Lecture 5 Overview

- Last Lecture
  - Name and address conversions
- This Lecture
  - IPv6
  - Broadcast and multicast sockets
  - Source: Chapters 12, 20, and 21
- Next Lecture
  - Introduction to wireless sensor networks

Anycast and multicast

- Anycast
  - A packet forwarded to an anycast address is delivered to only one interface of the set (the nearest to the source node, according to the routing metric).
  - Subnet prefix + 0000…000
  - Used for group of routers, not for hosts and as source addresses
- Applications
  - Domain Name System
  - Security
  - …

Multicast

- Multicast
  - 1111 1111 xxxT SSSS
  - The second byte consists of flag and scope.
  - Only one bit in the flag is used, other bits are for future use
- Flag bit—T
  - If T is 0, the multicast address is permanent (well-known addresses).
  - If T is 1, the address is temporary.
- Scope bits
  - 0000, reserved; 0001, node local; 0010, link local; 0101, site local; 1000, organization local; 1110, global; 1111, reserved
- Multicast addresses must not be used as source addresses

Multicast examples

- All node addresses
  - FF01:0000:0000:0000:0000:0000:0000:0001
  - FF02:0000:0000:0000:0000:0000:0000:0001
- All router address
  - FF01:0000:0000:0000:0000:0000:0000:0002
  - FF02:0000:0000:0000:0000:0000:0000:0002
  - FF05:0000:0000:0000:0000:0000:0000:0002
- Neighbor discovery
  - FF02:0000:0000:0000:0000:0001:FF00:0000 to FF02:0000:0000:0000:0000:0001:FFFF:FFFF

IPv6 addresses embedded with IPv4 addresses

- Each IPv4 address has corresponding IPv6 addresses
  - Belong to reserved (0x00)
  - 139.80.32.22
  - IPv4 compatible IPv6 address (for IPv6 tunneling in IPv4, deprecated)
    - 0000 0000 0000 0000 0000 0000 8B50 2016
    - ::139.80.32.22
  - IPv4 mapped IPv6 address (for IPv4 only nodes)
    - 0000 0000 0000 0000 0000 FFFF 8B50 2016
    - ::FFFF:139.80.32.22

Key Concept: *IPv4 address embedding* is used to create a relationship between an IPv4 address and an IPv6 address to aid in the transition from IPv4 to IPv6. One type, the *IPv4-compatible IPv6 address*, is used for devices that are compatible with both IPv4 and IPv6; it begins with 96 zero bits. The other, the *IPv4-mapped address*, is used for mapping IPv4 devices that are not compatible with IPv6 into the IPv6 address space; it begins with 80 zeroes followed by 16 ones.

Unspecified address

- ::
  - 0000 0000 0000 0000 0000 0000 0000 0000
  - It must never be assigned to any interface because it indicates the absence of an IPv6 address.
  - It can be used as a source address by a node during the configuration phase, when the node itself is trying to discover its IPv6 address.
  - It must never be used as the destination address or in the Routing header

