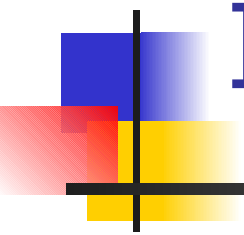
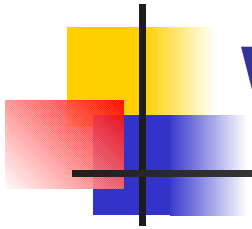


Introduction to IP Networking

Rik Wade

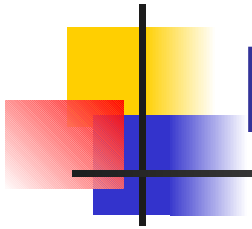
rik@rikwade.com

WYLUG 12/Jan/2004



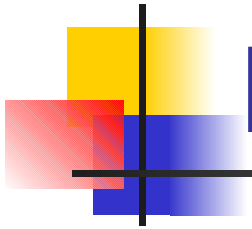
What We'll Cover

- Ethernet
- IP
- Routing



Ethernet

- Ethernet "packets" known as Frames
- MAC address (medium access control)
- 48 bits (6 bytes)
- Uniquely associated with hardware
- Special address for broadcast
  FF:FF:FF:FF:FF:FF hex or 48-bits of
  '1' in binary

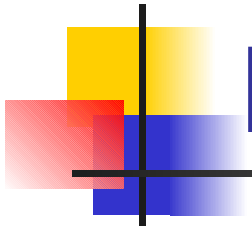


Ethernet

- Frame header contains Source and Destination Ethernet address

  (each 6 bytes)
- Just to confuse matters, this is generally represented as six 2-digit Hex numbers

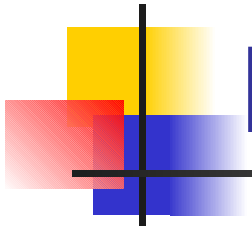
  e.g 00:60:5C:BC:09:15



Ethernet

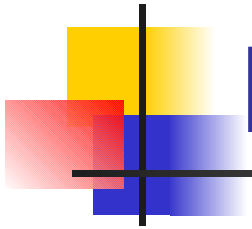
- ARP (address resolution protocol) http://www.ietf.org/rfc/rfc826.txt
- Discover MAC address for a given IP address

```
debian:/root# arp -an
? (192.168.1.1) at 00:60:5C:BC:09:15 [ether] on eth0
? (192.168.1.21) at 00:A0:CC:D0:B0:87 [ether] on eth0

debian:/root# tcpdump -i eth0 arp
18:08:36.727610 arp who-has 192.168.1.254 tell 192.168.1.10
18:08:36.728257 arp reply 192.168.1.254 is-at 0:b:6b:38:9:93
```



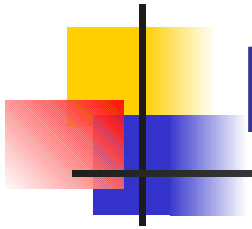
Ethernet

- ARP Process:
  - A wants to send to B
  - A sends broadcast ARP asking for the identity of B
  - B responds with MAC address
  - A uses the MAC address and saves it for later in its ARP table



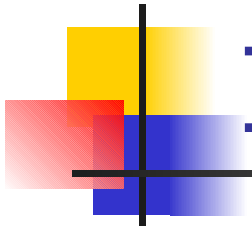
Ethernet

- What is a **hub**?
  - "A multiport repeater"
  - "A way of connecting many physical segments in to one logical segment"
  - "A waste of money"



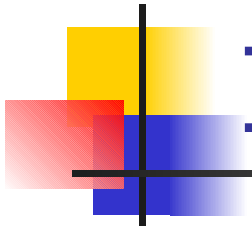
Ethernet

- What is a **switch**?
  - "More expensive than a hub"
  - "Faster than a hub"
  - "Switches frames based on DST MAC rather than broadcasting everything"
  - "The more expensive ones can contain many virtual hubs"
  - "Falling in price on a daily basis"



