Project 2 – Discussion 3
Simple Router
SO FAR WE TALKED ABOUT THE FOLLOWING:

- Pinging the router interfaces
  - When you, the router gets arp request, send back arp reply
  - Then, once you get echo request destined to the one of the router interfaces, send back echo reply (type 0)

- Pinging the servers with ttl as 1 (client ping -c1 -t1 server1)
  - Send back time exceeded icmp packet (type 11)

- Pinging a host which does not exist
  - Do LPM
  - When no entry is found, send back ICMP net unreachable (type 3, code 0)

- Pinging the servers
  - Do LPM
  - Once you find the entry, do sr_arpcache_lookup
  - If returns NULL, do sr_arpcache_queuereq and call handle_arpreq
    - Send arp request to determine the mac address of next hop
  - If returns an entry, forward the queued packets
If (times_sent < 5), then create an ARP request for the next hop ip address.
Use sr_send_packet() function to send the request.
Increment the times_sent variable of the request.
Update sent variable, which is the time this ARP request was sent
   Arp req->sent = time(NULL);
If (times_sent >= 5), then send back Destination host unreachable (type 3, code 1).
Destroy the arp req.

Note: Also make sure that the current time and the last time the packet was sent is
greater than or equal to 1(sec), in the start of this function.
   time_t current_time = time(NULL);
This function gets called every second. The arp sweepregs function would iterate through the ARP requests (cache->requests) every second and call handle_arpreq which will re-send any ARP requests that haven't been sent in the past second.
If Router receives an ARP reply

- Use `sr_arpcache_insert()` function. This will return a pointer to the arp request (`sr_arpreq`) for that particular IP address.

- The arp request has list of packets (`struct sr_packet *packets;`) waiting on this request to finish. So, for every packet in this list, you update destination mac address and forward the packet to next hop.

```c
struct sr_arpreq {
    uint32_t ip;
    time_t sent;
    uint32_t times_sent;
    struct sr_packet *packets;
    struct sr_arpreq *next;
};
```

- Use `sr_arpreq_destroy()` function which removes the request from the queue.
TCP /UDP PACKET

When you do a traceroute (client traceroute -n server1), the router will get a packet destined to it with protocol type as tcp or udp. Send back ICMP port unreachable (type 3, code 3).
BEST OF LUCK!