Broc Nickodemus

Final

1) Complete the [Wireshark.BitTorrent.Labreview the documentiew in a new window](https://oregonstate.instructure.com/courses/1602417/files/65221774/download?wrap=1)

See attachment

2) Complete the following code, by adding the correct lines in place of the bold, red text, to finish the UDP Pinger assignment on pages 179-180 of the book:

import sys, time

from socket import \*

# Get the server hostname and port as command line arguments

# Set up command line arguments

argv = sys.argv

# Assign host

host = argv[1]

# Assign Port

port = argv[2]

timeout = 1 # in seconds

# Create UDP client socket

# Note the use of SOCK\_DGRAM for UDP datagram packet

# Set up client socket

clientsocket = socket(AF\_INET, SOCK\_DGRAM)

# Set socket timeout as 1 second

clientsocket.settimeout(timeout)

# Command line argument is a string, change the port into integer

port = int(port)

# Sequence number of the ping message

ptime = 0

# Ping for 10 times

while ptime < 10:

ptime += 1

# Format the message to be sent

data = "Ping " + str(ptime) + " " + time.asctime()

try:

# Sent time

RTTb = time.time()

# Send the UDP packet with the ping message

clientsocket.sendto(data,(host, port))

# Receive the server response

message, address = clientsocket.recvfrom(1024)

# Received time

RTTa = time.time()

# Display the server response as an output

print "Reply from " + address[0] + ": " + message

# Round trip time is the difference between sent and received time

print "RTT: " + str(RTTa - RTTb)

except:

# Server does not response

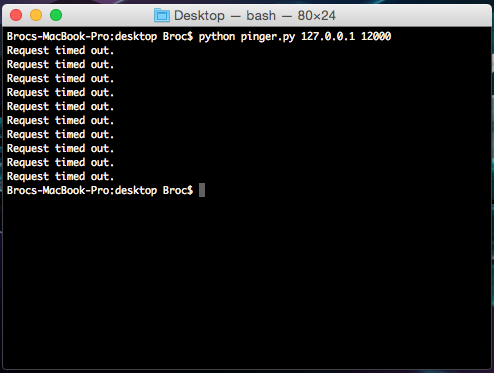
# Assume the packet is lost

print "Request timed out."

continue

# Close the client socket

clientsocket.close()



3. P1 on page 288

 P1. Suppose Client A initiates a Telnet session with Server S. At about the same time, Client B also initiates a Telnet session with Server S. Provide possible source and destination port numbers for

a. The segments sent from A to S.

source: 3600 destination: 23

b. The segments sent from B to S.

source: 3800 destination 23

c. The segments sent from S to A.

source: 23 destination 3600

d. The segments sent from S to B.

source: 23 destination 3800

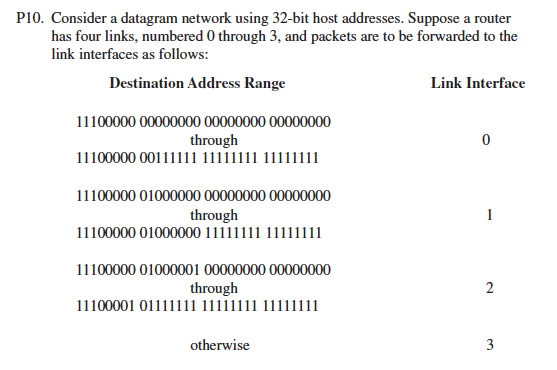
e. If A and B are different hosts, is it possible that the source port number in the segments from A to S is the same as that from B to S?

Yes, different hosts may have the same port number. They cannot have the same IP address and port number.

f. How about if they are the same host?

No, the IP and the port number would be the same.

4. P10 on page 419

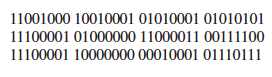


 a. Provide a forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.

|  |  |
| --- | --- |
| 11100000 00 | Interface 0 |
| 11100001 01000000 | Interface 1 |
| 1110000 | Interface 2 |
| 11100001 1 | Interface 3 |
| otherwise | Interface 3 |

b. Describe how your forwarding table determines the appropriate link interface

for datagrams with destination addresses:



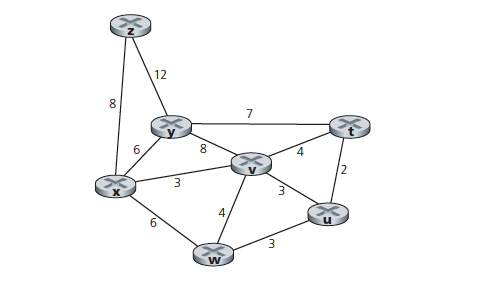
The first address would be link interface 3

The second address would be link interface 2

The third address would be link interface 3

5. P26 on page 422

P26. Consider the following network. With the indicated link costs, use Dijkstra’s shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 4.3.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | N’ | D(t),p(t) | D(u),p(u) | D(v),p(v) | D(w),p(w) | D(y),p(y) | D(z),p(z) |
| 0 | x | ∞ | ∞ | 3,x | 6,x | 6,x | 8,x |
| 1 | xv | 7,v | 6,v |  | 6,x | 6,x | 8,x |
| 2 | xvu | 7,v |  |  | 6,v | 6,v | 8,x |
| 3 | xvuw | 7,v |  |  |  | 6,v | 8,x |
| 4 | xvuwy | 7,v |  |  |  |  | 8,x |
| 5 | xvuwyt |  |  |  |  |  | 8,x |
| 6 | xvuwytz |  |  |  |  |  |  |

6. P3 on page 503

 P3. Suppose the information portion of a packet (D in Figure 5.3) contains 10

bytes consisting of the 8-bit unsigned binary ASCII representation of string

“Networking.” Compute the Internet checksum for this data.

