ΗΥ 335
Φροντιστήριο 9ο

Χειμερινό Εξάμηνο 2009-2010

Μακρογιαννάκης Αντώνης
makrog@csd.uoc.gr
Παπακωνσταντίνου Άρτεμις
artpap@csd.uoc.gr

11/12/2009

Roadmap

- IP Multicasting
  - IGMP
  - Multicast Routing
- DHCP vs ARP
- Obtaining an IP address
  - RARP
  - BOOTP
  - DHCP
- Traceroute

Roadmap

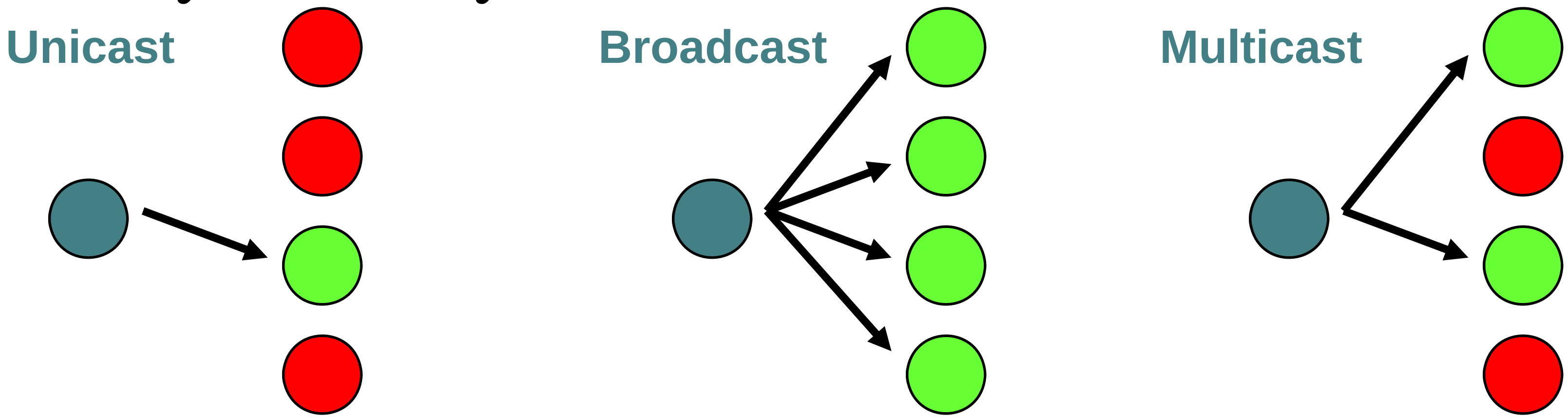
- IP Multicasting
  - IGMP
  - Multicast Routing
- DHCP vs ARP
- Obtaining an IP address
  - RARP
  - BOOTP
  - DHCP
- Traceroute

Muticasting

- What is multicasting?
- Why use multicasting?
- IGMP Protocol
- Multicast Routing
  - Source-Based tree
  - Group-Shared tree

Multicasting

- Multicast communications refers to one-to-many or many-to-many communications.



IP Multicasting refers to the implementation of multicast communication in the Internet

Multicast is driven by receivers: Receivers indicate interest in receiving data

Why use multicasting?

- Many applications transmit the same data at one time to multiple receivers
  - Broadcasts of Radio or Video
  - Videoconferencing
  - Shared Applications

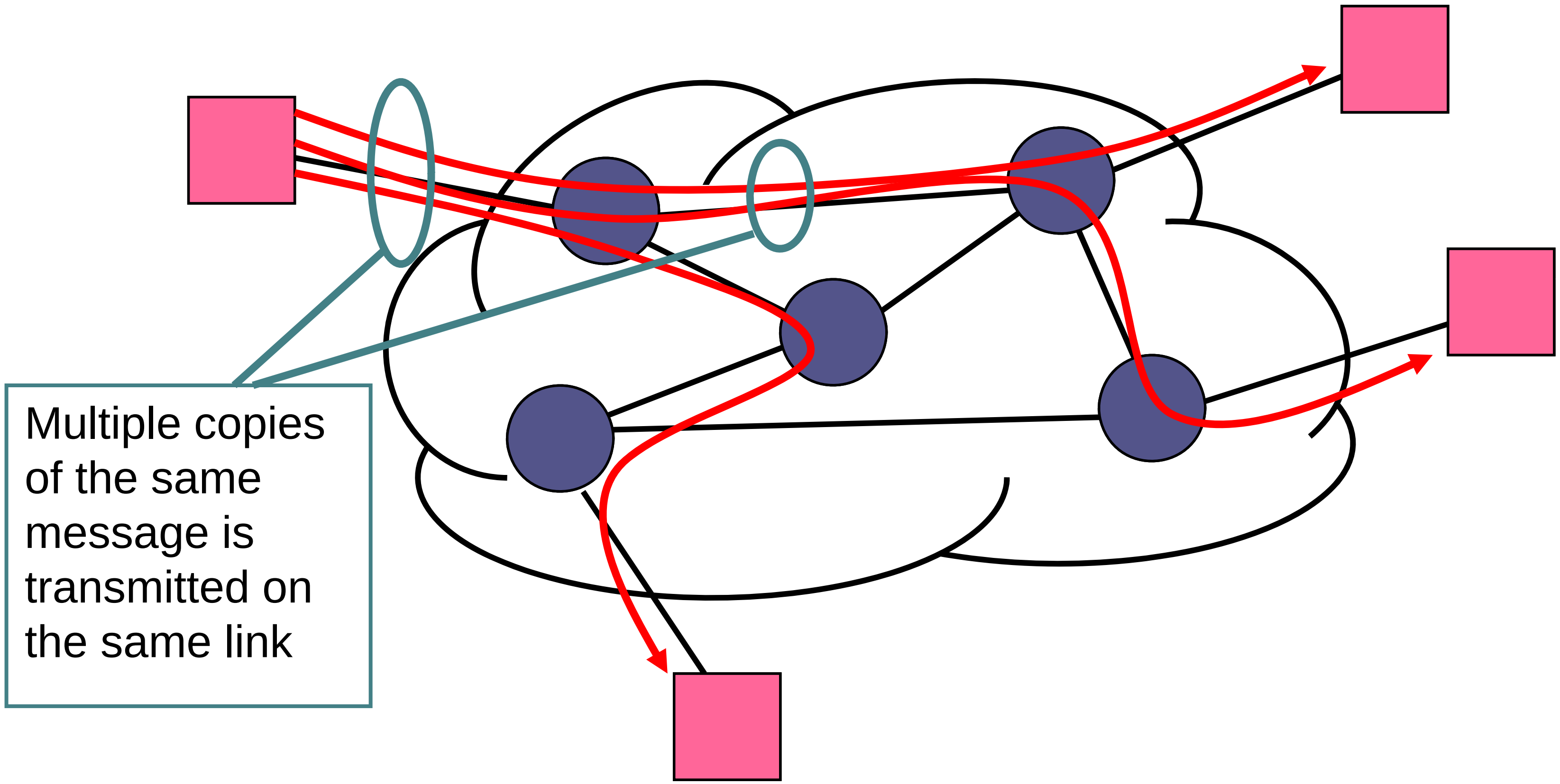
- A network must have mechanisms to support such applications in an efficient manner

Multicast Groups

- The set of receivers for a multicast transmission is called a **multicast group**
  - A multicast group is identified by a **multicast address**
  - A user that wants to receive multicast transmissions **joins** the corresponding multicast group, and becomes a **member** of that group
- After a user joins, the network builds the necessary routing paths so that the user receives the data sent to the multicast group

