Jonathan Chu 804141479
Eileen The 104141878
Joshua St. Clair 904187688

LAB 2: SIMPLE ROUTER

The second lab of CS118 was to create and write the code for a router in a network. This included handling both ARP requests/replies as well as IP packets, as well as the other various packets encountered in a network. This required a great deal of understanding of the various protocols like ARP, ICMP, and the ethernet frames. This project has taught all of us a great deal about the protocols and we have learned a lot about the interactions between hosts.

The solution that we submitted does not necessarily pass all of the test cases, but it does show an understanding of the underlying principles taught in the CS118 networking class. We were only able to make a few functionalities work, but we were close to getting the rest of them to work. If one were to examine our code, they could see that we understood the intricacies of the layouts and protocols of the packets. We had an understanding of all the principles from ICMP to IP to ARP to TCP/UDP packets. However, there are small bugs in our software that don't allow the code to work in practice. We hope that the graders will take this into consideration while grading our project; while we might not have been able to get a few of the functionalities to work, we understood the underlying principles of the course.

The following requirements were given to us:

- **The router must successfully route packets between the Internet and the application servers.**
    - The router does not successfully route the packets. This is due to our ARP requests not being answered correctly; we have them broadcasting with what we believe to be the correct data, but there are no responses that we can use to cache the data with.
- **The router must correctly handle ARP requests and replies.**
    - The router does correctly handle ARP requests and replies. When we initially ping the router, the client sends out an ARP request to find the IP; we take that request and respond to it appropriately
- **The router must correctly handle traceroutes through it (where it is not the end host) and to it (where it is the end host).**
    - The router does not successfully route the packets, for the same reason that the forwarding doesn't work. We have everything working except for the ARP response, so we cannot forward the traceroutes.
- **The router must respond correctly to ICMP echo requests.**

    - The router correctly responds to ICMP echo requests. We had some trouble with this because of how the casting worked, but the router is able to ping the client back.
- **The router must handle TCP/UDP packets sent to one of its interfaces. In this case the router should respond with an ICMP port unreachable.**
    - The router correctly handles TCP/UDP packets. The router responds to the host with ICMP port unreachable.
- **The router must maintain an ARP cache whose entries are invalidated after a timeout period (timeouts should be on the order of 15 seconds)**.
    - The router correctly maintains and ARP cache. The entries also time out after around 10 seconds.
- **The router must queue all packets waiting for outstanding ARP replies. If a host does not respond to 5 ARP requests, the queued packet is dropped and an ICMP host unreachable message is sent back to the source of the queued packet.**
    - The router does this successfully. The router uses a member variable for a packet to determine how many times the ARP request has been sent out; if it exceeds 5, it is dropped.
- **The router must not needlessly drop packets (for example when waiting for an ARP reply)**
    - The router successfully holds onto packets.
- **The router must enforce guarantees on timeouts--that is, if an ARP request is not responded to within a fixed period of time, the ICMP host unreachable message is generated even if no more packets arrive at the router. (Note: You can guarantee this by implementing the sr_arpcache_sweepreqs function in sr_arpcache.c correctly.)**
    - The router does timeout after a certain amount of time. The ICMP host unreachable message is generated and sent out to the appropriate host.

If there are any questions about these functionalities, please do not hesitate to contact anyone on the team. We have demonstrated that they work correctly on our virtual machines and can send proof via screenshot that they do work as intended.

The entire team has worked very hard for very long on this project, and we are saddened by the fact that it does not have all the functionalities that were requested of it. But through our many mistakes, we learned a lot about the fundamentals of network coding and we will not forget such a memorable experience.