CS183

Instructor: Ali Davanian

(Slides were adopted from Brian Crites and Alireza Abdoli)

L3 Networks

Review

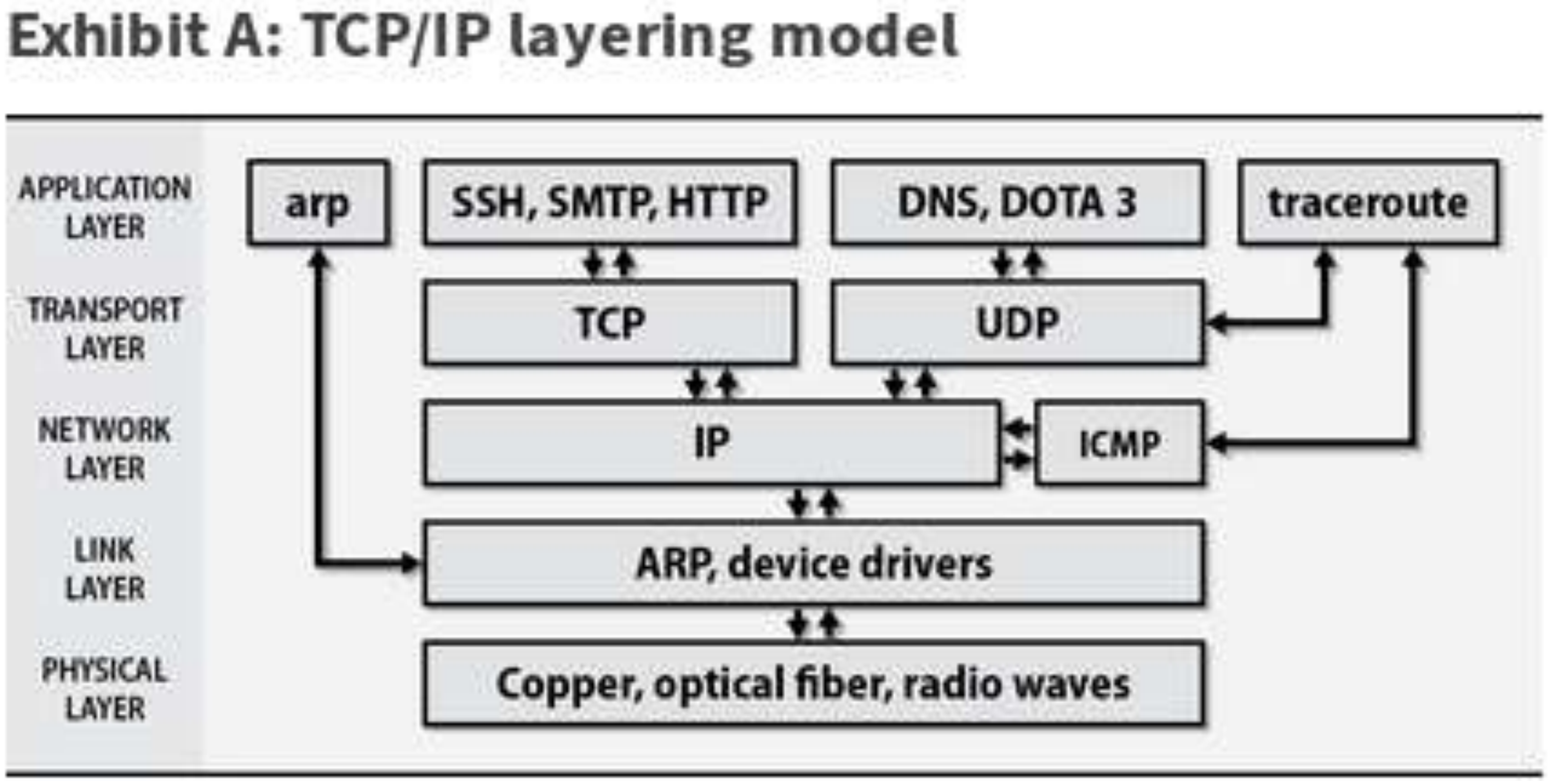


Exhibit A: TCP/IP layering model
APPLICATION LAYER
TRANSPORT LAYER
NETWORK LAYER
LINK LAYER
PHYSICAL LAYER
arp
SSH, SMTP, HTTP
DNS, DOTA 3
traceroute
TCP
UDP
IP
ICMP
ARP, device drivers
Copper, optical fiber, radio waves