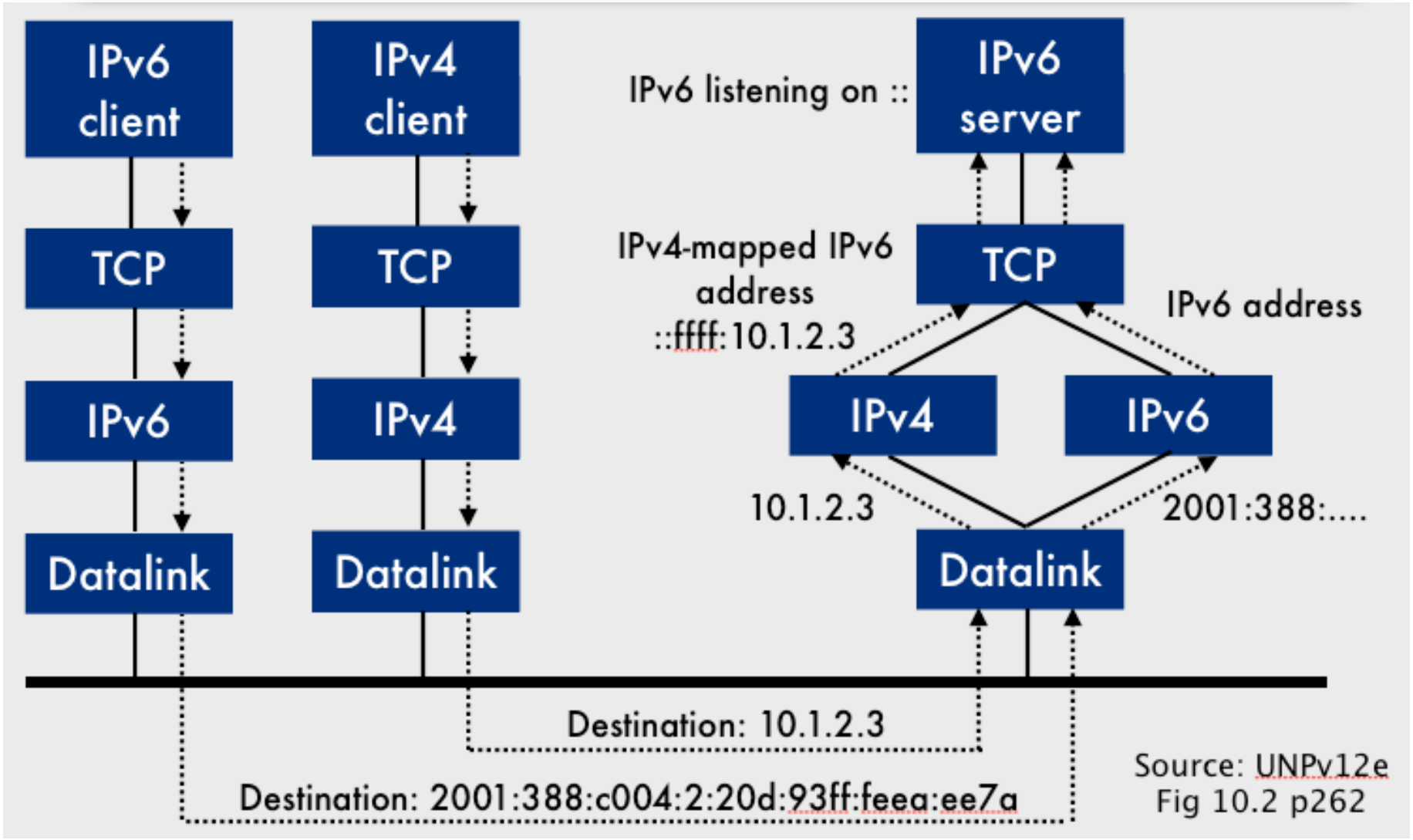
Differences between IPv6 & IPv4

- Interfaces have multiple IPv6 addresses
- Both handle addressing in much the same way. IPv6 has more special address ranges.
- IPv6 doesn't have broadcast. It does have multicast and anycast.
- IPv6 addresses are more auto-configuring

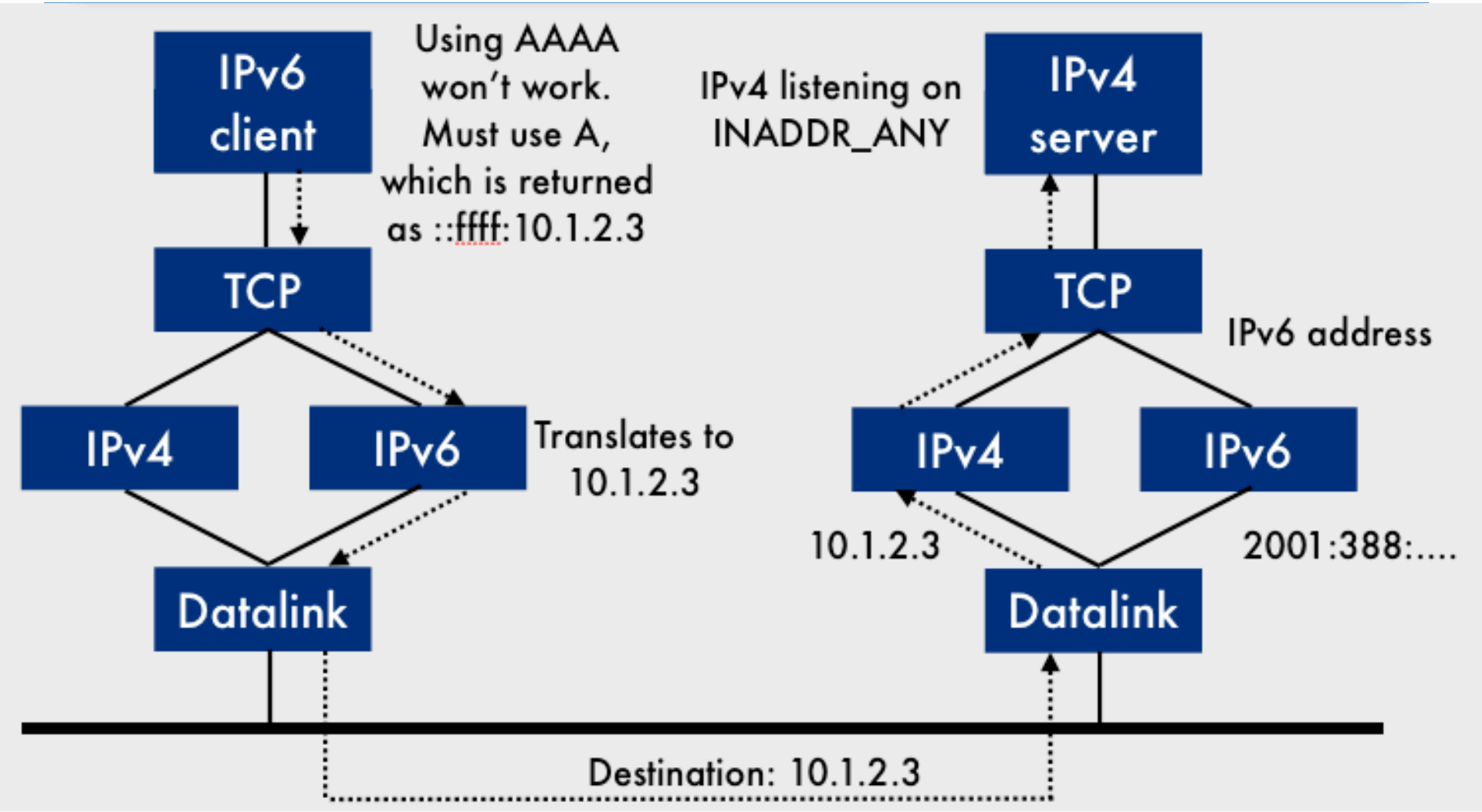
IPv6 Stateless Autoconfiguration

- Automatically configure their IP address and other parameters without the need for a server
- Steps:
  - **Link-Local Address Generation**
  - **Link-Local Address Uniqueness Test**
  - **Link-Local Address Assignment**
  - **Router Contact**
  - **Router Direction**
  - **Global Address Configuration**