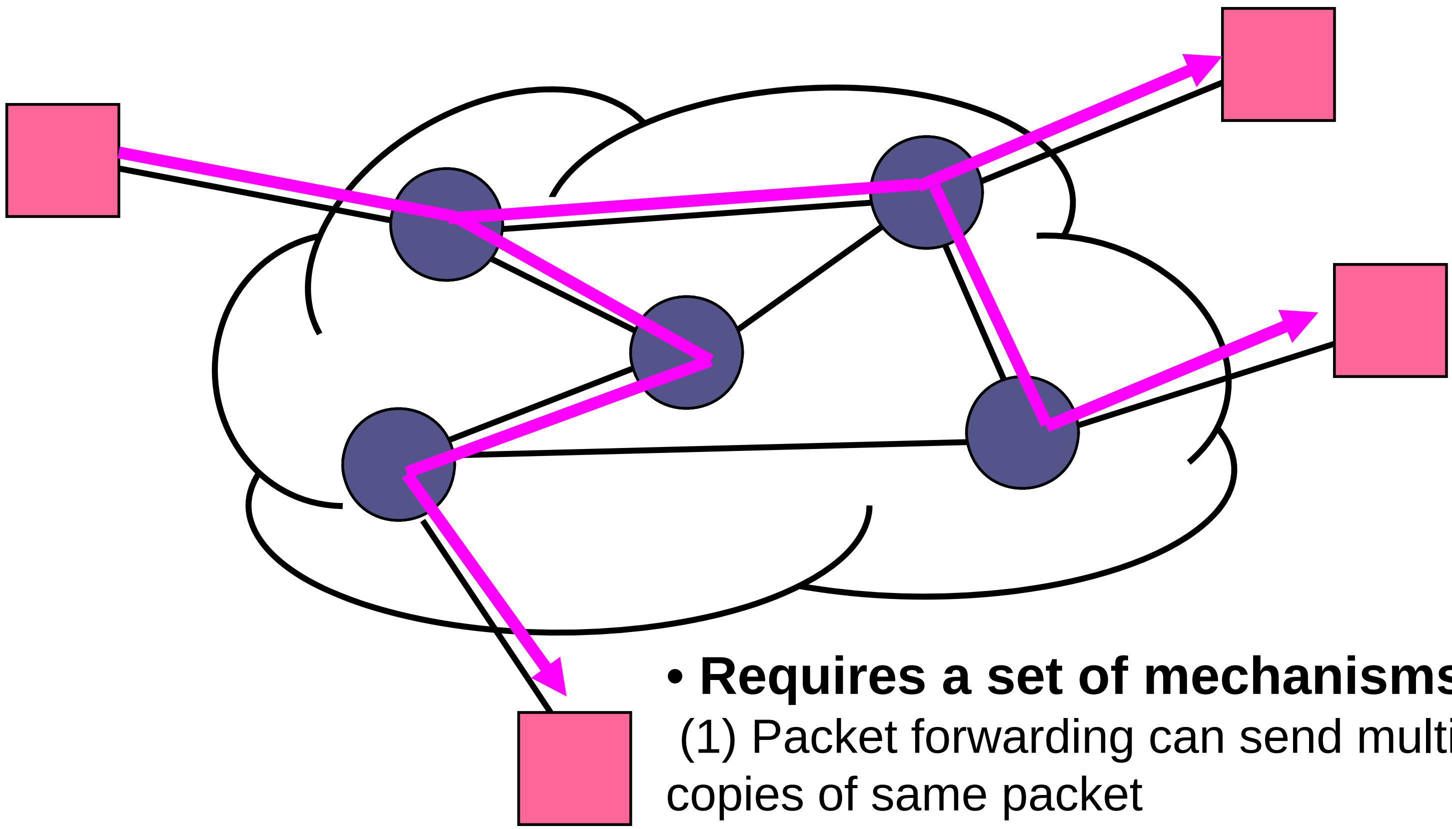
Multicasting over a Packet Network

- Without support for multicast at the network layer:



Multicasting over a Packet Network

- With support for multicast at the network layer:



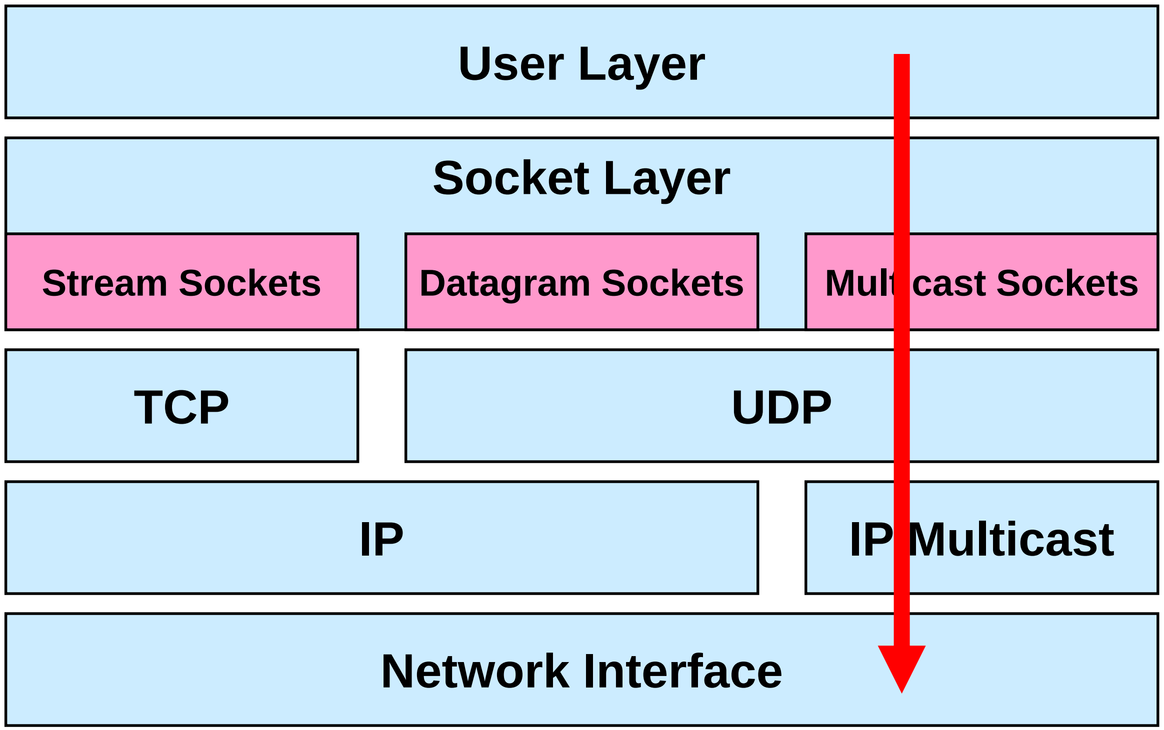
- **Requires a set of mechanisms:**
  (1) Packet forwarding can send multiple copies of same packet
  (2) Multicast routing algorithm which builds a spanning tree (dynamically)

Semantics of IP Multicast

- Multicast groups are identified by IP addresses in the range 224.0.0.0 - 239.255.255.255 (class D address)
- Every host (*more precisely:* interface) can join and leave a multicast group dynamically
    - no access control
- Every IP datagram send to a multicast group is transmitted to all members of the group
    - no security, no "floor control"
    - Sender does not need to be a member of the group
- The IP Multicast service is unreliable

The IP Protocol Stack

- IP Multicasting only supports UDP as higher layer
- There is no multicast TCP !