Layer 2 configurations in Linux

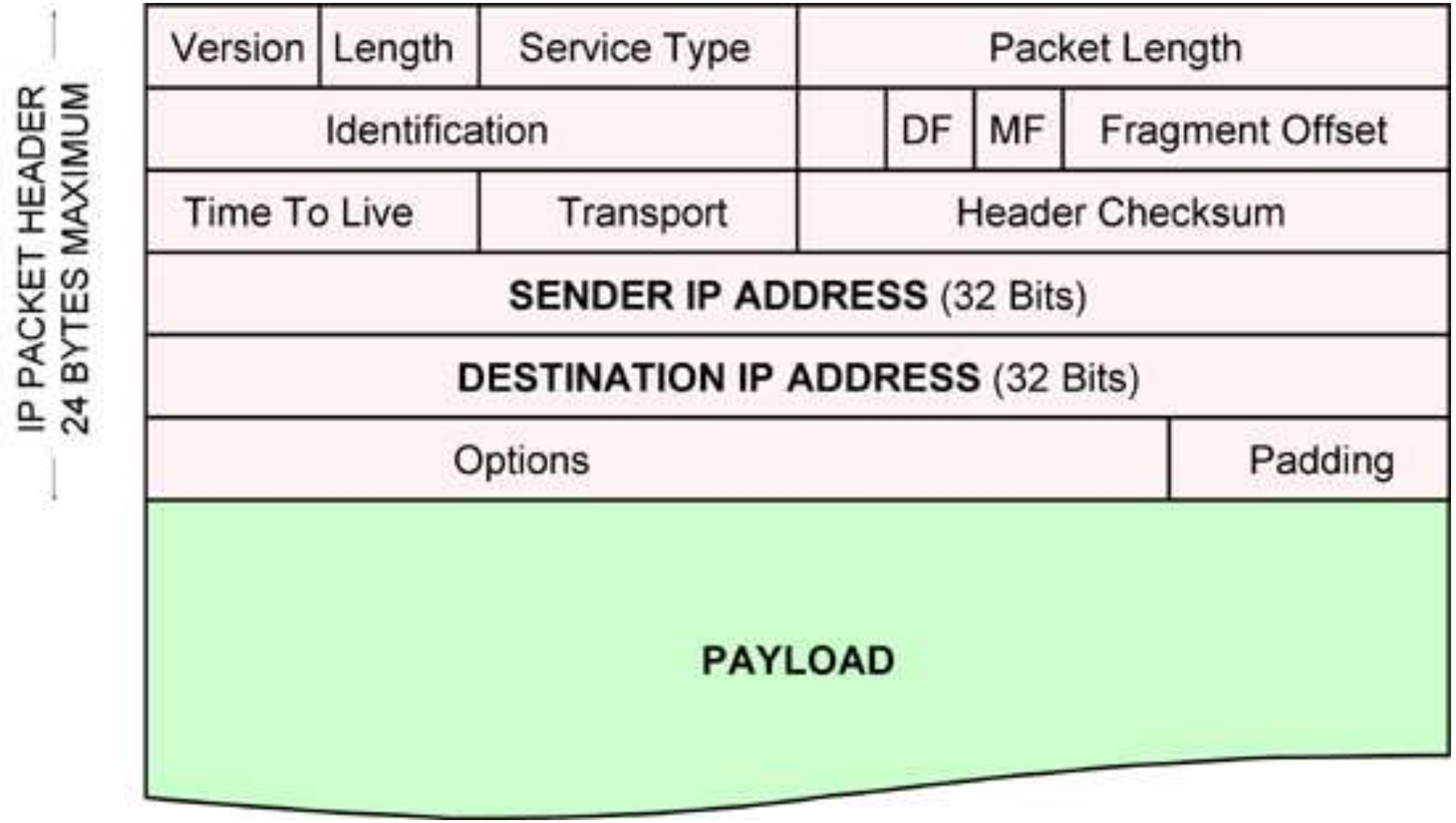
- **ethtool** can be used for viewing and setting ethernet layer parameters
  - ethtool enp0s3
  - ethtool –i enp0s3
  - You can use –s for changing generic options; it’s not advised to change link layer configurations though
- ip neighbour can be used for viewing arp table entries
  - In old Linux kernels, arp was the command to achieve the same goal

L3 Network Appliances

- Hardware that functions on the L3 (IP layer) layer (though not necessarily exclusively)
    - **Switches** (most modern ones operate on L3 to alleviate VLAN issues)
    - **Routers**
- Responsible for moving traffic from one network to another

The IP Packet

- Length: Internet Header Length (IHL) which is the length of the header
- Service Type: Type of Service (ToS) which contains a number of fields related to quality of service
- Packet Length: length of the packet (including data)
- Identification: ID of the packet

