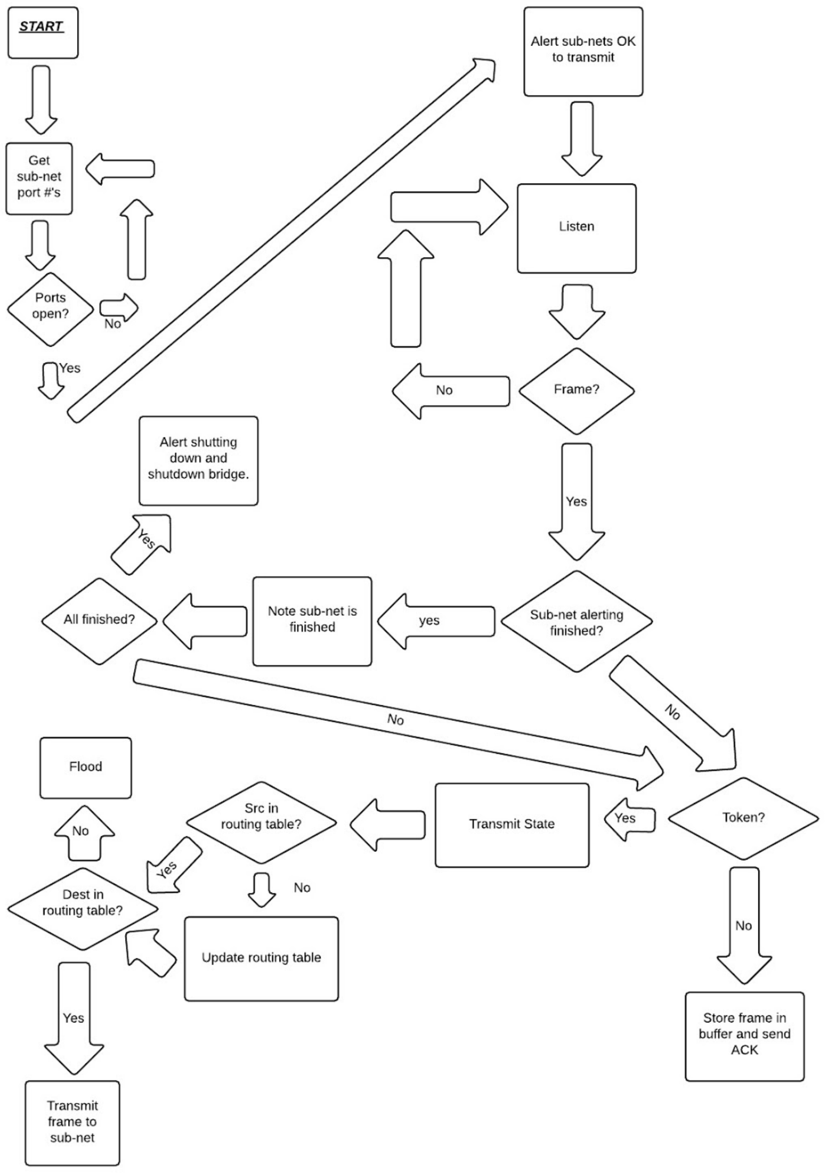
Steven Paustian
CSE 353
Project 3 Proposal

1) The token ring was and will continue to be implemented in Java. The bridge will be implemented in C++
2)



3)
    a) The bridge will only transmit frames being passed from one subnet to another when it has the token for that subnet. For example, when ring 1 passes a frame to ring 2, the bridge will buffer the frame and send an ACK. It will wait to transmit the frame into ring 1 until it receives the ring 2 token.
    b) The token ring will be initialized by executing the TokenRing executable twice; the executable will take 1 argument, the conf file. It will initialize itself based on that file (which will require no modifications). The bridge will initialize itself, taking only a log file as an arguments. Once initialized, the bridge will display 2 prompts asking for the listening port numbers for the monitors in ring 1 and 2. The bridge will then connect to the monitor of each sub-net and listen for incoming frames from the last node in the ring.
    c) The conf files will require no changes.
    d) There will be 2 executables: the token ring and the bridge. The token ring executable will be run twice for each sub-network. The bridge will be executed once.
    e) No modifications to the input/output files.
    f) The only command line modifications will be for the bridge. The bridge will only take 1 argument, the log file. I plan to implement dynamic bridge learning, so conf files will not be necessary.
4) The bridge will listen to frames coming from the last node in the subnet, and transmit to the monitor.

   When the monitor of each ring is alerted that none of it's nodes have any more data to send, it will then alert the bridge by sending a flag (format to be determined later). When the monitor receives shutdown flags from both sub-nets, it will alert both sub-nets it is shutting down. The bridge will then shutdown, and both sub-nets will shutdown
5) Bridge log files will contain timestamped events, where events include buffer levels (including thresholds being reached), identification of new nodes in a sub-net (dynamic network topology), relative buffer synch events, and anything else that is relevant.
6) I plan to use C++ to implement my bridge. Also, I plan to implement dynamic bridge learning. Frames being bridged with un-mapped source addresses will be dynamically

added to the routing table. Destination addresses which are unmapped will be flooded into both networks, and removed when seen as ‘orphaned.’

7) I’ll be following Google’s C++ coding style guide.
8) Risks are numerous. Errors in dynamic bridge learning, interaction between the C++ bridge and Java token rings, correct initialization of three different programs and order they are started, correct transmission of frames, keeping the tokens for different rings straight, buffer space, bridge ACK and NACKs and endless other possibilities for failure...
9) Modifications to my token ring will include different assignment of IDs to the nodes, waiting until the bridge gives the go-ahead before transmission begins, program output, the number and type of command line arguments, the connection of bridge nodes, and THT changes to simulate different transmission rates.