LAN Bridge Spanning Tree Animation
Final Year Dissertation
Deliverable one

Simon Arlott
Supervisor: Peter King

December 9, 2005

Contents

Chapter 1

Overview

The IEEE 802 committee has developed a series of algorithms for enabling bridges to be placed between LANs without the stations being aware of it. If the network formed has loops, the bridges run an election protocol to ensure that only sufficient bridges to form a spanning tree remain active. This project is to develop a system for specifying a configuration of LANs, simulating the operation of the spanning tree algorithm, and displaying the simulation in real-time using an interactive graphical environment.

Chapter 2

Aims and Objectives

The aim of this project is to create a program to set up a network configuration and run a simulation of the spanning tree algorithm on it to demonstrate what the algorithm does, and the status of the bridges and network when the algorithm is running.

2.1 Network configuration

To make the simulation will require a method of configuring the network that the simulation will be run on. This will require a graphical interface to be developed allowing the configuration to be created, edited, and displayed. Since the algorithm will be demonstrated with the same configuration multiple times a feature to save and reload will be required.

2.2 Simulation

Simulating a network requires some control over the speed of events, since in a real situation communication will occur faster than is suitable for a useful demonstration, therefore the simulation must have a variable speed and be able to be stopped and resumed at any time. The simulation of the algorithm itself needs to be interactive, allowing the current status of any part of the network to be obtained and modified (e.g. by disconnecting a bridge) to how the algorithm reacts to changes. Automated operation must also be provided to allow the simulation to run through certain situations on it's own.

2.3 Display

The current spanning tree that has been produced from running the algorithm should be shown on the display. When displaying the simulation packets between one part of the network and another should be shown to add animation to the system and to indicate where traffic is flowing. The program should be easy to use, without being too simple and unable to show enough detailed information at the same time.

Chapter 3

Background

3.1 Interconnections, Second Edition

Interconnections, Second Edition is a guide to Internet networking describing Bridges, Routers, Switches and Internetworking Protocols. [1]

The following is a summary of the content of chapter 3.3 to 3.5 and it's relevance to the network configuration, simulation and display of the spanning tree algorithm.

3.1.1 Spanning Tree Algorithm (Chapter 3.3)

Explains the purpose of the spanning tree algorithm - to have bridges dynamically discover a subset of the topology that is loop-free but still has enough connectivity so that there is a path between every LAN. Bridges transmit messages between each other that allow them to calculate a spanning tree.

Configuration Messages

Root Bridge election A single bridge is elected on all the LANs to act as the *Root Bridge*.

Shortest path to root All bridges calculate the shortest path from themselves to the Root Bridge.

Designated Bridge election Each LAN elects a *Designated Bridge* from among the bridges residing on that LAN, which is the one closest to the Root Bridge. The Designated Bridge will forward packets from that LAN to the Root Bridge.

Root port Each bridge will choose a port known as the *root port* that is the best path to the Root Bridge.

Select spanning tree ports Each bridge will select the ports to be included in the spanning tree, these will be the root port and any ports for which that bridge is a Designated Bridge.

Relevance to simulation

The simulation will need to model these messages and the algorithm to create an accurate simulation.

Relevance to network configuration

Configuration of the network will determine which bridge will become the Root Bridge based on an ID number.

3.1.2 Spanning Tree Algorithm Refinements (Chapter 3.4)

Failures

The algorithm based on handling one-time configuration messages only handles the initial start-up of the network, and new bridges or links, but it does not adapt to failures of bridges or links. Configuration messages for each port are stored with a *message age* field which incremented after a each unit of time. If the message age reaches a certain threshold it will be discarded. The Root Bridge transmits messages with an age of 0, and other bridges will retransmit it with the same age. When new bridges appear the Designated Bridge may need to retransmit the message which it will do using the current stored value of the message age.

Avoiding Temporary Loops

Since temporary partitions in a bridge topology are worse than loops, the algorithm minimises the probability of loops by requiring that bridges wait for a certain period of time before changing a bridge port to the forwarding state. The time required should be sufficient for the change in topology to spread to the rest of the network.