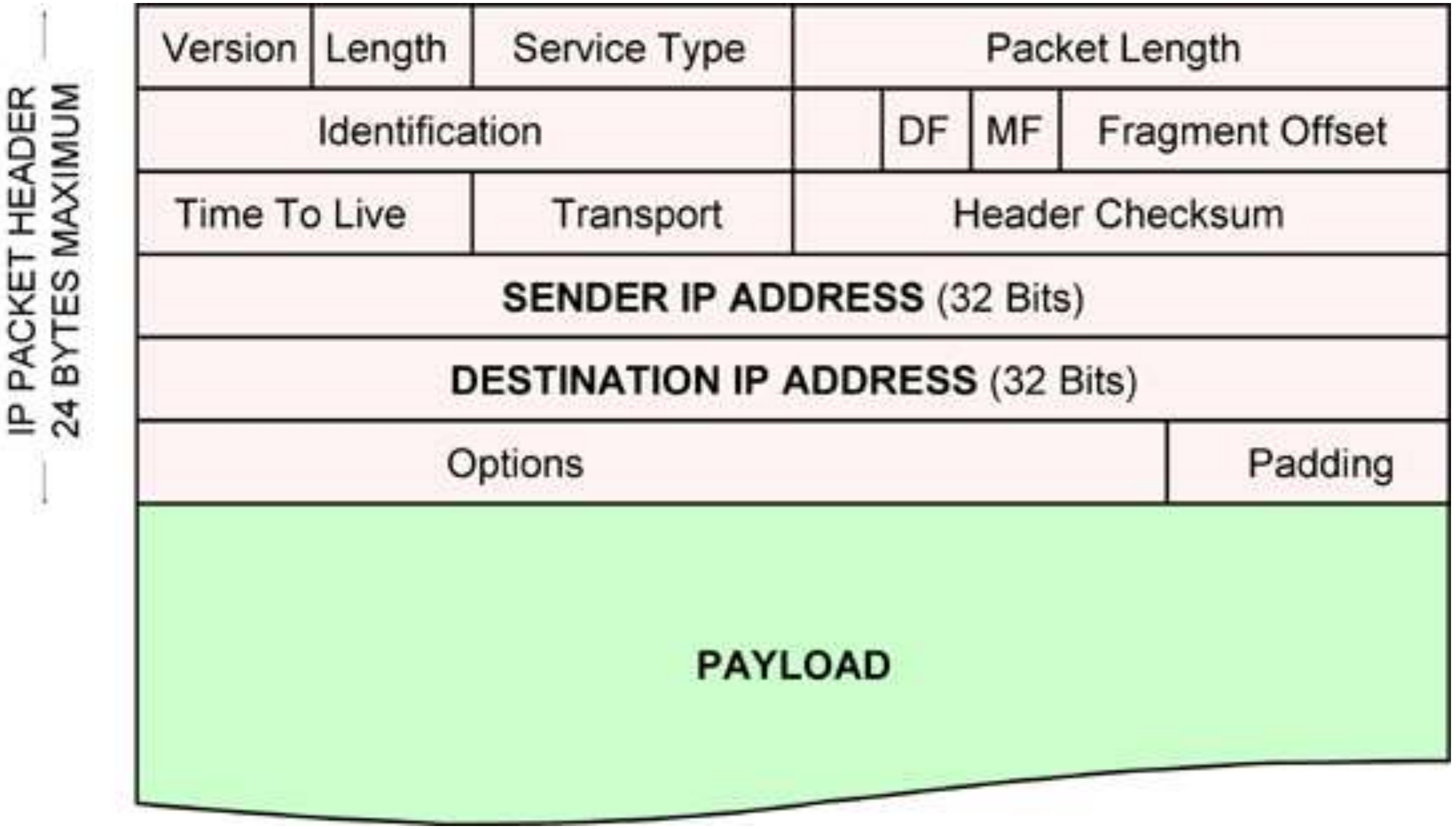


IP PACKET HEADER
24 BYTES MAXIMUM

Version	Length	Service Type	Packet Length			
Identification				DF	MF	Fragment Offset
Time To Live		Transport	Header Checksum			
SENDER IP ADDRESS (32 Bits)						
DESTINATION IP ADDRESS (32 Bits)						
Options						Padding
PAYLOAD						