The IGMP Protocol

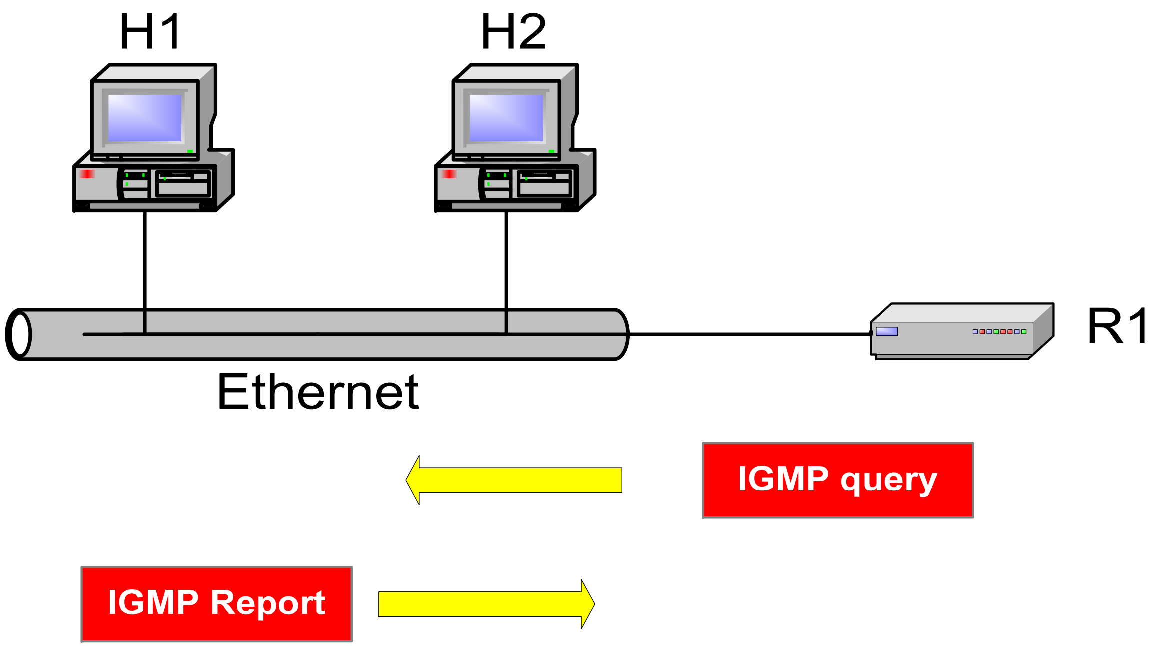
- The **Internet Group Management Protocol (IGMP)** is a simple protocol for the support of IP multicast.
- IGMP operates on a physical network (e.g., single Ethernet Segment.
- IGMP is used by multicast routers to keep track of membership in a multicast group.
- Support for:
  - Joining a multicast group
  - Query membership
  - Send membership reports

IGMP Protocol

- A host sends an IGMP report when it joins a multicast group (Note: multiple processes on a host can join. A report is sent only for the first process).
- No report is sent when a process leaves a group
- A multicast router regularly multicasts an IGMP query to all hosts (group address is set to zero).
- A host responds to an IGMP query with an IGMP report.

- Multicast router keeps a table on the multicast groups that have joined hosts. The router only forwards a packet, if there is a host still joined.
- Note: Router does not keep track which host is joined.

IGMP Protocol



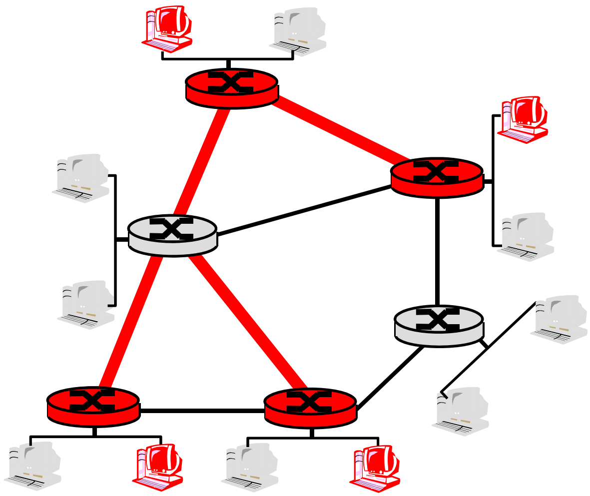
IP Multicasting

- There are three essential components of the IP Multicast service:

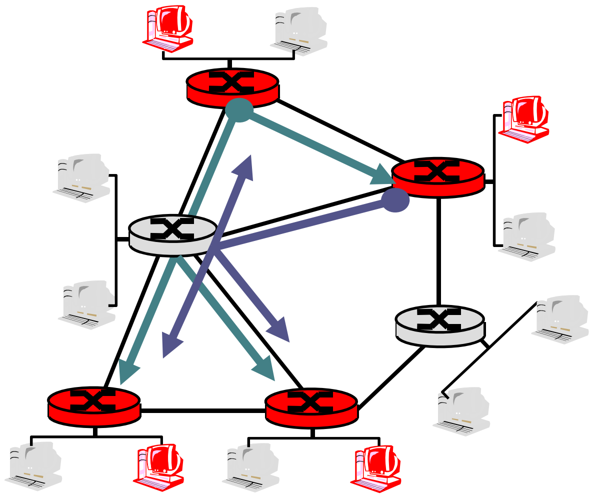
IP Multicast Addressing
IP Group Management
Multicast Routing

Multicast Routing: Problem Statement

- ***Goal:*** find a tree (or trees) connecting routers having local mcast group members
  - *tree:* not all paths between routers used
  - *source-based:* different tree from each sender to rcvrs
  - *shared-tree:* same tree used by all group members



Shared tree



Source-based trees

Approaches for building mcast trees

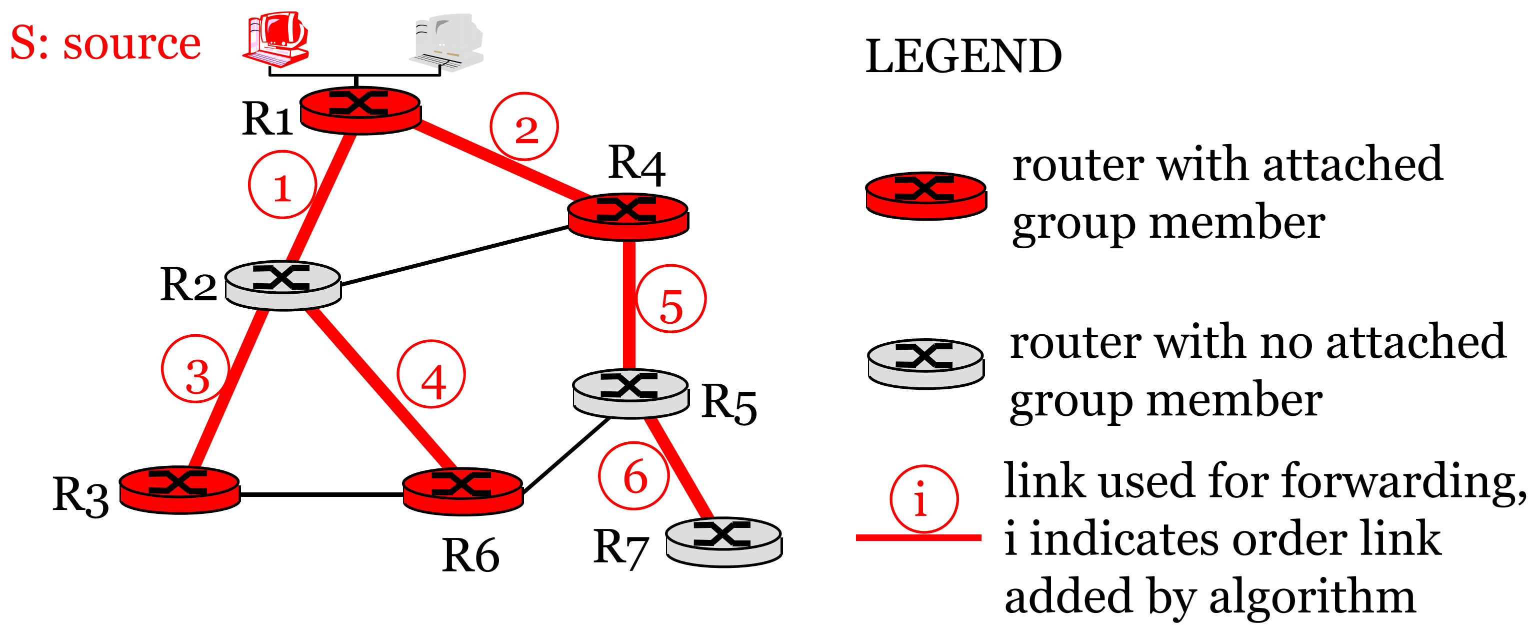
Approaches:

- source-based tree: one tree per source
  - shortest path trees
  - reverse path forwarding
- group-shared tree: group uses one tree
  - minimal spanning (Steiner)
  - center-based trees

…we first look at basic approaches, then specific protocols adopting these approaches

Shortest Path Tree

- mcast forwarding tree: tree of shortest path routes from source to all receivers
  - Dijkstra's algorithm

