Addressing and Forwarding

Kameswari Chebrolu

Example: Class C

216.12.1.*

222.5.11.2

216.12.1.4   216.12.1.3   216.12.1.2

216.12.3.2

222.5.11.6

R3

222.5.11.3

Intenet

216.12.1.1

R2

R1   216.12.3.1

216.12.2.1

| Network	Next Hop
222.5.11.*	R2
216.12.1.*	Inf1
216.12.3.*	Inf2
216.12.2.*	Inf3

216.12.3.3

216.12.2.4   216.12.2.3   216.12.2.2
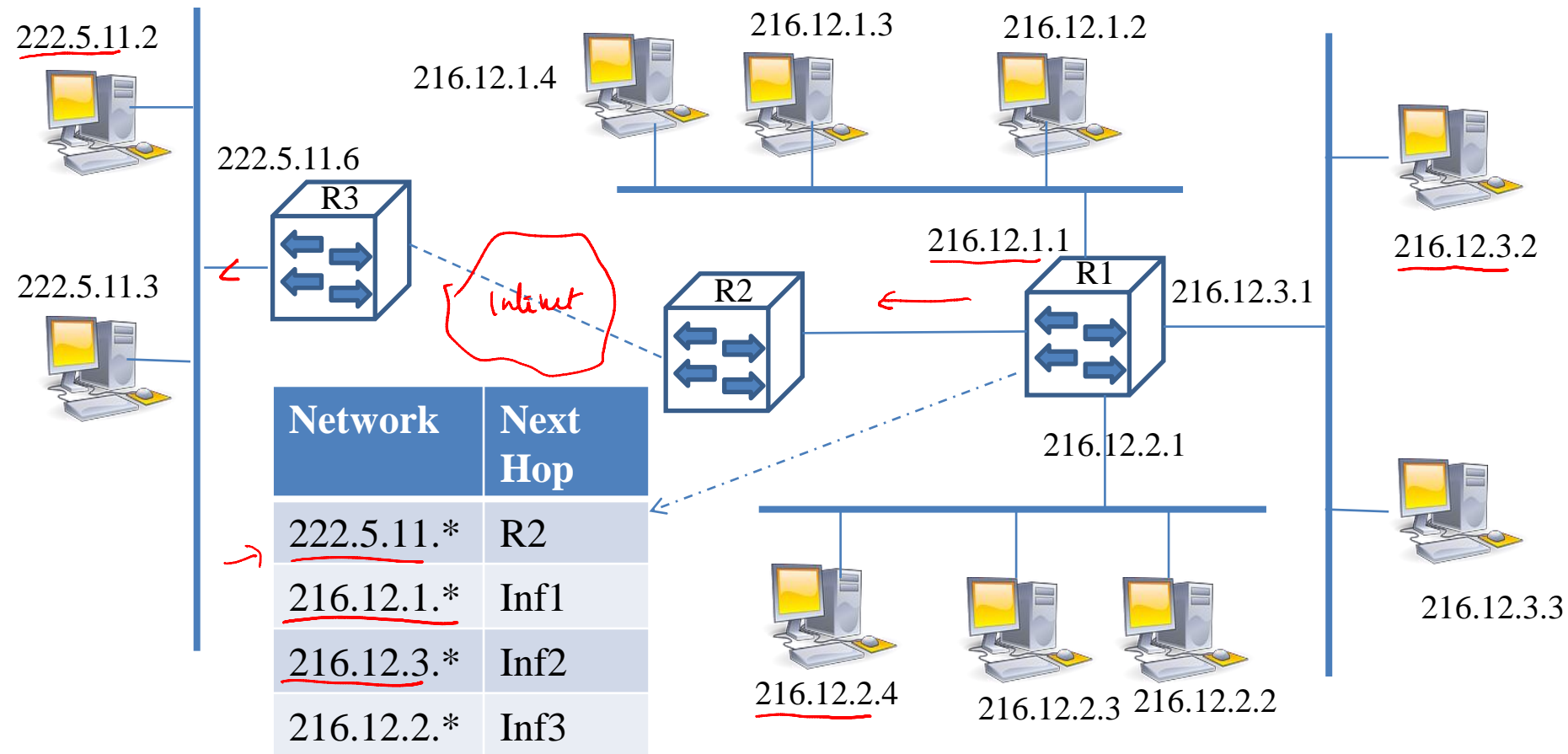
Points to Note

