Optimised Scheduling for Wireless Mesh Networks using Fixed Cycle Times

Ian Cooper, Stuart Allen, Roger Whitaker
School of Computer Science and Informatics
Cardiff University
Wales, UK.
Email: i.m.cooper@cs.cf.ac.uk stuart.m.allen@cs.cf.ac.uk r.m.whitaker@cs.cf.ac.uk

***Abstract*—In this paper, we consider the optimisation of transmission schedules for infrastructure Wireless Mesh Networks in which data is forwarded through mesh routers from a single Internet Gateway node. The mesh routers receive and aggregate data from local mobile devices and each mesh router has an assigned data allowance to ensure fairness, set depending on its geographical position or the predicted usage patterns. We examine the use of fair and efficient link scheduling for Wireless Mesh Networks and provide an integer program for maximising the throughput allowance for each mesh router in a network given the topology. The program uses a slotted time approach to maximise the throughput within a given number of slots N, thus allowing the network to be split into sub networks for local access to the mesh routers, and back-haul transmissions to the gateway. Results are presented showing the optimised throughput for a selection of networks and a range of values for N.**

***Keywords*-Wireless Mesh Networks; Link Scheduling.**

I. Introduction

Infrastructure Wireless Mesh Networks (WMN) are used to increase the accessibility of wireless clients to either the Internet or other services on that WMN. One of their biggest advantages is the ease and cost effectiveness of their installation; this is due, in part, to the lack of requirement for cable placement between the wireless access points. WMNs use the wireless access points as mesh routers that receive data from wireless clients and forward that data on to the WMN's Internet Gateway. This increases the potential interference within the network, as the forwarded transmissions add to the quantity of data vying for access to the channel. In a standard 802.11 Wireless Mesh Network the interference between the transmissions is controlled by the Distributed Coordination Function (DCF) [1]. The DCF is the 802.11 equivalent of the carrier sense multiple access as used in wired networks, it requires all transmitting stations to listen to the channel before transmitting to see if there is currently any transmission in progress. It can also use request to send (RTS) and clear to send (CTS) packets to ensure that hidden nodes will not interfere with each other.

The DCF functionality is a highly decentralised method that works well at low throughput levels, but it becomes inefficient, and unfair as the throughput starts to reach maximum capacity. The inefficiency is created by the random back off when signals collide and the unfairness is caused by the decision as to which station is to transmit being largely random. This means that data from routers closer to the Gateway is more likely to reach its destination than from those on the edges of the network. This effect is demonstrated by Jun et al. [2] who calculate the nominal capacity of Wireless Mesh Networks. All stations are allocated an amount of data T to be sent to the Gateway; the routing of the network is fixed and so the amount of forwarded data that each station is required to transmit can also be determined. Jun et al. describe a collision domain as the set of links that interfere with a given link; this can be used to obtain a maximum value of T by finding the collision domain that collectively needs to transmit the largest amount of data across its links. For example, Fig. 1 shows a grid network, the links are grouped into collision domains for a number of links. In this figure it can be seen that the largest collision domain is for link (3,4), the total amount of data required to be transferred in this collision domain is $18T$. For this network, Jun et al.'s work says that each link must transmit at a rate of at least $18T$ bits per second. From this we can deduce the value of T as the link rate of the network is generally known; as an example, a given link rate of 1024Mbps gives a data allocation T of 56.889 Mbps.

In order to increase the efficiency and fairness of the WMN, link lcheduling can be used. Link scheduling allocates the times when specific links are allowed to transmit. Schedules can be arranged to avoid any transmission collisions and to allow each link to have its required allocation of transmission time according to its traffic demand. The benefits of link scheduling are that the network topology and the routing can be considered when allocating the link activity, the link schedules can also increase the amount of data transmitted through the network by strategically allowing certain links to transmit concurrently. The nominal capacity method [2] is a fast method of calculating an estimated throughput of a network, but it does not provide a link schedule that can achieve this rate, nor does it provide the optimum throughput that can be given by using a schedule. Work to improve on this has been widely undertaken.