IPv6 on dual-stack host



IPv4 on dual-stack



6to4 tunneling

- IPv6 islands connecting automatically
  - Needs two routers with public IPv4 addresses as tunnel end-points.
  - Hosts don't need IPv4 addresses, only relay-routers.
  - 6to4 does not facilitate interoperation between IPv4-only hosts and IPv6-only hosts.
  - protocol field of the IPv4 header is set to 41 (decimal) indicating an encapsulated IPv6 packet
- Assigned address range 2002::/16
  - The destination IPv4 address is in the IPv6 prefix (32 bits after 2002::/16). The 6to4 relay router will extract the IPv4 address to send the encapsulated packet to the other router.

Teredo tunneling

- The same purpose as 6to4 tunneling
- Tunnel IPv6 over UDPv4/3544 through Network Address Translations (NATs) employing Skype-like NAT tricks.
- Why UDP?
  - Because a lot of SoHO kit doesn't pass protocol 41 (6to4)
- RFC 4380

Address data structures

- struct sockaddr_in6 {
  uint8_t sin6_len;// size of the structure
  sa_family_t sin6_family;// AF_INET6
  in_port_t sin6_port;// Transport layer port
  uint32_t sin6_flowinfo;// IP6 flow information
  struct in6_addr sin6_addr;// IP6 address
  uint32_t sin6_scope_id;// scope zone index
  };
- struct sockaddr_storage {
  uint8_t ss_len;
  sa_family_t ss_family;
  uint8_t padding[128-2];
  };

Initializing _in6

- struct sockaddr_in6 addr6;
- memset(&addr6, 0, sizeof(addr6));
- addr6.sin6_family = AF_INET6;
- addr6.sin6_port = htons(8742);
- addr6.sin6_addr = in6addr_any; **OR**
- inet_pton (AF_INET6, addrstr, (void *) &addr6.sin6_addr);

Using sockaddr_

- struct sockaddr_storage addr;
- struct sockaddr_in *addr4 = (struct sockaddr_in *) &addr;
- struct sockaddr_in6 *addr6 = (struct sockaddr_in6 *) &addr;
- switch(addr.sa_family) { ... }

Guess what?

- Everything else is just the same, because the Sockets API uses the opaque sockaddr{} structure.
- In the lab, you will improve tcp_connect and describe_socket to work with both IPv4 and IPv6.

Protocol Independent Code

- Pass around struct sockaddr *,
  not struct sockaddr_in *
- Declare as struct sockaddr_storage or
  union sockaddr_multi... sockaddr_storage{}
  more portable.
- Isolate parts that deal with sockaddr_in etc.
- Use get{host,addr}info, not gethostby*
- More work if using multicast, raw sockets, IP options

Changes in application protocols

- Changes to application protocol if IP addresses are passed in the protocol.
- These affect the same protocols as are affected by things like NAT (FTP, VoIP, P2P).
- NAT should never happen for IPv6, but is used more and more as IPv4 addresses become scarcer.
  - Upgrade pressure for IPv6.
- NAT is only an incidental security device; what you really want is a firewall, not IPv6 NAT.

MAC Layer Broadcast

- Might not support broadcast
  - Non-Broadcast Multiple Access (NBMA) such as ATM, Frame Relay, and X.25
  - Simulates IP broadcasts
- Ethernet
  - Specify destination address ff:ff:ff:ff:ff:ff (all 1s) for broadcast

