

Project 3: Routing Emulation

Internet is a collection of networks interconnected by routers where packets are routed based on their destination IP addresses. A host or a router performs *routing table lookup* to decide the *next-hop* to the destination. This next-hop will be the destination itself if it could be reached directly. Once the next-hop is determined the packet is delivered to the next-hop by the local link layer. However, this requires mapping from virtual IP address to physical hardware address of the next-hop. Each host or router maintains this mapping in an *arp cache* which is updated using address resolution protocol (ARP) requests and responses.

In this project, you are asked to emulate the functionalities of hosts and routers using BSD sockets for inter-process communication in place of a physical medium. Here, we do not distinguish between a host and a router except that they differ in their configuration, i.e., the routing table and the number of subnets they are attached. So there are only two components in this project: **station** and **hub**. Each subnet (LAN) is emulated using a hub and the hub code will be provided for you. You have to implement the *functionality of station only*, which is described below.

The job of a station is to watch out for packets from the subnets it is hooked to and input from the keyboard by user.

keyboard input First, figure out the destination IP address. The first word in the typed message is supposed to be the name of the destination station which is mapped to IP address using 'hosts' file. Then, form an IP packet filling in the address and data fields and process it like any other incoming IP packet.

packet from socket A station processes a packet only if the destination hardware address is either its interface hardware address or a broadcast hardware address. Otherwise it simply discards it. A station may receive three types of packets.

ARP-REQUEST Send an ARP-RESPONSE packet back to the sender if the target IP address in the request is its own. Also, add an entry to the ARP cache associating sender's IP address and hardware address. Otherwise simply ignore the packet.

ARP-RESPONSE Add an entry to the ARP cache associating sender's IP address and hardware address. Send any pending packets to this sender.

IP DATAGRAM Process an IP packet using the following steps.

- Check if this station is the destination comparing the destination IP address with the station's interface addresses. If so, display the contents of the packet.
- If the packet is not destined for this station, it needs to be routed (if the station is a router). Look up the routing table to determine the next-hop. This involves applying the network mask to the destination address and then comparing with the destination network in each entry in the routing table. The next-hop can be an address of all 0's in which case the destination itself is the next-hop. This way after the route look up, the next-hop IP address is known.
- Look up in the ARP cache to find the hardware address of the next-hop. If found, encapsulate the IP packet into an Ethernet packet filling in the source and destination hardware addresses appropriately and write on to the corresponding socket.

If the hardware address of the next-hop is not found in the ARP cache, queue the IP packet and generate an ARP request. The destination hardware address in the ARP packet is filled with all 1's implying broadcast.