A Frequency Control Algorithm for a Mobile Adhoc Network[1]

Andrew Coyle and Hung Nguyen[2]

Abstract. We present an algorithm that will allocate frequencies to nodes in a Mobile Adhoc Network. This algorithm uses the routing and the traffic information from the network and allocates to each node a frequency at which that node transmits in an optimal fashion. This algorithm has been used on a WiFi network using the BATMAN routing protocol. Results for this network are presented.

1. INTRODUCTION

Wireless networks have become ubiquitous in today's society with the advent of for example mobile phones, WiFi and WiMAX networks. Most of these systems use existing infrastructure such as base stations or access points for communicating. However in some military and civilian areas it would be advantageous to have the ability to deploy networks quickly without existing infrastructure. Networks of this type in which each node can be used to route traffic have the greatest flexibility. These adhoc networks may be rolled out in military tactical situations, in emergency relief operations, and in some third world countries.

There are many bottlenecks in these networks due to limited bandwidth being allocated across multiple networks and interference from other radio sources. The advantage of adhoc networks is that the traffic can route around these problems if alternative routes exist within the network. Alternatively if a non-jammed frequency is available traffic could route onto this frequency to escape jamming. Therefore an optimal route for a traffic flow will be a series of links of possibly different frequencies that both maximises the available bandwidth to the flow and minimises the interference that this flow causes to other network users.

In this paper we present a technique for allocating frequencies to nodes in a wireless network that optimises the performance of this network. A WiFi network has been created and the optimisation technique has been demonstrated using this network. This network uses the BATMAN routing protocol and the OpenFlow switching technology to route traffic over paths to, in some sense, optimise the traffic using this route.

Section 2 looks at related work in this field. In Section 3 the software and the hardware used in our network is described. A simple experiment to demonstrate how frequency switching leads to higher data rates through a network is shown in Section 4. The Frequency Control Algorithm is introduced in Section 5. Section 6 looks at some results using this algorithm and then a summary of this work is presented in Section 7.

2. RELATED WORK

There are literally thousands of papers that investigate Ad Hoc networks using simulation. These look at many different aspects of MANETs, including routing protocols and algorithms for optimising traffics. Many techniques for cross layer optimisation in which lower level protocols are changed to facilitate better higher level routing have been proposed. Allocating the network's resources at the lower layers to optimise the higher routing layer optimisation is a problem that has garnered much attention. A lot of this research has focussed on using physical layer information to better route traffic over better paths. For example [9] looks at using the signal to noise information to better route traffic using the SDR MANET protocol, and uses simulation to demonstrate this technique's efficacy. Other research has looked at routing around interference. For example cross layer planning has been discussed in [1]. An algorithm for modifying AODV to route through WiFi networks has been demonstrated on a test bed in [11].

The building and analysis of wireless networks using WiFi devices has become quite popular recently due to the ubiquity and low cost of these devices. A good review of the research laboratories around the world in 2007 is given in [7]. Some networks have been designed to look at how the BATMAN routing protocol works in large networks. For example in [4] a 49 node network has been set up to compare BATMAN with the OLSR routing protocol and in [5] an eight node network was set up to compare BATMAN with OLSR and BABEL. In [6] a small 5 node network was set up, but two of the nodes were moving so that a better understanding of how BATMAN and OLSR work with mobility was undertaken. In these works BATMAN has been shown to be an efficient and stable routing algorithm.

The OpenFlow switch consortium [2] has a number of publications that demonstrate the OpenFlow technology, although most of this work is concerned with wired IP networks. OpenRoads, the wireless extension of OpenFlow, has been presented in [8]. This paper describes a network of 30 WiFi access points connected by the Stanford University's Campus network.

1 This is supported by the South Australian Government through the Premier's Science and Research Fund
2 Centre for Defence Communications Information and Networking (CDCIN), University of Adelaide, Adelaide, Australia.

© MILCIS2010

3. THE NETWORK

The hardware network on which these experiments have been performed is a set of netbooks running Ubuntu with two 802.11 cards attached. The devices are all attached to two separate 802.11 networks, each on a separate WiFi channel, 1 and 6. Each netbook has two wireless IP addresses, one for each card. These addresses are taken from the sets 192.168.1.0/24 for the channel 1 cards and from 192.168.6.0/24 for the channel 6 cards.