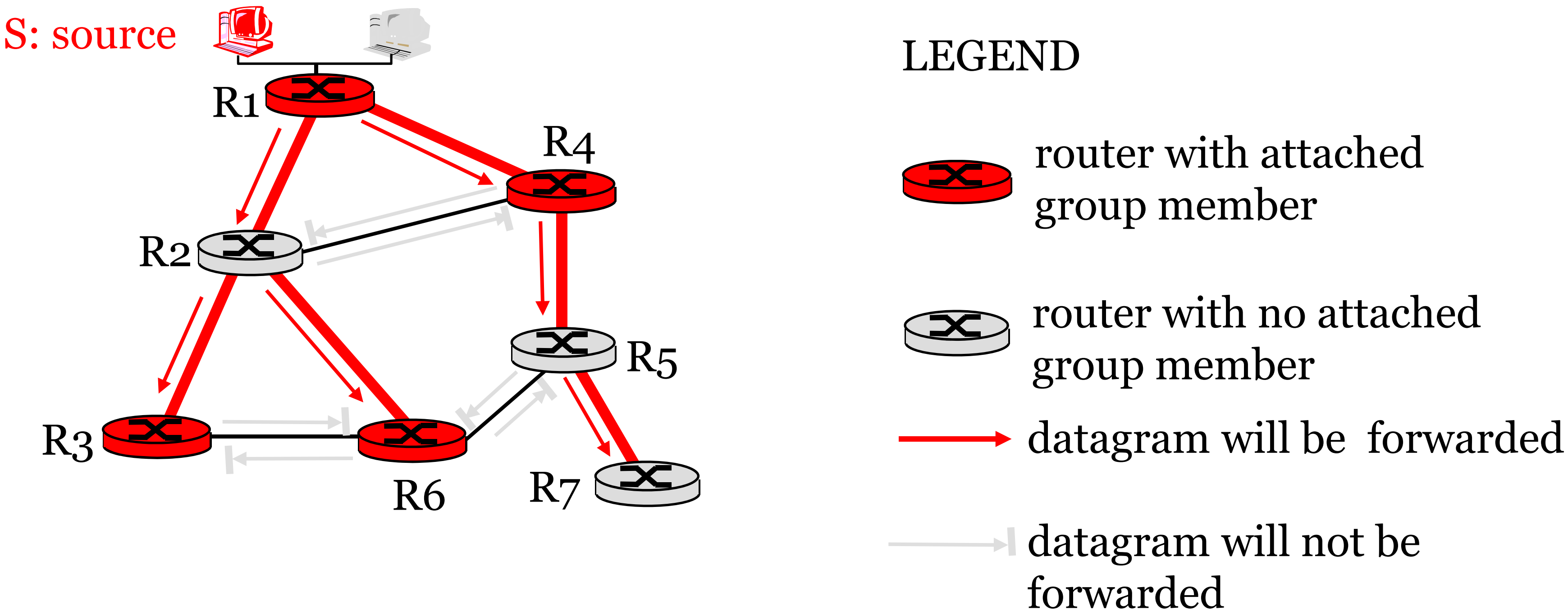


Reverse Path Forwarding

- rely on router's knowledge of unicast shortest path from it to sender
- each router has simple forwarding behavior:

if (mcast datagram received on incoming link on shortest path back to center)
then flood datagram onto all outgoing links
else ignore datagram

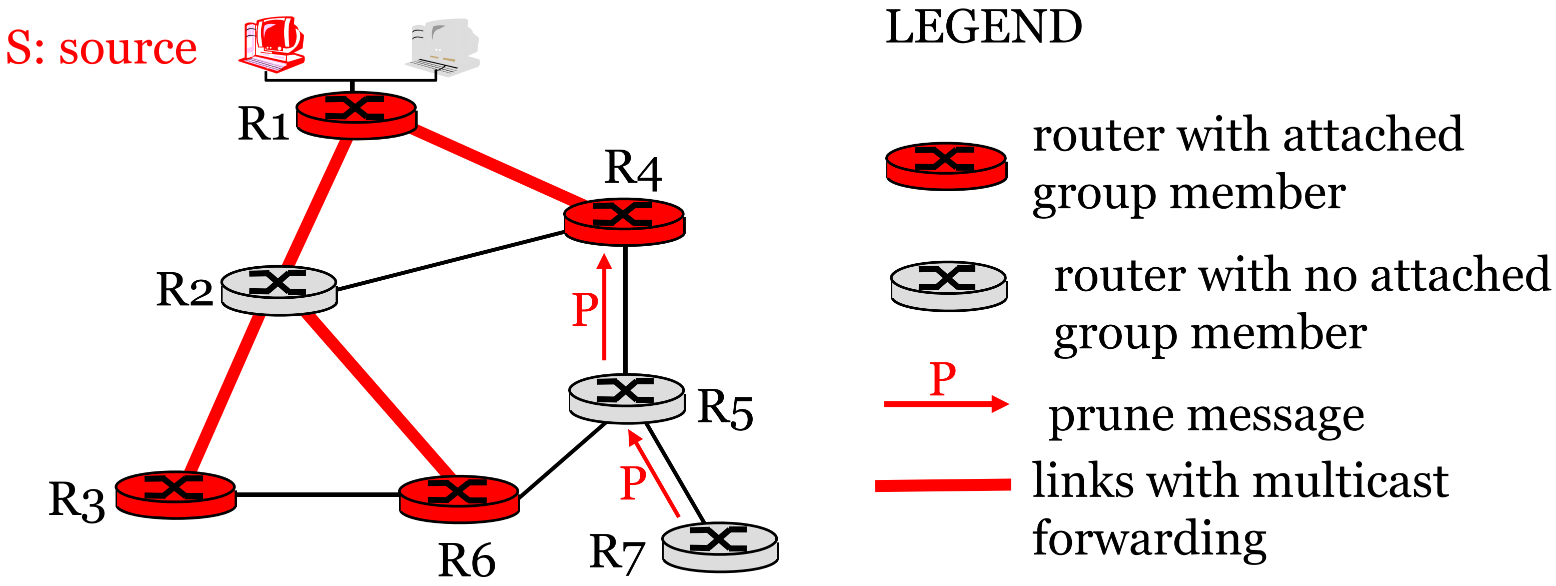
Reverse Path Forwarding: example



– may be a bad choice with asymmetric links

Reverse Path Forwarding: pruning

- forwarding tree contains subtrees with no mcast group members
  - no need to forward datagrams down subtree
  - "prune" msgs sent upstream by router with no downstream group members



Shared-Tree: Steiner Tree

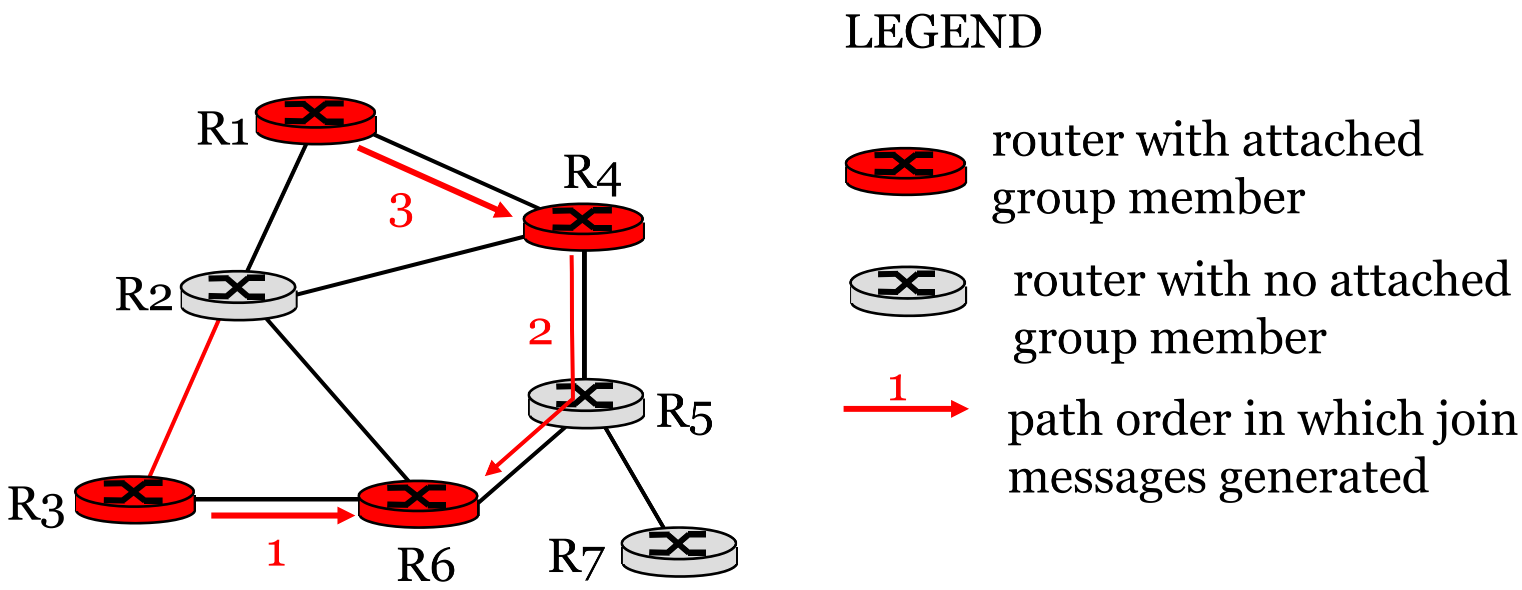
- Steiner Tree: minimum cost tree connecting all routers with attached group members
- problem is NP-complete
- excellent heuristics exists
- not used in practice:
  - computational complexity
  - information about entire network needed
  - monolithic: rerun whenever a router needs to join/leave

