

Interconnection Layer 3: IP

Jean-Yves Le Boudec

1

Contents

- 1. Principles
- 2. Addressing
- 3. Packet Delivery and Forwarding
- 4. IP header
- 5. ICMP
- 6. Fragmentation
- 7. Multicast IP
- 8. IPv6
- 9. Terminology
- 10. Static Configuration of Unix Host

2

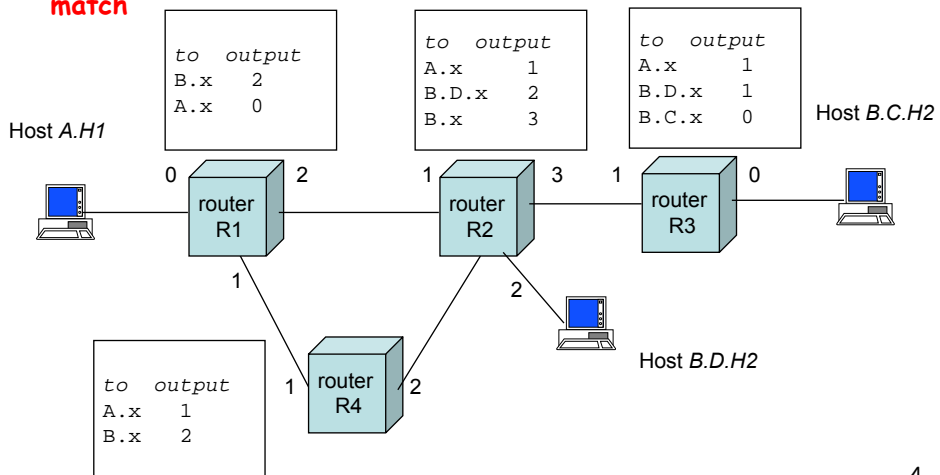
1. Why a network layer?

- ❑ MAC addresses and bridging are not sufficient
 - bridging does not scale well to large networks
 - MAC have no topological structure
- ❑ Solution: connectionless network layer (eg. Internet Protocol, IP):
 - every host receives a network layer address (IP address)
 - intermediate systems forward packets based on destination address

3

Connectionless Network Layer

- ❑ **Connectionless** network layer = no connection
- ❑ every packet contains destination address
- ❑ intermediate systems (= routers) forward based on **longest prefix match**



4

IP Principles

Homogeneous addressing

- ❑ an IP address is unique across the whole network (= the world in general)
- ❑ IP address is the address of an interface
- ❑ communication between IP hosts requires knowledge of IP addresses

Routers between subnetworks only:

- ❑ a subnetwork = a collection of systems with a common prefix
- ❑ inside a subnetwork: hosts communicate directly without routers
- ❑ between subnetworks: one or several routers are used

- ❑ Host either sends a packet to the destination using its LAN, or it passes it to the router for forwarding

Terminology:

- host = end system; router = intermediate system
- subnetwork = one collection of hosts that can communicate directly without routers

5

2. IP addresses

❑ IP address

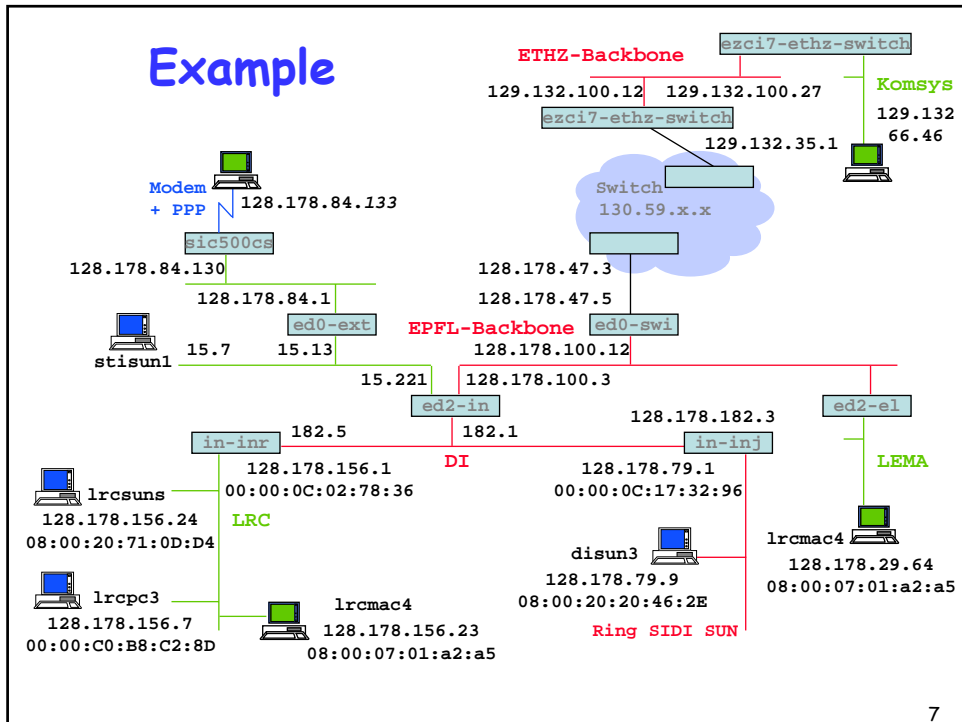
- Unique addresses in the world, decentralized allocation
- An IP address is 32 bits, noted in dotted decimal notation:
192.78.32.2

❑ Host and Prefix Part

- An IP address has a prefix and a host part:
 - prefix:host
- Prefix identifies a subnetwork
 - used for locating a subnetwork - routing
- Prefix is usually identified in a host using a "subnet mask"

6

Example



7

Binary, Decimal and Hexadecimal

- ❑ Given an integer B "the basis": any integer can be represented in "base B" by means of an alphabet of B symbols
- ❑ Usual cases are
 - decimal: 234
 - binary: b1110 1010
 - hexadecimal: xEA
- ❑ Mapping binary <-> hexa is simple: one hexa digit is 4 binary digits
 - xE = b1110 xA = b1010 xEA = b1110 1010
- ❑ Mapping binary <-> decimal is best done by a calculator
 - b1110 1010 = 128 + 64 + 32 + 8 + 2 = 234
- ❑ Special Cases to remember
 - xF = b1111 = 15
 - xFF = b1111 1111 = 255

8

Representation of IP Addresses

❑ **dotted decimal**: group bits in bytes, write the decimal representation of the number

- example 1: 128.191.151.1
- example 2: 129.192.152.2

❑ **hexadecimal**: hexadecimal representation -- fixed size string

- example 1: x80 BF 97 01
- example 2: x

❑ **binary**: string of 32 bits (2 symbols: 0, 1)

- example 1: b0100 0000 1011 1111 1001 0111 0000 0001
- example 2: b

9

A Subnet Prefix is written using one of two Notations: masks / prefixes

❑ Using a mask: address + mask :

- example : 128.178.156.13 mask 255.255.255.0
 - the mask is the dotted decimal representation of the string made of : 1 in the prefix, 0 elsewhere
 - bit wise address & mask gives the prefix
 - here: prefix is 128.178.156.0
- example 2: 129.132.119.77 mask 255.255.255.192
 - Q1: what is the prefix ?
 - Q2: how many host ids can be allocated ?

10

Prefix Notation

- ❑ prefix - notation: 128.178.156.1/24
 - the 24 first bits of the binary representation of the string, interpreted as dotted decimal
 - here: the prefix is 128.178.156.0

 - bits in excess are ignored
 - 128.178.156.1/24 is the same as 128.178.156.22/24 and 128.178.156/24
- ❑ example 2:
 - Q1: write 129.132.119.77 mask 255.255.255.192 in prefix notation
 - Q2: are these prefixes different ?
 - 201.10.0.0/28, 201.10.0.16/28, 201.10.0.32/28, 201.10.0.48/28
 - how many IP addresses can be allocated to each of the distinct subnets ?