The Better Approach To Mobile Adhoc Networking, BATMAN, routing protocol is currently being developed by the Freifunk Community in Germany [3]. It is a proactive distributed protocol that aims to quickly create stable routes with low network overheads. Each 802.11 network is linked together using the BATMAN routing protocol, which finds the optimal paths between nodes within this network. The BATMAN code is stable and works well for networks with high and varying link losses. With the two networks joined using the two BATMAN instantiations there now exists two routes between any two netbook, one on channel 1 and the other on channel 6.

An example of a BATMAN network is shown in Figure 1. This network has 8 nodes with IP addresses in the range 192.168.6.0/24. The link qualities between the different nodes are also shown. These values show the number of broadcast packets that must be transmitted for each single packet received at a destination. So for example from 192.168.6.30 it takes about 119 broadcast packets for 100 to be successfully received at 192.168.6.40. In the reverse direction it takes 440 broadcast packets from 192.168.6.40 for 100 to be successfully received at 192.168.6.30. Whilst an extreme case these numbers by themselves indicate the noisy, asymmetric nature of the wireless channels that we are looking at.

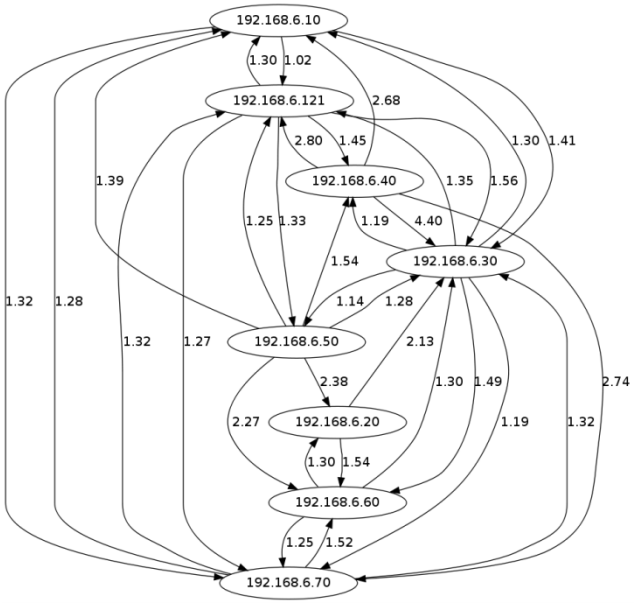


Figure 1 : A BATMAN network.

Further an OpenFlow switch [2] has been added to each node. This enables packets flowing through the network to be switched between 802.11 interfaces based on IP or MAC addresses or flow information. Whilst the switching could have been achieved without the use of an OpenFlow switch, this technique gives us a richness and flexibility for any future work. These various hardware and software components create a reasonable priced flexible and powerful network on which we can perform our various network experiments. Many technical issues needed to be overcome e to get the different technologies to work together. It is believed that this combination of 802.11, BATMAN and OpenFlow has never been attempted before and so some of the problems encountered was unique to this project.

For our experiments artificial streams of packets have been injected into the network. The Distributed Internet Traffic Generator software [10] has been used to do this. This software can generate traffic streams with different characteristics and can then measure the amount of traffic that is successfully carried from these traffic streams.

A central controller has been created from which we can distribute switching decisions. This controller holds code for the algorithm to calculate which device should use which frequency and then distributes this information over the wireless network using an ssh tunnel. Of course this means that if there is no link to a device then this information cannot get through. This is rectified partly by sending information over both wireless channels. Future work will look at distributed algorithms where this is no longer an issue.

The basic idea of the algorithm is to use traffic engineering to route traffic through the network. Note that the environment that we are using is a shared environment used by University networks used by students and employees. A typical snapshot of frequency usage in the 802.11 b/g range is shown in Figure 2. This was obtained using a WiSpy spectrum analyser and the Chanalyzer charting software. The usage of 802.11 networks at the standard 1, 6 and 11 frequency bands is apparent in this picture.

