HW6 - UDP

By Michael Almandeel

Consider the following sequence of events which comprise the lifecycle of a message traveling from host A to host B:

-An application on A decides to send some data to an application on host B. In order to do that, A must know the ip address of B(we haven't covered how that happens yet) and the port number of the application on B(also not yet covered). Therefore we assume we have this data and do not worry about how it is obtained.

- The application in A utilizes an api provided by the os to request network services provided by A. The data structure used to send and receive data is called a socket. The application in A configures the socket using the destination ip address and port number obtained above, and writes data to it(the payload).

-The udp subsystem in the transport layer, after examining the socket, isolates the payload data that is to be sent, and encapsulates it in a udp header, which itself contains the source port, destination port, message length(udp header + payload), and a checksum. After constructing this sequence of bytes, it passes the sequence to the internet layer of A.

-The internet layer of A encapsulates the bytes from the transport layer in an ip header, wherein the destination address is set to the value obtained from the socket, and the source address to A's own ip address. The resulting sequence of bytes is then passed to the network interface layer of A.

-At the network interface layer, the data received from the internet layer to determine the correct recipient on the local network. It is determined that the destination physical address will be the "default gateway" aka "Router R"(we have not covered precisely how this process works). The bytes from the internet layer are encapsulated in a frame, with the source address being A's own physical address, and the destination address being the router r's physical address. This frame is sent to router a via physical network 1.

-A short time later, the network interface of router r on physical network 1 receives the frame from A. It copies the payload of the frame(the internet layer datagram) and discards the header. This is passed to the internet layer of R , to make a routing decision about how to forward this datagram.

-The internet layer of R receives the datagram , and examines the destination ip address to compare to its routing table. R determines that this datagram should be forwarded out the network interface that corresponds to physical network 2. Note that as this point, R will decrement the value of the TTL in the datagram header by 1, before forwarding this data to the network interface.

-The network interface of router r on network 2 receives the datagram from R, and upon checking its arp cache, determines that the destination ip address(b's) has an entry in the cache, and from that entry the physical address is extracted. The network interface encapsulates the datagram into a frame, wherein the destination address of the frame is the physical address of B, and the source address is the physical address of the outgoing physical interface.

- A short time later, B’s network interface on network 2 receives this frame, and upon determining it is the recipient(same physical address as destination), it removes the frame header and passes the resulting datagram to the internet layer.

- The internet layer removes the datagram header and examines the payload, determining that it is a udp datagram. It passes the udp datagram to the transport layer.

-The transport layer, upon receiving the udp datagram, determines it is meant for the destination port indicated in the udp header. It passes this information to the os, along with the payload.

- The os will write this payload data to the buffer that corresponds to the incoming channel of the socket with that corresponding port, if one is open. Otherwise, the payload will be discarded.

Considering the above behavior, it is apparent that both the ttl field of the internet datagram and the source and destination addresses of the frames will change at router R.