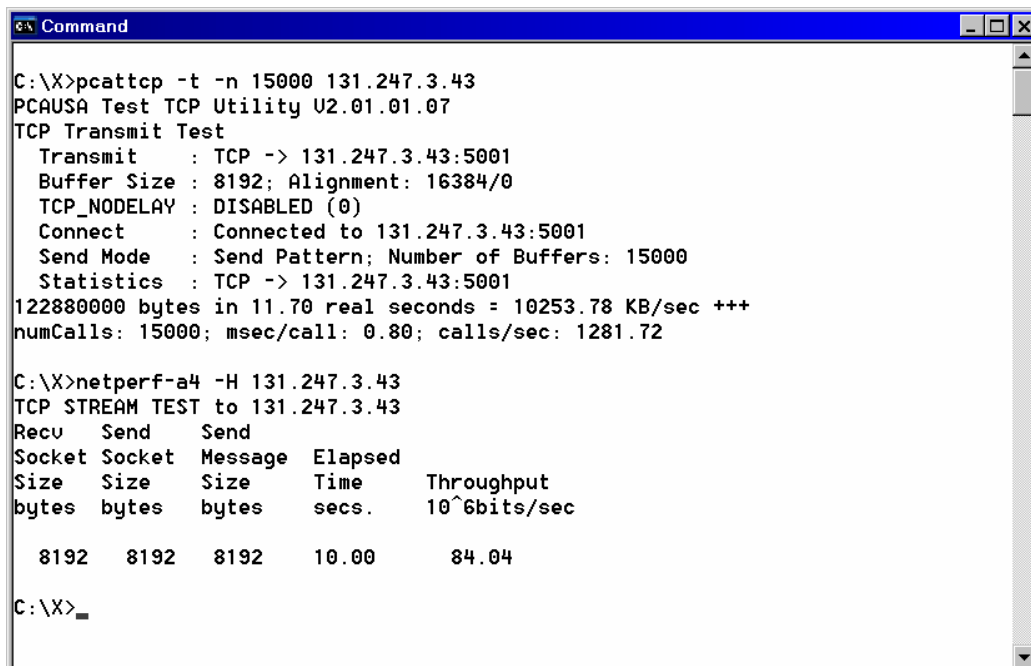


## Solution for Assignment #3

KJC (09/27/04)

#1) Here are two screen shots (sender is the first screenshot and the receiver is second) showing a run of `pcatttcp` (version 2.01.01.07) and `netperf` (version 2.1). The `pcatttcp` setting for buffers to send is such that the run time is about the same as for `netperf`. The sender was a Dell Simension 8300 Pentium4 3.2 Ghz with 1GB of RAM and Windows XP, the receiver was a Dell Latitude C840 Pentium4 2 Ghz with 256MB of RAM and WindowsXP. Both the sender and receiver were connected to the CSE department LAN at 100-Mbps in ENB 319. The transfer rates are almost the same (about 82-Mbps) as would be expected given the roughly same parameters (8192 byte buffers) for each program where each program implements essentially the same simple sockets `send()` and `recv()` loop. This experiment was run five times with similar results (within a few percent) in each run.



```
Command
C:\X>pcatttcp -t -n 15000 131.247.3.43
PCAUSA Test TCP Utility U2.01.01.07
TCP Transmit Test
  Transmit   : TCP -> 131.247.3.43:5001
  Buffer Size : 8192; Alignment: 16384/0
  TCP_NODELAY : DISABLED (0)
  Connect    : Connected to 131.247.3.43:5001
  Send Mode  : Send Pattern; Number of Buffers: 15000
  Statistics : TCP -> 131.247.3.43:5001
122880000 bytes in 11.70 real seconds = 10253.78 KB/sec +++
numCalls: 15000; msec/call: 0.80; calls/sec: 1281.72

C:\X>netperf-a4 -H 131.247.3.43
TCP STREAM TEST to 131.247.3.43
Recv  Send  Send
Socket Socket Message Elapsed
Size  Size  Size   Time   Throughput
bytes bytes bytes  secs.  10^6bits/sec

   8192   8192   8192   10.00    84.04

C:\X>_
```



```
Command
C:\X>pcatttcp -r
PCAUSA Test TCP Utility V2.01.01.07
TCP Receive Test
  Local Host : KJC_LAPTOP
*****
  Listening...: On port 5001

  Accept     : TCP <- 131.247.3.42:2323
  Buffer Size : 8192; Alignment: 16384/0
  Receive Mode: Sinking (discarding) Data
  Statistics : TCP <- 131.247.3.42:2323
122880000 bytes in 11.70 real seconds = 10259.04 KB/sec +++
numCalls: 17822; msec/call: 0.67; calls/sec: 1523.64

C:\X>netserver-a4
Starting netserver at port 12865

C:\X>_
```