Station Cache Timeout Values

Bridges learn the location of stations, and stations might move so it's important that a bridge forget the location of a station after a certain timeout period - this time can be several minutes. When a topology change occurs the bridges notify the Root Bridge which will set a topology change flag in its confirmation messages for a period of time to make all the bridges use a shorter station cache timeout of several seconds so that the new positions of stations are learned again and the network stabilises quickly.

Port ID

Each port has an associated ID to allow the spanning tree to be deterministic in that given the same topology the bridge will always select the same root port, and to allow a bridge to have two ports on the same LAN.

Performance Issues

Performance of the bridges is important for the spanning tree algorithm because lack of receipt of messages causes bridges to add connectivity, and extra connectivity (loops) is potentially disastrous.

One-way Connectivity

If the hardware of a bridge or bridges fails so that connectivity between two bridges on a LAN becomes one-way, then loops can be created. The standard does not define what to do in this situation.

Networkwide Parameters

For the algorithm to work properly there are several parameters which all the bridges must agree on, these are configurable. To ensure that all the bridges use the same parameters, the Root Bridge includes the values in it's configuration messages.

Settable Parameters

The following parameters can be set:

Bridge priority An ID used to influence the choice of the Root Bridge and the Designated Bridge.

Port priority A value used to influence the choice of ports when a bridge has two ports connected in a loop.

Hello time The time that elapses between generation of configuration messages by a bridge that assumes it is root.

Max age The message age value at which a stored configuration message is considered too old and discarded.

Forward delay The amount of time before a port is allowed to become part of the spanning tree.

Long cache timer The cache timeout used under normal conditions.

Path cost The cost to be added to the root path cost field in a configuration message received on this port in order to determine the cost of the path to the root through this port. (Set individually per port.)

Relevance to simulation

The simulation should allow a bridge to be disabled, as if it had failed, to show how the algorithm adapts. The stored message age needs to be incremented at a suitable speed. The time used for caching of station locations under a stable topology needs to be long enough to allow inspection of the current cache of addresses at each bridge before the entries are expired, yet also short enough to allow the bridge to detect when a station has been moved. Since moving stations can only be done while the simulation is temporarily paused, the cache timeout value may need to be artificially lowered when a station is moved so that it expires soon enough for the simulation to be useful - the 15 minute timeout suggested may be suitable in a real network where it could take 15 minutes to move a station from one LAN to another, but not in a simulation where stations can be moved instantly.

Performance issues of bridges are not relevant to the simulation since the whole simulation will need to be carried out in a single thread to allow the speed of the simulation to be changed and paused reliably. Packets may become queued up when the network is congested, this can be handled by a FIFO buffer with a suitable limit to ensure that a flood of data packets don't cause the same effect as a bridge not having enough processing power to respond to configuration messages quickly.

Relevance to network configuration

The network configuration will require a method of setting all the parameters for each bridge on a per-bridge basis.

Relevance to display

The status of the bridges, the network and the spanning tree topology need to be displayed in a graphical form so the effect of the algorithm can be seen. Information about the current status of a bridge including the station cache should be made available by selecting the bridge.

3.1.3 Bridge Message Formats (Chapter 3.5)

The spanning tree algorithm uses only two types of message: configuration messages and topology change notification messages, in this section the formats are described in detail.

Relevance to simulation

The simulation will need to implement both message formats and support sending and receiving of them for the simulated bridges.

3.2 IEEE 802.1D Standard: The Spanning Tree Algorithm and Protocol

The IEEE 802.1D Standard section 4 describes the algorithm and protocol of the Spanning Tree algorithm for a Bridged Local Area Network topology. The standard has requirements of bridges and the algorithm, and a detailed explanation of the protocol and how it works including sample pseudo-code in C. [2]

Relevance to simulation

Pseudo-code example in C and detailed explanation of the protocol and algorithm sets out what the simulation will have to do to model the spanning tree algorithm.

