Networking Endterm

Below 6 netmasks are shown in both binary and dotted-quad form.

Binary	Dotted-quad
11111111 11111111 11111111 10000000	255.255.255.128
11111111 11111111 11111111 11000000	255.255.255.192
11111111 11111111 11111111 11100000	255.255.255.224
11111111 11111111 11111111 11110000	255.255.255.240
11111111 11111111 11111111 11111000	255.255.255.248
11111111 11111111 11111111 11111100	255.255.255.252

1) The netmask that defines a network size of 32 is:
   a) 255.255.255.192
   b) 255.255.255.224
   c) 255.255.255.240
   d) 255.255.255.248

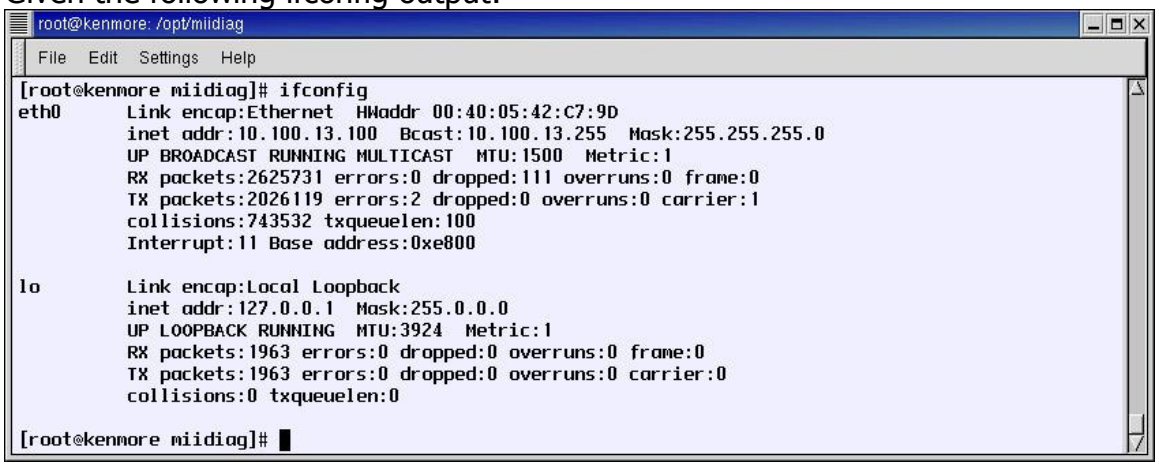
2) The netmask 255.255.255.128 defines a network size of:
   a) 32
   b) 64
   c) 96
   d) 128

3) A PC on a network has an IP address of 10.10.10.35 with a class C subnet mask (255.255.255.0). It's default gateway is at 10.10.10.1. It has just powered up and its ARP cache is empty.
   The user pings another PC at 207.188.86.185. What is the first packet that will be sent from his PC (10.10.10.35)?
   a) An ICMP request to 10.10.10.109
   b) An ICMP reply to 10.10.10.1
   c) An ARP request to 10.10.10.109
   d) An ARP request to 10.10.10.1

4) A PC on a network has an IP address of 10.10.10.35 with a class C subnet mask (255.255.255.0). Its default gateway is at 10.10.10.1. It has just powered up and its ARP cache is empty.
The user pings another PC at 10.10.10.109. What is the first packet that will be sent from his PC (10.10.10.35)?

a) An ICMP request to 10.10.10.109
b) An ICMP reply to 10.10.10.1
c) An ARP request to 10.10.10.109
d) An ARP request to 10.10.10.1

Given the following ifconfig output:

