Internet Programming & Protocols Lecture 2

Addressing

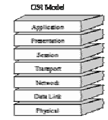
Ethernet

The Internet

IP

OSI reference model

- physical -- bit stream (wire, optical, wireless)
- data link -- packets on the link (FDDI, ethernet, token ring)
- network -- connects links, routers (IP)
- transport -- reliable stream (TCP, UDP)
- session -- more reliable (SSL)
- presentation -- canonical form (API, data conversion)
- application -- mail, telnet, http, ssh, etc.



Layers/encapsulation

Protocol Relationships

```
+------+ +-----+ +-----+     +-----+
|Telnet| | FTP | | TFTP| ... | ... |      Application
+------+ +-----+ +-----+     +-----+
|   |         |           |
+-----+     +-----+     +-----+
| TCP |     | UDP | ... | ... |      Transport
+-----+     +-----+     +-----+
|           |           |
+--------------------------+----+
|    Internet Protocol & ICMP   |    Network
+--------------------------+----+
|
+---------------------------+
|   Local Network Protocol  |      Data link
+---------------------------+
```

Protocol encapsulation

```
16       20       20/8                  4
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- ....... -+-+-+-+
| mac    |    IP   |TCP/UDP|  App/Data ..... | CRC |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- ....... -+-+-+-+
```

Data is carried in packets. Packets are intermixed.

The low levels

- Physical layer is concerned with putting bits on the media
  - A 1 megabit/second media, means bits are spaced 1 microsecond apart
  - Information bits encoded as change of voltage, amplitude, frequency
  - Telegraph, smoke signals, modem, ethernet ….
  - Part of NIC (network interface hardware)
- Link layer is concerned with combining bits into messages/packets/cells
  - Additional bits are added for addressing, error checking, type information
  - Error checking (often defined expected error rate: 1 bit loss in 10^9)
    - parity/CRC or ECC
    - Wireless is lossy, fiber is not
    - What to do if there is an error? … usually receiver NIC drops packet
  - Usually a "maximum message size" (MTU == Maximum Transmission Unit)
  - Media access protocol (e.g. CSMA/CD)
  - NIC manages link layer and often has link address encoded in hardware
- Need special equipment to diagnose low-level problems
  - Loose wire, full/half duplex mismatch, poor connection, RF interference

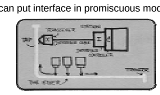
Ethernet

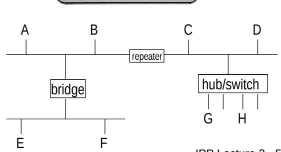
- Xerox, DEC, Intel, '76
- 10 million bits/sec (100, GigE, 10Gige)
- CSMA/CD
- thick, thin, fiber, twisted pair, wireless
- min packet (60 bytes)
- max pkt (1500) (9KB for jumbo-frame GigE)
- 6-byte address (vendor(3)+other(3)) (MAC)
- supports broadcast and multicast

```
0     7 8    15 16   23 24   31 32   39 40   47
+-------+-------+-------+-------+-------+-------+
|              Destination address              |
+-------+-------+-------+-------+-------+-------+
|                Source address                 |
+-------+-------+-------+-------+-------+-------+
|   type        |    data  ...
+-------+-------+-------+--...
....
+-------+-------+-------+-------+
|            checksum CRC       |
+-------+-------+-------+-------+
```

Microsoft stashes ether address in WORD documents – unique ID!

- inexpensive, pervasive
- physical and link layer spec (IEEE 802)
- carry IP, DECnet, appletalk, IPX (type field)
- packets travel by every interface, party line
- interface recognizes its own address and broadcast
- can program interface to recognize multicast
- can change interface address ! (impersonation)
- can put interface in promiscuous mode





CSMA/CD

- Ethernet is party-line, everyone hears
- Only ONE device can be talking at time!
- Carrier sense multiple access/ collision detect (CSMA/CD)
  - Manage shared media (only one NIC can transmit at a time)
  - Transmitter waits til no one transmitting, then transmits
  - Transmitter listens while it transmits (transmission delay)
  - If someone else starts at "same" time, transmitter sends a jam signal (48 bits) and backs off
  - Back off is exponential
    - After experiencing n^{th} collision in a row, sender chooses a backoff time randomly from 0 … 2^n

    ANIMATION
- Collisions are handled by link layer (NIC)
  - NIC usually keeps a count that can be queried by driver/OS
  - Collisions will SLOW performance
  - How your cable modem competes with your neighbors