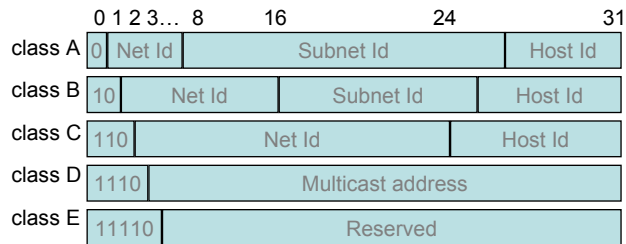
11

IP Address Hierarchies

- ❑ The prefix of an IP address can itself be structured into subprefix in order to support aggregation
 - For example:
 - 128.178.x.y represents an EPFL host
 - 128.178.156 / 24 represents the LRC subnet at EPFL
 - 128.178 / 16 represents EPFL
 - Used between routers by routing algorithms
 - This way of doing is called classless and was first introduced in inter domain routing under the name of CIDR (classless interdomain routing)
- ❑ IP address classes
 - IP addresses are sorted into classes
 - This is an obsolete classification - no longer used
 - Was used for automatic prefixing
 - a class A [resp. B] address was automatically exported to the rest of the network as an 8 bit [resp. a 16 bit] prefix

12

IP address classes



Examples: 128.178.x.x = EPFL host; 129.132.x.x = ETHZ host
 9.x.x.x = IBM host 18.x.x.x = MIT host

| Class | Range |
|-------|------------------------------|
| A | 0.0.0.0 to 127.255.255.255 |
| B | 128.0.0.0 to 191.255.255.255 |
| C | 192.0.0.0 to 223.255.255.255 |
| D | 224.0.0.0 to 239.255.255.255 |
| E | 240.0.0.0 to 247.255.255.255 |

- ❑ Class B addresses are close to exhausted; new addresses are taken from class C, allocated as continuous blocks

Address allocation

- ❑ World Coverage
 - Europe and the Middle East (RIPE NCC)
 - Africa (ARIN & RIPE NCC)
 - North America (ARIN)
 - Latin America including the Caribbean (ARIN)
 - Asia-Pacific (APNIC)
- ❑ Current allocations of Class C
 - 193-195/8, 212-213/8, 217/8 for RIPE
 - 199-201/8, 204-209/8, 216/8 for ARIN
 - 202-203/8, 210-211/8, 218/8 for APNIC
- ❑ Simplifies routing
 - short prefix aggregates many subnetworks
 - routing decision is taken based on the short prefix

Address delegation

□ Europe

- 62/8, 80/8, 193-195/8, ...

● ISP-1

- 62.125/16
- customer 1: banana foods
 - 62.125.44.128/25
- customer 2: sovkom
 - 62.125.44.50/24

● ISP-2

- 195.44/14
- customer 1:
 - 195.46.216/21
- customer 2:
 - 195.46.224/21

15

Renumbering?

□ Europe

- 62/8, 80/8, 193-195/8, ...

● ISP-1

- 62.125/16
- banana foods
 - 62.125.44.128/25

● ISP-2

- 195.44/14
- Customer 1
 - 195.46.216/21
- Customer 2
 - 195.46.224/21
- sovkom
 - 62.125.44.50/24

→ no aggregation possible
(explicit route to sovkom)

16

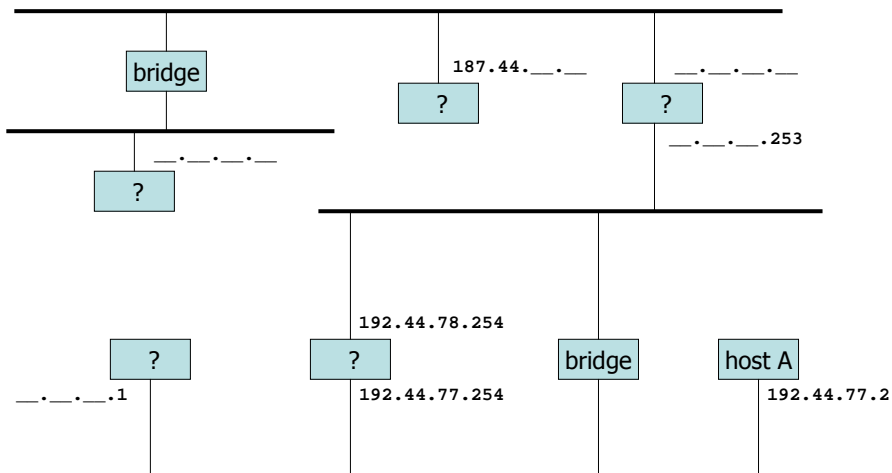
Special case IP addresses

- | | |
|------------------------------------|--|
| 1. 0.0.0.0 | this host, on this network |
| 2. 0.hostId | specified host on this net (initialization phase) |
| 3. 255.255.255.255 | limited broadcast (not forwarded by routers) |
| 4. subnetId.all 1's | broadcast on this subnet |
| 5. subnetId.all 0's | BSD used it for broadcast on this subnet (obsolete) |
| 6. 127.x.x.x | loopback |
| 7. 10/8 172.16/12 192.168/16 | reserved networks for internal use (Intranets) |

- 1,2: source IP@ only; 3,4,5: destination IP@ only

17

Test Your Understanding (1)



Q: Can host A have this address? (masks are all 255.255.255)

18

Test your Understanding (2)

- ❑ Q1: An Ethernet segment became too crowded; we split it into 2 segments, interconnected by a router. Do we need to change some IP host addresses ?
- ❑ Q2: same with a bridge.
- ❑ Q3: compare the two

19

3. IP packet forwarding

- ❑ Rule for sending packets (hosts, routers)
 - if the destination IP address has the same prefix as one of my interfaces, send directly to that interface
 - otherwise send to a router as given by the IP routing table

20

IP packet forwarding algorithm

```
destAddr = destination address /* unicast! */  
  
if /*case 1*/: a host route exists for destAddr  
    for every entry in routing table  
        if (destinationAddr = destAddr)  
            then send to nextHop IPAddr; leave  
  
else if /*case 2*/: destAddr is on a directly connected network (= on-link):  
    for every physical interface IP address A and subnet mask SM  
        if(A & SM = destAddr & SM)  
            then send directly to destAddr; leave  
  
else if /*case 3 */ there is a matching entry in routing table  
    find the longest prefix match for destAddr  
    send to nextHop IP addr given by matching entry; leave  
    /* this includes as special case the default route, if it exists */  
  
else /* error*/  
    send ICMP error message "destination unreachable" to source
```

21

Example

- ❑ Q1: Fill in the table if an IP packet has to be sent from lrcsuns

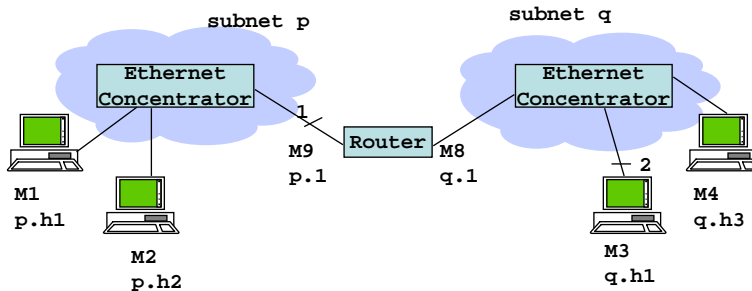
| final destination | next hop | case number |
|--|----------|-------------|
| 128.178.79.9 128.178.156.7 127.0.0.1 128.178.84.133 129.132.1.45 | | |

- ❑ Q2: Fill in the table if an IP packet has to be sent from ed2-in

22

Test Your Understanding (3)

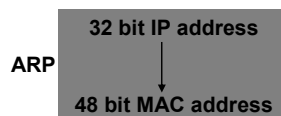
- ❑ Q1: What are the MAC and IP addresses at points 1 and 2 for packets sent by M1 to M3 ? At 2 for packets sent by M4 to M3 ? (Mx = mac address)
- ❑ Q2: What must the router do when it receives a packet to M2 for the first time?