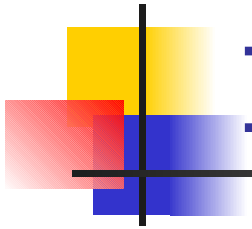
IP

- Major Components:
  - IP address (network part + host part, but we'll see later)
    - Linked to Ethernet addressing with ARP table
      [IP address : MAC address]
  - Subnet mask
    - Classful/Classless
  - Default gateway
    - Where to send packets if no specific route exists



IP

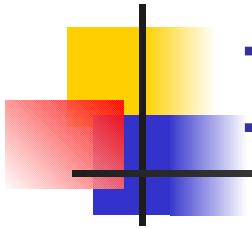
- IP Addressing
  - e.g. 192.168.0.1 255.255.0.0
  - 192.168.[1-255].[1-255]
- Easier for humans to understand in decimal
- IP addresses and masks represented as 4 blocks of 8 bits (32 bits in total)
- Each block of 8 bits is represented in decimal for human consumption



IP

- Computers work in binary however:

```
Address:   192.168.0.1          11000000.10101000 .00000000.00000001
Netmask:   255.255.0.0 = 16     11111111.11111111 .00000000.00000000
=>
Network:   192.168.0.0/16       11000000.10101000 .00000000.00000000
  (Class C)
Broadcast: 192.168.255.255      11000000.10101000 .11111111.11111111
HostMin:   192.168.0.1          11000000.10101000 .00000000.00000001
HostMax:   192.168.255.254      11000000.10101000 .11111111.11111110
Hosts:     65534
```

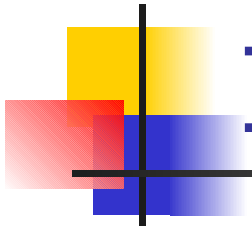


IP

- RFC1918 http://www.ietf.org/rfc/rfc1918.txt
  - Private addressing for non-public networks. Use these at home!

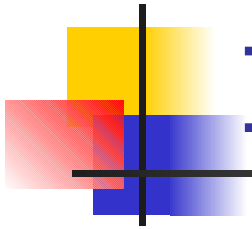
```
10.0.0.0        -   10.255.255.255  (10/8 prefix)
172.16.0.0      -   172.31.255.255  (172.16/12 prefix)
192.168.0.0     -   192.168.255.255 (192.168/16 prefix)
```

  - Gateway to the Internet using Network Address Translation (NAT)
  - With dial-up, an ISP will assign your PPP session a public IP address
  - Linux IP Masquerading or Microsoft Connection Sharing can be used to NAT and use a designated system as an IP router



