CHAPTER 7

CONCLUSIONS AND FUTURE WORK

The last chapter in this thesis report presents a summary of the accomplishments made throughout this project. The final conclusions and findings of the work done are also discussed followed by suggestions of possible areas for future work.

7.1 Final Conclusions

The main objective of this thesis is to improve the performance of wireless networks, specifically the transmission performance in ad hoc type WLANs. Ad hoc WLANs are particularly prone to suffering from exceptionally poor TCP performance because the nodes in these networks move at random and result in an ever changing and unpredictable network topology. This condition results in an increase of packet losses and link failures as the nodes move about and the routes between them keep changing and reconfiguring themselves.

The drawbacks of using a medium-less transmission method also contribute to the poor performance because the nodes are susceptible to interference and this causes false link failures. This is because, relying solely on feedback information provided by the network and other lower layers, the nodes in an ad hoc network cannot tell between packet losses that are caused by genuine errors such as link failures or network congestion.

In order to overcome this false link failure problem, a proposal was made to incorporate fuzzy logic control into a wireless network model to help identify packets lost due to congestion in the network. Basically, the control makes use of signal strength information from the PHY layer to determine if nodes are still and will remain within range of each other. If they are, then the nodes extend their attempts to establish a handshake to enable data transfers.

The fuzzy logic control was implemented into a new IEEE 802.11b model which modifies the original 802.11 model available in the NS network simulator used. The model was then tested and evaluated using multiple simulation scenarios that represented heavy and light traffic conditions in a dense and sparse network respectively. Each scenario measured the effects of the fuzzy logic control against increasing node mobility or velocity.

Based on the results average from the simulation runs, in the same period of time, the network with the fuzzy logic control managing the RTS-CTS handshake attempts manages to successfully deliver a higher amount of data to the destination nodes. The fuzzy logic control manages to successfully increase the amount of data sent from the source and received at the destination up to 4.4% when the network is moderately loaded. When heavily loaded, the improvement is better at over 6.0% for the sent and received amounts. The improvement is more obvious in congested networks since most packet losses in the sparse networks are due to actual link failures.

The packet delivery ratio which indicates the ratio of received packets to sent packets also sees an improvement in both scenarios. Without fuzzy logic, the ratio drops drastically as mobility increases because as the nodes move with greater velocity, the likelihood of packet loss increases affecting the number of packets received at the destination thus lowering the ratio. With fuzzy logic in place, the ratio experiences a less severe drop in the delivery ratio.

In a sparse network at low traffic, the ratio drops only around 10.0% at the maximum simulated 20m/s node velocity. This is in comparison with the 14.0% drop when the fuzzy logic control is not used. At high levels of congestion and traffic, the ratio drops by only 7.5% when the fuzzy logic control is used. Without it, the packet delivery ratio suffers a 13.0% drop when the nodes reach the 20m/s velocity – almost twice the amount. What this means is that even with the increase in packets being transmitted in the network, the amount being received is also being increased. Therefore the fuzzy logic control has managed to reduce the occurrence of false link failures especially in heavily congested networks.

The increase in RTS-CTS attempts and packets transmitted using the fuzzy logic control does not increase the level of packet drops even though in theory it should as extra packets are transmitted. On the contrary, the simulations show a slight decrease instead in the drop percentage. This is because the RTS messages are sent only when the channel is idle and follow the same collision and backoff rules as any other packet. Similarly, transmissions of the packets only take place if the handshake succeeds hence ensuring that the channel is free.

The fuzzy logic control implemented tries to utilize a route to a node for as long as possible before giving up and notifying the routing layer of a broken link. It does this by continuing its attempts to establish a handshake to a particular node using the same route it presently knows. Only once it has exhausted its attempts does it notify the routing protocol of the link condition and the routes affected removed from its routing table. With this said, the final results indicate that the fuzzy logic seems to be able to correctly identify false link failures because it manages to send and receive more packets but there is no following increase in routing overhead. Under normal conditions, to send these additional packets would have required a node to invoke its route discovery procedure because it had previously removed the affected link that was actually busy and not broken.

In conclusion, the implementation of fuzzy logic control into a new wireless network model in NS was successfully completed in this thesis. Simulations to evaluate this new model have also shown that the proposal of using fuzzy logic as a method of improving the wireless networks performance has its merits.

7.2 Suggestions for Future Work

There were areas in this project that could have been improved upon but were not due to lack of time and resources. The implemented fuzzy logic code is particularly difficult to fine-tune to suit all situations. This is a trait of the simplified inference method used because the singletons in the output narrow down the inference result to a single value. Use of more complex fuzzy inference methods such as using Mamdami's (Min-Max) inference method would probably have avoided this problem.

Calculations for distance and power variables are based upon the simpler free-space model which is less accurate and tends to experience fluctuating gains as the values get larger and distances grow. Using the more complex *Two-Ray* propagation model would have been preferable but would have involved time-consuming complex coding and is left for future implementations.

The current implemented IEEE 802.11b model is functional but can be further extended to include other characteristics of the whole model such as new antennae models and energy models. The code for channel interference is also implemented but remains unverified due to lack of comparison information. As of now, the tracing and debugging of the code is dumped onto the screen which is a crude but effective way. This was done to reduce the burden of parsing large text files but it remains an option that could be implemented in the future. A function could be made available to create logs of the fuzzy logic code's workings as this could help with the analysis of the trace output.

All the simulations run in this project were limited to scenarios running the AODV routing protocol and TCP Vegas. Future attempts could involve the other available routing protocols and TCP variations for comparison purposes. Different traffic types such as CBR, VBR or real-time traffic could also be tested to evaluate the overall impact of the fuzzy logic control in affecting wireless network performance.

Finally, the code could be implemented into a newer version of NS (version 2.27 as of this report) which might iron out any bugs related to coding errors in this version of NS used. Currently, the bandwidth and data transfer amounts are not working correctly in the simulator because the packets being generated keep growing in size as layer headers are added on. These headers are somehow never removed at the destination, thus making the amount of data being transferred reported wrongly. Perhaps the port could even be implemented into the newest IEEE 802.11g or 'a' standards instead of the current 'b' since it is likely that it will be phased out in the near future and replaced by the faster standards.