Ethernet NIC

- card/chip takes care of CSMA/CD, encoding, packet spacing, preamble/CRC – unique ethernet address “wired” into card
- control: commands and status
- control
  - add/delete multicast address
  - enable/disable promiscuous
  - set MAC address (DECnet)
  - Full/half duplex for twisted pair (10/100/1000)
- status: collisions, interrupts, ready
- drops "bad" packets (CRC failures)
- passes up own/broadcast/multicast pkts
- limited buffering
- kernel driver is the interface
- driver passes packet up to type handler

Ethernet type field

hex	
800	IPv4
806	ARP
600	XNS
8137	Novell
8035	reverse ARP
86DD	IPv6

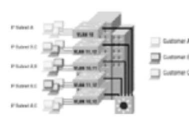
Ethernet NIC info from unix

```
 ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:C0:4F:6B:A5:52
          inet addr:160.36.58.221  Bcast:160.36.59.255 Mask:255.255.252.0
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:94907277 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35670805 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
```


smart link layer

- hubs pass all traffic to all ports ☹
- switches/bridges only pass multicast and matching destination traffic
- VLANs based on even smarter layer 2 switch
  - Ports tagged (802.1Q)
  - Ports can be grouped into virtual LANs
  - Control port to configure switch

- Note that different network layer protocols (e.g. DECnet, IP, SNA) may coexist on the same link. Ethernet type field distinguishes IEEE 802.3



VLAN for different customers dispersed within a building

addressing

- Simple point-to-point link, don’t need no stinkin’ address, but not a very interesting network
- Addresses are needed in data-link layer (e.g. Ethernet address)
  - As packet travels, physical addresses will change for each link
  - MTU may change from link to link …. a problem?
  - Worry about uniqueness? (ether: vendor+number)
- Network layer addresses don’t change (e.g. IP address) for a packet
  - Destination address is used for routing
  - People don’t like number, so there are “host names” for addresses
- Higher level addresses (application/server/process == port number)
- Issues of mapping addresses
  - Services to port number (predefined or portmap)
  - Names to network addresses (e.g. /etc/hosts or DNS)
  - Mapping network addresses to physical addresses (DHCP/ARP)
  - Current research: a host may have multiple addresses, you just want to talk to that host, don’t care which friggin’ address. Host id?

Addressing

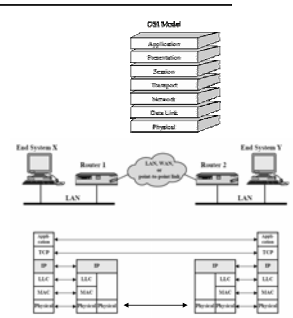
- Address: service (port), host
- network name to number translation (DNS)
- network to physical mapping (ARP)

32-bit internet address (IPv4)
 unique
 assigned by authority
 clumped in A, B, or C
 D is multicast
 net.255.255 is broadcast
Private (NAT) RFC 1918:
 10.0.0.0
 172.16.0.0
 192.168.0.0
IPv6 128-bit address

```
           +--------+--------+
DECnet     |area(6)  host(10)|
           +--------+--------+
               16 bits

           +-----------------------------------+
IPv6       |sub  |  subnet  |    interface     |
           +-----------------------------------+
                         128 bits

           +--------+--------+--------+--------+
IP    A    |0 net(7)|   host (24)              |
           +--------+--------+--------+--------+
                         32 bits
           +--------+--------+--------+--------+
      B    |10   net (14) |     host (16)      |
           +--------+--------+--------+--------+
           +--------+--------+--------+--------+
      C    |110   net (21)          |host(8)   |
           +--------+--------+--------+--------+
           +--------+--------+--------+--------+
      D    |1110     multicast (28)            |
           +--------+--------+--------+--------+

           +---+---+---+---+---+---+
Ethernet   | vendor(24)| local (24)|  broadcast: -1
           +---+---+---+---+---+---+
                         48 bits
IP multicast  0x01 00 5E
```


The Internet protocols

- Physical/data link layer: Ethernet, ATM, FDDI, …
- Network layer: IP
- Transport layer: ICMP/UDP/TCP
- Session/presentation: sockets/XDR
- Application: http/mail/ssh