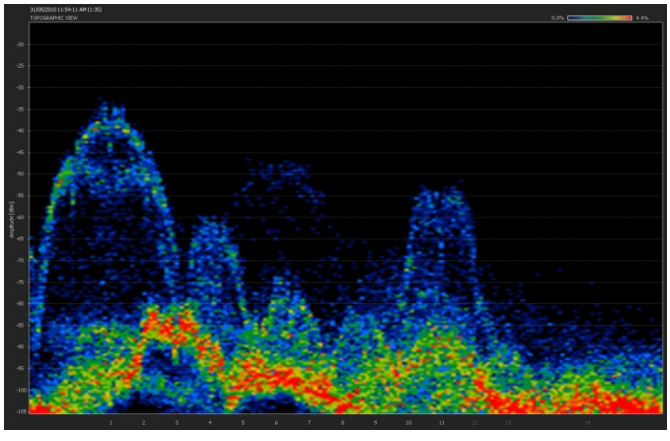


Figure 2 : A typical WiSpy mapping of the frequency usage at the Adelaide University.

The background 802.11 traffic is a two edged sword in that it means experiments are hard to replicate since other users of the spectrum will, at different times, use this resource.

© MilCIS2010

Simultaneously this noisy environment is exactly the type of situation that our algorithms have been designed to negotiate.

These various hardware, software and environment have been combined to create a research test bed. Various scripts have been written in Python to perform standard operations on the network, setting up a flow between two nodes for example. Using these scripts as an API to this network it is a relatively simple matter to set up and analyse various experiments.

4. A SIMPLE EXPERIMENT

To demonstrate in a simple fashion the ability of this network we set up a simple three node network and transmitted a data stream between two nodes through the third, see **Error! Reference source not found.**. This experiment shows how using different routes for a traffic stream leads to different throughputs.

Initially the traffic will be routed on a single frequency. Since the traffic on the first hop interferes with the second hop the throughput on this route is at the most half that of a single link's total bandwidth. Alternatively we can route the traffic on the second path on an alternative frequency so that there is now no interference between the traffic on the first hop and the second hop. Theoretically this path can then transmit data at the rate on a single hop, so that this route has doubled the possible rate of this traffic flow. Given that we are now using two frequencies where before we only used one this is to be expected. Nevertheless it is important to be able to verify this experimentally.

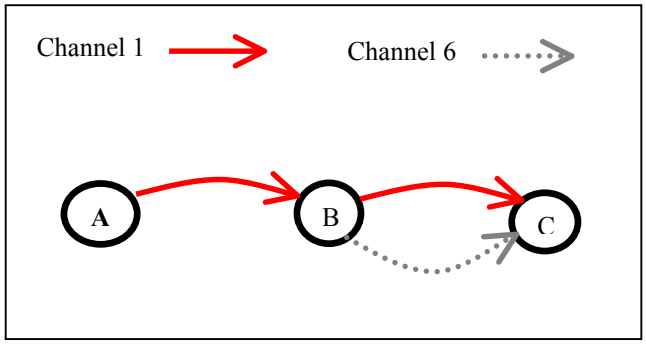


Figure 3: Simple Two Node Network

We have set up an experiment in which a stream of UDP traffics are transmitted over a two hop path. The Distributed Internet Traffic Generator (D-ITG) [10] has been used to create this stream of uniformly spaced, uniformly sized IP packets. For each experiment a stream of packets at a rate of 1 Mbps are sent from the origin to the destination over a three minute period. The D-ITG software analyses the results to determine the traffic rate over the route. This software measures the stream's throughput, so we will not expect this to be the channel capacities. IP and WiFi packet overheads alone will reduce the stream rate through the network. We alternate between an experiment using a single frequency and then one using the switched two frequency path. The rates on each link have been fixed and limited to 1 Mbps. An example of a WiSpy snapshot of the frequency usage when a single frequency is used is shown in Figure 4. In this case only channel 6 is used and so a strong single at the corresponding frequency can be easily seen.

The results from 20 repetitions are shown in Figure 5. The average rate for the single frequency experiments is about 381 kbps and that for the switched frequency case is 880 kbps, an increase of about 130%. Note that intuitively the increase would only be about 100%. That more than double the traffic is carried is indicative of the fact that there is no contention in the switched frequency scenario whilst there is in the single frequency scenario. Contention always reduces the efficiency of a channel and so in this case the lack of contention more than doubles the data rate.

