Mahedi Kaiser

Homework 2

Explain how the ARP protocol works. What problem does it solve? How does it solve the problem? Give examples and be as specific as possible.

The Address Resolution Protocol translates IP address to hardware address to help in communicating on a network. Devices on a network need to use the hardware address to communicate. However, the applications only use IP addresses. ARP solves this problem. When a device wants to communicate with another device on the network, it broadcasts a message containing its own IP address and hardware address, and the IP address of the intended recipient. All of the other devices receive this message. If their IP address matches that of the intended recipient, they respond back to the sender with a similar message. If not, they ignore it. For example, if A wants to communicate with B:

1. A sends B an address resolution request, which contains the following:
   1. I am IP\_A
   2. My Hardware Address is HW\_A (B needs this so that he can respond back)
   3. I am looking for IP\_B
2. This message is broadcast to every node on the network. They will each read the message and only reply back if their IP address matches IP\_B.
3. In this case, B’s IP address will match IP\_B. Using the hardware address that A sent, B will reply with a similar message:
   1. I am IP\_B
   2. My hardware address is HW\_B
   3. To HW\_A
4. B will also save A’s IP Address and Hardware Address in a table. This is called a table. It will also set a timer when it saves the binding. When the timer runs out, the binding is removed. This prevents too many unused bindings from taking up memory space.
5. A does the same with B’s Hardware address
6. After a device deletes a binding, it needs to send another ARP request if it wants to communicate with that device again.

Explain fragmentation as it relates to datagrams. Why is it necessary? Where does it occur? When and where are the fragments reassembled? Give examples and be as specific as possible.

Fragmentation is when a message is broken into fragments. This is necessary when the message is larger than the MTU (maximum transmission unit) of the network. When the message is larger than the MTU, the message is broken into fragments whose sizes are equal to the MTU, after accounting for the frame header. Of course, if the message size isn’t a multiple of the MTU, the last piece will simply be equal to the message size % MTU (again, accounting for header).

For example, if A wants to send a 1000 byte message over a network whose mtu is 500, each fragment will contain a maximum of 480 bytes + 20 bytes for the header:

1. The first fragment will contain bytes 0 – 479 of the original message
2. The second fragment will contain bytes 480 – 959 of the original message
3. The last fragment will contain the bytes 960 – 999 of the original message

It is possible that these fragments will go through several more networks with smaller mtu’s before reaching the recipient. In that case, the fragments are broken down into smaller pieces in a similar fashion. These fragments are reassembled only once they reach the recipient.

What is the function of the TTL field in a datagram header?

The TTL is a counter that starts off as 255 and decrements every time it passes from one network node to another. Once it reaches 0, it is discarded. Its purpose is to prevent faulty messages from entering an infinite loop and taking up network space (bandwidth?) and as a result of network errors.