The IP Packet

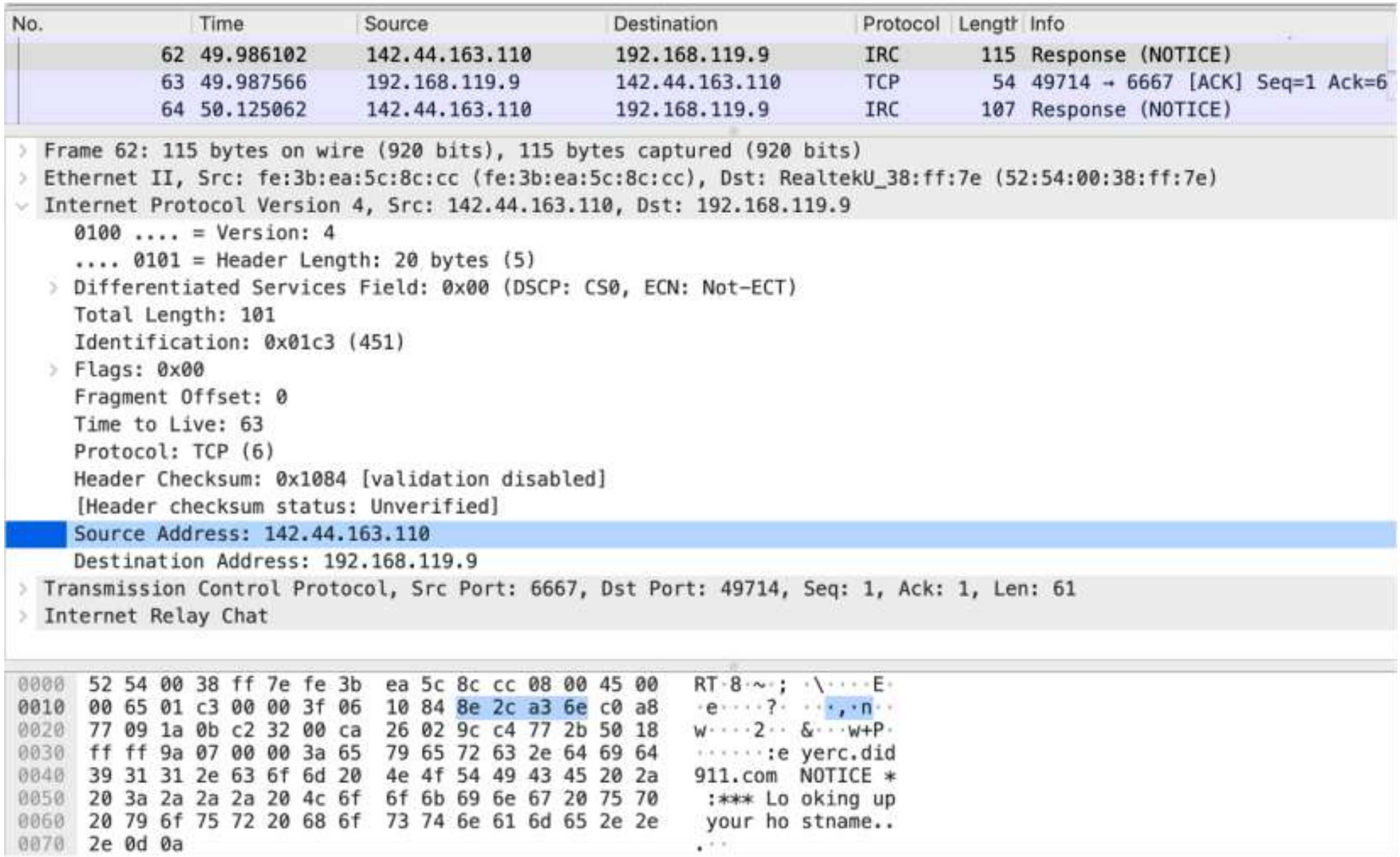
- Fragment: where in the order of packets this particular packet resides
- Time to Live (TTL): how many hops this packet has before it is discarded
- Transport: TCP or UDP
- Header Checksum: header validation test checksum
- Addresses: to and from IP addresses



IP PACKET HEADER
24 BYTES MAXIMUM

Version	Length	Service Type	Packet Length	
Identification		DF	MF	Fragment Offset
Time To Live	Transport	Header Checksum		
SENDER IP ADDRESS (32 Bits)				
DESTINATION IP ADDRESS (32 Bits)				
Options	Padding			
PAYLOAD				

IP layer



```
No.                  Time         Source            Destination       Protocol  Lengtr Info
                 62  49.986102    142.44.163.110    192.168.119.9     IRC        115 Response (NOTICE)
                 63  49.987566    192.168.119.9     142.44.163.110    TCP         54 49714 → 6667 [ACK] Seq=1 Ack=6
                 64  50.125062    142.44.163.110    192.168.119.9     IRC        107 Response (NOTICE)
```

```
> Frame 62: 115 bytes on wire (920 bits), 115 bytes captured (920 bits)
> Ethernet II, Src: fe:3b:ea:5c:8c:cc (fe:3b:ea:5c:8c:cc), Dst: RealtekU_38:ff:7e (52:54:00:38:ff:7e)
v Internet Protocol Version 4, Src: 142.44.163.110, Dst: 192.168.119.9
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 101
    Identification: 0x01c3 (451)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 63
    Protocol: TCP (6)
    Header Checksum: 0x1084 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 142.44.163.110
    Destination Address: 192.168.119.9
> Transmission Control Protocol, Src Port: 6667, Dst Port: 49714, Seq: 1, Ack: 1, Len: 61
> Internet Relay Chat
```

```
0000  52 54 00 38 ff 7e fe 3b  ea 5c 8c cc 08 00 45 00   RT·8·~·; ·\····E·
0010  00 65 01 c3 00 00 3f 06  10 84 8e 2c a3 6e c0 a8   ·e····?· ··,·n··
0020  77 09 1a 0b c2 32 00 ca  26 02 9c c4 77 2b 50 18   w····2·· &···w+P·
0030  ff ff 9a 07 00 00 3a 65  79 65 72 63 2e 64 69 64   ······:e yerc.did
0040  39 31 31 2e 63 6f 6d 20  4e 4f 54 49 43 45 20 2a   911.com  NOTICE *
0050  20 3a 2a 2a 2a 20 4c 6f  6f 6b 69 6e 67 20 75 70    :*** Lo oking up
0060  20 79 6f 75 72 20 68 6f  73 74 6e 61 6d 65 2e 2e    your ho stname..
0070  2e 0d 0a                                           .··
```
]