.

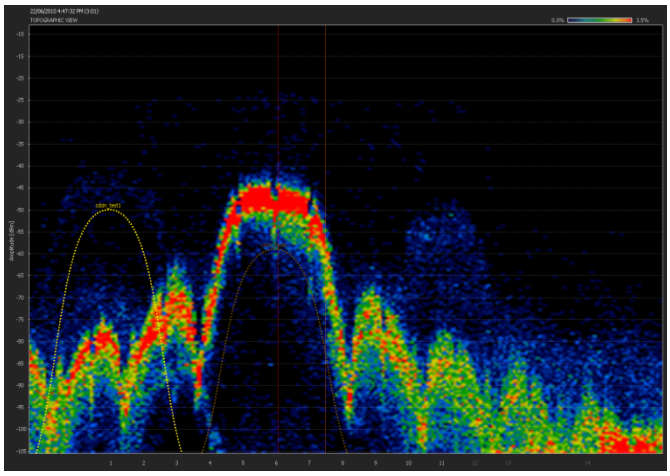


Figure 4 : WiSpy mapping during a single frequency experiment.

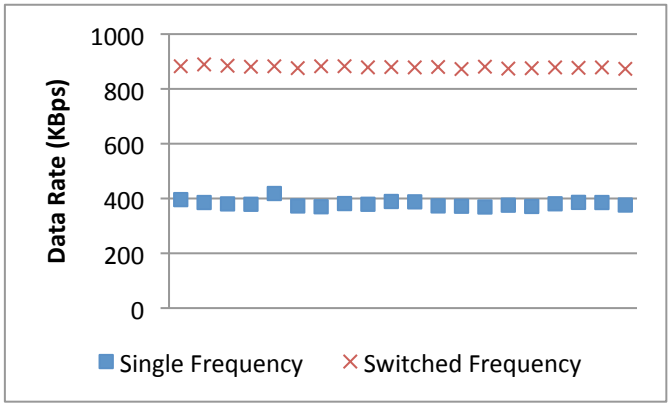


Figure 5 : A comparison of the data rates for the switched and single frequency 2 hop routes.

5. ALGORITHM

Far more interesting than the simple network is to create an algorithm that does not manually set up routes but automatically chooses in some sense the best route. This algorithm must be able to measure the state of the network, specifically the amount of traffic being lost on the different network links. The BATMAN routing protocol already measures these link qualities as part of its algorithm. These values can be sent to a central location where they can then

© MilCIS2010

be used by our algorithm to help choose an efficient path through a network for any traffic stream. There are many definitions of what an optimal path through a network would be. We do not attempt to unravel these issues but put forward an algorithm that has good utility and shows what can be achieved by choosing node frequencies. Future work will look at the routing problem from a more theoretical viewpoint, and see whether the theory flows into a more efficient algorithm. Also note that whilst various ways of choosing the best route were attempted the one presented here was the one which led to the best results. A purely utilitarian approach was taken when choosing the best solution.

Note that the BATMAN metrics are those for broadcast traffic. The WiFi protocol uses re-transmission mechanisms to significantly increase these loss rates, so the uni-directional traffic flow in a network is considerably higher than these loss rates may indicate. However this is at the cost of using more bandwidth – if a packet is re-transmitted 3 times before it is successfully acknowledged it has used up three times the bandwidth and has consequently interfered with other transmissions. Given this the BATMAN link qualities are still good proxies to the unicast link quality that an actual path will use.

Given a network of nodes firstly the valid non loop paths from the origin to the destination are calculated. These are calculated using an algorithm based on the Hamiltonian paths in the network. This algorithm ensures a non-exponential algorithm for finding these non loop paths. For each valid path the algorithm does two things:

1. Checks whether a path has the capacity to carry the offered traffic stream.
2. Finds the packet loss probability for this path when a new traffic stream is introduced.