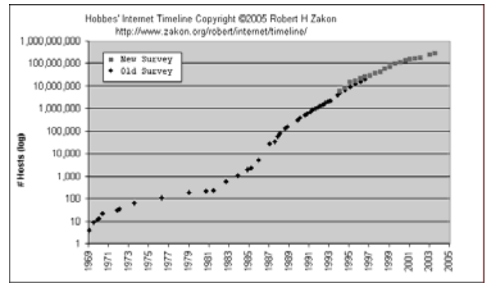


Ethernet	IP	TCP/UDP	Application

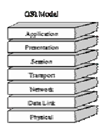
Internet history

- Developed in late 70's
  - Initially small community of users
  - Initial goals: scalability and ease of use
  - DARPA's interest: survivability
  - Protocols have grown and evolved
    - TCP/IP was originally a single protocol
    - TCP has been tweaked to accommodate new media and loads
  - Open design (non-proprietary)
  - Big boost from being distributed free as part of Berkeley UNIX in 80s
- Today Internet is a voluntary world-wide federation of networks
  - No central authority, no common culture
  - Links millions of people and organizations (competitors, enemies)
  - Voluntary (critical) services include routing and naming (DNS)
  - Routers and servers are just computers
  - As a packet travels across the internet it may pass thru several countries, over different media, and through different "administrative domains"

Internet growth



Network layer (IP)

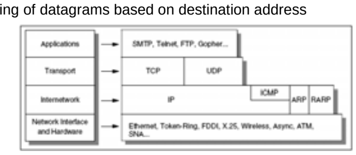


- Manage some end-to-end issues
  - Routing
  - Errors
- End node addressing
- Independent of link/physical layer
  - Almost, IP handles MTU issues (fragmentation)
- Transparently carries transport/application data
- Interfaces to data/link layer below (lots of media) and to the transport layer above

I Pee

DECnet
SNA OSI

- Internet Protocol (IP)
- Defined by RFC 791 (IP version 4)
- Network layer
  - Datagrams (more "survivable" than circuit based – DARPA)
  - Deliver datagrams from sender to receiver
  - Unreliable (best effort)
- End nodes distinguished by unique 32-bit address
- Routing of datagrams based on destination address



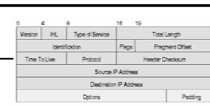
IP header

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

20 bytes to 60 (with options)

transmitted bits 0-7 first, network byte order

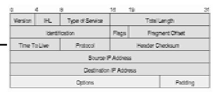
IP header



- Version, 4 bits

Version	Description
0	Reserved.
1 2 3	
4	IP, Internet Protocol.
5	ST, ST Datagram Mode.
6	SIP, Simple Internet Protocol. SIPP, Simple Internet Protocol Plus. IPv6, Internet Protocol.
7	TP/IX, The Next Internet.
8	PIP, The P Internet Protocol.
9	TUBA
10 - 14	
15	reserved.

IP header



- IHL, Internet Header Length (4 bits)
  - Units of words (32 bits)
  - Minimum is 5 (bigger if IP options)
- Type of Service (8 bits, really used?)

00	01	02	03	04	05	06	07
Precedence			D	T	R	M	0

Precedence

Value	Description
0	Routine.
1	Priority.
2	Immediate.
3	Flash.
4	Flash override.
5	CRITIC/ECP.
6	Internetwork control.
7	Network control.

D minimize delay (0 normal, 1 low delay)

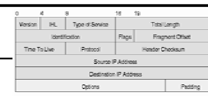
T maximize throughput (0 normal, 1 high)

R reliability (0 normal, 1 high)

M minimize cost (0 normal, 1 minimize cost)

•Total length (16 bits) – datagram size in bytes (max 65,535)

IP header



- Id, Flags, and Fragment offset are used for packet fragmentation
- Identification field (16 bits)
  - Incremented by 1 for each packet sent by host
  - Fragments will carry the same ID field, so they can be reassembled
- Flag field (3 bits)

00	01	02
R	DF	MF

  - R reserved
  - DF (0 may fragment, 1 DON'T fragment) (ICMP_UNREACH_NEEDFRAG)
  - MF (0 last fragment, 1 more fragments)
- Fragment Offset (13 bits)
  - Offset of this fragment in units of 8 bytes
  - Used to reassemble

IP fragmentation

- As packet travels from router to router, link layer changes, so MTU may change
- If next link has MTU smaller than packet, the packet must be fragmented by the router
- The receiving host is responsible for reassembling the fragments back into a complete IP packet
- UDP (NFS) can generate datagrams bigger than host MTU
- TCP goes to some effort to avoid IP fragmentation
  - Maximum segment size negotiation
  - MTU discovery protocol DF + ICMP (…. more later)
- Receiving host has to accumulate fragments and when (if) all arrive, assemble the fragments into an IP packet
  - Uses IP ID field to manage different frags
  - 30 second timer before it gives up (can be lost, out of order, dups …)
  - Fragments have been used to blue-screen Windows and to slip by firewalls
- IP v6 does away with this (network layer/link layer interaction)