The IP Address

- 32 bit unique network identifier, allowing for 4,294,967,296 (2^{32}) addresses
- Typically written in “quad dot” format, where each dot separates 8 bits
  - 142.44.163.110 == 8e.2c.a3.6e
  - 142.44.163.110 == 10001110.00101100.10100011.01101110
- The above is specifically for IPv4, an older IP standard which has been “replaced” with IPv6, a 128 bit unique network identifier
- While it is highly advised that you use IPv6 whenever possible, IPv4 is still very pervasive in the wild and is easier to use when working through examples

IP Address Notation

- **Address** (187.13.56.7) is unique and can identify a unique device
- **Subnet mask** (255.248.0.0) tells us how many hosts can be connected to a network and gives us an IP range to assign to those hosts
- **Network address** (187.8.0.0) defines what network an IP address belongs to, and is a subset of the full IP address
- **Broadcast address** (187.15.255.255) is a reserved address to send broadcasts on
- **CIDR notation** (187.13.56.7/13) specifies how many bits of the IP address are reserved for the network addressing and how many are reserved for the host addressing

IP Address Notation

187.13.56.7/13
IP Address

1	0	1	1	1	0	1	1	.	0	0	0	0	1	1	0	1	.	0	0	1	1	1	0	0	0	.	0	0	0	0	0	1	1	1

255.248.0.0
Subnet Mask

1	1	1	1	1	1	1	1	.	1	1	1	1	1	0	0	0	.	0	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0

187.8.0.0
Network Address

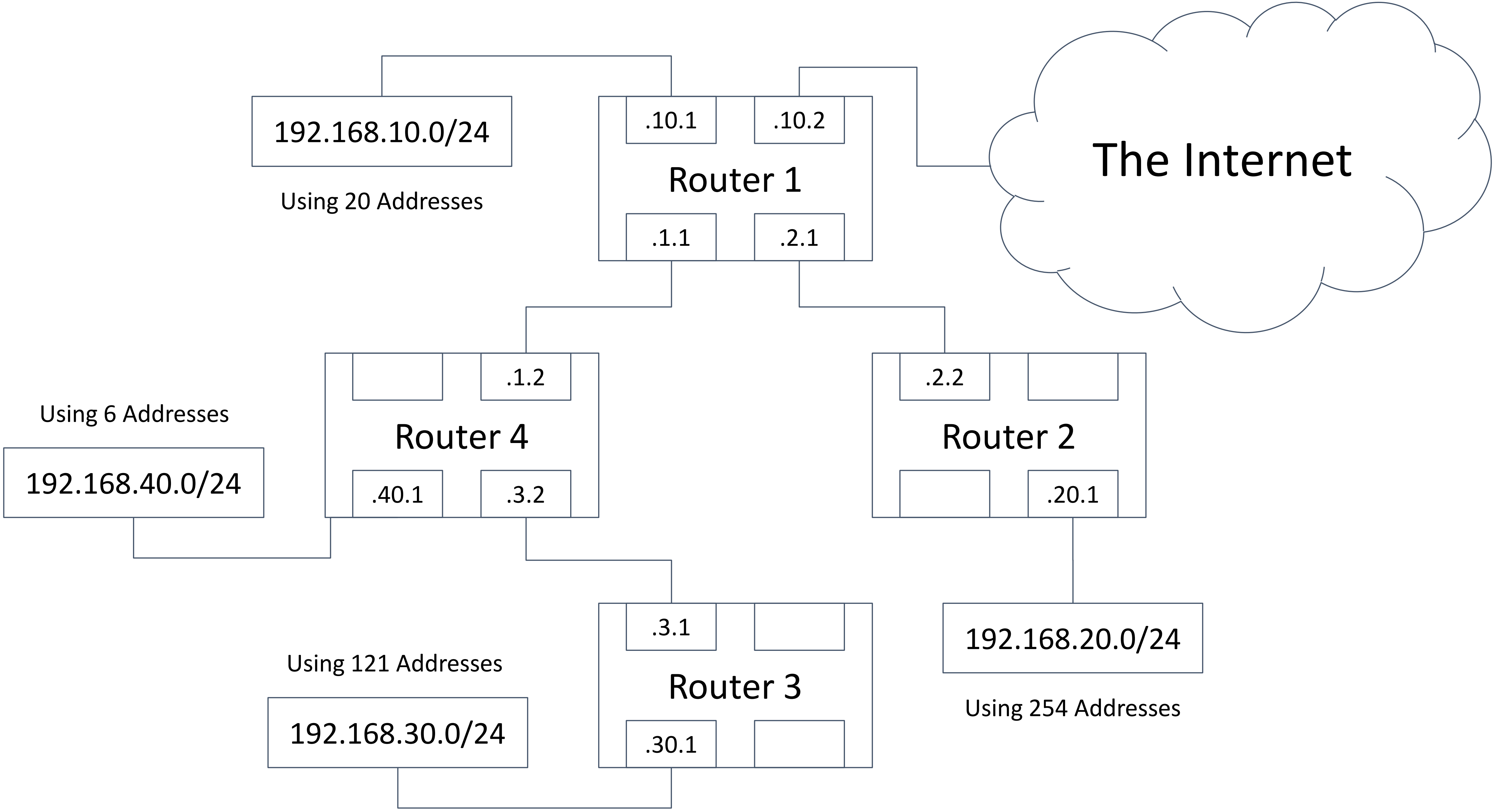
1	0	1	1	1	0	1	1	.	0	0	0	0	1	0	0	0	.	0	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0