The available bandwidth on a link is approximated by using the loss rate on a link . So for example in the network shown in Figure 1 it takes on average 1.3 packets to be sent from 192.168.6.121 to 192.168.6.10 for each successful packet. This implies a loss rate of about 30%. Note that the BATMAN packets that calculate this are small packets sent out periodically to measure the quality of the link. Therefore their loss rate can be taken as a measurement of the occupancy of the channel. So in this case 70% of our packets are carried successfully and we infer from this that the channel is available 70% of the time. Given this if the capacity of this link is 2 Mbps then the inferred available capacity is then approximately 1.4 Mbps. From the available capacities for all the links on the network the set of feasible paths for any traffic stream of a given rate can now be calculated.

The expected extra load on a link from the offered traffic stream depends on whether it uses the same or different channels that other channels use. If a route uses two or more hops using the same channel then as in the two hop network the links will interfere with one another. This effect is taken into account by dividing the available capacity in a channel by the number of hops using this channel.

So consider the network in Figure 6. To get from node A to node B there is a total of 6 paths. Two are single hops on channel 1 and channel 6 respectively, and the other 4 are all channel 1, channel 6 two hop combinations. Suppose also that every link is restricted to 2 Mbps. The available capacities on these routes have been calculated and are shown in Table 1. Consider for example the last route in this table. This is a two hop route with both hops on channel 6. The first hop has the full 2.0 Mbps available since it has a hundred percent BATMAN packet success rate. The second hop looses 20% of its BATMAN packets and so the capacity on this hop is 80% of 2 Mbps, that is 1.6 Mbps. So at most 1.6 Mbps is available on channel 6 for a new traffic stream Since both hops use the same channel we halve this value to get an available capacity of 0.8 Mbps. Note that the protocol overheads must now be taken into consideration, this will further reduce the capacity available and is a bit more complex since the overheads depend on the characteristics of the traffic stream; is it TCP or UDP, how big are the packets? For this network we can see that any stream less than 0.4 Mbps can be carried on any route. However any stream more than 1 Mbps can only be carried on the two hop paths with different channels. This is not unexpected due to the interference effects on the single channel paths.

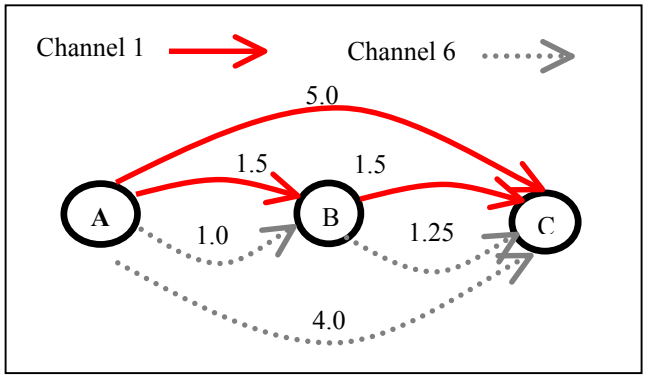


Figure 6 : A simple 2 node network.

Link 1	Link 1 Available Capacity	Link 2	Link 2 Available Capacity	Route Available Capacity
A-C Channel 1	0.4	-		0.4
A-B Channel 1	1.33	B-C Channel 1	1.33	0.67
A-B Channel 1	1.33	B-C Channel 6	1.6	1.33
A-C Channel 6	0.5	-		0.5
A-B Channel 6	2.0	B-C Channel 1	1.33	1.33
A-B Channel 6	2.0	B-C Channel 6	1.6	0.8

© MilCIS2010

Table 1 : The available capacities in Mbps from the network in Figure 6

For each feasible path in the network the expected loss rate for a new traffic stream can now be calculated. These loss are calculated from the product of the transmission probabilities. Once again consider the simple 2 link network in Figure 6. The route loss probabilities for each route are shown in Table 2.

The algorithm chooses from the set of routes with the available bandwidth that which has the least packet loss probability. This route is then distributed to the network using ssh – to mimimise the probability of the path information not getting through they are sent on multiple channels. The switch at the different nodes will then implement the route that the algorithm has chosen.

The algorithm is re-run every 30 seconds to ensure that routes adapt to a changing environment. Specifically if a link were to be overloaded with other WiFi (or other) traffic then the algorithm will measure the change in the network and the resultant path will take this into consideration. The resultant path will then most likely switch channel at that link to circumvent this interference or jamming. Experiments in which a link is overloaded with traffic from an independent network have been run to verify that the algorithm does re-route traffic using a route from a different channel.