- Every datagram contains IP address of destination host

  - Network part of IP address uniquely identifies a single physical network

- All nodes that share the same network part are connected to the same physical network

- Every physical network has at least one router that is connected to at least one other physical network.
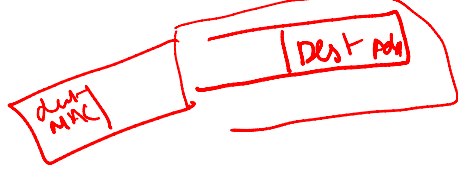
Forwarding at Host

- If (NetNum of Dest = my NetNum) then

  - deliver packet to destination directly

  - use ARP to get MAC address corresponding to dest IP address
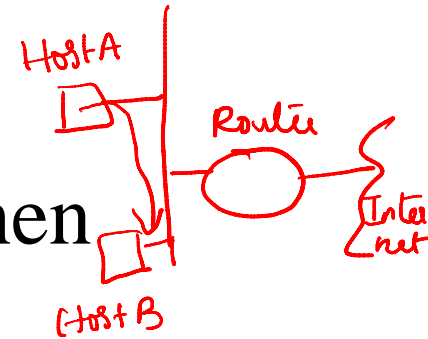
  Else deliver packet to default router

  - use ARP to get MAC address corresponding to router IP address

Linux Usage


```
kameswari@asterix:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.129.0.0      *               255.255.0.0     U     1      0        0 eth0
link-local      *               255.255.0.0     U     1000   0        0 eth0
default         router.it.iitb. 0.0.0.0         UG    0      0        0 eth0
kameswari@asterix:~$
kameswari@asterix:~$
kameswari@asterix:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.129.0.0      0.0.0.0         255.255.0.0     U     1      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
0.0.0.0         10.129.250.1    0.0.0.0         UG    0      0        0 eth0
kameswari@asterix:~$
```
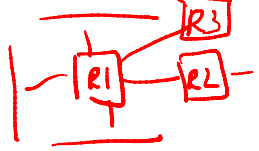
Windows Usage


```
IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway        Interface  Metric
          0.0.0.0          0.0.0.0      10.129.250.1   10.129.154.135     40
      10.129.128.0    255.255.128.0         On-link    10.129.154.135    296
    10.129.154.135  255.255.255.255         On-link    10.129.154.135    296
    10.129.255.255  255.255.255.255         On-link    10.129.154.135    296
         127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
         127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
   127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
         224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
         224.0.0.0        240.0.0.0         On-link    10.129.154.135    296
   255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
   255.255.255.255  255.255.255.255         On-link    10.129.154.135    296
===========================================================================
```

Handwritten annotations (in red):
- IP address → 10.129.154.135
- Broadcast → 10.129.255.255
- Loopback [127.0.0.0 and 127.0.0.1]
- → 127.255.255.255
- Multicast [224.0.0.0 and 224.0.0.0]
- Broadcast [255.255.255.255

Obtained via "route print" command

Forwarding at Router

- If (NetNum of <u>De</u>st = NetNum of one of my interfaces) then

    – deliver packet to destination over that interface

    Else if (NetNum of Dest is in my forwarding table) then

    – deliver packet to NextHop router

    Else deliver packet to default router

Summary

- Top concern: Scalability

- Handled via hierarchical addressing

  – IP address has a network and a host part

  – Significantly reduces entries in forwarding table

- Looked at how forwarding is done at host and router based on the addressing scheme

- Ahead: Address assignment inefficiency