187.15.255.255
Broadcast Address = IP | !Subnet Mask

1	0	1	1	1	0	1	1	.	0	0	0	0	1	1	1	1	.	1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1

The Routing Table

- The routing table performs a similar function to the switching table in L2 and is responsible for making sure traffic gets to where it needs to go
- The routing table is primarily made up of the following fields:
    - Network ID & Mask represents an available network a device can route traffic to (0.0.0.0 used to route traffic not otherwise in the table)
    - Gateway (Next Hop) is the next L3 appliance interface's IP address that must be traversed to reach the destination
    - Interface is the port* used to reach the next hop
    - Metric is the cost of getting to the next hop through the interface


192.168.10.0/24
Using 20 Addresses
.10.1
.10.2
Router 1
.1.1
.2.1
The Internet
Using 6 Addresses
192.168.40.0/24
.1.2
Router 4
.40.1
.3.2
.2.2
Router 2
.20.1
.3.1
Router 3
.30.1
Using 121 Addresses
192.168.30.0/24
192.168.20.0/24
Using 254 Addresses

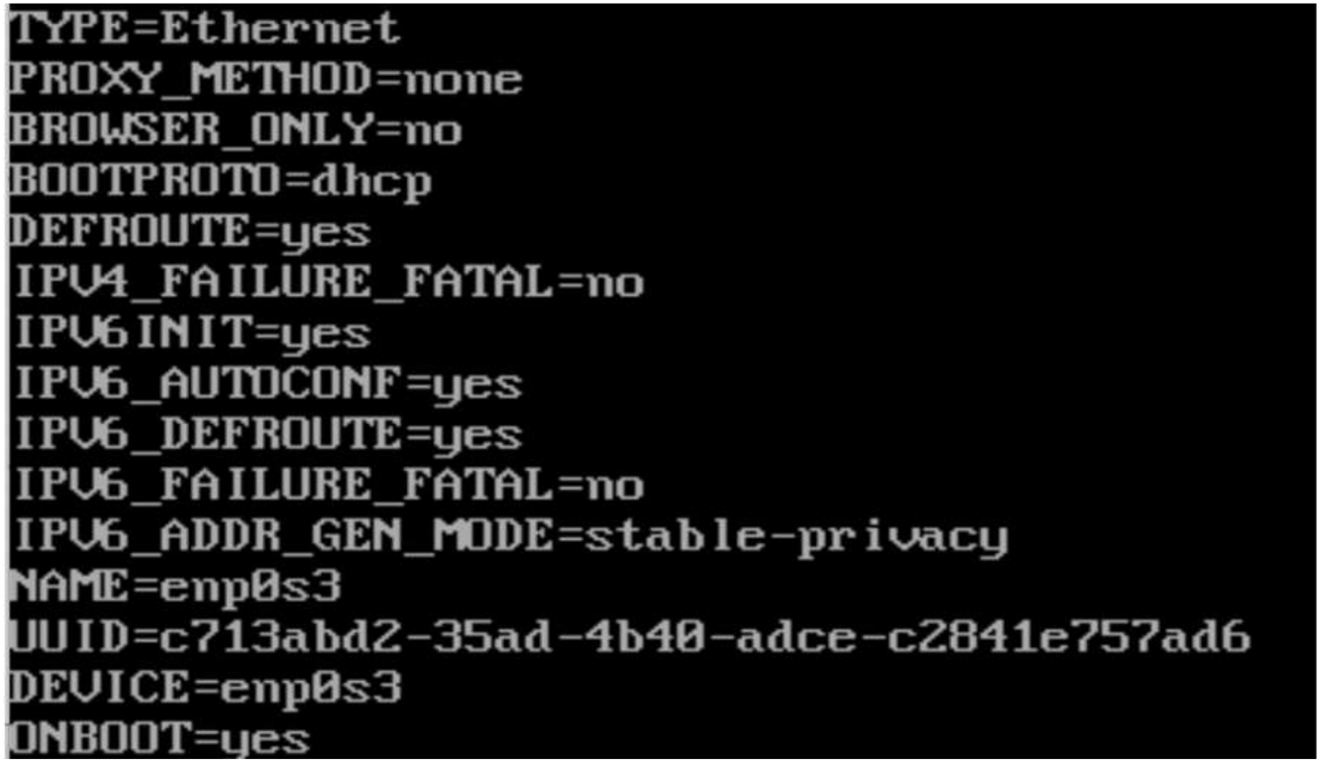
Setting up Network configurations

- Assuming that the network card is installed, and the cable is connected:
  - We need to setup the IP address
  - We need to setup the default gateway
  - We need to setup the name servers (DNS servers)
- After the setup, we need to verify the configurations:
  - ping: checking whether a host can receive our traffic
  - nslookup: checking whether the DNS server can resolve our requests

Network Interface IP Address