Salem and Haubaux [3] show a heuristic method of obtaining a fair schedule, such that the mesh access points are allocated the required amount of data to service each of their clients in the same manner. Their solution deals

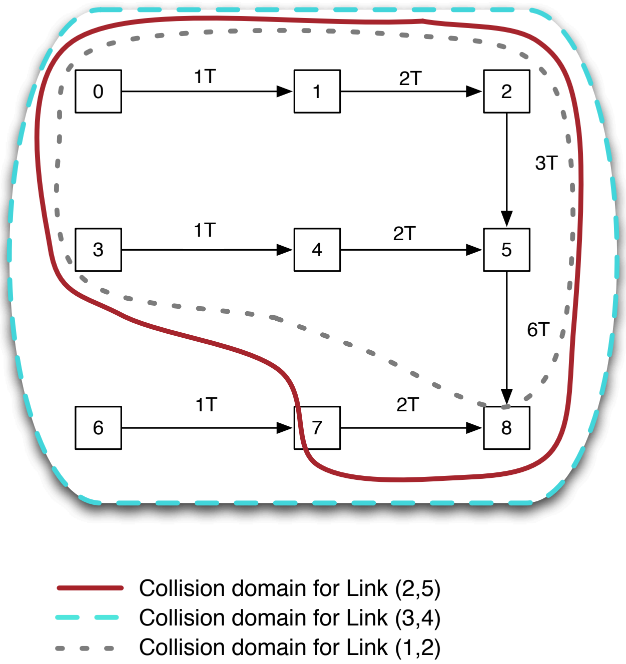

Figure 1. An example Network graph with collision domains as defined in Jun et al. [2]

with up and down stream traffic, but in doing so they allocate a separate channel for each direction, thus reducing the problem to a simpler unidirectional transfer problem. Salem and Haubaux use a compatibility matrix, which is a matrix of all the links that can be transmitting at the same time, the compatibility matrix can be thought of as an inverse representation of the collision domains. From this compatibility matrix, all possible cliques are constructed (a clique is a set of links that can be concurrently transmitting). Each clique is scored as to how much data can be transmitted and how much time it takes. These metrics are then used to decide which cliques are used subject to the constraint that each link is activated once and only once, and that each clique is activated one after the other. Salem and Haubaux relate their work to a scheduling that has no spatial reuse which effectively means that each link transmits one after the other. They show that by using the clique method, the total time for the cycle is reduced from the non spatial reuse method.

Guan and Zhu [4] outline a weighted vertex colouring problem where the vertices represent the links and the edges represent potential collisions. The main idea is to create vertex colour sets and then to chose the links in these colour sets based on minimising the sum of the maximum weights in each used colour set. This is of course subject to the constraints that each vertex is assigned to at least one colour set. Guan and Zhu use the solution to solve a bus network problem.

Malaguti et al. [5] provide 2 solutions to the weighted vertex colouring problem. The first solution is to use colour assignment variables and a cost variable. This cost is equal to the maximum cost of any link in the colour set. Similar to Guan and Zhu [4] the objective function is to minimise the cost of all colour classes used, where all links are included in at least one colour class. This is effectively the minimum cycle time. This first solution is very similar to Guan and Zhu, the novelty of this solution in its own right is the application to scheduling on a batch machine. The second model reduces the solution space by utilising the fact that there are many ways of ordering the same solution, the colours can be different for the sets and effectively it is the same solution. So if the colours are ordered, solutions can only be accepted if they have the colours for the sets in the order of the colour numeration. This reduces the time needed to optimise the solution.

Cicconetti et al. [6] present a power based scheduling model in order to guarantee fair bandwidth. The raw SINR is used to generate and derive the collision domains. As a performance metric, they consider several aspects, including: the end to end throughput of a traffic flow, the MAC layer throughput of a node (irrespective of its traffic flow), the end to end delay of the a packet between the sending node and the destination node, and finally the fairness of the schedule.

The delay of a packet from the source to its destination is also extensively covered in the literature. Badia et al. [7] provide a flow based linear program which monitors the inputs to, and the outputs from nodes to ensure that the data is transferred from the source to the destination within the cycle period. The metric used in this model is the number of slots per cycle, but due to the extra constraints on the destination of the data, this metric includes the delay between the source and the destination. Shetiya et al. [8] also look at the packet delays from arrival in source queue to the destination. They provide a centralised routing and scheduling scheme that first routes the network using shortest path algorithms, then schedules the packets based on the number of slots used. Sanchez et al. [9] use end to end packet delay to provide the scheduling in order to adapt the networks routing.

