IP control protocols

1. Explain how a forwarding table for multicast would differ from a forwarding table for unicast.

- a forwarding table for unicast would have a series of networks and subnets, not particular to a host, but particular to a network i.e. you would have a 2-tuple between hosts
- a forwarding table for multicast would have an n-tuple long list of specific hosts associated with a multicast group

2. Finding an optimal multicast routing tree is NP-hard. (a) Does that mean that multicast requires a lot of processing per packet? Why or why not?

- multicast tree is just ARP-table. No notion of routing at link layer, just routing between adjacent hosts (b) Does that mean that computing a multicast tree requires a lot of processing? Why or why not?
- no: every adjacent node maintain its own ARP table

3. Why is an ARP query sent within a broadcast frame? Why is an ARP response sent within a frame with a specific destination MAC address?

- An ARP query is sent within a broadcast frame because you don't know in advance the MAC address of the IP address you are looking for, so you need to send it to everyone
- the response is sent with a specific destination MAC address is used because this was the host that requested the ARP query - prevent unnecessary traffic on network

4. Suppose you walk into a room, connect to Ethernet (a layer-2 network), and want to download a Web page. What are all the protocol steps that take place, starting from powering on your PC to getting the Web page? Assume there is nothing in your DNS or browser caches when you power on your PC. (Hint: the steps include the use of DHCP, ARP, DNS, TCP, and HTTP protocols.) Explicitly indicate in your steps how you obtain the IP and MAC addresses of a gateway router.

- Connecting to network
- Need an IP address: make a DHCP request
  - encapsulate in UDP segment with DHCP destination port/source port
  - place in IP datagram with destination address 255.255.255.255, source address 0.0.0.0
  - place IP datagram in Ethernet frame with destination MAC address FF:FF:FF:FF:FF:FF for broadcast, and source MAC of the network adapter on the computer
  - send Ethernet frame to the switch: switch broadcasts frame on all outgoing ports
  - router receives broadcast Ethernet frame containing DHCP request on its adapter with a MAC address, then extracts IP datagram. Broadcast IP address means upper layer should be handled here, so UDP segment is demultiplexed to UDP

  - DHCP request is extracted from UDP segment, and read by the DHCP server
  - DHCP server allocates an available IP address, and creates a DHCP ACK including: allocated IP address, DNS server IP address, default gateway router IP address, subnet mask
  - DHCP request placed inside UDP segment, IP datagram, Ethernet frame with source MAC address of the router interface to the LAN and destination MAC address of the PC interface
  - Ethernet frame is sent by router to the switch, which now knows where to forward the frame addressed to the PC
  - PC receives the frame, unpacks IP datagram, extracts UDP segment, extracts DHCP ACK and passes to DHCP client. This reads the IP addresses provided, storing the IP address of default gateway in its IP forwarding table. All datagrams with a destination outside the network subnet get sent to the default gateway

- requesting web page
- browser receives request for webpage
- TCP socket created
- DNS request for hostname of resource:
  - compose DNS query message for hostname
  - encapsulate in UDP segment (destination port 53)
  - encapsulate in IP packet with source address as DNS server provided by DHCP
  - as DNS server is outside network, forward to default gateway
  - encapsulate in Ethernet frame: need MAC address of default gateway
  - ARP query: target IP of default gateway sent encapsulated in Ethernet frame with broadcast address FF:FF:FF:FF:FF:FF, sent to switch
  - ARP reply sent by default gateway indicating MAC address, again encapsulated in Ethernet frame
  - send Ethernet frame to switch, which forwards to default gateway
  - default gateway extracts IP datagram, examines destination IP address and looks it up in forwardng table
  - IP datagram then placed in link-layer frame and it is passed on to DNS server
  - DNS server unpacks DNS message and forms a reply (which may be in cache, or require further DNS queries)
  - This is sent back to default gateway and passed to computer
  - TCP connection requested to destination IP address port 80
  - three-way handshake
- HTTP get request made
- encapsulated in TCP segment
- encapsulated in IP datagram
- encapsulated in Ethernet frame

  - ARP table lookup for default gateway

- forwarded to default gateway
- routed to destination (BGP)

5. You have just explained the ARP protocol to a friend. When you are all done, he says: "I've got it. ARP provides a service to the network layer, so it is part of the data link layer." What do you say to him?