IP

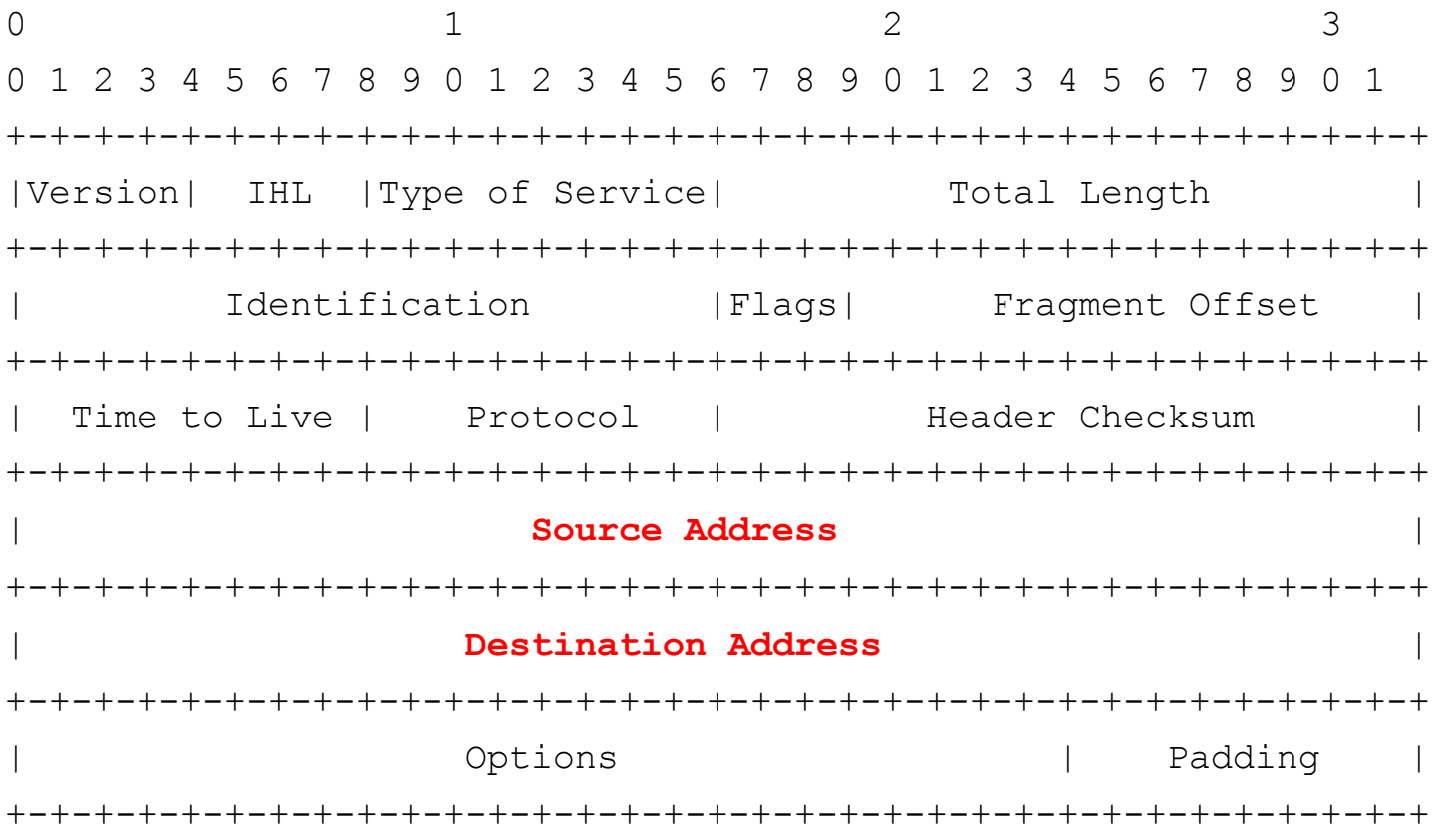
- What is a **router**?
  - A machine connected to one or more networks
  - Every connected network has a router
- IP routing is performed hop-by-hop according to Routing Tables
  - Static routing
  - Dynamic routing

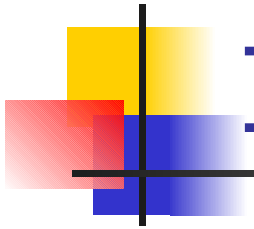


IP

- IP packet
  - http://www.ietf.org/rfc/rfc791.txt
  - 1981!
- 20 bytes header minimum size
- Variable data portion, minimum 1 byte

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
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding

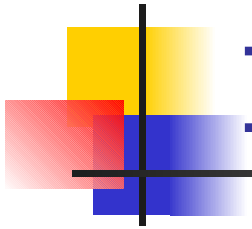


IP

- **10ms in the life of an IP packet**

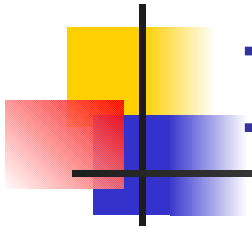
```
                          A    B
A----------------------Router----------------------B
.2   192.168.0.0/24     .1   .1    192.168.1.0/24   .2
```

  - A = 192.168.0.2/24 (255.255.255.0)
    - default gateway 192.168.0.1
  - B = 192.168.1.2/24 (255.255.255.0)
    - default gateway 192.168.1.1
- Router has interfaces:
  - A 192.168.0.1/24 CC-CC-CC-CC-CC-CC
  - B 192.168.1.1/24 DD-DD-DD-DD-DD-DD