IP fragmentation

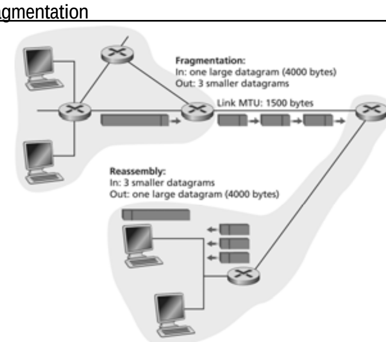
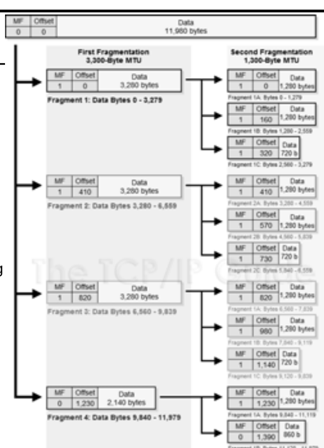


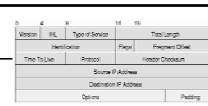
Figure 4.14 ♦ IP fragmentation and reassembly

Network	MTU
Hyperchannel	65535
ATM/AAL5	65535
token ring	17914
IP/ATM	9180
GigE jumbo	9180
FDDI	4352
Ethernet	1500
PPP	296

Big MTU is good thing



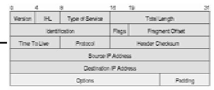
IP header



- Time to live (8 bits)
  - Counter that is decremented at each router hop
  - When counter goes to zero, packet dropped (ICMP sent to sender)
  - Keeps packets from bouncing around the internet forever!
  - OS's differ in setting initial value (64, 255, …)
  - traceroute messes with TTL
- Protocol (8 bits)
  - Indicates what the payload is (so the receiving OS can pass it to proper transport module)
- Header checksum (16 bit)
  - One's complement checksum of JUST the IP header
  - Checked and recalculated at each hop
    - Changing IP fields (TTL, possibly frag fields)
    - Checksum fails – packet is dropped (silently)

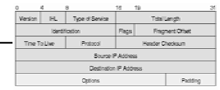
Protocols	
1	ICMP
4	Encapsulated IP
6	TCP
8	EGP
17	UDP
47	GRE
50	ESP (encrypted)
51	AH (authenticated
89	OSPF

IP header



- Source address (32 bits)
  - IP address of sender
  - NAT (network address translation) may muck with this
  - Hackers may set this randomly (spoof/impersonate)
  - For UDP, used to send a reply
  - Ignored by routers (like “return address” on USmail)
- Destination address (32 bits)
  - Where packet is destined
  - “network portion” of address is used by routers
- 32 bits is 4 billion hosts

IP header



- Options (optional, infrequent, check IHL)
  - **C** copy flag (1 ➔ copy to all fragments)
  - **Class** (0 control, 1 reserved, 2 debugging, 3 reserved)
  - **Option**

Option	Copy	Class	Value	Length	Description	References
0	0	0	0	1	End of options list	
1	0	0	1	1	NOP	
2	1	0	130	11	Security	
3	1	0	131	variable	Loose Source Route	
4	0	2	68	variable	Time stamp	RFC 781, RFC 791
5	1	0	133	3 to 31	Extended Security	RFC 1108
6	1	0	134		Commercial Security	
7	0	0	7	variable	Record Route	RFC 791
8	1	0	136	4	Stream Identifier	RFC 791, RFC 1122
9	1	0	137	variable	Strict Source Route	RFC 791

- padding
  - If options are used, header must be a multiple of 4 bytes
  - Fill with NOP and end with EOL

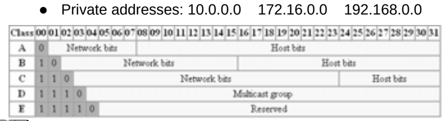
IP options

- Mostly unused
- Extends header from 20 bytes up to 60 bytes (IHL=15)
- Source routing options are “dangerous”, usually blocked by firewall
- Dropped by IPv6
- Program interface is setsockopt() with IP_OPTIONS
  - Example, record route option ping –R

```
rspace[IPOPT_OPTVAL] = IPOPT_RR;
rspace[IPOPT_OLEN] = sizeof(rspace)-1;
rspace[IPOPT_OFFSET] = IPOPT_MINOFF
if (setsockopt(s, IPPROTO_IP, IP_OPTIONS, rspace, sizeof(rspace)) < 0) {
     perror("ping: record route");
     exit(1);
}
```