23

Direct Packet Forwarding: ARP

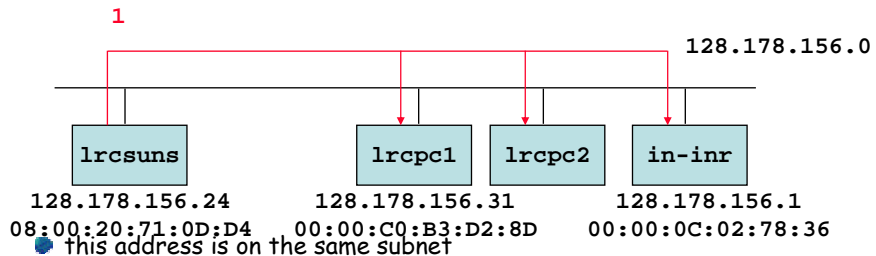
- ❑ Sending to host on the same subnet = direct packet forwarding
 - does not use a router
- ❑ Requires the knowledge of the MAC address on a LAN
 - on LANs: uses the Address Resolution Protocol



24

ARP Protocol

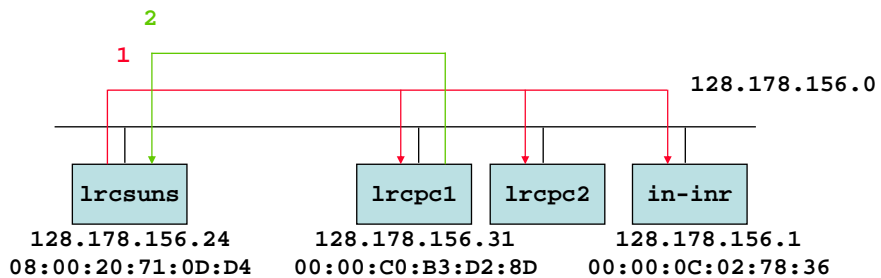
1: lrcsuns has a packet to send to 128.178.156.31 (lrcpc1)



- this address is on the same subnet
- lrcsuns sends an ARP request to all systems on the subnet (broadcast)
- target IP address = 128.178.156.31
- ARP request is received by all IP hosts on the local network
- is not forwarded by routers

25

ARP Protocol

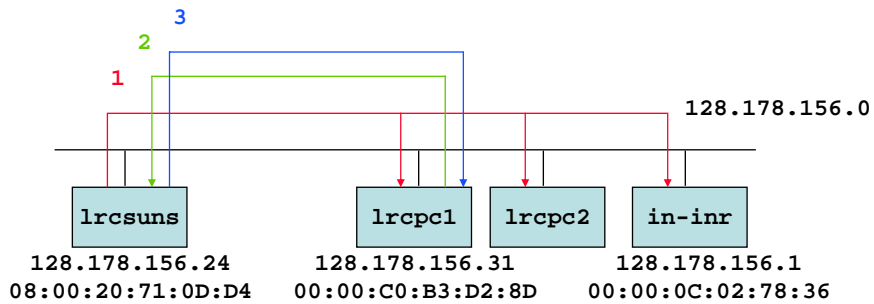


2: lrcpc1 has recognized its IP address

- sends an ARP reply packet to the requesting host
- with its IP and MAC addresses

26

ARP Protocol

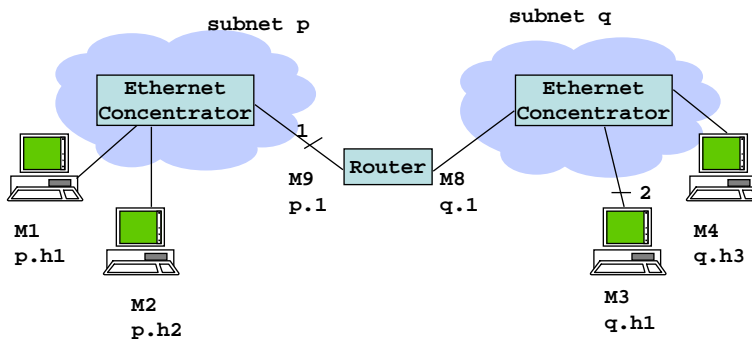


- 3: lrcsuns reads ARP reply, stores in a cache and sends IP packet to lrcpc1

27

Test Your Understanding

- ❑ Q1: What are the MAC and IP addresses at points 1 and 2 for packets sent by M1 or M4 to M3 (Mx = ? mac address)
- ❑ Q2: What must the router do when it receives a packet to M2 for the first time?



28

Etherreal

Ethernet II

Destination: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)

Source: 00:03:93:a3:83:3a (Apple_a3:83:3a)

Type: ARP (0x0806)

Trailer: 00000000000000000000000000000000...

Address Resolution Protocol (request)

Hardware type: Ethernet (0x0001)

Protocol type: IP (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (0x0001)

Sender MAC address: 00:03:93:a3:83:3a (Apple_a3:83:3a)

Sender IP address: 129.88.38.135 (129.88.38.135)

Target MAC address: 00:00:00:00:00:00 (00:00:00_00:00:00)

Target IP address: 129.88.38.254 (129.88.38.254)

29

ARP sent

- Wait for reply
 - if received, put target information into the cache
 - if not, repeat on timeout, increase timeout on each failure
- ARP cache
 - entry maintained for 20 minutes (on BSD), or less (Linux)
 - incomplete entry (no reply), 3 minutes

30

ARP receive

- ❑ If IP address = self address
 - put the sender information into the cache
 - reply with our Ethernet address as target
- ❑ Otherwise
 - if IP address in the cache
 - update entry
 - otherwise
 - do not update

31

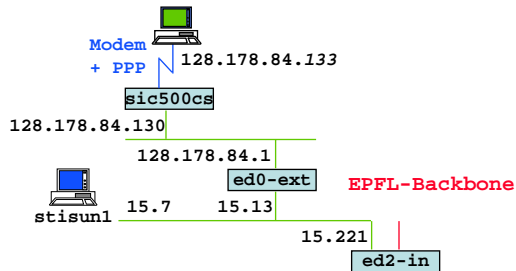
Gratuitous ARP

- ❑ When a network interface is configured, the host sends an ARP request for the IP address being set
 - check if the IP address is unique
 - if no response, the address is unique
 - inform other hosts about the change of the MAC address

32

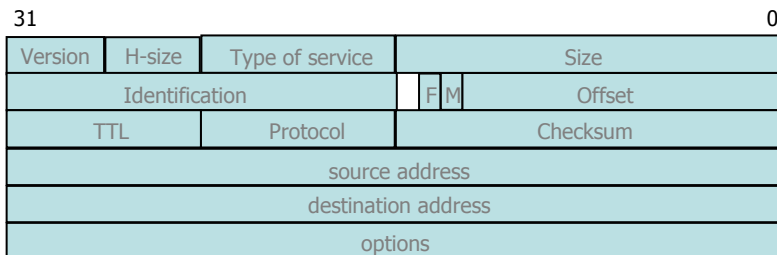
Proxy ARP

- ❑ Proxy ARP = a host answers ARP requests on behalf of others
 - example: sic500cs for PPP connected computers
 - manual configuration
 - works well for "stub" networks only
- ❑ Q1: how must sics500cs routing table be configured ?
- ❑ Q2: explain what happens when ed2-in has a packet to send to 128.178.84.133