Center-based trees

- single delivery tree shared by all
- one router identified as "*center*" of tree
- to join:
  - edge router sends unicast *join-msg* addressed to center router
  - *join-msg* "processed" by intermediate routers and forwarded towards center
  - *join-msg* either hits existing tree branch for this center, or arrives at center
  - path taken by *join-msg* becomes new branch of tree for this router

Center-based trees: an example

Suppose R6 chosen as center:



Internet Multicasting Routing

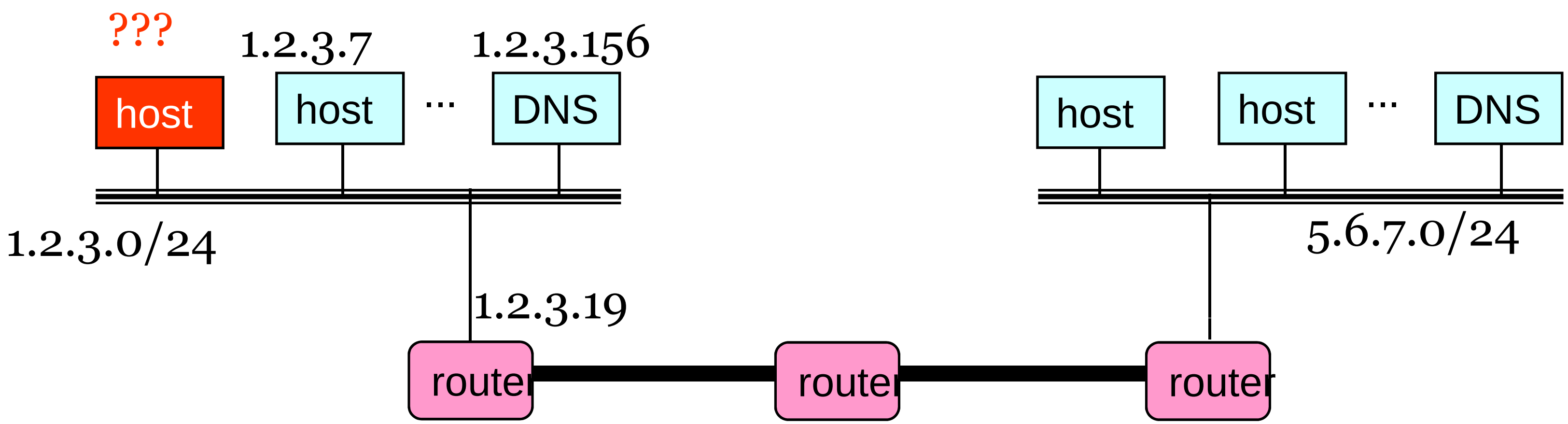
- DVMRP
- MOSPF
- PIM

Roadmap

- IP Multicasting
  - IGMP
  - Multicast Routing
- DHCP vs ARP
- Obtaining an IP address
  - RARP
  - BOOTP
  - DHCP
- Traceroute

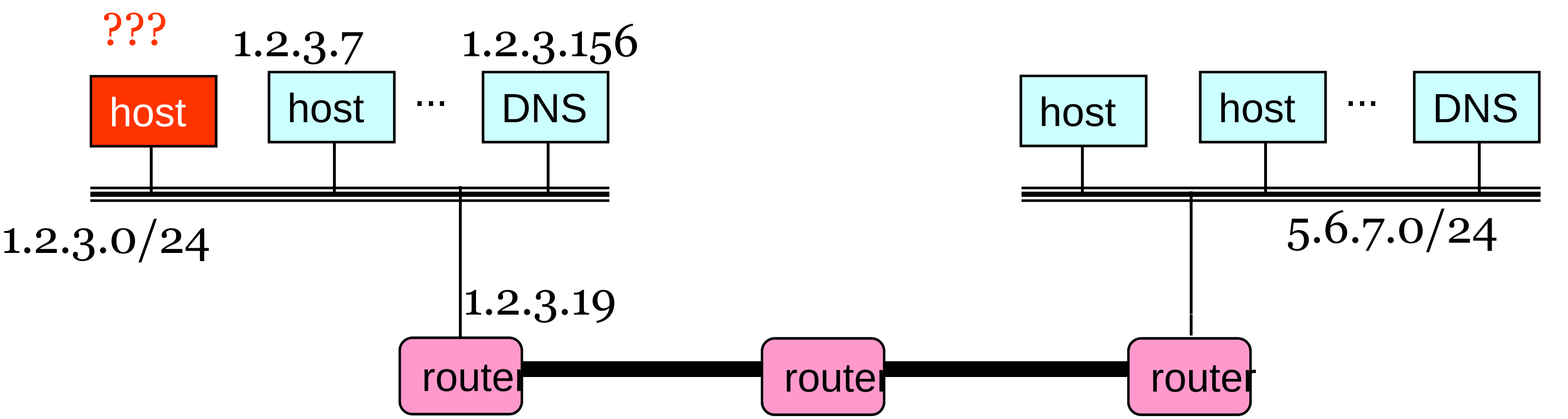
How To Bootstrap an End Host?

- What local Domain Name System server to use?
- What IP address the host should use?
- How to send packets to remote destinations?
- How to ensure incoming packets arrive?



Avoiding Manual Configuration

- Dynamic Host Configuration Protocol (DHCP)
  - End host learns how to send packets
  - Learn IP address, DNS servers, and gateway
- Address Resolution Protocol (ARP)
  - Others learn how to send packets to the end host
  - Learn mapping between IP address & interface address

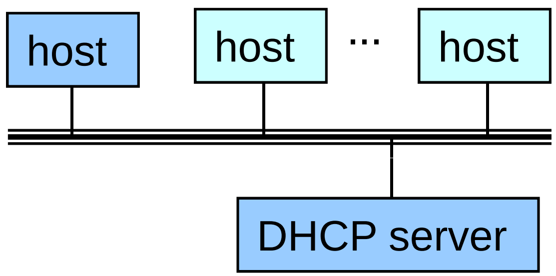


Key Ideas in Both Protocols

- Broadcasting: when in doubt, shout!
  - Broadcast query to all hosts in the local-area-network
  - … when you don’t know how to identify the right one
- Caching: remember the past for a while
  - Store the information you learn to reduce overhead
  - Remember your own address & other host’s addresses
- Soft state: … but eventually forget the past
  - Associate a time-to-live field with the information
  - … and either refresh or discard the information
  - Key for robustness in the face of unpredictable change

