



Project 2

Simple Router

[Part 3]

SHREEJA KUMAR

CSE 123-FALL 2015

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 the destination mac address and send the packet using `sr_send_packet` function.
- ▶ Use `sr_arpreq_destroy(&(sr->cache), arp req)` function (already defined for you) which basically removes the request from the queue and frees all memory associated with this arp request entry

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

```
struct sr_packet {  
    uint8_t *buf;  
    unsigned int len;  
    char *iface;  
    struct sr_packet *next;  
};
```

Handle_arpreq

- ▶ When you are doing forwarding of an IP packet and you don't have an entry for it in the arp cache, you need to use `sr_arpcache_queuereq()` function to add an ARP request to the queue.

Note: Gateway address and outgoing interface, you obtain from the routing table after you find the LPM.

- ▶ A pointer to the ARP request is returned. Send the sr instance and the arp queue to ***handle_arpreq***.
- ▶ If (`times_sent >= 5`), then send back Destination host unreachable (type 3, code 1). Destroy the arp req without lock.

Use the 'buff' pointer (available in `sr_packet`, which is present in `sr_arpreq`), to get the required information.

- ▶ If ($times_sent < 5$), then create an ARP request for the destination ip address (the *ip* variable in the arp req).
- ▶ Use `sr_send_packet()` function to send the request.
- ▶ Increment the *times_sent* variable of the request.
- ▶ Update *sent* variable, which is the last time this ARP request was sent

```
Arp req->sent = time(NULL);
```

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 curr_time = time(NULL);
```

sr_arpcache_sweepreqs

This function gets called every second. The arp sweepreqs function would iterate through the ARP requests (cache->requests) every second and re-send (**handle_arpreq(arp req, sr)**) any outstanding ARP requests that haven't been sent in the past second.

To test Destination Host Unreachable

- ▶ With some changes to the routing table, you can trigger a Destination Host Unreachable ICMP (type 3, code 1) by making your router send ARP request to a host which is not present. To simulate that scenario, change the next hop gateway to a non-existing IP address.



Thank You