33

4. IP header



- ❑ Transmitted "big-endian" - bit 31 first
- ❑ Version
 - IPv4, futur IPv6
- ❑ Header size
 - options - variable size
 - in 32 bit words

34

IP header

- ❑ Type of service
 - priority : 0 - normal, 7 - control packets
 - short delay (telnet)
 - high throughput (ftp)
 - high reliability (SNMP)
 - low cost (NNTP)
- ❑ Redefined in DiffServ (Differentiated Services)
 - 1 byte codepoint determining QoS class
 - Expedited Forwarding (EF) - minimize delay and jitter
 - Assured Forwarding (AF) - four classes and three drop-precedences (12 codepoints)

35

IP header

- ❑ Packet size
 - in bytes including header
 - ≤ 64 Kbytes; limited in practice by link-level MTU (Maximum Transmission Unit)
 - every subnet should forward packets of $576 = 512 + 64$ bytes
- ❑ Id
 - unique identifier for re-assembling
- ❑ Flags
 - M : more ; set in fragments
 - F : prohibits fragmentation

36

IP header

- ❑ Offset
 - position of a fragment in multiples of 8 bytes
- ❑ TTL (Time-to-live)
 - in secondes
 - now: number of hops
 - router : --, if 0, drop (send ICMP packet to source)
- ❑ Protocol
 - identifier of protocol (1 - ICMP, 6 - TCP, 17 - UDP)
- ❑ Checksum
 - only on the header

37

IP header

- ❑ Options
 - strict source routing
 - all routers
 - loose source routing
 - some routers
 - record route
 - timestamp route
 - router alert
 - used by IGMP or RSVP for processing a packet

38

Ethereal

Ethernet II

```
Destination: 00:03:93:a3:83:3a (Apple_a3:83:3a)
Source: 00:10:83:35:34:04 (HEWLETT-_35:34:04)
Type: IP (0x0800)
Internet Protocol, Src Addr: 129.88.38.94 (129.88.38.94), Dst Addr: 129.88.38.241
(129.88.38.241)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 1500
Identification: 0x624d
Flags: 0x04
Fragment offset: 0
Time to live: 64
Protocol: TCP (0x06)
Header checksum: 0x82cf (correct)
Source: 129.88.38.94 (129.88.38.94)
Destination: 129.88.38.241 (129.88.38.241)
```

39

5. ICMP: Internet Control Message Protocol

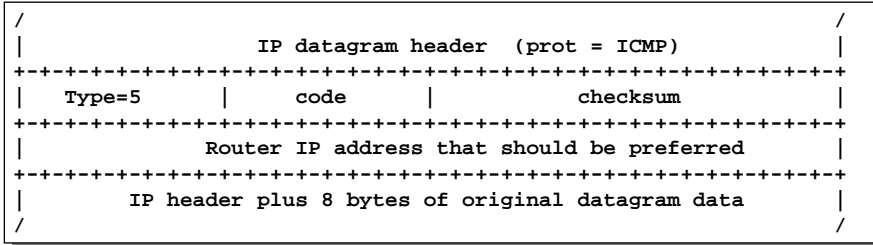
- used by router or host to send error or control messages to other hosts or routers
- error or control messages relate to layer 3 only
- carried in IP datagrams (protocol type = 1)
- ICMP message types
 - echo request (reply) -> used by ping
 - destination unreachable
 - time exceeded (TTL = 0) -> used for traceroute responses
 - address mask request/reply
 - source quench
 - redirect - router discovery
 - timestamps
 - ICMP messages never sent in response to
 - ICMP error message - datagram sent or multicast or broadcast IP or layer 2 address - fragment other than first

40

ICMP Redirect

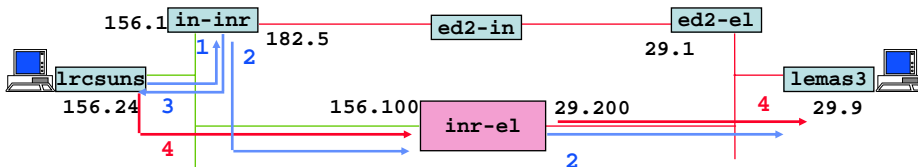
- ❑ Sent by router to source host to inform source that destination is directly connected
 - host updates the routing table
 - ICMP redirect can be used to update the router table (eg. in-inj route to LRC?)

ICMP Redirect Format



- ❑ General routing principle of the TCP/IP architecture:
 - host have minimal routing information
 - learn host routes from ICMP redirects
 - routers have extensive knowledge of routes

ICMP Redirect Example



| | dest IP addr | srce IP addr | prot | data part |
|----|----------------|----------------|------|--|
| 1: | 128.178.29.9 | 128.178.156.24 | udp | xxxxxxx |
| 2: | 128.178.29.9 | 128.178.156.24 | udp | xxxxxxx |
| 3: | 128.178.156.24 | 128.178.156.1 | icmp | type=redir code=host cksum 128.178.156.100 xxxxxxx (28 bytes of 1) |
| 4: | 128.178.29.9 | 128.178.156.24 | udp | |

ICMP Redirect Example (cont'd)

After 4

```
lracsuns:/export/home1/leboudec$ netstat -nr
Routing Table:
  Destination          Gateway                Flags  Ref    Use  Interface
-----
127.0.0.1              127.0.0.1             UH     0  11239  lo0
128.178.29.9          128.178.156.100      UGHD   0     19
128.178.156.0         128.178.156.24       U       3  38896  le0
224.0.0.0             128.178.156.24       U       3     0  le0
default               128.178.156.1        UG     0  85883
```

6. MTU

Link-layer networks have different maximum frame length

- ❑ MTU (maximum transmission unit) = maximum frame size usable for an IP packet
- ❑ value of short MTU ? of long MTU ?

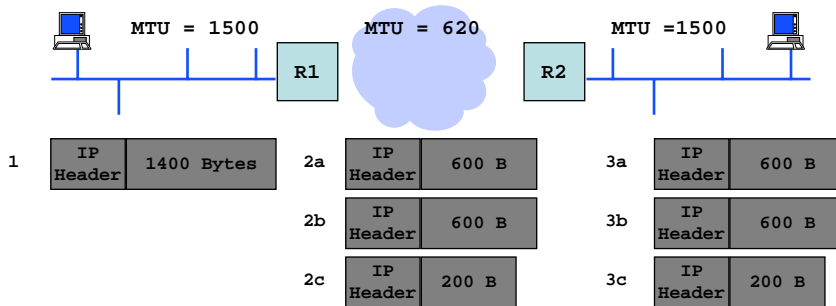
| Link-layer Network | MTU |
|---------------------|-------------|
| Ethernet | 1500 |
| 802.3 with LLC/SNAP | 1492 |
| Token Ring 4 Mb/s | 4464 |
| 16 Mb/s | 17914 |
| FDDI | 4352 |
| X.25 | 576 |
| Frame Relay | 1600 |
| ATM with AAL5 | 9180 |
| Hyperchannel | 65535 |
| PPP | 296 to 1500 |

```
lracsuns:/export/home1/leboudec$ ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 128.178.156.24 netmask fffffff0 broadcast
128.178.156.255
    ether 8:0:20:71:d:d4
```