- The configurations are stored in:
  - /etc/sysconfig/network-scripts/ifcfg-[interface-name]

- BOOTPROTO says the IP will be acquired from DHCP
- ONBOOT says this interface will be enabled after the boot
- Static IP address can be set by:
  - IPADDR=X.X.X.X
  - NETMASK=Y.Y.Y.Y



```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp0s3
UUID=c713abd2-35ad-4b40-adce-c2841e757ad6
DEVICE=enp0s3
ONBOOT=yes
```

Connecting to Internet

- We need to setup the default gateway and a DNS server
- Default Gateway can be set in the /etc/sysconfig/network file
    - GATEWAY=10.0.2.2
    - HOSTNAME=example.cs183.org
- DNS server can be set in the /etc/resolv.conf
    - nameserver 8.8.8.8

L3 ACLs and Firewalls

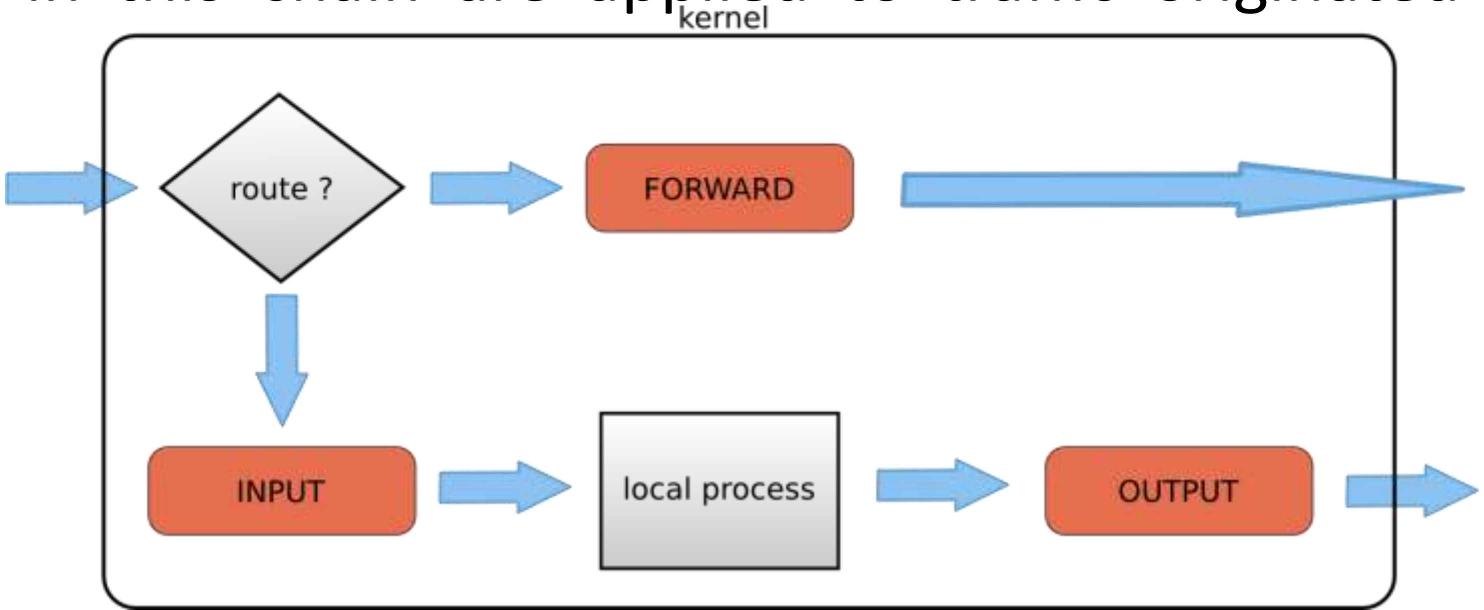
- While ACLs and Firewalls can be deployed on L2, 3, and 4 they are most often utilized in L3 (IP traffic) and L4/7 (Port traffic) to perform either packet blocking/shaping or “application” blocking/shaping.
- ACL rules can be based on source address, destination address, and port (although technically this is an L4/7 ACL, most routers can use this info)
    - Incoming packets are compared to ACL entries based on the order that the entries occur in the router they are passing through
    - If the packet does not match an Access Control Entry (ACE), the packet is then matched against the next ACE in the list
    - If a packet and an access list statement match, the rest of the statements in the list are skipped
    - If no conditions match an ACE, the packet is dropped