Previous works that provide integer programs for the production of link scheduling use graph based vertex colouring methods to create time divisions during which groups of links can transmit. In this paper we present an integer program that uses slotted time allocation. Our motivation for using a slotted approach is its ability to be more flexible in terms of link allocation; in the vertex colouring model, for a particular colour class, the links are set for the duration of that class. For the example in Fig. 2 we define the cycle time of the schedule as the period of time that the schedule needs to transfer the data requirements from all links in the network. The time slots are uniform time divisions used to transmit portions of data from concurrently transmitting links, when the timeslots are but together they form the

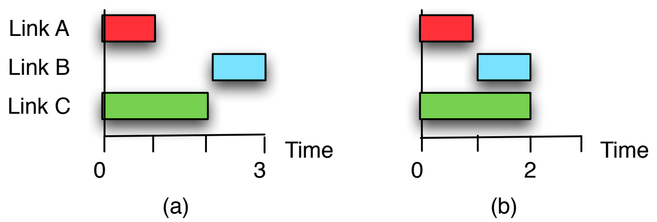

Figure 2. An example schedule. a) Using vertex colouring, b) Using slotted set allocation.

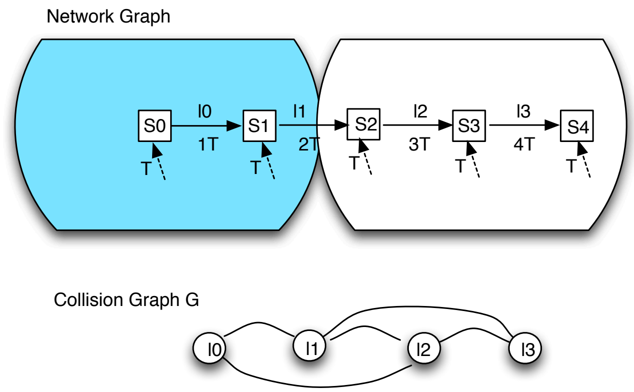

Figure 3. The transposition of a network graph to a collision graph.

schedule. Fig. 2 (a) shows a schedule for a simple network where link A interferes with link B, and link C interferes with neither link. In the schedule links A and C are in a colour class and link B is in a separate colour class. By looking at the schedule, it can be seen that there is a missed opportunity to combine link B with link C in the second third of the cycle. In the vertex colouring models this would not be permissible as the sets are fixed for their entire duration. The use of slotted time allocation allows this problem to be resolved; during each slot, a separate set of links can be used, and so as seen in Fig. 2 (b), the cycle time of the slotted schedule can be reduced.

We present an integer program formulation for the slotted approach, which maximises the allocated throughput allowance for each station when given a fixed number of slots. This formulation allows us to either acquire the maximum throughput for the network for a given number of slots per cycle, or to decide the optimum number of slots by cycling through the number of available slots.

II. Collision Model

Previous work in the area of Wireless Mesh Networks has considered ways of calculating the potential collisions occurring in Wireless Mesh Networks. There are two commonly used models described by Gupta et al. [10]; the Physical model, and the Protocol model. In the physical model, signal to interference and noise ratios (SNIR) are calculated for each link in the network, which can be time consuming for large networks. The protocol method uses interference ranges to assess whether a transmitting station is with in range of a receiving station. Most related work using an integer programming approach uses the physical method to calculate the interference within the network. Our model uses the protocol model for increased speed, because the collision domains of the network can be pre-computed (Shi et al. [11] provide extra guidance on signal propagation models to use in order to achieve more accuracy).

Our model of the collisions is defined as follows. Consider a Wireless Mesh Network that has a set of stations connected by wireless transmissions. In our model, data is passed from one station to another station in a point to point fashion and these transmission channels are referred to as the set of links $L = \{l_1, \ldots, l_n\}$. Due to the broadcast nature of wireless communication, a station transmitting data across a link will affect all stations within a specified range of that transmitting station. All affected stations will be unable to receive data reliably.

In a standard graphical representation of a network, the set of stations would be defined as the vertices of a graph and the set of links between those stations would be defined as the set of edges of the graph. Instead we define a collision graph, where the set of links L are the vertices and edges join those vertices whose links cannot concurrently transmit without collisions. That is, the set of links with receiving stations within the specified range of another link's transmitting station, are said to be in the interference range of that other link. For our collision model, the collision domain of a specified link is defined by the union of the set of links within interference range of that link and the set of links for which the specified link is in interference range. $E = \{e_1 \ldots e_m\}$ is the set of all interference edges in the collision graph as described above. Given a pre-defined routing for a Wireless Mesh Network, the network graph of stations and links, can be transposed to a collision graph of links and collisions (as shown in Fig. 3).