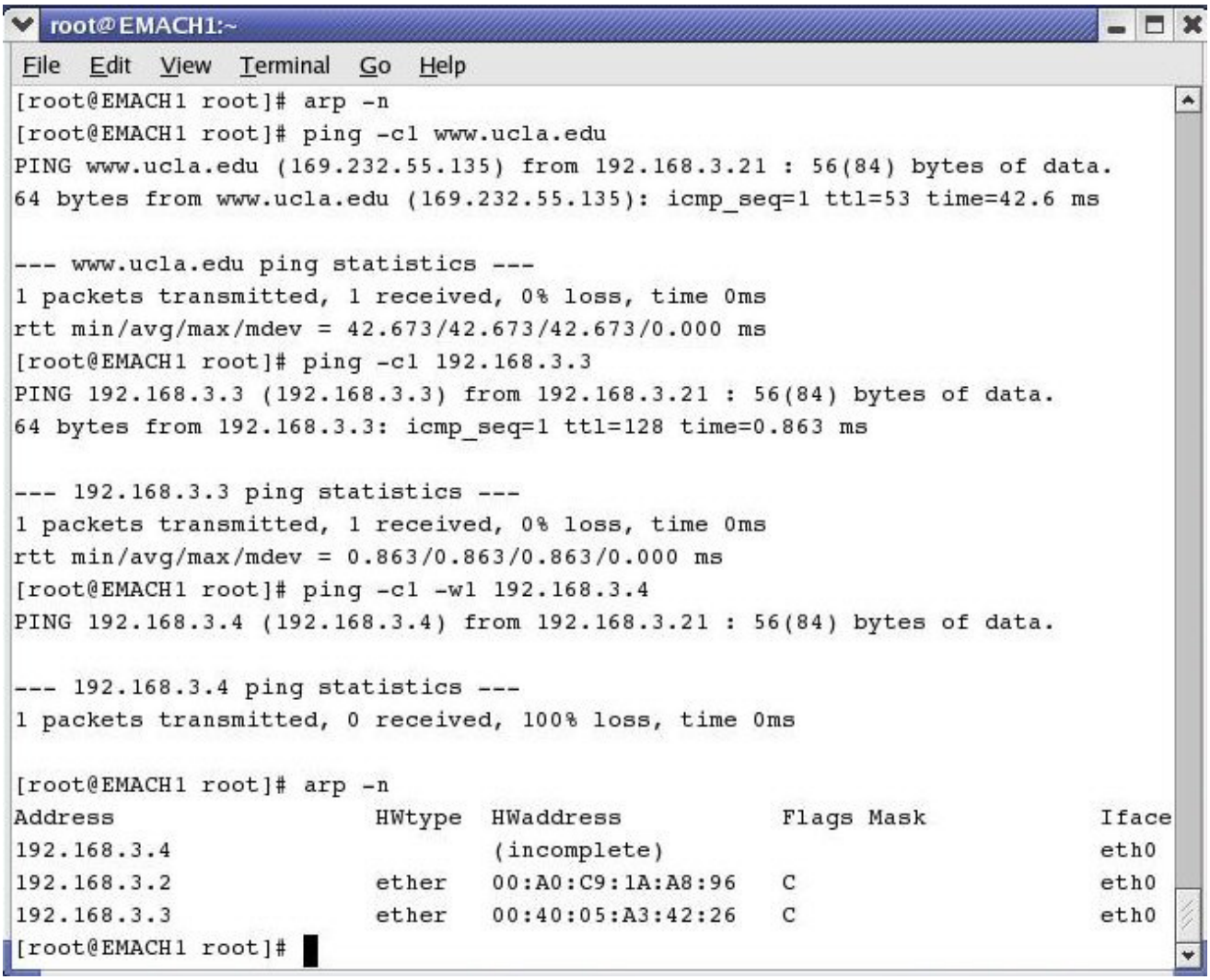


5) The IP address of interface eth0 is:

a) 00:40:05:42:C7:9D
b) 0xe800
c) 10.100.13.255
d) none of the above

6) The MAC address of interface eth0 is:

a) 00:40:05:42:C7:9D
b) 0xe800
c) 10.100.13.255
d) none of the above



Note the sequence of commands shown in the screen image above, and their effect on the contents of the arp table. The LAN is 192.168.3.0/255.255.255.0.

7) Unlike the other 2 IP addresses shown, 192.168.3.4's hardware address shows as "incomplete." Choose the best reason for this.

a) At 192.168.3.4, ping is responding but arp is not
b) At 192.168.3.4, arp is responding but ping is not
c) At 192.168.3.4, arp is not responding
d) At 192.168.3.4, ping is not responding

8) During the activities shown on the screen the arp table acquires an entry for IP address 192.168.3.2. The command responsible for it is:

a) arp –n
b) ping –c1 www.ucla.edu
c) ping –c1 192.168.3.3
d) ping –c1 –w1 192.168.3.4

9) Ethernet is a data link layer protocol, while IP is a network layer protocol.

a) IP is the only network protocol Ethernet can carry, and Ethernet is the only data link protocol that can carry IP
b) Ethernet can carry other network protocols, but is the only data link protocol that can carry IP
c) IP is the only network protocol Ethernet can carry, but IP can also be carried by other data link protocols
d) Ethernet can carry other network protocols, and other data link protocols can carry IP

10) The IP packet structure provides a time-to-live field. A host creates and sends a packet to a destination, via a series of intervening routers.

a) IP in the host initializes TTL to 255
b) ICMP in each router decrements TTL
c) TTL should reach 0 by the time of arrival at the destination
d) IP in each router decrements TTL