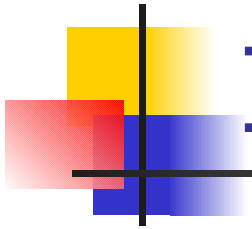
IP

- A wants to send to B
- A builds an IP packet with:
  - SRC: 192.168.0.2
  - DST: 192.168.1.2
- A does not have a route in its routing table for 192.168.1.2
- Uses its default route of 192.168.0.1



IP

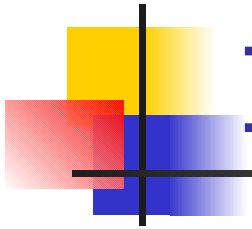
- A looks in its ARP table and does not have the MAC address for Router
- A ARPs for Router and receives a reply with CC-CC-CC-CC-CC-CC
- A puts the MAC address in its ARP table along with 192.168.0.1
- Router does not have 192.168.0.2 in its ARP table, so puts AA-AA-AA-AA-AA-AA in along with 192.168.0.2



IP

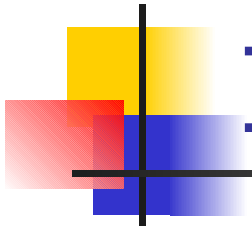
- A encapsulates the IP packet in an Ethernet Frame with:
  - SRC: AA-AA-AA-AA-AA-AA
  - DST: CC-CC-CC-CC-CC-CC
- Router receives Ethernet frame because its interface has MAC address

  CC-CC-CC-CC-CC-CC



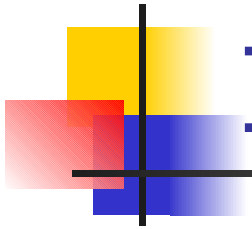
IP

- Router looks up 192.168.1.2 in its routing table
- It has an interface in this network, so ARPs for 192.168.1.2 to obtain the MAC address
- B receives this ARP (as it is a broadcast Ethernet frame) and replies
- Router puts BB-BB-BB-BB-BB-BB in its ARP table along with 192.168.1.2



IP

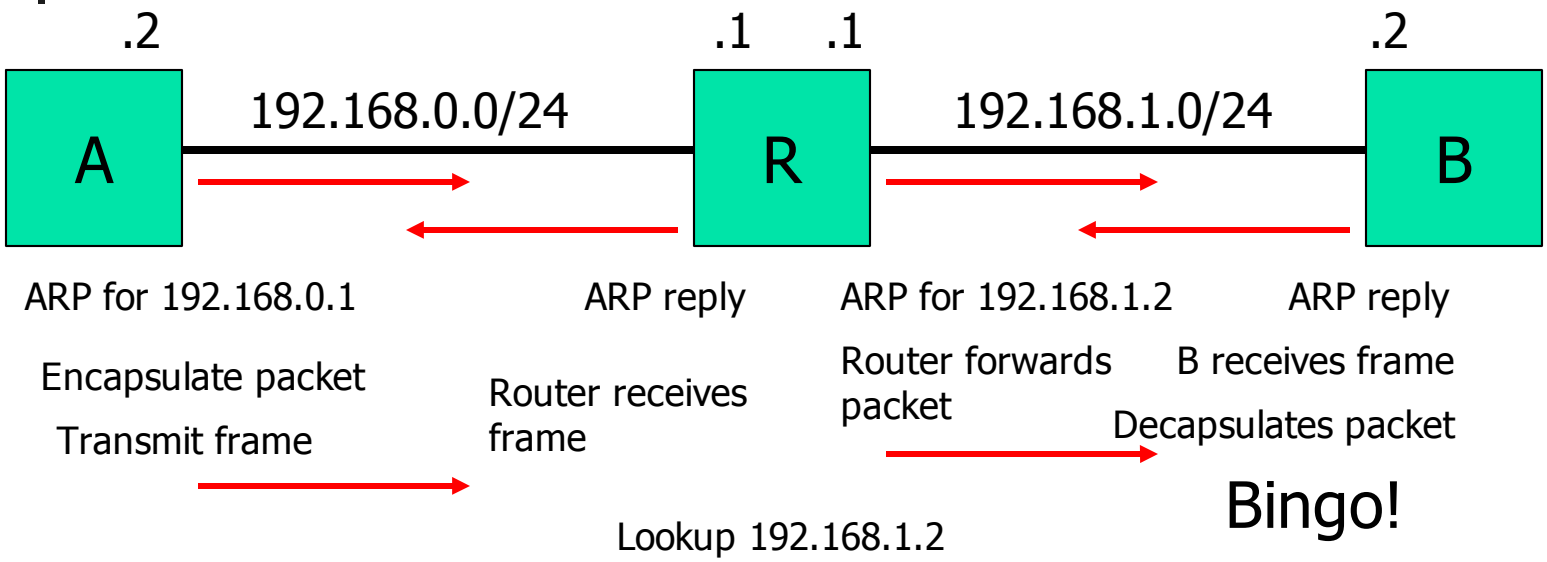
- Router forwards the IP packet to 192.168.1.2 out of 192.168.1.1 with:
  - IP SRC: 192.168.0.2
  - IP DST: 192.168.1.2
  - ETH SRC: DD-DD-DD-DD-DD-DD
  - ETH DST: BB-BB-BB-BB-BB-BB

