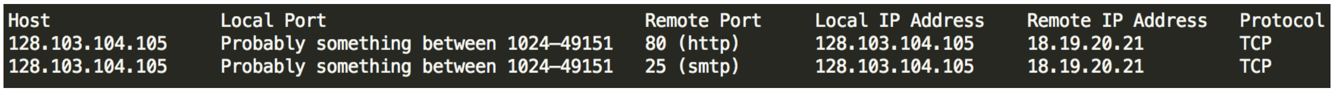
a.) Describe this specific 5-tuple in detail.

This 5-tuple that we talked about in lecture 8 allows for unique identification of layer-4 protocol connections on a given host. There needs to be a notion of unique-ness when dealing with these connections so that one isn't sending information through a connection it shouldn't be. Any modern Unix-y or Windows host will have information about its current connections. One can find useful information about these connections on Unix-y machines using: `netstat -an` . Some entries in the output of `netstat` may be counterintuitive at first: i.e. there may be many connections with the same local & remote IP... how is this? It should be noted that inter-process communication is commonly done using layer-4 communication protocols.

The information in an entry for a single connection that, as a whole, deems it unique is: `(local_ip, remote_ip, local_port, remote_port, protocol)` . Where the local designation is for a given host and the remote designation is for the host on the other side of the connection.

The protocol portion of the tuple is simply the protocol that is providing the connection. There is no definitive answer as to what the source port should be. It's usually determine from a range of ports in the 1024–49151 range. The remote port will be specific to the service that the connection is for. For example, web servers generally "talk" on TCP port 80, and ssh is generally done over TCP port 22.

b.) Assume that a user working on a laptop has both an email connection and a web connection open simultaneously to a remote server. Assume that both the laptop and web server are directly connected to the Internet (in other words, they are not behind a NAT device) and that the IP address of the laptop is 128.103.104.105 and that the IP address of the server is 18.19.20.21. Describe in detail the 5-tuple information that you would find in the connection table of both the laptop and the server. (As shown in lecture, you would be able to view these details using the Netstat command.)



```
Host             Local Port                            Remote Port     Local IP Address    Remote IP Address   Protocol
128.103.104.105  Probably something between 1024–49151  80 (http)       128.103.104.105     18.19.20.21         TCP
128.103.104.105  Probably something between 1024–49151  25 (smtp)       128.103.104.105     18.19.20.21         TCP
```

3.)

a.) Describe the type of information that would be found in a distance vector routing table. In other words, what are the typical column headings of the routing table in a router using the RIPv2 protocol?

RIPv2 is a routing protocol that counts hops to measure efficiency of a given route. It then populates a routing table with the most efficient routes that it knows of, and shares this information with all other routers that it can "see". The typical column headings for such a table are:

```
Network # | Distance (Cost) | Outgoing Port # | Next Hop's IP Address
```

b.) Assume that you had to choose either RIPv2 or OSPF as the routing protocol for a large private enterprise network. Which one would you pick? Provide three substantive technical reasons for your choice.

I would choose OSPF for this task for a few reasons:

- RIPv2 (and other distance vector routing protocols) cause routing loops which at a large scale can be inefficient and costly. This type of loop is also known as a "Count to Infinity" problem and can happen in certain cases when link between routers are disconnected and other routers with information about the disconnected router keep on propagating its information after its death.
- OSPF supports multiple types of routes, which could be helpful in a larger network topology where it's harder to immediately know all the different route types.
- OSPF is known to be easier to manage as it can be used in partitioned fashion over a network with each partition independently running OSPF.

4.)

UDP uses a pseudo-header when calculating the checksum of a datagram. (We are referring to the checksum that is included in the UDP header.) Describe the pseudo-header and how it is used to calculate this checksum. What potential problem or problems does calculating the checksum in this manner solve?

The UDP pseudo-header is comprised of the originating host's source and destination IP addresses, IP Protocol field, and the UDP length field. It is prefixed to the UDP header. This pseudo-header protects against misdelivery due to corrupted IP addresses. The receiver can use the pseudo-header info to silently discard the datagram if it was not meant to receive it.

5.)

TCP includes functionality for both flow control and congestion control. Describe the motivation for each of them. In other words, explain why both are necessary and how do they differ. Also describe how each of them is implemented and works.

- Flow Control:

    - Flow Control happens from "end-to-end". The host receiving the information is the direct cause for the flow of TCP traffic needing to be controlled/managed.
    - Flow control is necessary as to not overwhelm the receiving host.
    - Flow control happens from End-to-End rather than on the network itself.
    - The "window" or the amount of bits that a receiver can receive at a given time is changed dynamically based on its ability to process said information.

- Congestion Control:

    - Congestion control happens on the network itself.
        - Network congestion occurs at routers and other network devices due to bursts of traffic
    - Network congestion is addressed by TCP's "Slow Start" (a misnomer, the slowness is relative)
    - TCP was designed to be very efficient. In designing an efficient protocol to transfer information, it makes sense that one would want to send as much as possible in a reasonable fashion.
        - TCP does this with "Slow Start". Basically, the sender starts sending a number of packets, exponentially increasing the number of packets as full sets of ACKs are received until packet loss starts to happen. At that point the transmission rate is chopped down, and a linear increase in packets is favored rather than an exponential one. This way connections that would potentially cause network congestion are suppressed before they could ever do so.