IP addresses

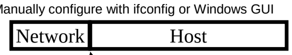
- Only 32 bits, aggregated into network classes (A, B, C)
- Assigned by Internet authority (in “network” chunks)
- Running out of addresses! (IPv6 has 128 bit address)
- Routing based on “network portion” of destination address
- No world-wide “broadcast” address (Whew!) 255.255.255.255
- Multicast addresses (class D)
  - Send once, many receivers
  - Handy for audio/video
  - UDP-based, messy
- Private addresses: 10.0.0.0 172.16.0.0 192.168.0.0



10.0.0.0-127.255.255.255
128.0.0.0-191.255.255.255
192.0.0.0-223.255.255.255
224.0.0.0-239.255.255.255
240.0.0.0-254.255.255.255

Assigning IP addresses

- Enterprise requests class A, B, or collection of class C’s
  - Most nets have been allocated ☹
- UT has a class B, 160.36.0.0 (65,535 hosts)
  - Enterprise can (and usually does) subnet their class B
  - Subnet defined my subnet mask and a default router within the subnet
  - hosts are assigned IP address or dynamically acquire (DHCP)
    - DHCP (later) will configure IP address, mask, and default router
    - Manually configure with ifconfig or Windows GUI



Boundary is flexible, and defined by subnet mask

```
ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:C0:4F:6B:A5:52
          inet addr:160.36.58.221  Bcast:160.36.59.255 Mask:255.255.252.0
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:94907277 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35670805 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
```


IP routing (your host)

- When you send a an IP packet to a host, your OS inspects the destination IP address
  - If it’s on the same subnet as your host (e.g. on the same Ethernet), OS checks ARP table for Ethernet address of destination host
  - If not in ARP table, OS sends an ARP request (broadcast), requesting the Ethernet associated with the destination IP address
  - If host is not on local subnet, OS usually sends the packet to the default router for the subnet (OS needs Ethernet address of router too!)
  - Routers ARP for hosts on their subnets

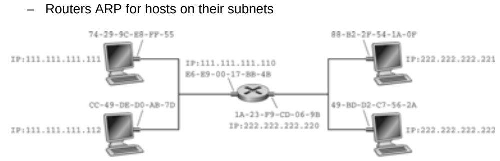


Figure 5.19 ♦ Two subnets interconnected by a router

IP routing (your host)

- You can examine ARP table with arp, host routes with netstat –r -n

```
netstat -r -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
160.36.56.0     0.0.0.0         255.255.252.0   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         160.36.56.1     0.0.0.0         UG       40 0          0 eth0

arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
160.36.56.154            ether   00:06:5B:8E:81:E0   C                     eth0
160.36.57.8              ether   00:09:6B:02:CE:C2   C                     eth0
160.36.56.70             ether   00:06:5B:8E:81:E2   C                     eth0
160.36.56.72             ether   08:00:20:7E:78:5D   C                     eth0
160.36.56.1              ether   00:D0:04:77:4C:00   C                     eth0
```


Address Resolution Protocol (ARP)

•If Ether address not in ARP cache, broadcast an ARP request

•All hosts on subnet hear broadcast, designated host responds

•Cache for 20 minutes

•Operation request/reply

•tcpdump next time

DECnet didn't require an ARP protocol, they changed the Ether address on the NIC to the network address and host address!

RFC826   Ether type 806 (not IP)

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Hardware                    |          protocol             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  HLEN         |   PLEN        |         operation             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              sender hardware address (0-3)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| sender HA (4-5)               |  sender internet addr (0-1)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  sender IA (2-3)              |  target HA (0-1)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     target hardware addr (2-5)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     target internet addr (0-3)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```


Can you impersonate other hosts?

- Can you impersonate a host not on your subnet?
  - www.amazon.com?
- Can your impersonate a host on the local subnet?
  - Sure, just manually configure in the other hosts IP address and reboot
  - Messy if other host is active
    - Multiple ARP replies
    - Hosts will complain about conflicting ARP's
- Hackers send gratuitous ARP replies to trick local hosts
  - e.g., impersonate the default router

Next time …

- routing
- tcpdump/ethereal
- ICMP