Csc72010

Network Protocols and Distributed Algorithms

Assignment 2

Due March 2, 2006

Objectives:

Understand impact of different reliability techniques on performance
Model important protocols (ARQ, STP).
Simple proofs of invariants

Instructions:
Please indicate the method used or give diagrams to illustrate computations. This permits partial credit if your method is close to right but your computation is wrong. Please keep answers concise but clear. You will get more sympathy if you explain what you're doing clearly but we don't have to read volumes to grade papers.
Please simulate automata if at all possible. This is not required in question 2, but strongly recommended. I doubt that you will get the automaton right without simulation.
Copying from code in the book, in the manual, in lectures, or in handouts is permitted, as long as you credit the source. No other sources are permitted. We will question idential code unless we know it has a shared, legitimate source.

Questions:

1. How long does a take to transfer a 10-MB file over a link having an RTT of 100 ms and a frame size of 1500 bytes, in each of the following cases. Hint: approximate answers to a similar problem, problem 6, are in the back of the book. Better answers to 6a and 6b are in the attached diagrams.
   a. The bandwidth is 10Mbps and frames can be sent continuously as long as none is lost. [Use this for part e: If one is lost, the previous 666 frames must be re-sent.]
   b. The bandwidth is 10Mbps and after sending each frame, we must wait an RTT to send the next (e.g., stop and wait).
   c. The bandwidth is infinite, ie transmit time is 0, and we can send 30 frames, then wait an RTT for an ACK. [Use this for part e: If the ACK doesn't arrive, we send the previous 30.]
   d. The bandwidth is infinite. Durint the first RTT, we can send 1 frame (2^{1-1}). During the second RTT, we can send the 2 frames (2^{2-1}). During the third RTT, we can send 4 frames (2^{3-1}) . [For part e: If a frame is lost, re-send the previous batch and cut the number of frames by half in the next RTT.
   e. (Extra credit) If 1 frame in each 100,000 is lost, how does that affect the worst-case time for each of the above questions? Alternatively, determine

the expected time to transfer the file.

2. Define one or two IOAs for the Sliding Window Protocol with Cumulative Acknowledgment.

   The data part of the message may be defined as an Int, for simplicity.
   Define a tuple type for messages, including sequence number, acknowledgment number (next expected sequence number), and data.
   You may either initialize the state with a buffer of messages to send or create an init action that initializes the state. To test, make sure you test with the initial array much larger than the window size.
   Note that in simulating, you need only two processes connected by a lossy, reordering, duplicating channel (you can compose a reording channel with a lossy channel and a duplicating channel to achieve this).

   Ideally, you will define a single two-way IOA, but you will get 75% credit for defining a sending IOA and a receiving IOA. I recommend trying the two-way IOA, but if you find that hard another possible strategy is to define the two IOA's first, then combine them if you have time.

3. State and prove the invariant relating window size (SWS), last acknowledgment received (LAR), and next frame (NEXT) to be sent: NEXT <= LAR+SWS. You can test the invariant using the simulation.

4. The attached IOA is an approximate model of the Cisco Spanning Tree Protocol.

   An important part of the algorithm is to send a BPDU every 2 seconds for a number of rounds. Since we can't keep time in this IOA model, the IOA models this by having the scheduler fire the actions in rounds. We only are only interested in the executions that this scheduler can produce.

   Note also that the number of rounds that a process spends in `electingRoot` state (i.e., leader election) is a parameter of the process, and that the process changes state to `designatingPorts` after this many rounds have passed. It stays in `designatingPorts` for only one round.

   a. Assume that you know the configuration of the network, i.e., the graph representing the connectivity of the bridges. How many rounds should the processes stay in `electingRoot`? Give a general rule for the number of rounds.

   b. Prove that the set of bridges in the network together with the edges incident on a `rootPort` is a spanning tree, after running the STP algorithm on the network, assuming that the number of rounds satisfies the rule in part a.