Relevance to network configuration

The standard suggests values for settable parameters, which can be used as defaults in the network configuration part of the program.

Chapter 4

Requirements

4.1 Network Configuration

Loading and saving

Save network configurations to a file so they can be loaded again later.

Graphical edit interface

Create and modify a network of LANs, hosts and bridges using the mouse.

Ability to reconfigure network while the simulation is paused

This allows the possibility of adding more bridges and hosts to see the effect on the existing network.

Bridge identification

Each bridge and port requires an ID, so these must be configurable.

Bridge parameters

Individual settings of the algorithm parameters for each bridge. Required to show the effect of changing the parameters on the way in which the bridges discover other bridges on the same LAN.

4.2 Simulation

Variable speed

The overall speed of the simulation can be changed to see the changing state more clearly.

Pause function

The simulation can be paused at any time so that the state of bridges in the network can be viewed without it changing.

Stepping function

Runs the simulation until the next major event, such as a bridge changing it's root port.

Automation using scripting

Define the initial state and changes in the state of bridges and hosts using a list of events to follow. The automated script should be saved with the network configuration.

Random packet sending state for hosts

Automatically sends out packets from hosts to other hosts on the network at random.

Trigger a packet from a host to another host

Manually send a packet from one host to another.

Accessibility

It must be possible to control the simulation and switch between network objects using only the keyboard.

4.3 Display

Show/modify current status of a bridge

Information about the current state of bridges should be shown when they are selected and allow modification of the operation of the bridges as required for the simulation.

View log of a bridge's status

Information about the past states of the bridge as it changes in response to the spanning tree configuration packets.

Visual indication of packets

Animate packets moving around the network to show activity.

Spanning tree

Show the current spanning tree configuration to indicate which parts of the network are being connected to the rest of the network by the bridges.

Control hosts

Hosts need to be selectable and made to create data packets with a random or specified destination.

Control simulation

The simulation will require a way to control the speed, pause, and stepping functions and also the scriptable automation function.

Chapter 5

Testing

After completing all the sections of the program it will require testing to evaluate how good the interface is and how useful the simulation is at demonstrating the spanning tree algorithm.

5.1 Network Configuration

The network configuration interface needs to be user-tested to evaluate if it is easy to use and changed if any operations could be made easier to use. Basic testing of the requirements to check that the functionality is there and works properly.

5.2 Simulation

The simulation of the algorithm needs to be checked against the specification of the protocol in certain test situations to check that the results are correct, i.e. adding a bridge, moving a station, failure of a bridge or LAN.

5.3 Display

The display interface needs to be user-tested to evaluate how well it displays the simulation, performs the required functions and if there are any problems with how the user is expected to interact with the running simulation.

Chapter 6

Preliminary Implementation

6.1 Network Configuration

The screenshot in Figure 6.1 shows a preliminary implementation of the network configuration GUI.