IPv4 broadcast

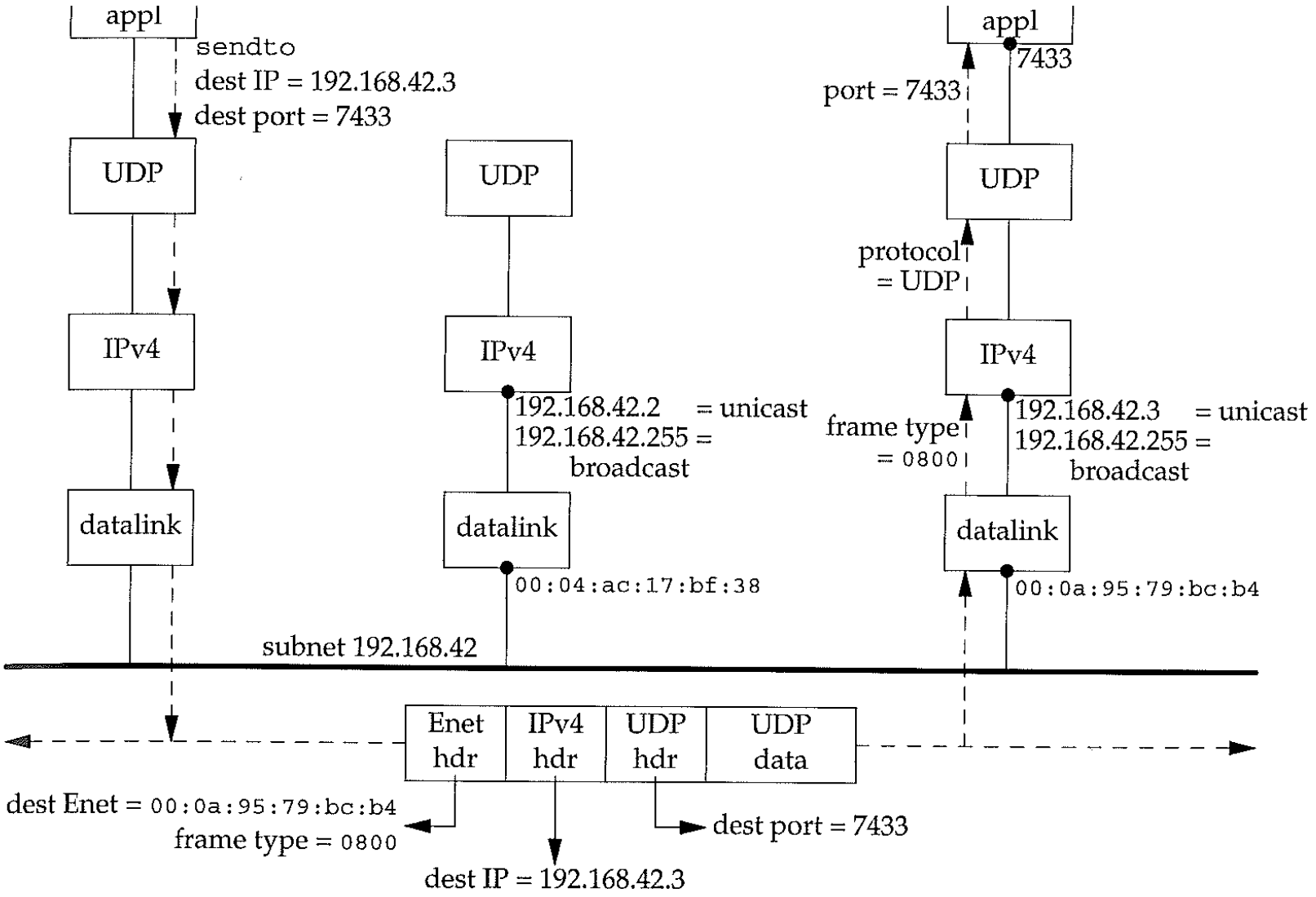
- IPv4addr = {netid, subnetid, hostid}
  - Some parts can be -1 (wraps around to maximum value), which means “all ones”
- Two common broadcast address types
  - Subnet-directed broadcast
    - {netid,subnetid,-1}, last address in subnet (not always .255), eg. 10.18.2.31 for 10.18.2.16/28
    - Commonly not forwarded by routers
  - Limited broadcast:
    - {-1,-1,-1} = 255.255.255.255
    - Use when network details unknown
    - bootstrapping — DHCP, ARP etc.
    - Must not be forwarded

IPv4 broadcast

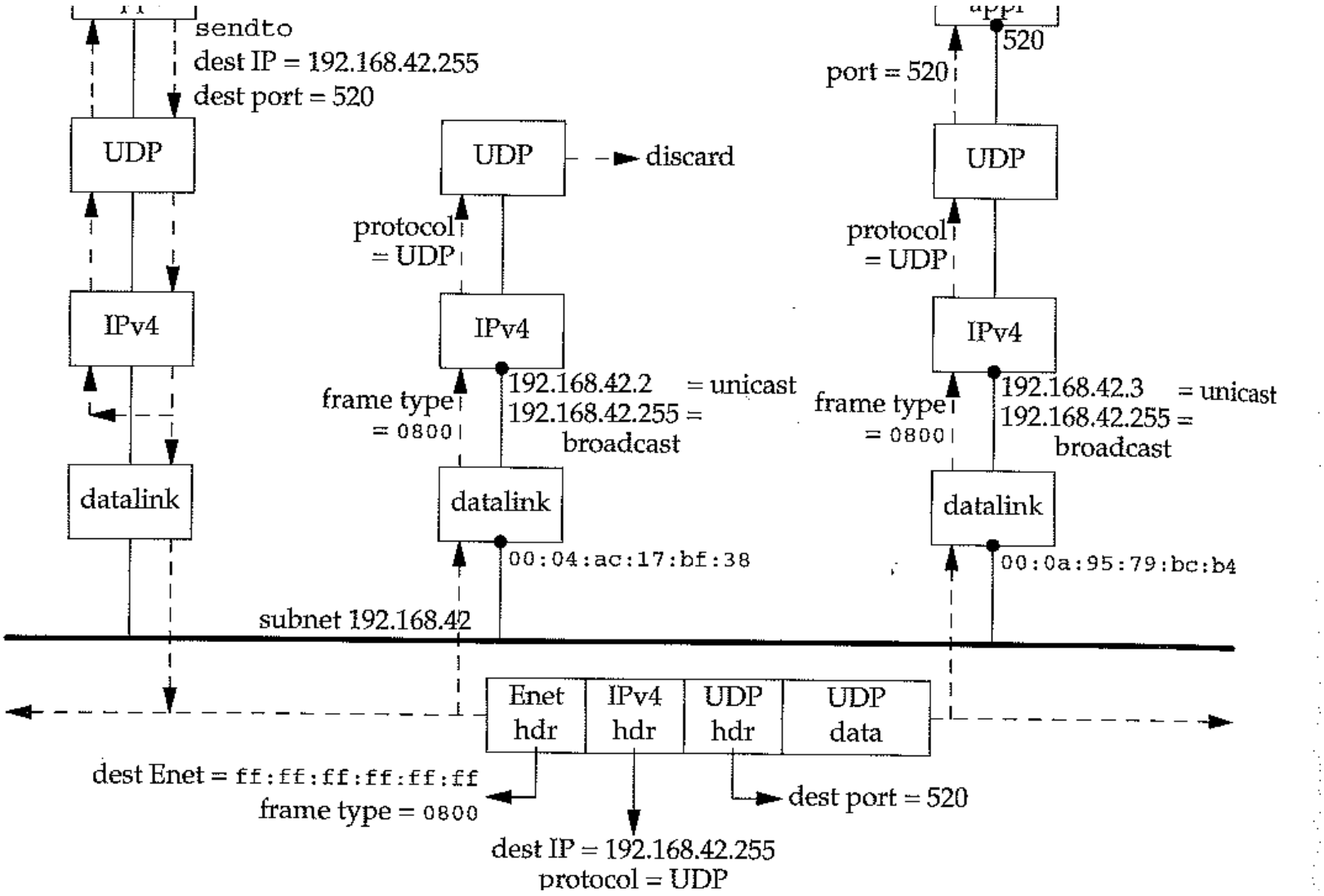
- How does a host do when an application sends a UDP datagram to {-1,-1,-1}?