IP Fragmentation

IP hosts or routers may have IP datagrams larger than MTU

- Fragmentation is performed when IP datagram too large
- re-assembly is only at destination
- fragmentation is in principle avoided with TCP



45

IP Fragmentation (2)

- IP datagram is *fragmented* if
 $MTU \text{ of interface} < \text{datagram total length}$
- all fragments are self-contained IP packets
- fragmentation controlled by fields: Identification, Flag and Fragment Offset
- IP *datagram* = original ; IP *packet* = fragments or complete datagram

| | 1 | 2a | 2b | 2c |
|--------------------|------|-----|-----|------|
| Length | 1420 | 620 | 620 | 220 |
| Identification | 567 | 567 | 567 | 567 |
| More Fragment flag | 0 | 1 | 1 | 0 |
| Offset | 0 | 0 | 75 | 150 |
| 8 * Offset | 0 | 0 | 600 | 1200 |

Fragment data size (here 600) is always a multiple of 8
 Identification given by source

46

Fragmentation Algorithm

- ❑ Repeated fragmentations may occur
- ❑ Don't fragment flag prevents fragmentation
- ❑ Fragmentation Algorithm:

```
procedure sendIPp(P0):  
  
  if P0.totalLength > MTU then  
    datalength = (MTU-P0.HLEN rounded to multiple of 8)  
    data1= first datalength bytes of P0 data part  
    data2= remainder of P0 data part  
    header1 = P0.header with  
              More bit set  
              totalLength = P0.HLEN + datalength  
    P1= new (IPPacket; header1; data1)  
    send P1 on data link layer  
    header2 = P0.header with  
              totalLength = P0.totalLength - datalength  
              fragmentOffset += datalength/8  
    P2= new(IPPacket; header2; data2)  
    sendIPp(P2)  
  else  
    send P0 on data link layer
```

47

Fragment Re-Assembly

- ❑ Re-assembly is performed at the final destination only, never at intermediate points
- ❑ Re-assembly issues:
 - packet misordering
 - packet loss
 - others?

48

```

IP packets are sorted in fragment lists
    one fragment list per (Identification, source IP @)
    sorted by increasing Fragment Offset
Fragments F1 and F2 are contiguous iff
    F1.moreBit = 1
    F1.fragmentOffset + F1.dataLength/8 = F2.fragmentOffset
Fragment List F0...Fn is complete iff
    F0.fragmentOffset = 0
    Fi and Fi+1 are contiguous for i=0...(n-1)
    Fn.moreBit = 0

```

```

IP packet arrival (P0) /* and packet is not a complete datagram */ ->
if (P0.(identification, source address)) is new
then if (new(fragmentList, P0.(identification, source address), fl))
    then insert P0 in fl
        start reassemblyTimer(fl)
else
    fl = fragmentList(P0.(identification, source address))
    insert(fl,P0)
    if fl is complete
        then deliver IP datagram
        else start reassemblyTimer(fl)

reassemblyTimer(fl) expires ->
    send ICMP error message to source
    delete(fl)

```

49

Fragmentation Problems

Fragmentation requires re-assembly

- deadlocks
- identification wrapping problem
- unit of loss is smaller than unit of re-transmission

Solution = avoid fragmentation

- Path MTU = minimum MTU for all links of one path
- Discovery of path MTU
 - heuristics: local -> 1500; other : 576 (subnetsarelocal variable)

Path MTU discovery avoids fragmentation

50

Path MTU Discovery

- ❑ Method for Path MTU (PMTU) discovery
 - 1. host sets Don't Fragment bit on all datagrams and estimate PMTU to local MTU
 - 2. routers send an ICMP message: "destination unreachable/ fragmentation needed"
 - 3. host reduces PMTU estimate to next smallest value
 - 4. after timeout, host increases PMTU estimate
 - route changes may cause 2

51

TCP, UDP and Fragmentation

- ❑ The UDP service interface accepts a datagram up to 64 KB
 - UDP datagram passed to the IP service interface as one SDU
 - is fragmented at the source if resulting IP datagram is too large
- ❑ The TCP service interface is stream oriented
 - packetization is done by TCP
 - several calls to the TCP service interface may be grouped into one TCP segment (many small pieces)
 - or: one call may cause several segments to be created (one large piece)
 - TCP always creates a segment that fits in one IP packet: no fragmentation at source
 - fragmentation may occur in a router, if IPv4 is used, and if PMTU discovery is not implemented

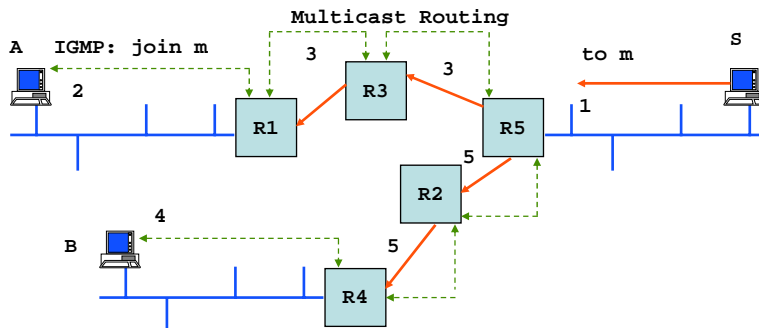
52

7. Multicasting

- ❑ Broadcast = send to all
 - sent to all hosts on one net/subnet; used by NetBIOS for discovery
- ❑ Multicast = send to a group
 - IP multicast address = class D = 224.0.0.0 to 239.255.255.255
 - 224.0.0.1 = all multicast capable systems on subnet
 - 224.0.0.2 = all multicast capable routers on subnet
 - used for: conferencing, radio distribution, ...
- ❑ IP uses open group paradigm
 - multicast IP addresses are logical (= non topological)
 - for receiving data sent to multicast address m , a host must subscribe to m
 - for sending to multicast address m , a host simply puts m in the dest addr field

53

IP Multicast Principles



- ❑ hosts subscribe via IGMP join messages sent to router
- ❑ routers build distribution tree via multicast routing
- ❑ sources do not know who destinations are
- ❑ packet multiplication is done by routers

54

Multicast Address Scopes

| IPv6 SCOP | RFC 1884 Description | IPv4 Prefix |
|-----------|--------------------------|---------------------------|
| 0 | reserved | |
| 1 | node-local scope | |
| 2 | link-local scope | 224.0.0.0/24 |
| 3 | (unassigned) | 239.255.0.0/16 |
| 4 | (unassigned) | |
| 5 | site-local scope | |
| 6 | (unassigned) | |
| 7 | (unassigned) | |
| 8 | organization-local scope | 239.192.0.0/14 |
| A | (unassigned) | |
| B | (unassigned) | |
| C | (unassigned) | |
| D | (unassigned) | |
| E | global scope | 224.0.1.0-238.255.255.255 |
| F | reserved | |

55

IP Multicast Forwarding Algorithm

Packet Forwarding (host, router)

Read address MA = destination IP@

```
/* assume it is multicast */
for every physical interface PI
    if MA is enabled on PI then
        send directly to PI
```

At lrcsuns: Physical Interface Tables

| IP | subnetMask |
|----------------|---------------|
| 128.178.156.24 | 255.255.255.0 |
| 224.2.166.207 | |
| 224.2.127.255 | |

Send directly (Ethernet, FDDI)

```
send directly(MA, MAC@):
    map last 23 bits of MA to last 23 bits
    of MAC address
    send MAC frame with      DA = 01-00-5E-xx-xx-xx,
                             SA = own i/f address
```

