Final exam–December 11

Open book section (64 points)

The exam is to be turned in at 3:00 pm. The closed book section should be turned in before you open your books and notes to work the open book section. For the open book section, write your answers on separate pieces of paper.

I. (9 points)

You are taken to a large room which contains 100 computers labeled $M_0, M_1, \ldots, M_{99}$. In a stack in one corner of that room is 110 Ethernet interface cards and transceivers and an infinite supply of Ethernet cable and terminators. Each of the computers can hold up to two Ethernet interface cards. You are asked to connect these 100 computers into a network using the cable and 110 interface cards to run a mysterious distributed application.

All you are told about the distributed application is that each machine M_i runs a program P_i. Each program P_i is a loop that looks like.

```
var B: array[0:999] of character;
while true do
    Compute B for about .1 second;
    Send B as a datagram to P_{i+1 mod 100};
    Receive new B as a datagram from P_{i-1 mod 100}.
```

That is, the programs are connected in a loop and about every tenth of a second, each program sends 1000 bytes to its right neighbor and receives 1000 bytes from its left neighbor.

Describe a way of connecting these 100 computers to efficiently run this distributed application. Estimate the traffic on each Ethernet in your network. Estimate the shortest, longest, and average time required to transmit the 1000 byte packets between neighboring programs.

II. (6 points)

To quote Comer (problem 2.9): "A *bridge* is a computer that connects two Ethernets by sending all packets from one to another and vice versa. An *adaptive bridge* monitors source and destination address in all packets it encounters, gradually building a table of addresses on each network so it knows whether transfer is needed."

The question you are to answer is also from Comer (problem 2.10): "Multiple adaptive bridges can be used to join more than two Ethernets. For example, two bridges can be used to interconnect three Ethernets. Characterize the interconnection topologies that are allowed, and those that are not." Be sure to explain why your characterization is correct.

III. (6 points)

Once upon a time, the princes of ARPANET decreed there would be two kinds of addresses: hardware address and protocol address (Internet numbers). Discuss what is lost and what is gained in having two different addresses identify machines.

IV. (6 points)

Packets send by connectionless datagram service may be unreliably delivered, unsequenced, and redundant. Describe a way of taking an unreliable datagram service and turning it into a reliable one. By reliable, we mean that every packet sent will be delivered at least once. Try to design your new, improved service so that is does not unnecessarily constrain the concurrency of datagram service, that is, if it takes a very long time to deliver one particular datagram, don't hold up other independent datagram transmissions.

V. (28 points)

Suppose we want to build a "shared memory" server on a network. The shared memory will be divided into about one thousand blocks of some interesting size (say one thousand bytes).

a. (6 points)

Describe an *abstract* RPC interface that allows remote processes to read and write blocks of this shared memory. Allow processes writing blocks to set write locks. An *abstract* interface is a listing of procedures and arguments and an informal description of what the procedures do.

b. (10 points)

Outline implementations of your interface using (i), Unix sockets, (ii), Ada, and (iii), ARGUS.

c. (2 points)

Is your implementation *stateless* or *statefull*? Would either choice make it more reliable?

d. (5 points)

Describe how your implementation could be made more reliable by replicating the shared data. Explain how this would effect processes using the shared data, that is, how is the "user" interface changed? What happens when one of the replicated servers crash?

e. (5 points)

Suppose the network holding the shared memory is very large and widely distributed. For example, the network could be the ARPANET. How could you make the shared memory more efficient? (Hint: Think about similar problems faced with file servers.) Again, how does this effect the interface?

VI. (9 points)

Suppose you have 10 diskless workstations each with four million words of memory and one file server (also with four million words of memory) on a network (either CSMA/CD or token ring, your choice). A certain file on the server's disk contains ten million integers randomly and uniformly distributed within the range zero to one billion.

Describe an efficient distributed algorithm for sorting that file. Estimate how long it will require to run. (Assume each machine can perform an *in-core* sort of N integers in $10\,N \log_2 N$ microseconds.)