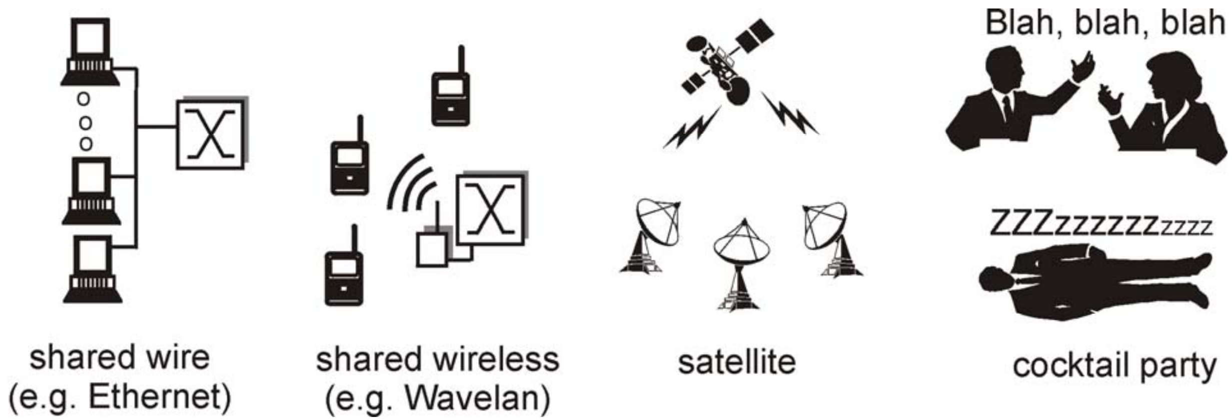
Bootstrapping Problem

- Host doesn't have an IP address yet
  - So, host doesn't know what source address to use
- Host doesn't know who to ask for an IP address
  - So, host doesn't know what destination address to use
- Solution: shout to discover a server who can help
  - Broadcast a server-discovery message
  - Server sends a reply offering an address



Broadcasting

- Broadcasting: sending to everyone
  - Special destination address: FF-FF-FF-FF-FF-FF
  - All adapters on the LAN receive the packet
- Delivering a broadcast packet
  - Easy on a "shared media"
  - Like shouting in a room – everyone can hear you



Response from the DHCP Server

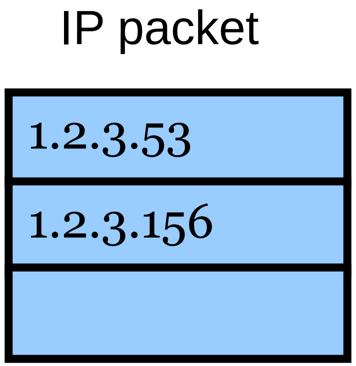
- DHCP "offer message" from the server
  - Configuration parameters (proposed IP address, mask, gateway router, DNS server, ...)
  - Lease time (the time the information remains valid)
- Multiple servers may respond
  - Multiple servers on the same broadcast media
  - Each may respond with an offer
  - The client can decide which offer to accept
- Accepting one of the offers
  - Client sends a DHCP request echoing the parameters
  - The DHCP server responds with an ACK to confirm
  - ... and the other servers see they were not chosen

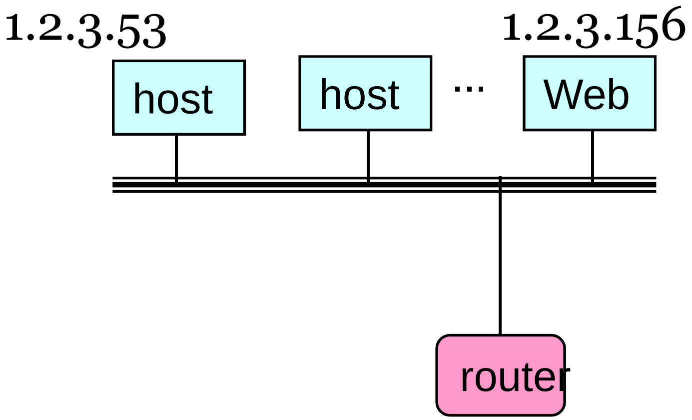
So, Now the Host Knows Things

- IP address
- Mask
- Gateway router
- DNS server
- …

- And can send packets to other IP addresses
  - But, how to learn the MAC address of the destination?