- What does a multihomed host do when an application sends a UDP datagram to {-1,-1,-1}?

Unicast example



Broadcast example



Applications

- Network Management
  - ARP, DHCP
- Service Advertisement / Discovery
  - LAN gaming server discovery
- Reducing network traffic
  - Network Time Protocol (NTP)
  - single broadcast vs. many unicasts

Broadcast problems

- Overhead for disinterested hosts
  - embedded systems in large networks
- Broadcast storms
  - bad to acknowledge broadcasts
- Inflexible scope
  - should use multicast instead
- 255.255.255.255 Routing
  - Determine outgoing interface behavior varies depending on OS
- No IP fragmentation for broadcast

Broadcast client

- Assure the kernel that you mean to use broadcast (sanity check)
  - setsockopt(sock, SOL_SOCKET, SO_BROADCAST, &one, onelen)
  - EPERM (Permission Denied) if you don't.
- Use sendto to send datagrams
- Use recvfrom in a loop
  - Number of responses unknown
  - Implement a time-out using select
  - Check for loopback if bound to destination port

Broadcast server

- Get interface list to bind to individual interfaces or bind to any (0.0.0.0)
- Create a UDP socket
  - Set SO_BROADCAST before bind
- Call recvfrom in a loop
- Send response back using sendto

Multicast

- One (or more) senders
- Receivers (>0) subscribe to a set of multicast addresses.
- Broadcasting is normally limited to a LAN, whereas multicasting can be used on a LAN or across a WAN.
- Optional in IPv4, mandatory in IPv6
- Multicast replaces broadcast in IPv6

Typical applications

- Now (all in a LAN or private WAN)
  - Network Management
  - Video Conferencing/VoIP
  - Workstation Imaging
- Later
  - IP TV - subscriber based
  - Large scale push services like messaging, email, video-conferencing

IPv4 multicast addresses

- Class D: 224.0.0.0/4 (1110...)
  - group ID - the low-order 28 bits
- 224.0.0.0 – 239.255.255.255
  - 224.0.0.0/24 is link local
  - 239.0.0.0/8 is administratively scoped
    - 239.255.0.0 – 239.255.255.255 is site local
    - 239.192.0.0 – 239.195.255.255 is organization local
  - 224.0.1.0 – 238.255.255.255 is global
- Common addresses
  - 224.0.0.1 = all-systems.mcast.net
  - 224.0.0.2 = all-routers.mcast.net

IPv4 multicast addresses (cont.)

- 224.0.0.0/24 for link local
  - Network management…
- Multicast Addresses (assigned by IANA)
  - Identifies a service (eg. ntp.mcast.net)
- Network administrators control LAN assignments
- For public/general use you should apply for an address (free).

IPv6 multicast addresses

- Multicast
  - 1111 1111 xxxT SSSS
  - The second byte consists of flag and scope.
  - Only one bit in the flag is used, other bits are for future use
- Flag bit—T
  - If T is 0, the multicast address is permanent (well-known addresses).
  - If T is 1, the address is temporary.
- Scope bits
  - 0000, reserved; 0001, node local; 0010, link local; 0101, site local; 1000, organization local; 1110, global; 1111, reserved

IPv6 multicast examples

- All node addresses
  - FF01:0000:0000:0000:0000:0000:0000:0001
  - FF02:0000:0000:0000:0000:0000:0000:0001
- All router address
  - FF01:0000:0000:0000:0000:0000:0000:0002
  - FF02:0000:0000:0000:0000:0000:0000:0002
  - FF05:0000:0000:0000:0000:0000:0000:0002
- Neighbor discovery
  - FF02:0000:0000:0000:0000:0001:FF00:0000 to FF02:0000:0000:0000:0000:0001:FFFF:FFFF
- Multicast addresses must not be used as source addresses