In Fig. 3's network graph, the dashed arrows to the stations represent the amount of data being injected into the network by the mobile nodes attached to that station, and the solid arrows between stations represent the links. The multiples of T associated with the links are the amount of data required to be transmitted across that link due to the data forwarding . $w_u T$ is the total amount of data that each station is required to transmit, for each link $u \in L$ (where L is the set of links, or edges, in the network graph) The two shaded areas show the interference range for the stations $S0$ (the darker shading) and $S3$ (the lighter shading), in this simplified example the stations are 100 meters apart, each station can successfully transmit for 100 meters (receive range) and the distance that a signal will cause interference within (interference range) is 150 meters. It can be seen from the figure that if link 3 was to be transmitting, station

$S3$ would cause interference at receiving stations $S2$ and $S3$, Station $S4$ is the station receiving on link 3. This interference causes link 2 and link 1 to be affected by link 3's transmission. Our collision graph is a non-directional graph $G = (L, E)$ where the collision domain for link u is defined as $N_G(u)$ which is the neighbourhood of u in the collision graph, ($Collision\ Domain\ of\ link\ u = N_G(u) \quad \forall u$). As an example from Fig. 3, $N_G(1) = \{0, 2, 3\}$.

III. Slotting Model: Maximise T using N Slots

We now present an integer program to maximise the allocated throughput for a routed graph. In this model the number of slots that can be used is predefined as the value N and we maximise the allocated throughput allowance T for each station. The IP is listed below, where $x_{u,i}$ are a set of binary decision variables (4), they represent the assignment of link u to slot i, w_u is the weighting of Slot u and r_u is the maximum data rate of link u.

$$\text{Max } T \tag{1}$$

$$\text{subject to: } x_{u,i} + x_{v,i} \leq 1 \quad \forall i, u, v \in N_G(u) \tag{2}$$

$$\sum_{1 \leq i \leq N} x_{u,i} \geq \frac{N\ w_u\ T}{r_u} \quad \forall u \tag{3}$$

$$x_{u,i} \in \{0, 1\} \tag{4}$$

The approach maximises the throughput T allocated to each client (1) whilst ensuring that no colliding links are transmitting at the same time (2) where the assignment of any Slot i can not be allocated to link v if it is also allocated to link u where u and v are in the same neighbourhood ($v \in N_G(u)$) This is applied to all values of v, u and i.

The value of T is constrained in (3), ensuring that each link receives at least its fair share of the allocatable time. The ratio of allocated time to available time $\sum_{1 \leq i \leq N} x_{u,i}/N$ is defined by the number of slots allocated to link u to the total number of slots allocated N, this must be at least equivalent to the required amount of data for a link compared to the maximum amount of data that link could send, ignoring interference, $(w_u T / r_u)$.

One of the key points with this method is that the cycle time is fixed by the predefined number of slots. If the maximum throughput value is required, all the values of N need to be examined in order to ascertain the most optimum result. As with the vertex colouring method, this provides a rudimentary link schedule that can be used at the MAC layer of wireless routers.

IV. Results

We evaluate the optimisation model through comparison with two existing models on published network topologies. The evaluation aims to assess the model's potential to provide an optimum schedule for a Wireless Mesh Network.

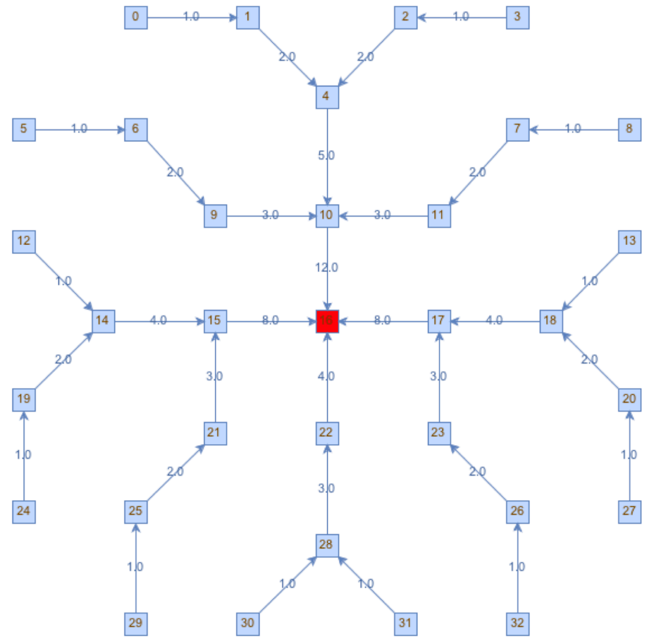

Figure 4. The routing and network configuration of the Grid network (unused links not shown).