6.)

a.) Imagine that you are designing a NAT box which used port-mapped network address translation (sometimes known as NAPT.) As you do your design, you realize that the 5-tuple discussed in class is not sufficient to keep track of the connections which traverse the NAPT box. It is a good starting point, but the "translation table" in the NAPT box needs additional information. What information is needed in this table and how is the table structured and managed? Describe the use of the table in detail.

The additional information needed in the translation table is the:

- Mapped source IP address
- Mapped source port

This mapping is done to make hosts on private networks accessible in limited fashions without exposing routable information. This table is usually managed by network admins who are exposing applications/services running on said hosts in the private network.

Let's say that a network admin has configured NAT to map port 8888 on host 192.168.1.100 running a webserver on the private network behind a router to port 80 on edge router's public IP. An end user accessing http://:80 would then be able to see the web traffic that host 192.168.1.100 is serving.

b.) The NAT box you design is implemented and will soon be used to provide IP address sharing for users of cable modems and DSL service. During the testing though, it becomes obvious that port mapping is not the only capability that is required for a NAT box. For example, some ICMP diagnostics do not work when the packets are sent through your box. Explain in detail what is happening, and what changes must be made to the various packet headers by your software so that the packets flow properly across the NAT box. Given what you have learned, describe any limitations you might expect with your implementation. Be specific.

ICMP has no port number information in its header like TCP/UDP and it isn't the only layer-4 protocol to not include said info. Therefore, the 7-tuple mentioned before needs to be utilized in a manner specific to the protocol. In the case of ICMP, Traditional NAT is done by translating the IP header within a given ICMP packet. While NAPT has to further translate the ICMP sequence number and associated checksum as well.

The situation described in the example could be caused by a fault in the ICMP query sender whos job it is to set the ICMP query identifier field. If this field isn't being set to a unique value, then it isn't viable to use this value as an analog for TCP/UDP ports during translation. It is also known that ICMP error packets need to be modified to work with NAT [rfc 3022 section 4.3] so it could also be the case that the outer layer isn't making the proper modifications there either.

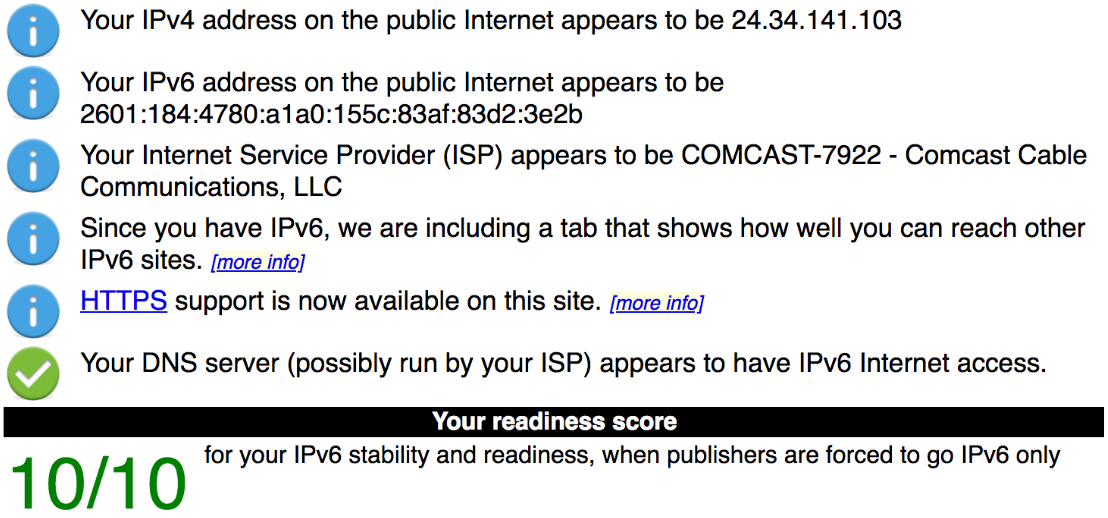
7.)

a.) What does it mean from a technical standpoint to say that a network supports IPv6? In other works, what are the critical components and systems that need to be in place for your network to support IPv6. (This is more than just saying that OSX or Windows 7 supports IPv6.)

To say that a network supports IPv6 is to say that it as a whole (all devices, transmission medium, and device configurations) are able to send/transmit and receive IPv6 traffic, and support IPv6-specific implementations (such as its neighbor discovery). The networks edge routers should specifically be able to translate any internal link-local IPv6 addresses to IPv4 since there is no guarantee that the outer network also supports IPv6.

b.) Determine whether the network you use at work or school supports IPv6, and report your results. How did you figure this out? Note that if you work from home and do not use a corporate or campus network then you should determine whether your home network supports IPv6.

I used test-ipv6.com from my home network as mentioned in lecture.



ℹ Your IPv4 address on the public Internet appears to be 24.34.141.103

ℹ Your IPv6 address on the public Internet appears to be 2601:184:4780:a1a0:155c:83af:83d2:3e2b

ℹ Your Internet Service Provider (ISP) appears to be COMCAST-7922 - Comcast Cable Communications, LLC

ℹ Since you have IPv6, we are including a tab that shows how well you can reach other IPv6 sites. *[more info]*

ℹ HTTPS support is now available on this site. *[more info]*

✓ Your DNS server (possibly run by your ISP) appears to have IPv6 Internet access.

Your readiness score

10/10 for your IPv6 stability and readiness, when publishers are forced to go IPv6 only