11) A host participates in a tcp connection with a partner. The values it places in the "acknowledgement number" fields of the packets it send to the partner:

a) come from the partner's sequence number stream
b) come from the host's own sequence number stream
c) are a function of both sequence number streams
d) have to do with neither; acknowledgment and sequence numbers are independent of each other

12) Given an Ethernet frame carrying an IP packet bearing a host destination address, a switch

a) sends it out all ports
b) sends it out a particular port according to the destination IP address
c) sends it out a particular port according to the destination Ethernet address
d) sends it out a subset of ports according to the IP address's subnet

13) The DNS system amounts to:

a) a single physical database composed of multiple logical databases
b) a single logical database composed of a single physical database
c) multiple logical databases composed of a single physical database
d) a single logical database composed of multiple physical databases

14) When an application acts as a DNS client, querying a nameserver for the address of training.support.microsoft.com, the nameserver that it queries is:

a) one that is named in the local DNS configuration
b) a root nameserver
c) a Microsoft nameserver
d) a com nameserver

15) When an application acts as a DNS client, querying some nameserver directly for an address, the nameserver that answers the query (originating that answer, not merely relaying it) is:

a) always the one that the client queries directly
b) always a root nameserver
c) sometimes the one that the client queries directly
d) sometimes the application itself

16) TCP client and corresponding server processes, running on different machines or the same, interact. When they do so, they distinguish and identify one another as specific processes by:

a) name, from the command line that launched them
b) hardware address
c) IP address
d) phone number
e) port number

17) Which of the following is a client for the http protocol:

a) named
b) httpd
c) Internet Explorer
d) Apache

18) The difference between a hub and a switch is that:

a) switches forward fewer frames overall
b) hubs consume more of the wire's inherent bandwidth
c) switches are more network-aware
d) all of the above

19) in linux, addresses of the nameservers to be queried are kept:

a) in /etc/hosts
b) in /etc/named.conf
c) in /etc/resolv.conf
d) in separate files per each under /var/named/

20) A dhcp client:

a) must accept the first IP address it is offered
b) must accept all IP addresses it is offered
c) may accept the first IP address it is offered
d) none of the above

21) Which of the following is an HTTP request:

a) CGI
b) <html>
c) GET
d) http://www.yahoo.com

22) A web server can:

a) provide return pages to browsers without using any .html files
b) return different content to different requesting clients
c) listen to ports other than 80
d) all of a, b, and c
e) none of a, b, and c

23) Host 192.168.3.1 and 192.168.3.2 can ping each other. Server3.c is executed on 192.168.3.2 and client3.c is executed on 192.168.3.1. The source code for both appears below, at bottom. They are the ones we studied in class. Study them carefully. The expected screen message on client 192.168.3.1 is:

a) char from server = B
b) char from server = C
c) oops: client1: no route to host
d) oops: client1: connection refused

CLIENT3 CODE (for question 23)**:**

```
int main()
{
        int sockfd;
        int len;
        struct sockaddr_in address;
        int result;
        char ch = 'B';
        sockfd = socket(AF_INET, SOCK_STREAM, 0);
        address.sin_family=AF_INET;
        address.sin_addr.s_addr = inet_addr("192.168.3.2");
        address.sin_port = htons(1101);
        len = sizeof(address);
        result = connect(sockfd, (struct sockaddr *)&address, len);
        if (result == -1) {
                perror("oops: client1");
                exit(1);
        }
        write(sockfd, &ch, 1);
        read(sockfd, &ch, 1);
        printf("char from server = %c\n", ch);
        close(sockfd);
        exit(0);
}
```

SERVER3 CODE:

```
int main()
{
        int server_sockfd, client_sockfd;
        int server_len, client_len;
        struct sockaddr_in server_address;
        struct sockaddr_in client_address;
        server_sockfd = socket(AF_INET, SOCK_STREAM, 0);
        server_address.sin_family = AF_INET;
        server_address.sin_addr.s_addr = htonl(INADDR_ANY);
        server_address.sin_port = htons(1100);
        server_len = sizeof(server_address);
        bind(server_sockfd, (struct sockaddr *)&server_address, server_len);
        listen(server_sockfd, 5);
        while(1) {
                char ch;
                printf("server waiting\n");
                client_sockfd = accept(server_sockfd, (struct sockaddr
*)&client_address, &client_len);
```

```
        read(client_sockfd, &ch, 1);
        ch++;
        write(client_sockfd, &ch, 1);
        close(client_sockfd);

        }
}
```