Ethernet mapping

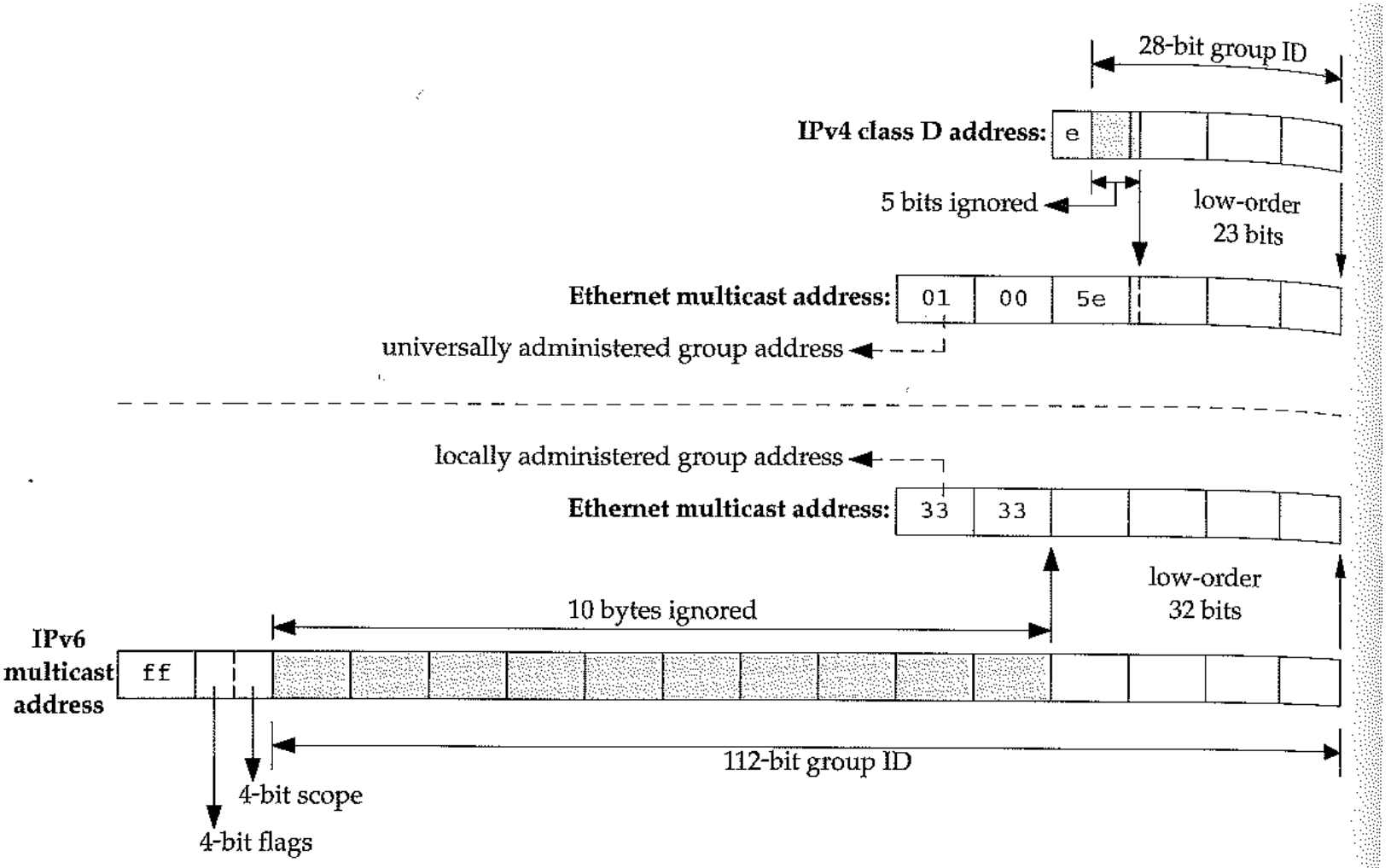
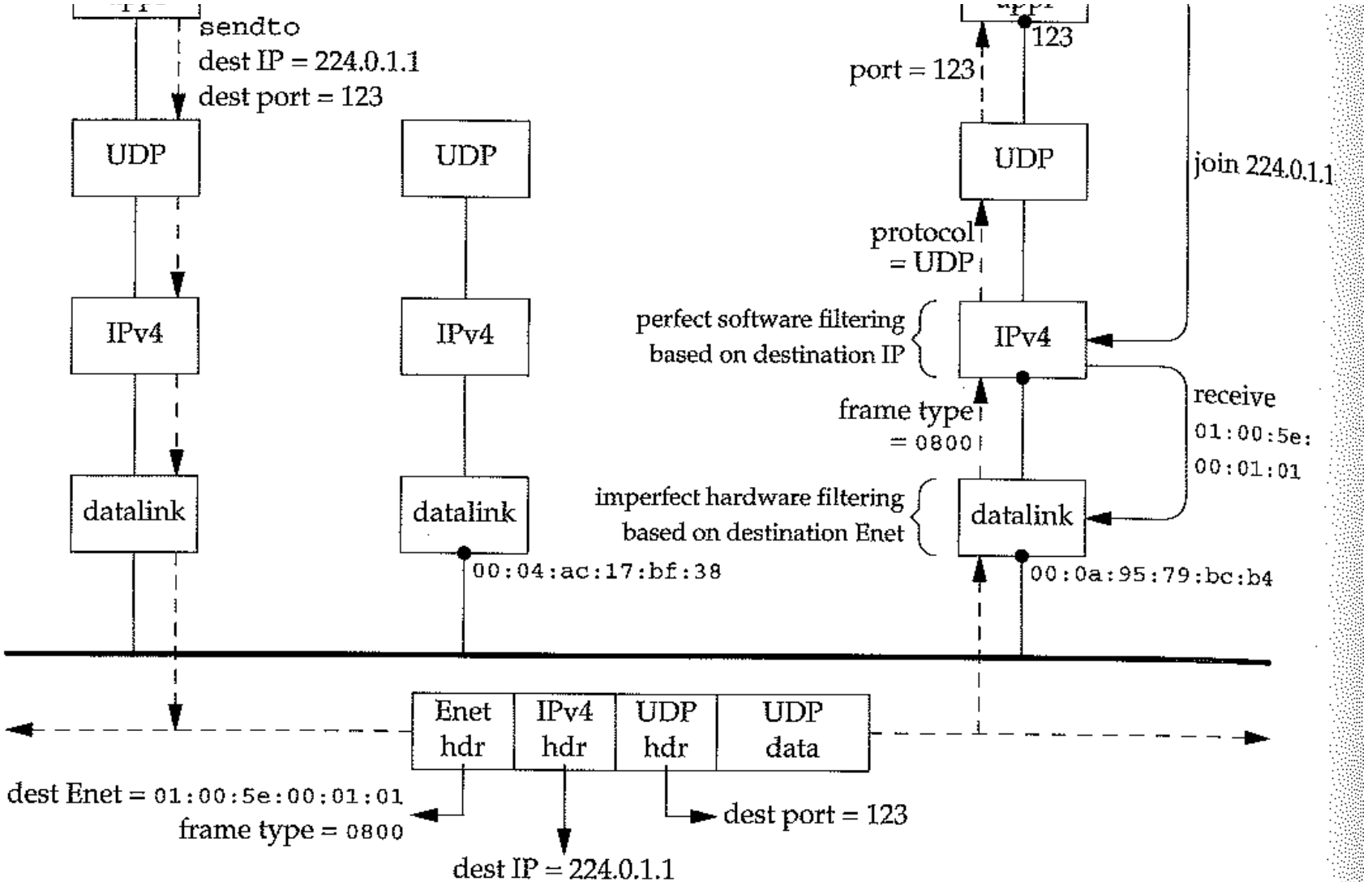