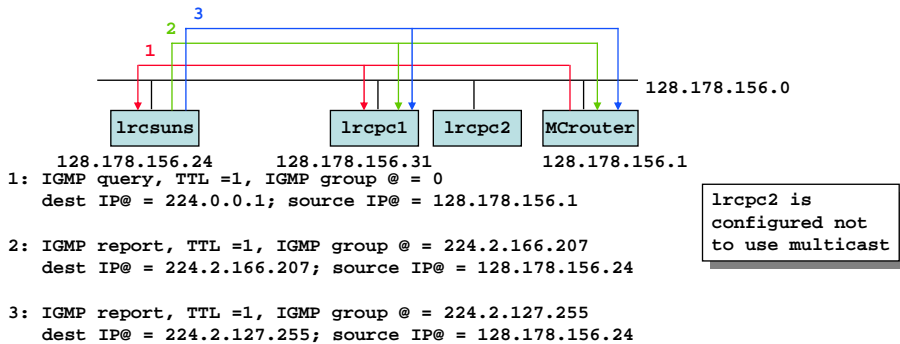
- ❑ Systems have to know which group they belong to
 - Hosts: application processes register to IP
 - Routers: learn if members present with IGMP
- ❑ Direct send to link layer:
 - algorithmic mapping of 23 last bits : ex : 224.2.166.207 -> 01-00-5E-02-A6-CF

56

IGMP: Internet Group Management Protocol

Purpose: manage group membership inside one subnet

- ❑ routers: know if group is present on an interface
 - know whether to forward locally or not
- ❑ hosts: know if a multicast address is already in use locally

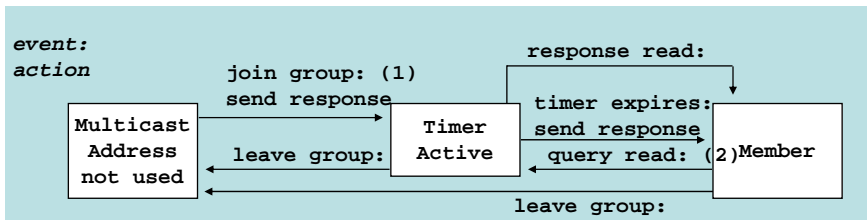


IGMP Host Implementation

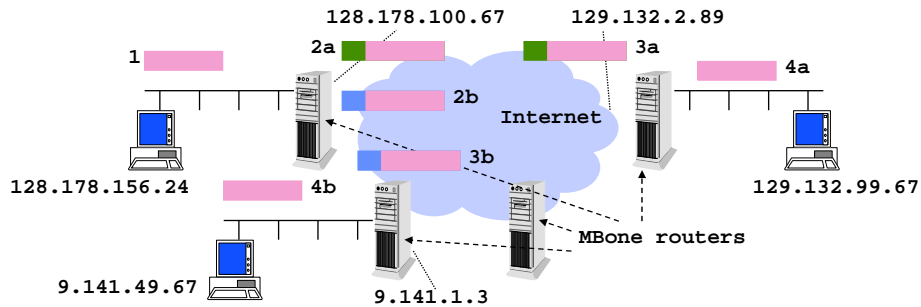
Host Implementation

- ❑ goal: avoid avalanche effects - one router originated query might cause a burst of reports
- ❑ solution = **synchronization avoidance protocol**
 - 1. hosts delay responses randomly
 - 2. hosts listen to responses, only first one answers

Host IGMP Finite State Machine



MBone (1)



| | dest IP addr | srce IP addr | prot | IP packet data part |
|----|---------------|----------------|------|--|
| 1 | 224.2.165.231 | 128.178.156.24 | UDP | bla bla |
| 2a | 129.132.2.89 | 128.178.100.67 | IP | 224.2.165.231 128.178.156.24 UDP bla bla |
| 3a | 129.132.2.89 | 128.178.100.67 | IP | 224.2.165.231 128.178.156.24 UDP bla bla |
| 4a | 224.2.165.231 | 128.178.156.24 | UDP | |

59

MBone (2)

- Global Multicast not available
 - no stable routing protocol implemented in all routers of the Internet
- Mbone = a network of "routers" supporting multicast
- Tunneling used to build virtual links
 - protocol = 4 in IP header
 - example of the use of a network layer as a layer 2 by another network
 - other examples: IPv6 over IPv4, IP over Frame Relay, over ATM, AppleTalk over IP, etc.
 - MBone "hacks"
 - limitation of multicast enforced by Mbone routers on TTL field
 - multicast routing with DVMRP
 - each router computes SPT from each source using distance vector algorithm
 - reverse path forwarding (RPF)

60

8. Routing

- ❑ Packet Forwarding
 - for every packet
 - done in real time
- ❑ Routing
 - computation of routing tables or data structures for unicast and multicast
 - normally only between routers
 - non-real time: latency up to 2 minutes
 - uses protocols such as RIP, OSPF, EIGRP (Cisco) for unicast and DVMRP, M-OSPF, PIM for multicast
 - ICMP-redirect may alter host routing tables

61

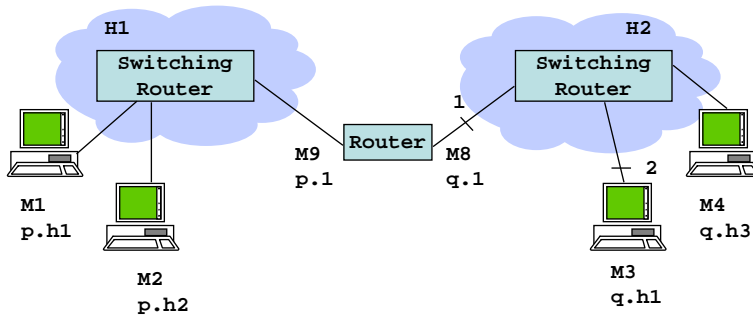
Router Definitions

- ❑ **Definition:** IP router
 - a system that forwards packets based on IP addresses
 - performs packet forwarding + control method
 - routing, configuration management, DHCP relay, IPv6 router advertisements...
- ❑ **Implementation:**
 - any UNIX machine can be configured as IP router
 - normally, dedicated packet forwarder called router
- ❑ **Multiprotocol router**
 - a system that forwards packets based on layer 3 addresses for various protocol architectures (ex: IP, Appletalk)
 - CISCO, IBM, etc...
 - most multiprotocol routers perform both bridging and routing
 - architecture: bridge + router
 - implementation: one CISCO
 - IP router boxes also perform other functions: port filtering, DHCP relay, ...

62

Routers and Bridges

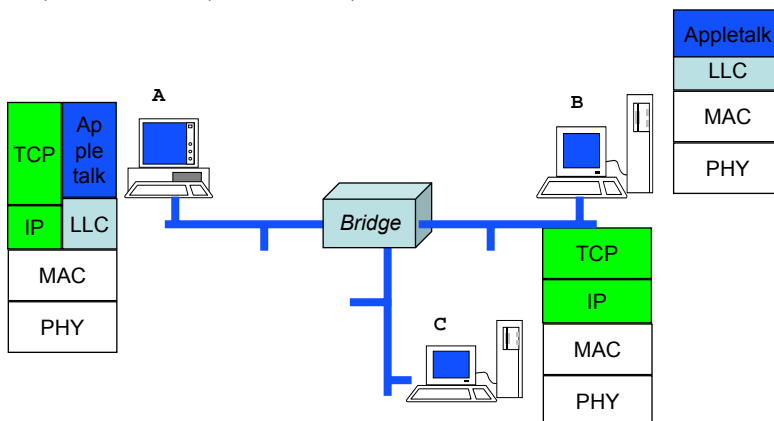
- ❑ Routers extend the scale limitations of bridges
- ❑ But bridges are "plug and play" and are simpler to manage
- ❑ Intelligent products combine advantages of both
 - example: "switching router" - knows the MAC addresses of directly attached hosts



