

## Chapter 3: Internet Protocol

A *simple* network layer protocol.

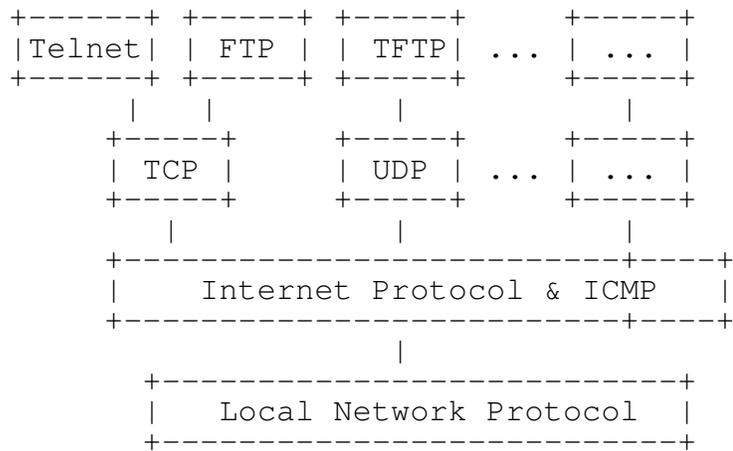
### Services

Connectionless  
Unreliable

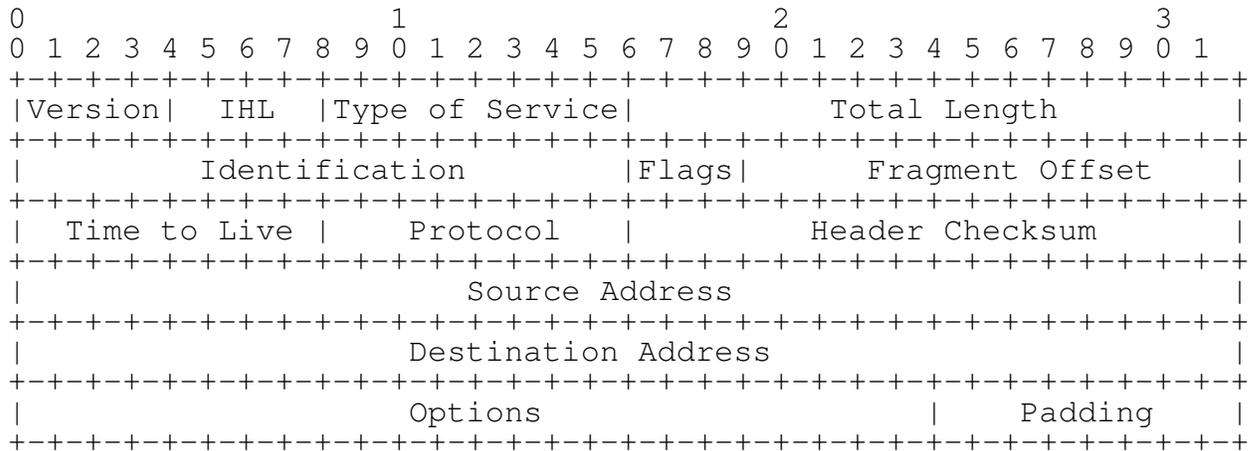
### Functions

Routing  
Congestion control via packet discarding  
Fragmentation and reassembly at the destination

### Relation to other Protocols



## IP Header



Example Internet Datagram Header

Version of IP	4
Length of Header in words	Typically 5
Type of service options (Typically ignored!)	Precedence (3 bits) .. followed by the DTRC bits (Minimize delay (telnet.. interactive stuff) Maximize throughput (ftp data) Maximize reliability (IGP, SNMP) Minimize cost (NNTP)
Datagram length in bytes	Maximum of 64K
Datagram id	Typically incremented by 1 by each host. (Used in reassembly)
Fragmentation flags	(Reserved, DF, MF)
Fragment offset	In 8 byte units
Time to live	Decrement by 1 at each hop At 0 packet is discarded and sender notified (tracerte)
Protocol	TCP, UDP, see IANA RFC for values
Checksum	Header checksum.. 1's complement of the one's complement sum (See RFC 1071).. Checksum error -> discard datagram. Must be recalculated each time.
Source IP address	
Dest IP address	
Options	Security and handling restriction, record route, time stamp, loose/strict source routing.

### The checksum algorithm:

```
int in_cksum(u_short *p, int n)
{
    register u_short answer;
    register long sum = 0;
    u_short odd_byte = 0;

    while( n > 1 )
        { sum += *p++; n -= 2; }

    /* mop up an odd byte, if necessary */

    if( n == 1 )
    {
        *(u_char *)&odd_byte = *(u_char *)p;
        sum += odd_byte;
    }

    sum = (sum >> 16) + (sum & 0xffff); /* add hi 16 to low 16 */
    sum += (sum >> 16);                /* add carry */
    answer = ~sum;                      /* ones-complement, truncate */
    return (answer);
}
```

## Routing in IP

A hierarchical routing scheme is now in use:

Exterior Gateway Protocols (EGPs) (e.g. BGP) provide connectivity among different *Autonomous Systems (AS)*

Interior Gateway Protocols (IGPs) (e.g. OSPF) provide connectivity within an Autonomous system..

Static routing tables (Used for Host level routing)

Routing decisions made by edge routers and hosts

Router need not know the entire path  
==> Routing function is to *determine the next hop*

Based on contents of *routing table*

(Dest IP) (Next Hop IP) (Netmask/Pfx Len) (Flags) (Intf)

If *Packet Destination* best matches a *Table Destination*, send to the the *Next Hop* specified in the table entry

Routing decisions are typically simple for a host:

If dest directly connected via LAN or Point to Point  
Send to Dest  
else  
Send to default router.