#2) The below screenshot shows `pcattcp` runs for TCP and UDP throughput. This is for the same configuration as in problem #1 above. The UDP throughput is about 7.5% higher than the TCP throughput. This experiment was run five times with similar results (within a few percent) in each run. A hypothesis for this result is that UDP requires less CPU overhead than does TCP and thus can achieve slightly higher throughput. For both the TCP and UDP runs, the throughput is less than the link capacity (100-Mbps), so there might be a bottleneck in the PCs (e.g., the CPUs) or waiting for network events (however, the RTT between the two PCs on a LAN is very small – probably in the 100s of milliseconds). To test this hypothesis, measurements should be made on CPU utilization of both the sender and receiver. Also, a packet trace of the network could be used to look for waiting delays.

```

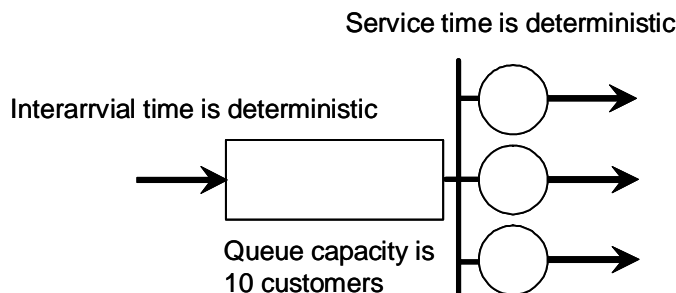
C:\X>pcattcp -t -n 15000 131.247.3.43
PCAUSA Test TCP Utility U2.01.01.07
TCP Transmit Test
  Transmit      : TCP -> 131.247.3.43:5001
  Buffer Size   : 8192; Alignment: 16384/0
  TCP_NODELAY  : DISABLED (0)
  Connect      : Connected to 131.247.3.43:5001
  Send Mode    : Send Pattern; Number of Buffers: 15000
  Statistics   : TCP -> 131.247.3.43:5001
122880000 bytes in 11.44 real seconds = 10491.34 KB/sec +++
numCalls: 15000; msec/call: 0.78; calls/sec: 1311.42

C:\X>pcattcp -t -u 15000 131.247.3.43
PCAUSA Test TCP Utility U2.01.01.07
UDP Transmit Test
  Transmit      : UDP -> 131.247.3.43:5001
  Buffer Size   : 8192; Alignment: 16384/0
  Send Mode    : Send Pattern; Number of Buffers: 2048
  Statistics   : UDP -> 131.247.3.43:5001
16777216 bytes in 1.45 real seconds = 11275.98 KB/sec +++
numCalls: 2050; msec/call: 0.73; calls/sec: 1410.87

C:\X>_

```

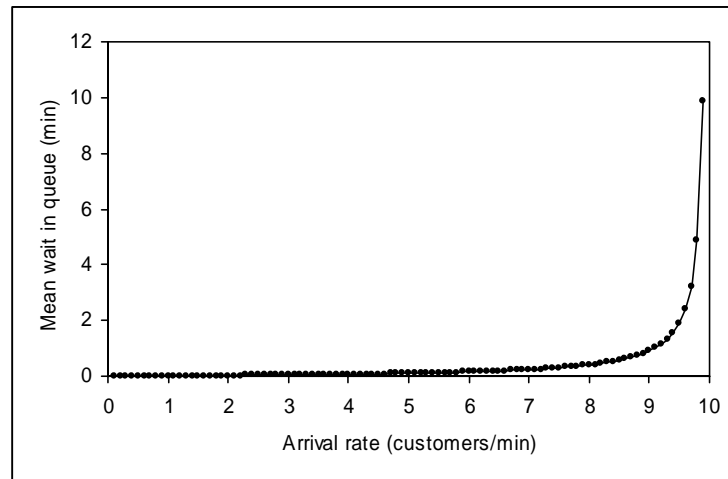
#3) The D/D/3/10 queue is shown below. Increasing the queue size of a D/D queue cannot reduce steady-state delay or loss because in a D/D system the queuing delay and loss are either zero (for utilization less than or equal to one) or infinite (for utilization greater than 1).



#4) From Kleinrock (1993, page 1189): “We define this boundary to be the place where the two terms in our equation are exactly equal; namely, where the propagation delay equals the queueing-plus-transmission time delay.”

#5) There will be 20 customers queued at the end of 10 minutes  $((12-10) \cdot 10)$ .

#6) For an M/M/1 queue with arrival rate  $\lambda$ , service rate  $\mu$ , and  $\rho = \lambda/\mu$  we know that  $L = \rho/(1-\rho)$ ,  $W = L/\lambda$ , and  $W_q = W - 1/\mu$ . For  $\mu = 10$  the arrival rate can obviously be no more than  $\lambda = 10$ . We range  $\lambda$  for 0 to 9.9 in steps of 0.1 and plot  $W$  and  $W_q$ . The plot is below. The mean queue wait times ( $W_q$ ) for arrival rates of 6, 7, 8, and 9 customers per minute are 0.150, 0.233, 0.400, and 0.900 minutes, respectively.



---