63

Protocols Other than TCP/IP

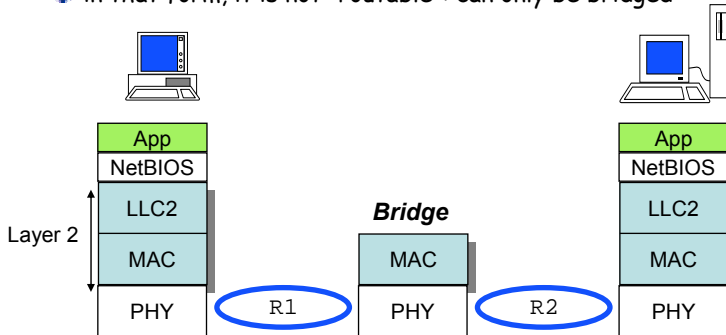
- ❑ Some other protocol families (ex: Appletalk, IPX) are not compatible with TCP/IP
- ❑ routers must be multiprotocol
- ❑ MAC interface is standard -> bridges are not aware of higher layer protocol (they are "multiprotocol")



64

NetBIOS

- ❑ NetBIOS was originally developed to work only in one bridged LAN
 - uses LLC-2, similar to TCP but located in layer 2 (also called NETBEUI)
 - in that form, it is not "routable": can only be bridged

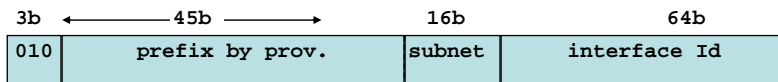


- ❑ NetBIOS today is offered as a TCP/IP application
 - uses the NBT reserved port
 - Windows machines at EPFL use TCP/IP only

65

9. IPv6

- ❑ The current IP is IPv4
- ❑ IPv4 address space is too small (32 bits)
 - will be exhausted some day
- ❑ IPv6 is the new version of IP
 - addresses are 128 bit longs
 - draft standard



allocated by org / provider allocated by customer

- ❑ IPv6 is incompatible with IPv4

66

Plug and Play and DHCP

- ❑ IPv6 address is allocated automatically by negotiation with routers
 - "stateless allocation"
- ❑ alternatively, DHCP can be used
- ❑ DHCP can be used with IPv4 also
 - DHCP server on LAN has a list of IP addresses that can be allocated dynamically
 - MAC address used to identify a host to DHCP server
 - renumbering is possible
 - more complex to use than IPv6 stateless allocation

67

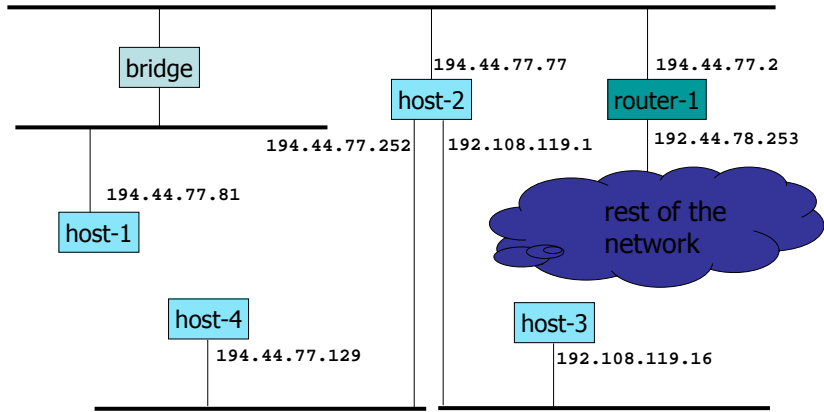
10. Static Configuration of a Unix host

```
/usr/etc/ifconfig interface [ address_family ]
    [ address [ dest_address ] ] [ netmask mask ]
    [ broadcast address ] [ up ] [ down ] [ trailers ]
    [ -trailers ] [ arp ] [ -arp ] [ private ]
    [ -private ] [ metric n ] [ auto-revarp ]

host-1# ifconfig le0 host-1 netmask +
Setting netmask of le0 to 255.255.255.128
# + means netmask from /etc/netmask
host-1# ifconfig -a
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING>
    inet 192.44.77.81 netmask ffffffff broadcast 192.44.77.0
    ether 8:0:20:1c:74:84
lo0: flags=849<UP,LOOPBACK,RUNNING>
    inet 127.0.0.1 netmask ff000000
```

68

Example interconnection



69

Routing tables

host-1 (192.44.77.81) :

```
>netstat -n -r
```

Routing tables

| Destination | Gateway | Flags | Refcnt | Use | Interface |
|----------------|--------------|-------|--------|-------|-----------|
| 192.108.119.16 | 192.44.77.77 | UGHD | 1 | 1683 | le0 |
| 127.0.0.1 | 127.0.0.1 | UH | 2 | 12971 | lo0 |
| default | 192.44.77.2 | UG | 3 | 16977 | le0 |
| 192.44.77.0 | 192.44.77.81 | U | 13 | 5780 | le0 |

U - up

G - gateway (next router)

H - host route

D - route from ICMP Redirect

70

Routing tables

```
host-2 (192.44.77.77) :
>rsh host-2 netstat -n -r
Routing tables
```

| Destination | Gateway | Flags | Refcnt | Use | Interface |
|---------------|---------------|-------|--------|----------|-----------|
| 127.0.0.1 | 127.0.0.1 | UH | 3 | 351344 | lo0 |
| default | 192.44.77.2 | UG | 3 | 17388997 | 1e0 |
| 192.44.77.128 | 192.44.77.252 | U | 26 | 504768 | 1e2 |
| 192.44.77.0 | 192.44.77.77 | U | 24 | 10702069 | 1e0 |
| 192.108.119.0 | 192.108.119.1 | U | 2 | 249777 | 1e1 |

71

Modifying routing tables

```
/usr/etc/route [ -fn ] add|delete [ host|net ] destination [gateway [
metric ] ]
host-1# netstat -r
Routing tables
```

| Destination | Gateway | Flags | Refcnt | Use | Interface |
|-------------|-----------|-------|--------|-------|-----------|
| localhost | localhost | UH | 2 | 13569 | lo0 |
| 192.44.77.0 | host-1 | U | 18 | 13272 | 1e0 |

```
host-1# ping 133.11.11.11
sendto: Network is unreachable
host-1# route add 0.0.0.0 router-1 1
add net 0.0.0.0 gateway router-1
```

72

Modifying routing tables

```
host-1# netstat -r
Routing tables
Destination      Gateway         Flags          Refcnt Use      Interface
localhost        localhost      UH             2      13591    lo0
default          router-1       UG             0      0        le0
192.44.77.0      host-1         U              16     13566    le0
host-1# ping 133.11.11.11
133.11.11.11 is alive
```

73

Facts to Remember

- IP is a connectionless network layer
- IP addresses are 32 bit numbers
- One IP address per interface
- A unicast IP address has a topological meaning
- Routers scale well because they can aggregate routes
- Multicast addresses are logical
- Hosts on the Internet exchange packets with IP addresses
- Non routable protocols use only MAC addresses

74

Solutions

75

Representation of IP Addresses

❑ **dotted decimal**: group bits in bytes, write the decimal representation of the number

● example 1: 128.191.151.1

● example 2: 129.192.152.2

❑ **hexadecimal**: hexadecimal representation -- fixed size string

● example 1: x80 BF 97 01

● example 2: x81 C0 98 02

❑ **binary**: string of 32 bits (2 symbols: 0, 1)