In order to do this we use a number of standard network scenarios, as used in previous work. The chain network [2][3] is a one dimensional line of equidistant network stations, with a gateway station at one end. In our chain scenario, the stations are separated by a distance of 10 units.

The receive range of the stations is set to 11; the cTx distance (the interference range) for the transmissions can be calculated as 34 by using the TwoRayGround model [12]. For our results we use a Chain network of 5, 10, 15 and 20 stations. The chains are a simplified network configuration that are commonly used to illustrate wireless network interference. We also present results from a more realistic network configuration similar to the configuration used in Salem and Haubaux's paper [3] and illustrated in Fig. 4. The numbers associated with the links in this figure are the required traffic multiples carried by each link. For the grid network, we use configurations of 9, 17 and 25 stations, always using the innermost stations in the figure. In these configurations we use the same receive and interference ranges as in the Chain network configuration.

We also show nominal capacity results using Jun's [2] method and results from the Malaguti weighted vertex colouring model [5] in order to show a comparison to existing work. The resulting integer programs have been solved using the CPLEX [13] software suite.

The graphs in Fig. 5 show that for a grid network of 25 stations the maximum throughput of 1.174Mbps is obtained using 46 slots. In comparison the nominal throughput using Jun's calculations is 0.964Mbps and the throughput that

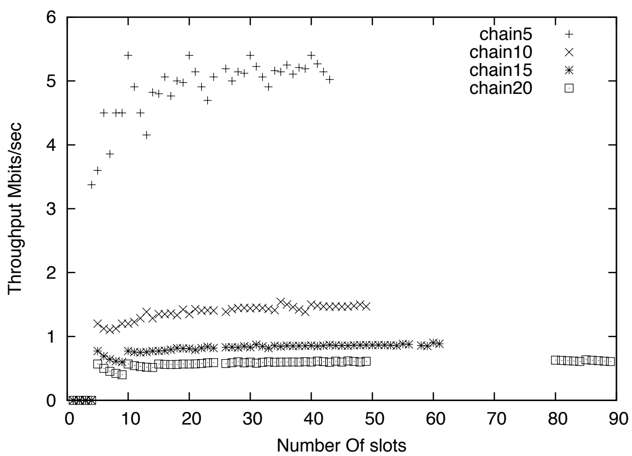

Figure 5. The throughputs for the IP approaches on the Chain network.

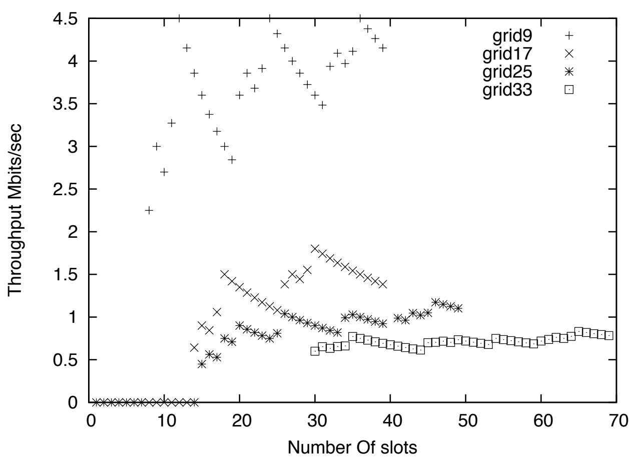

Figure 6. The throughputs for the IP approaches on the Grid network.

can be obtained using our implementation of Malaguti's weighted vertex colouring method is 1.125Mbps. These results are given as a comparison, the figure calculated for Jun's nominal capacity approximation [2] is given as a benchmark for systems that do not use scheduling and the throughput allocation provided by Malaguti's method [5] is given as an example of non slotted vertex colouring schedules. Table I shows the comparisons for the other networks, and that as the size of the more complex Grid network increases, the slotting approach becomes more optimal.

Due to the time taken to solve individual cases of the algorithms, we only present solutions in the close vicinity of the optimum solution, approximate results can be obtained quickly by using CPLEX's 'gap' functionality and extrapolating from a suboptimal solution. The results in Table II show approximate timings for a single run of each integer program The results given are to be treated as approximate as the execution time of the slotted maximum T program varies with the value of N. It can be seen that the algorithms

Table I
SUMMARY OF RESULTS AND COMPARISONS WITH EXISTING WORK.