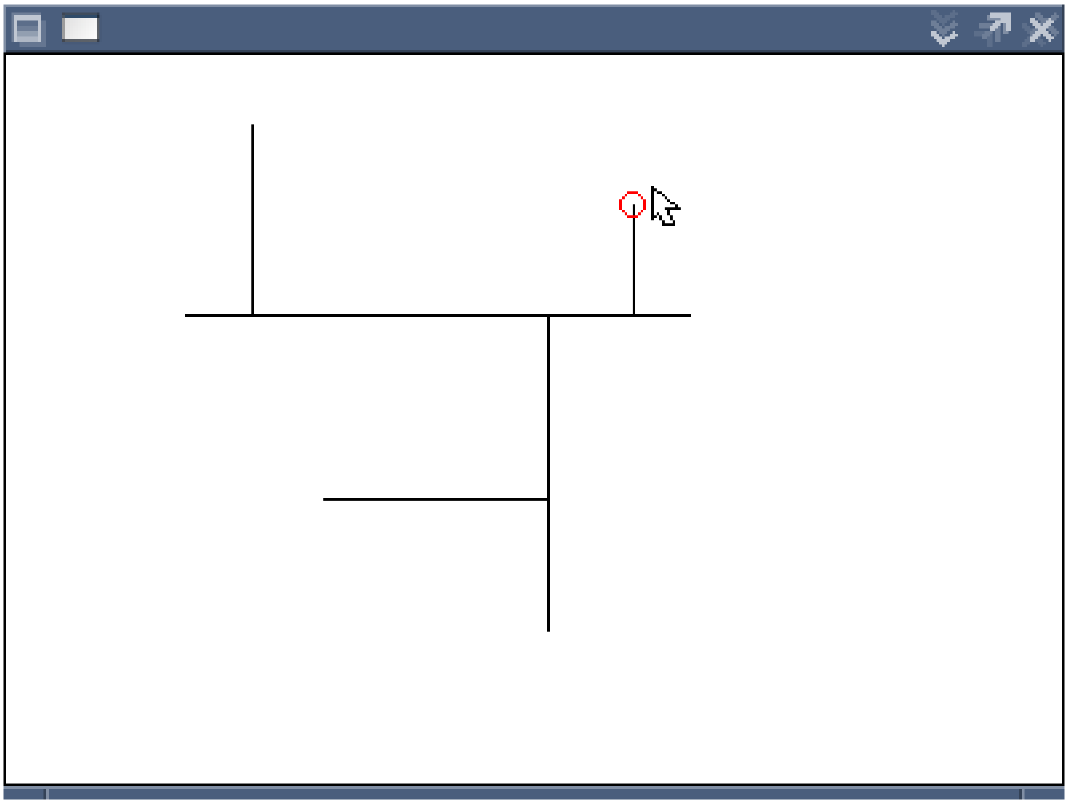

Figure 6.1: Network Configuration interface

LANs can be added by clicking to start a line and then again to end it, all the LANs will only be made horizontal or vertical to simplify the display. New LANs can easily be connected to existing LANs by a snap-like mechanism. LANs cannot overlap other LANs (i.e. two perpendicular LANs crossing each other) because it wouldn't be clear that they weren't connected.

In this implementation there is no toolbar for selecting what type of network component to add next, this would be required to insert bridges and hosts.

6.2 Simulation Display

Figure 6.2 is a drawing of how the simulation might display packets as they move around the network.

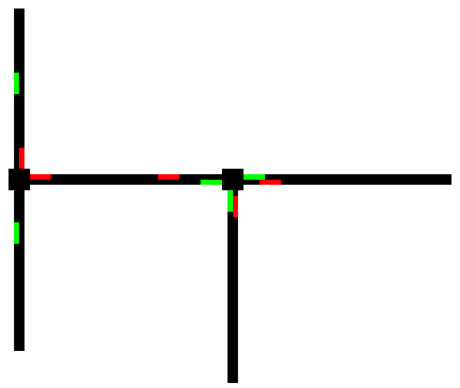

Figure 6.2: Simulation Display

Packets are shown in different colours depending on the direction they are travelling, it may be more suitable to show spanning tree algorithm protocol packets in a different colour from other data packets.

Chapter 7

Timetable

Deadline	**Activity/Deliverable**
Term 1	
14th October	Dissertation allocation form
2th December	First draft of requirements document
9th December	Requirements document (deliverable 1)
6th January	Network configuration requirements implemented
Term 2	
20th January	Interview with supervisor and second reader
3rd March	First draft of design document
17th March	Design document (deliverable 2)
24th March	Simulation and Display requirements implemented
31st March	Software ready for user testing
Term 3	
17th April	*Exams*
24th April	*Exams*
15th May	Testing completed including bug fixes
23rd May	First draft of dissertation
31st May	Second draft of dissertation
5th June	Dissertation finished
11th June	Presentation complete
23th June	Presentation

Bibliography

[1] Radia Perlman. *Interconnections Second Edition*. Addison-Wesley, 2000.

[2] *ANSI/IEEE Std 802.1D, 1993 Edition*. ISO/IEC 10038 : 1993.