Linux host routing table:

```
local/westall ==> netstat -nr
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt  Iface
130.127.48.113  0.0.0.0         255.255.255.255 UH      40  0        0    eth0
192.168.2.0     0.0.0.0         255.255.255.0  U       40  0        0    lec0
130.127.48.0   0.0.0.0         255.255.255.0  U       40  0        0    eth0
127.0.0.0      0.0.0.0         255.0.0.0      U       40  0        0    lo
0.0.0.0        130.127.48.1   0.0.0.0        UG      40  0        0    eth0
```

SUN host routing table:

Destination	Gateway	Flags	Ref	Use	Interface
localhost	localhost	UH	0	3840	lo0
130.127.48.0	jmw	U	3	42289	le0
BASE-ADDRESS.MCAST.NET	jmw	U	3	0	le0
default	130.127.48.1	UG	0	2058	

## Routing table elements

Destination host or network address (host bits = 0 => network address)

IP address of next hop router or directly connected network

Need this so that ARP can resolve *mac* address.

Next hop *must* be directly attached.

Flags

Interface specification... access to the output queue of the selected link driver.

## Routing table lookup scheme

Search for complete (network, host) match.

Search for matching network address (may have to consider subnetworks).

Search for a default router (Citron)

If none found return "host unreachable".

## Subnet addressing

Now required for all hosts (RFC 950)

Example of a typical class B subnet mask 255.255.255.0

0 bits in the subnet mask identify the bits reserved for host id.

Given IP address and subnet mask a host can tell if any other address is

- a host on its own subnetwork
- on a different subnet but on the same network
- on a different network.

## Advantage of subnetworking

Provides for hierarchical routing

External routers need know about only 130.127 to reach any system at CU

Internal routers at Clemson know about 130.127.n

n = 1, 254 (total subnetworks possible)

n = 48 => Computer Science Dept.

**Example in book (p. 44):**

Host address is 140.252.1.1  
Subnet mask is 255.255.255.0

IP address 140.252.4.5 is on same network but a different subnetwork  
Packet must be sent to the default router  
ARP can't be used to obtain MAC address

IP address 140.252.1.22 is on the same network and subnet  
Packet doesn't need to be sent to a router  
ARP can be used to obtain MAC address

IP address 192.43.235.6 is on the a different network  
Packet must be sent to the default router

Exercises:

What if 140.252.4.5 really *is* on the same ethernet  
What if 140.252.1.22 really *isn't* on the same ethernet

Subnet-directed broadcast includes subnet id and host of 255 (-1)

### Example in book p.46 (multiple subnet masks)

Note that Sun has a different subnet mask associated with each interface

The other (non 140.152.13) residents of 140.252 view the systems *slip*, *bsd1*, and *svr4* as residents of net 140.252.13

Residents of 140.252.1 must know (via routing tables) to send 140.252.13 traffic to 140.252.1.29 (normally only the default router for 140.252.1 will have a static definition of this route).

Residents of 140.252.n (n != 13 and n != 1) must have a default router on 140.252.n that has:

- An interface on 140.252.1 or
- Knows how to reach 140.252.1

Using the 255.255.255.0 subnet mask — subnet 13 can have up to 254 hosts

By changing the subnet mask to 255.255.255.224

- Subnet 13 can be further subdivided into 6 networks (3 additional bits – but can't use 0 and 7)

- Network ids are 32, 64, 92, 128, 160, 192

- Each subnet can have 30 hosts (5 bits)

- And sane routing can be employed.

- There is nothing special about the *slip* net — it just as well be another ethernet with up 30 hosts.

Exercise: Suppose a netmask of 255.255.255.0 *were* used on net 13. How would routing tables have to differ to make everything work correctly..

## Host commands used to manage interfaces and routing

The *ifconfig* command is used to

- Bind an link level interface to a network address
- Associate a netmask with an interface
- Display status of interfaces
- It is normally run at system boot time via */etc/rc.d* scripts

```
==> /sbin/ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:B0:D0:E9:0F:5B
          inet addr:192.168.2.33  Bcast:192.168.2.63  Mask:255.255.255.224
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4771 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22017 errors:0 dropped:0 overruns:0 carrier:0
          collisions:2814 txqueuelen:100
          Interrupt:16 Base address:0xec80

lec0      Link encap:Ethernet  HWaddr 00:00:77:8E:6D:B9
          inet addr:192.168.2.8   Bcast:192.168.2.255  Mask:255.255.255.224
          UP BROADCAST RUNNING MULTICAST  MTU:1492  Metric:1
          RX packets:61704714 errors:0 dropped:0 overruns:0 frame:0
          TX packets:32632157 errors:0 dropped:61 overruns:0 carrier:0
          collisions:0 txqueuelen:100

lo        Link encap:Local Loopback
          inet addr:127.0.0.1   Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:987 errors:0 dropped:0 overruns:0 frame:0
          TX packets:987 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

/sbin/ifconfig eth1 node1 up mtu $1 broadcast 10.1.255.255 netmask 255.255.0.0
```

The *route* command can be used to

- Add new routes
- Delete existing routes
- Display the current routing table

```
route add -net 192.168.3.0 netmask 255.255.255.0 gw 192.168.2.14
route add -net 192.168.2.0 netmask 255.255.255.0 dev lec1
```

The *netstat* command can also be used to display  
interface information (*-i*) and  
routing table information (*-r*)

Exercises: Run *ifconfig* and *netstat* on your favorite system and determine what information is provided.

Exercises: 3.2, 3.3, 3.4