Sending Packets Over a Link



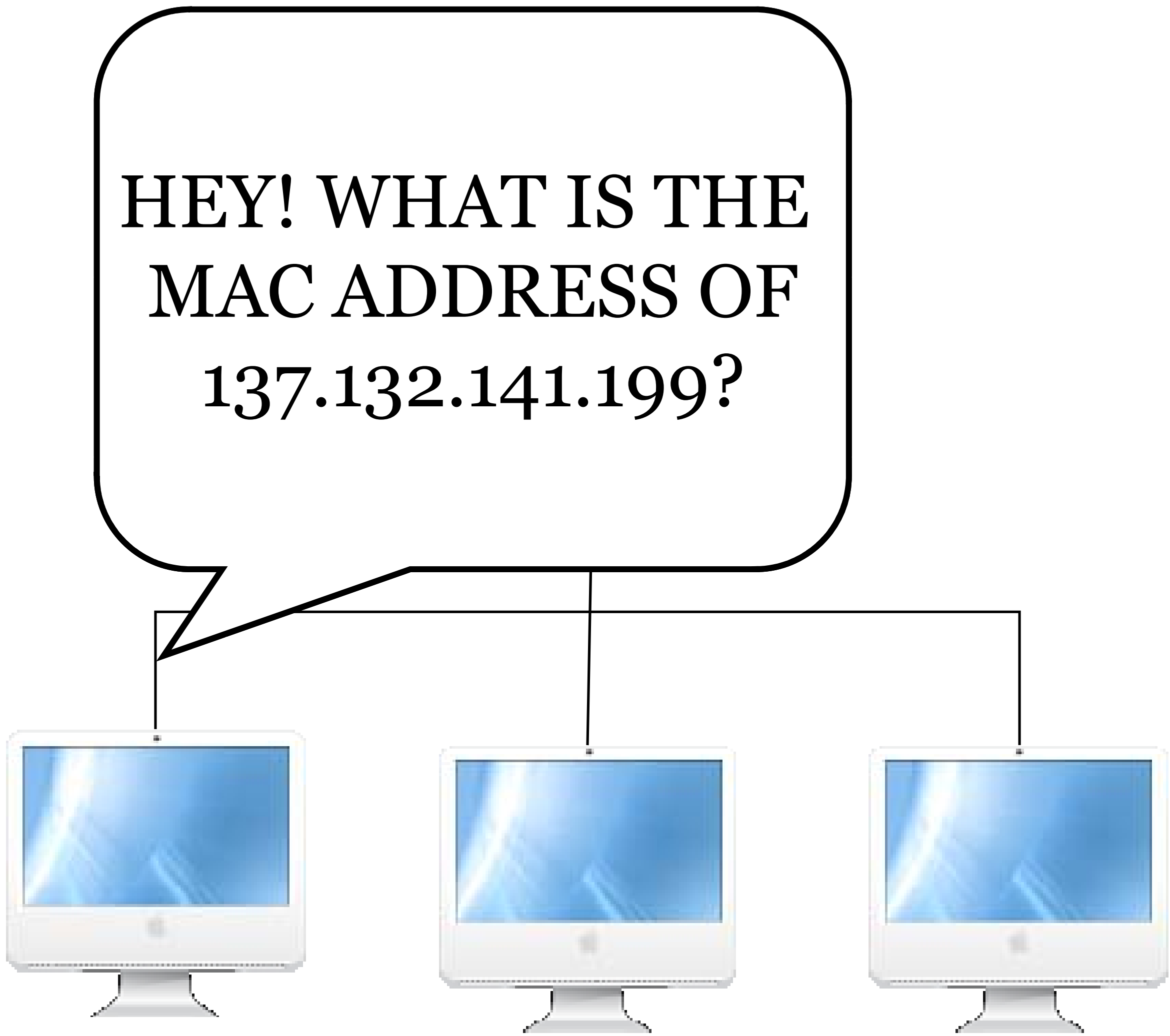


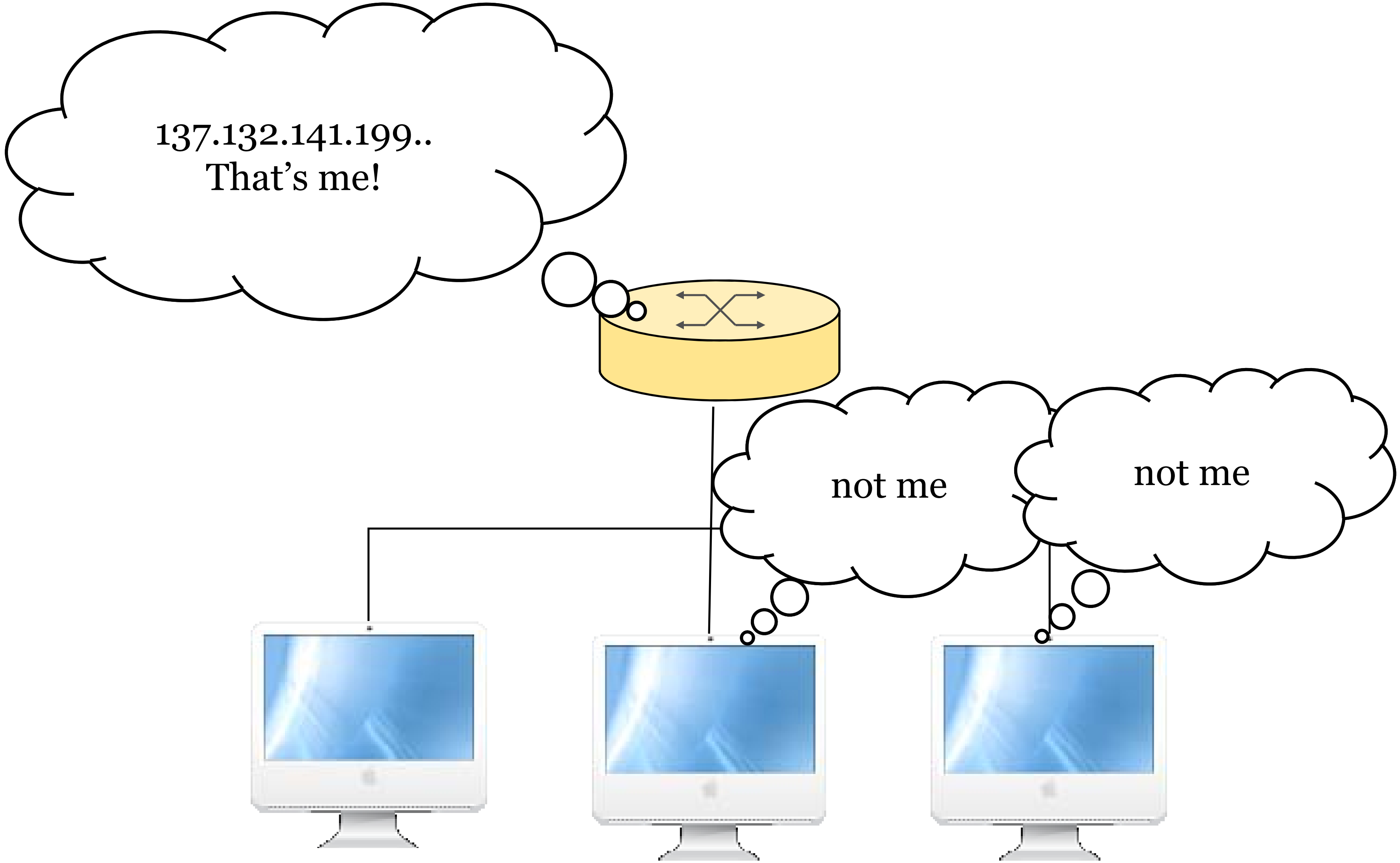
- Adaptors only understand MAC addresses
  - Translate the destination IP address to MAC address
  - Encapsulate the IP packet inside a link-level frame

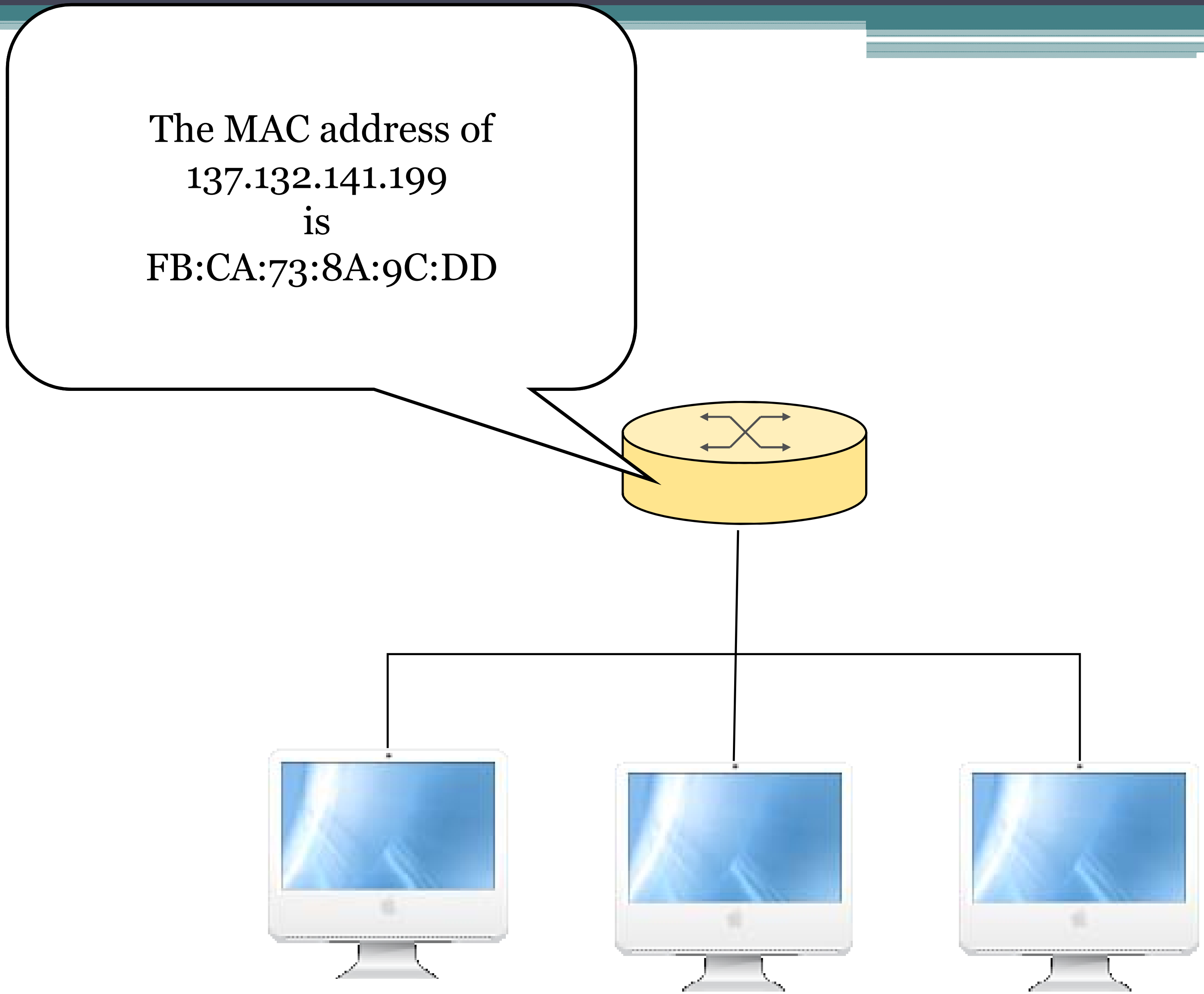
Address Resolution Protocol Table

- Every node maintains an ARP table
  - (IP address, MAC address) pair
- Consult the table when sending a packet
  - Map destination IP address to destination MAC address
  - Encapsulate and transmit the data packet

- But, what if the IP address is not in the table?
  - Sender broadcasts: "Who has IP address 1.2.3.156?" (ARP query)
  - Receiver responds: "MAC address 58-23-D7-FA-20-B0" (unicast)
  - Sender caches the result in its ARP table
    - Entries in ARP table have a timer and an entry is removed when its timer expires
- No need for network administrator to get involved


HEY! WHAT IS THE MAC ADDRESS OF 137.132.141.199?