Link 1	Link 1 loss rate	Link 2	Link 2 loss rate	Route loss rate
A-C Channel 1	80 %	-		80%
A-B Channel 1	33 %	B-C Channel 1	33 %	55.11%
A-B Channel 1	33 %	B-C Channel 6	20 %	46.40%
A-C Channel 6	75 %	-		75%
A-B Channel 6	0 %	B-C Channel 1	33 %	33%
A-B Channel 6	0 %	B-C Channel 6	20 %	20%

Table 2 : The route losses for the network in Figure 6

6. RESULTS

To demonstrate the efficacy of the algorithm a WiFi network of eight nodes has been created. As in the simple two hop experiment D-ITG has been used to create IP traffic streams between different nodes in a network. The frequency allocation algorithm has then been used to determine which channels should be used to route the traffic through the network.

For this experiment an eight node network has been created inside the CDCIN's offices at the University of Adelaide. Two BATMAN networks have been set up, one running on channel 1 the other on channel 6. A snapshot of the BATMAN network for channel 1 is shown in Figure 1. This indicates the link connectivity and quality for our test network. Note that this is just a snapshot, as different traffic from the other University WiFi networks is carried by these channels the quality of the links constantly changes. This in turn can lead to dynamic routing behaviour for the BATMAN networks which has been observed.

The BATMAN routing data are forwarded to a central location. These data have both the connectivity and quality information about the links in the BATMAN networks. From these the algorithm will work out which combination of links and channels will lead to the optimal path from an Origin Destination pair for a specific UDP data stream. The channel capacity for each link in this experiment has been fixed and limited to 2 Mbps.

Thirty experiments have been performed. The first part of the experiment is to route a 1 Mbps traffic stream over the fixed frequency BATMAN route, the optimal single frequency route according to the BATMAN criteria. The second part of the experiment is to use the Frequency Control Algorithm to choose the route through the network. Interleaving these two route choice techniques we then compare their results to determine the increased throughput that can be achieved using the frequency control algorithm.

Figure 7 shows the bitrate in Kbps for our 1 Mbps traffic streams. The average rate using a single frequency is 580 Mbps whilst that using the frequency switching algorithm is 740 Mbps, an increase of about 27%. The packet loss probability for these streams is shown in Table 2. The average single frequency loss rate is 20.5% whilst that from the algorithm is 29.6%. What is apparent from the graphs is that the algorithm is more likely to find a route that enables most of the traffic to be carried. However when interference from other sources means that no clear route can be found then neither approach is guaranteed to find a good route. If there is a route that can carry the traffic then the frequency selection algorithm has a high probability of finding that route.

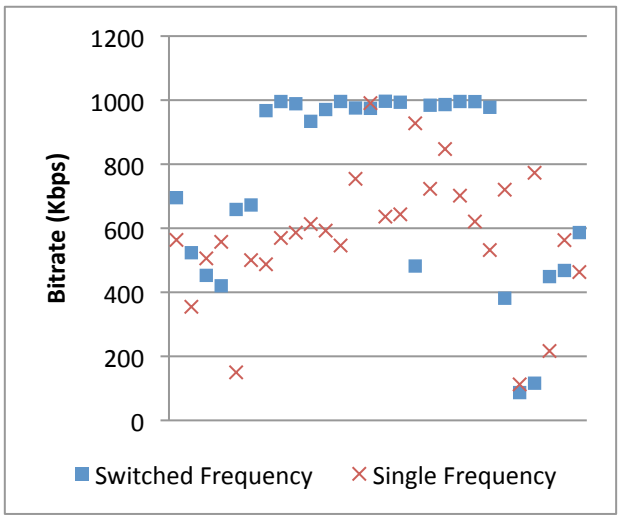


Figure 7 : The rate of the traffic through a network using the switched frequency algorithm versus a single frequency path.