● example 1: b0100 0000 1011 1111 1001 0111 0000 0001

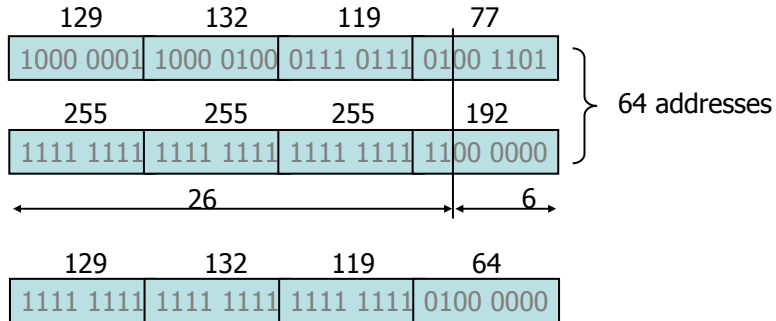
● example 2: b0100 0001 1100 0000 1001 0111 0000 0010

76

A Subnet Prefix is written using one of two Notations: masks / prefixes

● example 2: 129.132.119.77 mask 255.255.255.192

● Q1: what is the prefix ? A: 129.132.119.64



● Q2: how many host ids can be allocated ? A: 64 (minus the reserved addresses: 62)

Prefix Notation

example 2:

● Q1: write 129.132.119.77 mask 255.255.255.192 in prefix notation

A: 129.132.119.77/26 or 129.132.119.64/26

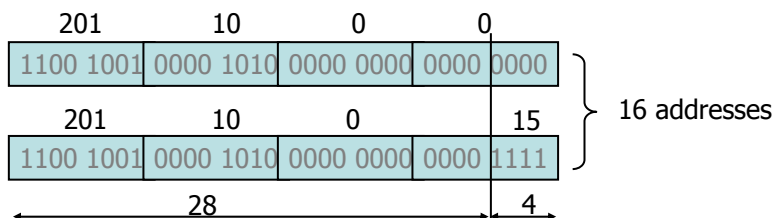
● Q2: are these prefixes different ?

● 201.10.0.0/28, 201.10.0.16/28, 201.10.0.32/28, 201.10.0.48/28

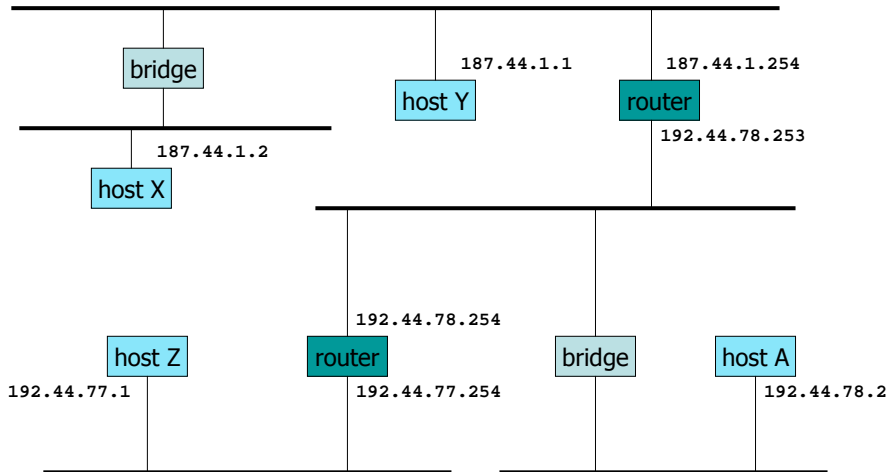
A: they differ in bits that are not the last 4 ones, thus they are all different prefixes

● how many IP addresses can be allocated to each of the distinct subnets ?

A: 14 (16 minus 2 reserved)



Test Your Understanding (1)



- A: Host A is on subnetwork 192.44.78

79

Test your Understanding (2)

- ❑ Q1: An Ethernet segment became too crowded; we split it into 2 segments, interconnected by a router. Do we need to change some IP host addresses ?
A: yes in general. Two different subnets cannot have the same prefix
- ❑ Q2: same with a bridge
A: no, bridging is transparent.
- ❑ Q3: compare the two
A: bridging is plug and play but the network performance is more difficult to guarantee (broadcasts + spanning tree)

80

Example

- Q: Fill in the table if an IP packet has to be sent from lrscuns

| final destination | next hop | case number |
|-------------------|---------------|-------------|
| 128.178.79.9 | 128.178.156.1 | 3 |
| 128.178.156.7 | 128.178.156.7 | 2 |
| 127.0.0.1 | loopback | 2 |
| 128.178.84.133 | 128.178.156.1 | 3 |
| 129.132.1.45 | 128.178.156.1 | 3 |

- Q: Fill in the table if an IP packet has to be sent from ed2-in

| final destination | next hop | case number |
|-------------------|---------------|-------------|
| 128.178.79.9 | 128.178.182.3 | 3 |
| 128.178.156.7 | 128.178.182.5 | 3 |
| 127.0.0.1 | loopback | 2 |
| 128.178.84.133 | 128.178.15.13 | 3 |
| 129.132.1.45 | 128.178.100.3 | 3 |

81

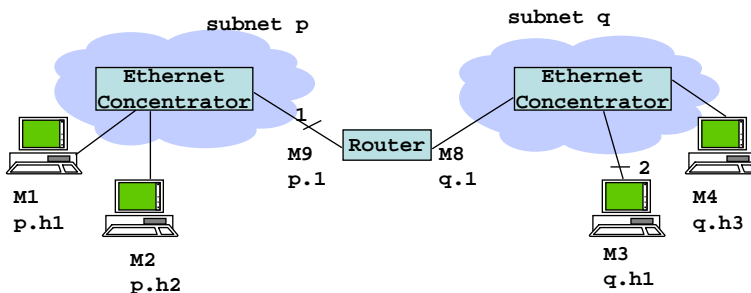
Test Your Understanding (3)

- Q1: What are the MAC and IP addresses at points 1 and 2 for packets sent by M1 to M3 ? At 2 for packets sent by M4 to M3 ?(Mx = mac address)

A: at 1: srce IP@=p.h1, dest IP@=q.h1, MACsrce=M1, MACdest=M9
 at 2: srce IP@=p.h1, dest IP@=q.h1, MACsrce=M8, MACdest=M3
 at 2: srce IP@=q.h3, dest IP@=q.h1, MACsrce=M4, MACdest=M3

- Q2: What must the router do when it receives a packet to M2 for the first time?

A: send an ARP request to the LAN p



82

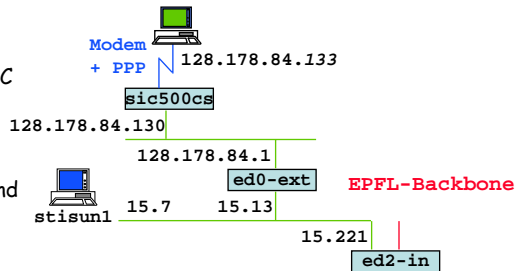
Proxy ARP

❑ Q1: how must sics500cs routing table be configured ?

- A: one host route per host such as 128.178.84.133

❑ Q2: explain what happens when ed2-in has a packet to send to 128.178.84.133

- packet sent to ed0-ext
- ARP sent by ed0-ext for target address = 128.178.84.133
- sics500cs responds with MAC addr = sics500cs's MAC addr
- packet sent ed0-ext to sics500cs
- sics500cs reads host route and forwards to 128.178.84.133 (case 1 of IP forwarding algorithm)



83

MTU

❑ value of short MTU ?

- reduces queue lengths and delays
- on lossy links (radio) reduces proba of packet error

❑ of long MTU ?

- reduces per packet processing

84