137.132.141.199..
That's me!
not me
not me


The MAC address of
137.132.141.199
is
FB:CA:73:8A:9C:DD

Roadmap

- IP Multicasting
  - IGMP
  - Multicast Routing
- DHCP vs ARP
- Obtaining an IP address
  - RARP
  - BOOTP
  - DHCP
- Traceroute

Determining an IP Address at Startup- RARP

- How does a machine without permanent storage determine its IP address?
  - Diskless workstation employs network booting to load its OS from a server
  - OS images with specific IP's cannot be used on multiple machines
  - Critical for network appliances or embedded systems
- Use the network to obtain an IP from a remote server
  - System must use its physical address to communicate
  - Requests address from server which maintains table of IP's
  - System doesn't know the server - sends broadcast request for address

Reverse Address Resolution Protocol

- RARP is part of the TCP/IP specification
- RARP operates much like ARP
  - A requestor broadcasts is RARP request
  - Servers respond by sending response directly to requestor
  - Requestor keeps IP delivered by first responder
  - Requestor keeps sending requests until it gets an IP
- Clearly there is a need for redundant RARP servers for reliability
  - Timeouts can be used to activate backup RARP servers
    - Backup servers reply to a RARP request if they don't hear the RARP response from the primary server after some time

Alternatives to RARP

- RARP has shortcomings
  - Serious problem due to broadcasting at the data link layer-- RARP packet is encapsulated directly into a data link frame
  - RARP server required for each network or subnet.
  - Multiple RARP servers needed for reliability, but unlike ARP where only one reply is sent, each RARP server sends a unicast reply => additional traffic
  - Possibility of collision between RARP replies
- BOOTstrap Protocol (BOOTP) was developed as an alternative to RARP – moves process to network level
  - Uses UDP/IP packets to carry messages

BOOTP

- How can UDP running over IP be used by a computer to discover its IP address?
  - IP broadcast (to 255.255.255.255) even if local IP address unknown => client broadcasts BOOTP request
- A BOOTP server receives the broadcast
  - It looks up the sender's MAC address in a database txt file.
  - If there's a match, it replies with an IP broadcast.
- The client receives a datagram and checks the MAC address.
  - If it finds its own MAC address in the destination address field, then it takes the IP address in that datagram

Dynamic Configuration

- BOOTP was designed for relatively static environment where each host has a permanent network connection
  - Net manager creates a BOOTP config file with parameters for each host – file is typically stable for long periods
- Wireless networking enables environments much more dynamic
  - BOOTP does not provide for dynamic address assignment
- Dynamic configuration is the primary method for IP address allocation used today
  - Not only facilitates mobility but also efficient use of IPs

Roadmap

- IP Multicasting
  - IGMP
  - Multicast Routing
- DHCP vs ARP
- Obtaining an IP address
  - RARP
  - BOOTP
  - DHCP
- Traceroute

Traceroute output (from frapa.csd.uoc.gr to lg.nexlinx.net.pk)

```
traceroute to nasa.nexlinx.net.pk (202.59.80.52), 30 hops max, 40 byte packets
1  147.52.19.1 (147.52.19.1)  14.915 ms  0.503 ms  0.287 ms
2  147.52.1.11 (147.52.1.11)  0.467 ms  0.292 ms  0.283 ms
3  grnetRouter.uoc.heraklio-2.access-link.grnet.gr (195.251.25.201)  0.443 ms  0.428 ms  0.362 ms
4  Syros-to-Heraklio2.backbone.grnet.gr (195.251.27.81)  4.698 ms  4.573 ms  4.523 ms
5  athens3-to-Syros.backbone.grnet.gr (195.251.27.145)  6.899 ms  6.824 ms  6.873 ms
6  grnet.rt1.ath2.gr.geant2.net (62.40.124.89)  7.004 ms  6.989 ms  6.948 ms
7  so-1-1-0.rt1.sof.bg.geant2.net (62.40.112.197)  22.646 ms  22.533 ms  22.521 ms
8  so-2-3-0.rt1.bud.hu.geant2.net (62.40.112.202)  36.646 ms  44.107 ms  36.532 ms
9  bpt-b2-link.telia.net (80.239.134.1)  36.652 ms  36.633 ms  36.564 ms
10 bpt-b1-link.telia.net (213.248.96.97)  36.755 ms  36.838 ms  37.051 ms
11 ffm-bb1-link.telia.net (80.91.251.182)  56.206 ms  56.332 ms  56.170 ms
12 ffm-b7-link.telia.net (80.91.251.230)  56.331 ms ffm-b7-link.telia.net (80.91.249.105)  56.376 ms ffm-b7-link.telia.net (80.91.254.249)  56.320 ms
13 ge-6-1-3.BR1.FFT1.alter.net (146.188.112.41)  56.337 ms  56.216 ms  56.194 ms
14 so-1-0-0.XT1.PAR2.ALTER.NET (146.188.14.233)  64.347 ms  64.334 ms  64.282 ms
15 so-5-0-0.XR2.PAR2.ALTER.NET (146.188.10.46)  64.460 ms  64.372 ms  64.441 ms
16 POS1-0-0.GW3.PAR2.ALTER.NET (146.188.9.30)  64.281 ms  64.192 ms  64.170 ms
17 uuk203403.uk.customer.alter.net (158.43.65.34)  180.241 ms  180.140 ms  180.415 ms
18 tw112-static74.tw1.com (221.132.112.74)  199.218 ms  198.857 ms  199.077 ms
19 tw21-static22.tw1.com (117.20.21.22)  198.609 ms  198.712 ms  198.738 ms
20  * * *
21 nasa.nexlinx.net.pk (202.59.80.52)  201.968 ms  201.000 ms  201.272 ms
```

Traceroute output (from lg.nexlinx.net.pk to frapa.csd.uoc.gr

```
traceroute to frapa.csd.uoc.gr (147.52.19.28), 30 hops max, 40 byte packets
 1  10.10.12.2 (10.10.12.2)  0.883 ms   0.725 ms   0.920 ms
 2  10.10.80.4 (10.10.80.4)  2.451 ms   2.192 ms   3.695 ms
 3  tw21-static21.tw1.com (117.20.21.21)  3.671 ms   4.704 ms   5.573 ms
 4  tw112-static121.tw1.com (221.132.112.121)  21.722 ms   20.833 ms   20.065 ms
 5  pos0-3-1.gw3.par2.alter.net (158.43.65.33)  136.624 ms   136.649 ms   135.933 ms
 6  so-3-0-0.CR1.PAR2.ALTER.NET (146.188.9.25)  136.314 ms   135.814 ms   136.487 ms
 7  so-2-0-0.XT2.PAR2.ALTER.NET (146.188.10.49)  136.107 ms   136.057 ms   137.211 ms
 8  POS7-0.BR1.PAR2.ALTER.NET (146.188.8.122)  136.912 ms   137.767 ms   136.554 ms
 9  146.188.69.58 (146.188.69.58)  157.974 ms   158.727 ms   157.067 ms
10  te1-4-10G.ar2.VIE1.gblx.net (67.16.131.194)  174.053 ms   174.106 ms   174.102 ms
11  DANTE.tenGigabitEthernet1-3.ar2.VIE1.gblx.net (64.214.145.146)  165.364 ms   164.619 ms
164.473 ms
12  as1.rt1.ath2.gr.geant2.net (62.40.112.166)  193.932 ms   195.302 ms   195.382 ms
13  grnet-gw.rt1.ath2.gr.geant2.net (62.40.124.90)  194.105 ms   193.965 ms   195.853 ms
14  Syros-to-athens3.backbone.grnet.gr (195.251.27.146)  195.419 ms   196.324 ms   196.751 ms
15  Heraklio2-to-Syros.backbone.grnet.gr (195.251.27.82)  202.156 ms   202.710 ms   201.003 ms
16  clientRouter.uoc.heraklio-2.access-link.grnet.gr (195.251.25.202)  202.704 ms   200.450 ms
201.706 ms
17  olympos-e45.lanh.uoc.gr (147.52.1.9)  200.016 ms   200.631 ms   200.645 ms
18  frapa.csd.uoc.gr (147.52.19.28)  201.385 ms   202.026 ms   202.188 ms
```