Figure 21.1 Mapping of IPv4 and IPv6 multicast address to Ethernet addresses.

IPv4 multicast example



TTL scoping

- A multicast address is unique within a particular scope
- Overload TTL field in IPv4 header
- Routers decrement and check TTL.
- IPv4 TTL Scope values
  - 0 = Local, 1 = Link, <32 = Site, <64 = Region, <128 = Continent, <255 = Global
- TTL scoping inflexible
  - esp. when networks aren't concentric.
  - Better to define the boundary of a network explicitly…

Administratively scoped mcast

- Shape scope to fit region defined by local administrator
  - Applications don't need to know TTL
- Simply configure routers to act as a boundary for particular mcast addresses
- Examples
  - 239.255.0.0 – 239.255.255.255 is site local
  - 239.192.0.0 – 239.195.255.255 is organization local
  - Addresses in the specified ranges are assigned locally, but are not guaranteed to be unique globally.

Multicast on a WAN

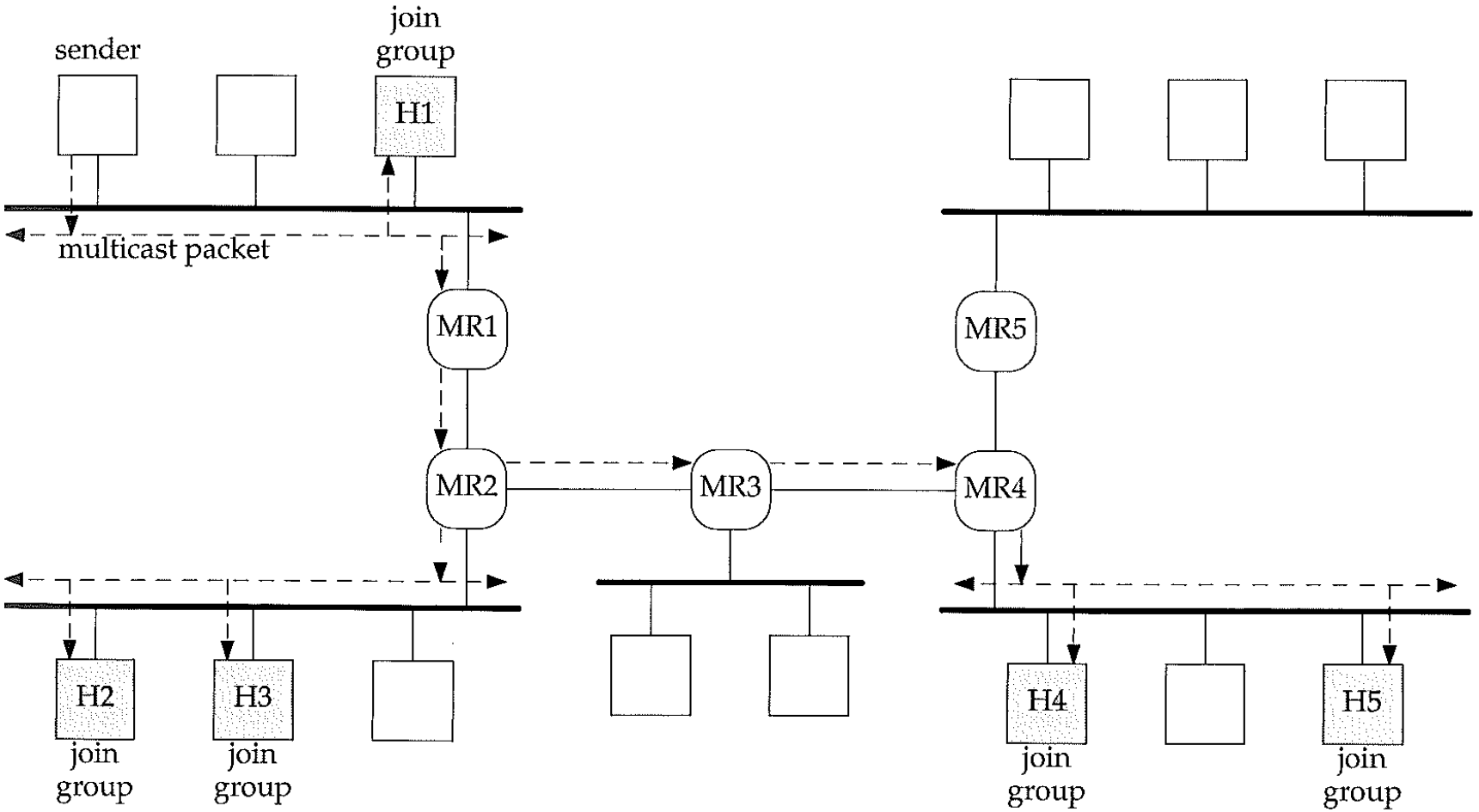


Figure 21.7 Sending multicast packets on a WAN.

IGMP

- Internet Group Management Protocol
  - Host to router protocol for joining and leaving multicast groups
  - Current = v2, v3 coming, v1 still around
- Programmer requests the kernel to join or leave a group.
  - Property of kernel, not socket.

MRP

- Challenges for multicast routing protocol
  - Get data from all the senders located anywhere to receivers located anywhere
  - Not enough IPv4 multicast addresses to statically assign to everyone
- Source-specific multicast (SSM)
  - Combines the multicast address with the sender's address
  - The receivers supply the sender's address to the routers when joining the group.

Multicast socket options

- Options for multicast
  - IP_MULTICAST_LOOP, IPV6_MUlTICAST_LOOP
    - Enable or disable local loopback of multicast datagrams
    - Loopback is enabled by default
  - IP_MULTICAST_TTL, IPV6_MULTICAST_HOPS
    - Set the IPv4 TTL or the IPv6 hop limit for outgoing multicast datagrams
    - Default is 1
  - IP_MULTICAST_IF, IPV6_MULTICAST_IF
    - Specify the interface for outgoing multicast datagrams sent on this socket
    - Specified as either an in_addr structure for IPv4 or an interface index of 0 for IPv6