- ARP doesn't fit neatly into the layering model: broadcasts are sent using the link layer directly, and are not IP packets, but it does handle network layer addresses as well as link layer addresses
- on boundary between link-layer protocol and network layer protocol

6. When the IPv6 protocol is introduced, does the ARP protocol have to be changed? If so, are the changes conceptual or technical?

- IPv6 may technically allow for each device to have an globally unique IP address, thus making the MAC address redundant and arguably less useful (due to the flat structure of MAC addresses) however backward support with IPv4 would make it basically impossible to eliminate, and devices having globally unique MAC addresses also means they aren't tied to upper layer protocols.
- from a technical standpoint there still needs to be a way to map between MAC addresses and IP addresses. There is need for technical changes in terms of ensuring it can handle the extra bits associated with IP addresses

7. How is ICMP used? Give examples from the lecture slides, plus any you find on the web.

- Internet Control Message Protocol: used by hosts/routers to communicate network-layer info between each other. Typical use: error reporting e.g. "Destination network unreachable". ICMP messages are carried as IP payload, so it sits just about IP layer
- `ping`: ICMP echo
- `tracert`: uses a series of `ping`s with `TTL` increasing from 1 to determine all hops along the route

Process management and multithreading

8. What is the difference between kernel and user mode? Explain how having two distinct modes aids in designing an operating system.

- the operating system runs in kernel mode, giving it full, privileged access to hardware and the ability to execute any instruction the machine is capable of executing

- user mode runs the rest of software and imposes restrictions upon what can be executed
- mode is controlled by a bit in the Program Status Word register
- this distinction allows you to maintain a clean interface of abstract resources rather than an application needing to deal with hardware directly
- this helps to impose security, reliability, and fairness on processes

9. What is a purpose of a system call in an operating system?

- a system call is used by a user program to obtain services from the operating system, which then moves to kernel mode to execute a privileged operation. Once complete, control returns to user program

10. To a programmer, a system call looks like any other call to a library procedure. Is it important that a programmer knows which library procedures result in system calls? Under what circumstances and why?

- yes: when performance is critical, as switching between modes take time
- yes: when security is important, as switching to kernel mode introduces potential vulnerabilities that could affect more than the isolated program

11. What is the biggest advantage of implementing threads in user space? What is the biggest disadvantage?

- implementing threads in user space compared to kernel space allows you to have potential performance gains because you don't have to make system calls to switch threads. However, if they are implemented in user space and one thread makes a system call, all threads will be blocked. If threads are implemented in the kernel, and system calls are made by one thread, the kernel is aware of other threads and can keep them running.

12. In the lectures, a multi-threaded text editor was shown. If the only way to read from a file is the normal blocking read system call, do you think user-level threads or kernel-level threads are being used for the text editor? Why?

- kernel level threads, as otherwise the entire text editor would be blocked when a file read system call was made

13. Consider designing a server for a Website where requests for pages come in and the requested page is sent back to the client. At most Websites, some pages are more commonly accessed than other pages. For example, a home page is accessed far more than a page deeper in the tree structure of the Website. Web servers use this fact to improve performance by maintaining a collection of heavily requested pages in main memory to eliminate the need to go to disk to get them. Such a collection is called a cache and provides much faster access to data than disk (we

will learn more about caches in the upcoming lectures). One possibility is to have the server operate as a single thread: The main loop of the Web server gets a request, examines it, and carries it out to completion before getting the next one. While waiting for the disk, the server is idle and does not process any other incoming requests. What problems does this solution have? Design a multi-threaded server to improve over a single-threaded design.

- problems: long delay between anyone sending a request and getting a response, you may also overload the server if many requests come in while the first request is being served
- multi-threaded solution: for each request, spin up a thread managed in kernel space to go and serve the request. This way, new requests can continue to be handled while other requests are being served