iptables

- Since Linux Kernel version 2.4, a packet handling engine called Netfilter comes with every Linux flavor
    - iptables is the command-line tool to manage it
- iptables applies ordered “chains” of rules to network packets
- Sets of chains make up “tables”, and hence the name iptabales
- iptables has “filter”, “nat” and “mangle” tables
    - We briefly review the “filter” table today
- Chains have rules and each rule has a “target” clause
    - The “target” clause determines what to do with matching packets
    - Targets in the filter table are ACCEPT, DROP, REJECT, LOG, ULOG, REDIRECT, RETURN, MIRROR and QUEUE

iptables filter table

- FORWARD chain: rules in this chain are applied to all packets that arrive on one network interface and need to be forwarded to another
- INPUT chain: rules in this chain are applied to traffic addressed to the local host
- OUTPUT chain: rules in this chain are applied to traffic originated from local host



iptables firewall setup

- Three main forms for rules:
  - iptables –F [chain-name]: flushes all prior rules
  - Iptables –P [chain-name] target: default chain target
  - Iptables –A [chain-name] -i [interface] -j target
- The rules are usually placed in the rc startup script

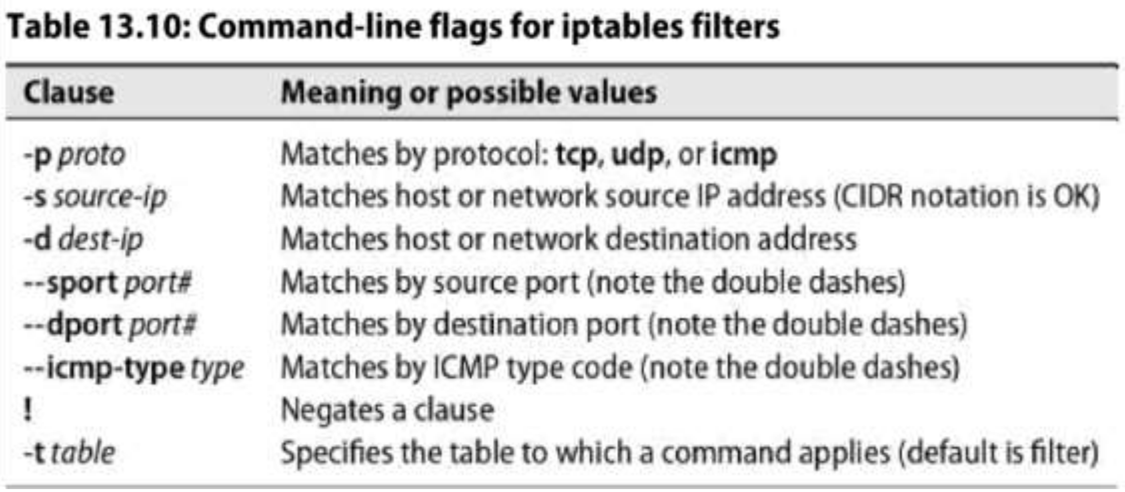


Table 13.10: Command-line flags for iptables filters

Clause	Meaning or possible values
-p *proto*	Matches by protocol: **tcp**, **udp**, or **icmp**
-s *source-ip*	Matches host or network source IP address (CIDR notation is OK)
-d *dest-ip*	Matches host or network destination address
--sport *port#*	Matches by source port (note the double dashes)
--dport *port#*	Matches by destination port (note the double dashes)
--icmp-type *type*	Matches by ICMP type code (note the double dashes)
!	Negates a clause
-t *table*	Specifies the table to which a command applies (default is filter)

iptables packet filtering - Example

- We want to allow traffic only to port 22 of a particular target
  - iptables –F
  - iptables –P INPUT DROP
  - Iptables –P FORWARD DROP
  - Iptables –A INPUT –i enp0s3 –d 10.0.2.22 –p tcp --dport 22 –j ACCEPT
  - Iptables –A INPUT –i enp0s3 –d 10.0.2.22 –p icmp --icmp-type 8 –j ACCEPT

Additional reading

What is IP (Internet Protocol)

Subnet Calculator

Network configuration and information

Masquerading Made Simple