Multicast socket options (contd)

- IP_ADD_MEMBERSHIP, IPV6_JOIN_GROUP, MCAST_JOIN_GROUP
  - Join a multicast group on a specified local interface.
  - MCAST_JOIN_GROUP is protocol-independent
- IP_DROP_MEMBERSHIP, IPV6_LEAVE_GROUP, MCAST_LEAVE_GROUP
  - Leave a multicast group on a specified local interface.

Multicast socket options (contd)

- IP_BLOCK_SOURCE
  - Block receipt of traffic on this socket from a source given an existing any-source group membership on a specified local interface.
- IP_UNBLOCK_SOURCE
  - Unblock a previously blocked source
- IP_ADD_SOURCE_MEMBERSHIP, MCAST_JOIN_SOURCE_GROUP
  - Join a source-specific group on a specified local interface
- IP_DROP_SOURCE_MEMBERSHIP, MCAST_LEAVE_SOURCE_GROUP
  - Leave a source-specific group on a specified local interface

- These all use IPPROTO_IP for the level.

Sender's code

- For every multicast interface
  - Create a UDP socket
  - array of {socket, iface_addr}, one per iface
  - Enable SO_REUSEADDR
  - Bind socket to interface's unicast address
  - Joining is not necessary for sending
  - Set TTL scope (even for Admin Scoped)
- Write the data to each socket.

Receiver's code

- Create a UDP socket
  - Enable SO_REUSEADDR on that socket.
  - Bind socket to the multicast address.
- For every multicast interface, set the following option to the socket
  - IP_ADD_MEMBERSHIP
- Receive data, removing duplicates
- Replies generally go via unicast socket

References (1)

- IPv6 address space
  - http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml
- IPv6—the new protocol for Internet and Intranets, Chap. 4
  - http://www.ip6.com/us/book/Chap4.pdf
- The TCP/IP Guide — IPv6 Addressing
  - http://www.tcpipguide.com/free/t_IPv6Addressing.htm
- IPv4-to-IPv6 Transition and Co-Existence Strategies
  - http://ipv6.bt.com/Downloads/bt_wp_IPv6_Transition_Strategies_2011.pdf
- RFC 3056
  - 6to4 tunneling; STD 0066 – URI Generic Syntax
- IPv6 deployment in New Zealand
  - http://en.wikipedia.org/wiki/IPv6_deployment#New_Zealand

References (2)

- Multicast Addresses
  - http://www.iana.org/assignments/multicast-addresses
- RFC Site
  - http://www.rfc-editor.org
- RFC 2365
  - Administratively Scoped IP Multicast