Network	MaxTN No.Slots	MaxTN T	Jun T	Malaguti T
Chain 5	10	5.4	5.4	5.4
Chain 10	35	1.543	1.286	1.543
Chain 15	60	0.9	0.701	0.9
Chain 20	85	0.635	0.482	0.635
Grid 9	12	4.5	4.5	4.5
Grid 17	30	1.8	1.686	1.8
Grid 25	46	1.174	0.964	1.125
Grid 33	65	0.831	0.614	0.794

have the same order of execution time, although the slotted maximum T program has to be run multiple times to get the optimum throughput.

Table II
TIMES FOR ALGORITHMS ON VARIOUS SIZED NETWORKS.

Algorithm	10 station (ms)	15 station (ms)	20 station (ms)	25 station (ms)
SlotMaxT_ N	92	214	163	370
Malaguti	58	303	414	816

V. CONCLUSION

In this paper we have contrasted an integer program scheduler that uses fixed size slotted time divisions, and a vertex colouring integer program using fixed proportions of the cycle time for each link in a colour class. We conclude that the approach using fixed size slotted time divisions achieves better throughput results than the vertex colouring solution for networks that are complex (at least 25 stations arranged in a 2 dimensional topology) and that it produces no worse solutions for simpler networks. The use of an IP that relies on a given value for the number of slots, and hence a given cycle time has its uses for scheduling sub networks into larger schedules. The methods and results that are given in this paper are optimum results for networks using a slotted time allocation and can be used to validate future, faster integer programs and heuristics that do not require the multiple values of N to be explored. As future work we are developing an integer program formulation that minimises the cycle time of a schedule and also a heuristic implementation of the slotted time allocation scheduler.

ACKNOWLEDGMENT

This research project "Integrated Scheduling for Wireless Mesh Networks" was funded by the EPSRC, grant number EP/G036454/1. The authors would like to thank our project partners at Cass Business School, London: Prof Celia. A. Glass and Dr Eun-Seok Kim, for their periodic views on the progressing work.

REFERENCES

[1] M. S. Gast, *802.11 Wireless Networks*. O'Reilly, 2002.

[2] J. Jun and M. Sichitiu, "The nominal Capacity Of Wirless Mesh Networks," *Ieee Wireless Communications*, no. October, pp. 8–14, 2003.

[3] N. B. Salem and J.-P. Hubaux, "A fair scheduling for wireless mesh networks," in *Proc. of WiMesh*. Citeseer, 2005.

[4] D. J. Guan and X. Zhu, "A coloring problem for weighted graphs," *Information Processing Letters*, vol. 61, no. 2, pp. 77–81, 1997.

[5] E. Malaguti, M. Monaci, and P. Toth, "Models and heuristic algorithms for a weighted vertex coloring problem," *Journal of Heuristics*, vol. 15, no. 5, pp. 503–526, 2009.

[6] C. Cicconetti, I. F. Akyildiz, and L. Lenzini, "Bandwidth Balancing in Multi-Channel IEEE 802.16 Wireless Mesh Networks," *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pp. 2108–2116, May 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4215826

[7] L. Badia, A. Botta, and L. Lenzini, "A genetic approach to joint routing and link scheduling for wireless mesh networks," *Ad Hoc Networks*, vol. 7, no. 4, pp. 654–664, Jun. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2008.04.005

[8] H. Shetiya and V. Sharma, "Algorithms for routing and centralized scheduling to provide QoS in IEEE 802.16 mesh networks," in *Proceedings of the 1st ACM workshop on Wireless multimedia networking and performance modeling*. ACM, 2005, p. 149.

[9] M. Sánchez, J. Zander, and T. Giles, "Combined routing & scheduling for spatial tdma in multihop ad hoc networks," *Wireless Personal Multimedia Communications (WPMC)*, 2002.

[10] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[11] Y. Shi, Y. Hou, and J. Liu, "How to correctly use the protocol interference model for multi-hop wireless networks," in *The 28th Conference on Computer Communications IEEE*, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1530782\&dl=ACM

[12] T. Rappaport, *Wireless Communications: Principles and Practice*, 1st ed. Prentice Hall, 1999.

[13] IBM, "Cplex." [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/