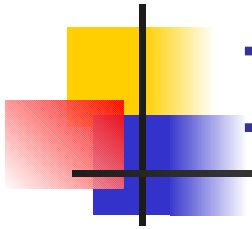


IP

- B receives the frame as the destination MAC matches its MAC address of

  BB-BB-BB-BB-BB-BB
- It decapsulates the IP packet and accepts the DST address of 192.168.1.2
- **Bingo!**


.2
.1
.1
.2
A
192.168.0.0/24
R
192.168.1.0/24
B
ARP for 192.168.0.1
ARP reply
ARP for 192.168.1.2
ARP reply
Encapsulate packet
Transmit frame
Router receives frame
Router forwards packet
B receives frame
Decapsulates packet
Bingo!
Lookup 192.168.1.2



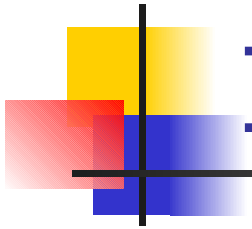
IP

- More complex network example

```
A---Router1---Router2---B
```

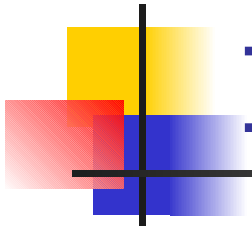
- SRC: 192.168.0.2/24
- DST: 192.168.2.2/24

- When Router1 receives the IP packet, it does not have an interface in 192.168.2.0/24 network. It therefore consults its routing tables



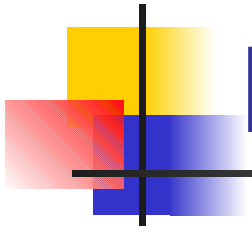
IP

```
Router1>sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C    192.168.0.0/24 is directly connected, Ethernet0
     192.168.253.0/30 is subnetted, 2 subnets
O       192.168.1.4 [110/128] via 192.168.1.2, 00:03:33, Serial0
C       192.168.1.0 is directly connected, Serial0
O E2 192.168.2.0/24 [110/20] via 192.168.1.2, 00:02:07, Serial0
     192.168.252.0/32 is subnetted, 2 subnets
O       192.168.252.4 [110/65] via 192.168.1.2, 00:03:33, Serial0
O       192.168.252.1 [110/129] via 192.168.1.2, 00:03:33, Serial0
```



IP

- Router1 therefore has a route in its routing table for the network 192.168.2.0/24 via 192.168.1.2
- We have an interface 192.168.1.1 in 192.168.1.0/30 (which only contains .1 and .2!) so forward the packet to Router2
- Router2 receives the packet and has an interface in 192.168.2.0/24, so performs the function as before



FIN

- OBLinux bits:

- Networking HOWTO
  - http://en.tldp.org/HOWTO/Net-HOWTO/
- IP Masquerading (NAT) HOWTO
  - http://en.tldp.org/HOWTO/IP-Masquerade-HOWTO/
- Slightly old Home Networking HOWTO
  - http://en.tldp.org/HOWTO/Home-Network-mini-HOWTO.html