Joohan Lee
7711577119
joohanl@usc.edu

Lab1 Report
CSCI 551

09-08-2023

1 Design

1.1 Goal

- Build a static router that can perform basic routing functions and is runnable as a single shell command(./router/sr).

- Handle both ARP and IP packets after receiving a raw ethernet frame.

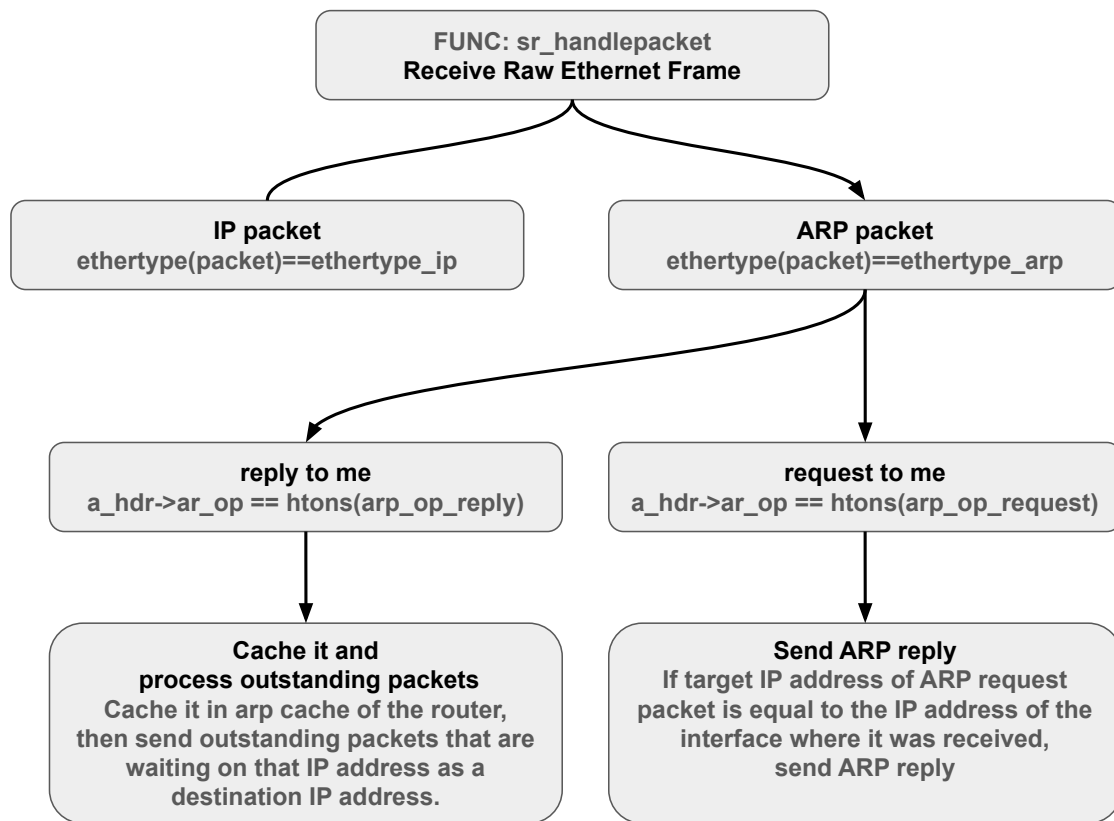- Process and forward packets to correct outgoing interfaces same as lab manual page 12.

1.2 Limitation

- Can handle only five ICMP messages.

- Cannot handle IP packets with many other protocols including IPv6.

- Does not support fragmentation. (All the packets should not be fragmented)

- It is a static router, so router configuration(routing table) is initialized only when the static router runs. Other routing information cannot be updated, inserted, or deleted.

- All the TCP/UDP packets would not be processed, and the router will just send back the ICMP port unreachable message.

- When I tested, iperf results varied. When I tested it several times, the best result was '2.48Mbits/s' and the worst result was '1.56Mbits/s'.

```
*********************************

*********** TEST BEGIN: iperf
*** Running iperf between 10.0.1.100 and 172.64.3.18
*** Iperf: testing TCP bandwidth between client and server3
*** Results: ['2.48 Mbits/sec', '2.77 Mbits/sec']
*********** TEST END: iperf: PASSED (20 points)!
```

2 Implementaion

2.1 Handling ARP packet

```
┌─────────────────────────────────┐
│        FUNC: sr_handlepacket     │
│      Receive Raw Ethernet Frame  │
└─────────────────────────────────┘
```

IP packet
ethertype(packet)==ethertype_ip

ARP packet
ethertype(packet)==ethertype_arp

reply to me
a_hdr->ar_op == htons(arp_op_reply)

request to me
a_hdr->ar_op == htons(arp_op_request)

**Cache it and
process outstanding packets**
Cache it in arp cache of the router,
then send outstanding packets that are
waiting on that IP address as a
destination IP address.

Send ARP reply
If target IP address of ARP request
packet is equal to the IP address of the
interface where it was received,
send ARP reply

2.2 Handling IP packet

```
FUNC: sr_handlepacket
Receive Raw Ethernet Frame
```

```
IP packet
ethertype(packet)==ethertype_ip
```

```
ARP packet
ethertype(packet)==ethertype_arp
```

```
To one of router's IP addresses
is_router_ip(sr, dst_ip) == 1
```

```
To NOT one of router's IP addresses
is_router_ip(sr, dst_ip) == 0
```

```
ICMP echo reply
If IP packet is ICMP echo
request(type:8, code:0),
send ICMP reply(type:0,
code:0).
```

```
TCP/UDP
If IP packet is TCP/UDP(ip_p==7
or ip_p==16) packet,
create and send ICMP port
unreachable (type:3, code: 3).
```

```
ICMP Time Exceeded
If TTL of IP packet to
forward is zero, discard
it and create and send
ICMP message.
```

```
Check routing table. Run LPM.
find_sr_rt_by_ip()
```

Found

FUNC: sr_arpcache_sweepreqs

Not found

```
Check ARP cache
sr_arpcache_lookup()
```

```
ICMP net unreachable
Create and send ICMP message.
find_sr_rt_by_ip()==NULL
```

Hit

Miss

```
Send frame to next hop
If cache hits,
send the IP packet to next hop.
sr_send_packet()
```

```
Send ARP request
If cache miss,
add an ARP request to the ARP
request queue.
sr_arpcache_queuereq()
```

```
ICMP host unreachable
If ARP request was sent
> 5 times,
create and send ICMP host unreachable message
to the host which was triggered that packet.
```