© MilCIS2010

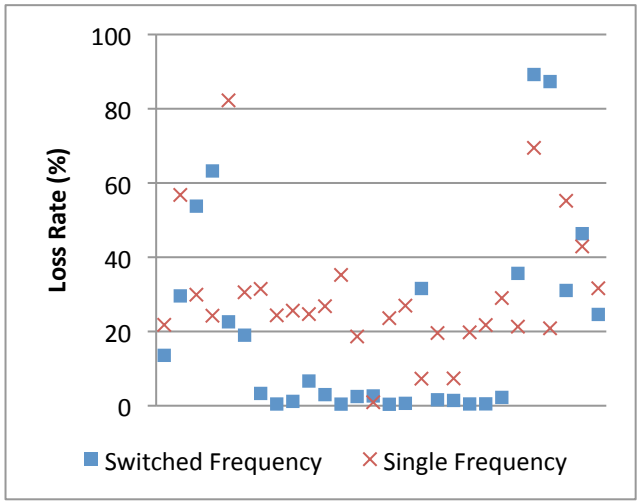


Figure 8 : The average packet loss probability through a network using the switched frequency algorithm versus a single frequency path.

7. SUMMARY

In this paper we have presented some results that have been obtained from a WiFi network that we have created. This network was created as part of the South Australian Network Laboratory (SANlab) so that experiments with real networks could take place. The centrally controlled frequency selection algorithm has been able to route the traffic through the network in a more efficient manner than just using a single frequency.

There are a number of future directions that this research can take. A closer analysis of our results could lead to a better understanding of some of the complex interactions taking place and hance a more efficient algorithm. A distributed algorithm could also be created. This is a more realistic situation for a MANET and would also mean that measurements and control information need not have to be transported over a possibly lossy network. Other PHY and MAC layer parameters may also be investigated, such as power, to see how they could be changed to better route traffic through this network. In all of this we can use the present software and hardware to ensure that the results work in a real network and not just in a simulation.

8. REFERENCES

[1] S. Naguleswaran and S. Perrau, Cross-Layer Planning and Optimization for Military Applications, *MILCIS 2009*.

[2] The OpenFlow Switch Consortium, http://www.openflowswitch.org.

[3] The Open Mesh networks consortium, http://www.open-mesh.org.

[4] D. Johnson, N. Ntlatlapa and C. Aichele, Simple Pragmatic approach to mesh routing using BATMAN, *2nd IFIP International Symposium on Wireless Communications and Information Technology in developing Countries,* Pretoria, 2008.

[5] M. Abolhasan, B. Hagelstein and JC-P. Wang, Real-world performance of current proactive multi-hop mesh protocols, *IEEE APC*, Shanghai, 2009.

[6] L. Barolli, M. Ikeda, G. DE Marco, A. Durresi and F. Xhafa, Performance Analysis of OLSR and BATMAN Protocols Considering Link Quality Parameters, *2009 Conference on Advanced Information Networking and Applications*, Bradford, 2009.

[7] W. Kiess and M. Mauve, A survey on real-world implementations of mobile ad-hoc networks, *Ad Hoc Networks*, vol. 5, no. 3, pp. 324-339, 2007.

[8] K-K. Yap, M. Kobayashi and D. Underhill, The Stanford OpenRoads Deployment, WiNTECH, Beijing, 2009.

[9] F. Alnajjar and Y. Chen, SNR/RP Aware Routing Algorithm: Cross-Layer Design for MANETS, *International Journal of Wireless & Mobile Networks*, Vol. 1, No. 2, November 2009.

[10] D-ITG, Distributed Internet Traffic Generator, http://www.grid.unina.it/software/ITG.

[11] A. Pirzada, M. Portmann, R. Wishart and J. Inulska, SafeMesh: A wireless mesh network routing protocol for incident area communications, Pervasive and Mobile Computing, 5, 2009, pp 201-221.

Dr Andrew Coyle is the Deputy Director of the Centre for Defence Communications Information and Networking at the University of Adelaide. He is currently the Director of the South Australian Networking Laboratory (SANlab). This laboratory has built up a suite of technologies that demonstrate wireless networking capability. Andrew.coyle@adelaide.edu.au

Dr Hung Nguyen is a research fellow at Centre for Defence Communications Information and Networking at the University of Adelaide. He holds a PhD in Computer and Communication Science from the Swiss Federal Institute of Technology-Lausanne (EPFL), Switzerland. His main research interest is on network measurement, network tomography and routing algorithms on wireless networks.