78(N)+101(e)+116(t)+119(w)+111(o)+114(r)+107(k)+105(i)+110(n)+103(g) = 1064

0100 1110 +

0110 0101

0111 0100

0111 0111

0110 1111

0111 0010

0110 1011

0110 1001

0110 1110

0110 0111

100 0010 1000 = 1064

00101000

100 +

00101100 = wraparound sum

11010011 = one’s compliment

7. P7 on page 581

P7. Suppose an 802.11b station is configured to always reserve the channel with the RTS/CTS sequence. Suppose this station suddenly wants to transmit 1,000 bytes of data, and all other stations are idle at this time. As a function of SIFS and DIFS, and ignoring propagation delay and assuming no bit errors, calculate the time required to transmit the frame and receive the acknowledgment.

802.11 frame with no data = 32 bytes. With a transmission rate of 11mbps the transmit frame is 32\*8bytes/11mbps = 23usec. The data is 1000bytes so the time is 1000\*8/11

= 727usec.

DIFS + RTS + SIFS + CTS + SIFS + FRAME + SIFS + ACK

Transmission time is DIFS + 3SIFS + 3 \* 23 + 23 + 727 = DIFS + 3SIFS + 820usec

8. P20a on pages 665

P20. Consider the figure below, which shows a leaky bucket policer being fed by a stream of packets. The token buffer can hold at most two tokens, and is initially full at t = 0. New tokens arrive at a rate of one token per slot. The output link speed is such that if two packets obtain tokens at the beginning of a time slot, they can both go to the output link in the same slot. The timing details of the system are as follows:

 1. Packets (if any) arrive at the beginning of the slot. Thus in the figure, packets 1, 2, and 3 arrive in slot 0. If there are already packets in the queue, then the arriving packets join the end of the queue. Packets proceed towards the front of the queue in a FIFO manner.

2. After the arrivals have been added to the queue, if there are any queued packets, one or two of those packets (depending on the number of available tokens) will each remove a token from the token buffer and go to the output link during that slot. Thus, packets 1 and 2 each remove a token

from the buffer (since there are initially two tokens) and go to the output link during slot 0.

3. A new token is added to the token buffer if it is not full, since the token

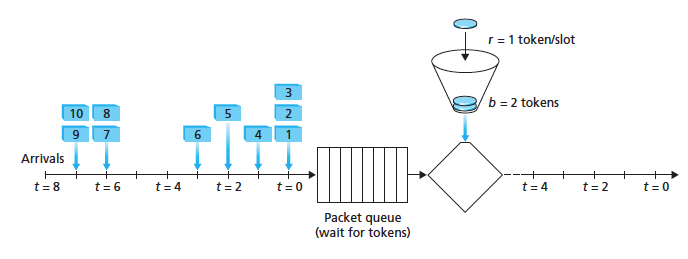
generation rate is r = 1 token/slot.

4. Time then advances to the next time slot, and these steps repeat.

Answer the following questions:

a. For each time slot, identify the packets that are in the queue and the number of tokens in the bucket, immediately after the arrivals have been processed (step 1 above) but before any of the packets have passed through the queue and removed a token. Thus, for the t = 0 time slot in the

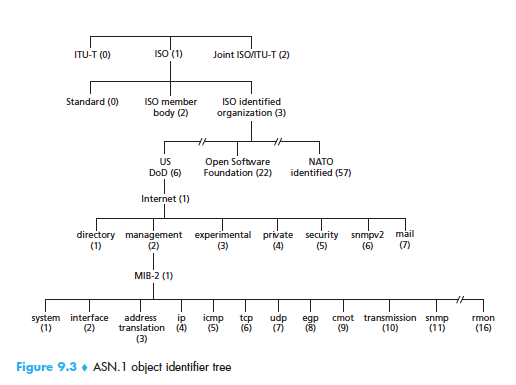
example above, packets 1, 2 and 3 are in the queue, and there are two tokens in the buffer.



|  |  |  |  |
| --- | --- | --- | --- |
| Time slot | Packets in queue | Tokens in buffer before output | Output |
| 0 | 1, 2, 3 | 2 | 1, 2 |
| 1 | 3, 4 | 1 | 3 |
| 2 | 4, 5 | 1 | 4 |
| 3 | 5,6 | 1 | 5 |
| 4 | 6 | 1 | 6 |
| 5 | none | 1 | none |
| 6 | 7, 8 | 2 | 7, 8 |
| 7 | 9, 10 | 1 | 9 |
| 8 | 10 | 1 | 10 |

9. P3 on page 784

P3. What is the ASN.1 object identifier for the ICMP protocol (see Figure 9.3)?



For icmp protocol it is 1.3.6.1.2.1.5