Optimising Multi-Rate Link Scheduling for Wireless Mesh Networks

Stuart M Allen, Ian M Cooper, Roger M Whitaker

School of Computer Science and Informatics, Cardiff University, Wales UK.

Abstract

Traffic in an infrastructure wireless mesh network is routed over multiple hops between clients and gateways, hence performance can be significantly reduced where links interfere with each other. In this paper we consider the problem of optimising link scheduling for wireless mesh networks, making a number of contributions. Adopting a protocol-based model, we introduce an integer programming approach for an optimised schedule using a time-slot model. This model compares favourably against previously published methods and we introduce a rapid heuristic approximation that can present near-optimal solutions in a fraction of the time. We show that taking into consideration the affect of varying data rates across individual links during different time slots can further enhance the throughput achieved. This decreases the local data rate on some links but concurrently reduces the interference range of the transmitted signal which increases spatial reuse across the network. We present efficient heuristics to rapidly find near-optimal solutions to an integer programming model of this problem and provide rigorous justification on benchmark problems.

Keywords: Wireless Mesh Networks, Scheduling, Optimisation

1. Introduction

Wireless Mesh Networks (WMNs) represent a technology that economically increases the geographical area within which mobile clients may access broadband communication. Mesh routers facilitate multi hop wireless transmission to relay data over extended distances without need for the cost, delay and disruption of installing cabled access points.

In this paper we consider infrastructure WMNs, which aim to provide internet access from a single gateway to a number of mesh routers (which may be clients themselves or support a number of local clients). We assume 802.11b/g/n is used for wireless connectivity, where control of the transmissions is governed by the Distributed Coordination Function (DCF) [1]. The DCF ensures that each router avoids transmitting data in the presence of another mesh routers interfering transmission. This is achieved with a combination of first-come first-served access and the use of back off periods. This method, whilst highly decentralised and scalable with respect to the number of mesh routers in the network, starts to become inefficient when the network is heavily loaded. Furthermore, the use of DCF within wireless mesh networks means that the decision on which link is allowed to transmit is largely random in nature, giving no guarantee of fair access for mesh routers which may compromise quality of service. DCF is effectively giving equal forwarding priority to all mesh routers. However these networks are predominantly acting as a back-haul network to an internet gateway and therefore some mesh routers, particularly those close to a gateway, will be required to forward their neighbour's data in addition to their own traffic [2].

To get high performance throughput utilisation from a wireless mesh network it is possible to adopt a schedule for communication via the medium access control (MAC) layer that allows certain router pairs to transmit and receive data at specified times. This imposed scheduling of the communication activity is commonly referred to as *link scheduling*. The overall benefit of this approach is that optimal link activity can be coordinated via the schedule taking into account a given network's topology. Determining link schedules is a complex task with many possible degrees of freedom for modelling. As such there is much current work seeking to find optimal methods [3, 4, 5]

Consistent with previous literature, we start by modelling WMN scheduling in terms of graph colouring and propose an initial model for reference which is compared with previous work. This initial graph uses weighted vertices to represent the network links and the relative amount of data that those links are required to carry. Graph edges are used to represent the potential for interference between links. In Section 3 we present an integer program for vertex colouring to maximise the throughput that can be fairly allocated to each mesh router. We show that further improvements in throughput are possible by imposing discrete time slots on the model by developing further integer programs. Given the high performance of the slotted approach and the significant time required to determine solve the integer program, we introduce a fast heuristic approximation that presents near optimal solutions in a fraction of the time. As well as determining individual schedules, these fast heuristics are also suitable to be used as objective functions in other optimisation problems, such as routing.

Note that we do not address the practical issues of implementing a scheduled transmission framework in this paper as issues such as synchronisation of network nodes, and dissemination of scheduling information, are discussed in other papers. For example, Soft-TDMAC [6] presents a TDMA scheduling protocol which is run over 802.11 commodity hardware. This layer provides synchronisation between network nodes, sending scheduling information in dedicated time slots. Our work would only require slight modificationto fit into this or similar models (e.g. [7]).

In Section 4 we further enhance the time allocation concept to take into consideration the effect of varying data rates across individual communication links used during different time slots. The power of the signal is adjusted to keep the transmission range constant thereby eliminating the need to recalculate the network routing. This potential reduction in the data rate has the effect of decreasing the amount of data transmitted along an individual link, but concurrently it reduces the interference range of the transmitted signal which improves spatial reuse across the network. These effects are shown to combine to provide an overall benefit to the network's performance. Integer programming is used to formally model this as an optimisation problem and we provide a further efficient heuristic to achieve a near optimal solution with fractional computational effort. The fast heuristic models could then also be used in the evaluation of routing and deployment decision making schemes such as [9, 10]. Throughout we provide rigorous comparisons between our work and with others from the literature.

2. Related Work

Establishing the capacity of a wireless mesh network is very useful in network evaluation. A method to calculate the nominal capacity of a wireless mesh network is detailed by the work of Jun and Sichitiu [2]. In Jun and Sichitiu's paper, each mesh router is allowed to contribute a maximum allocation of data, T bits per second. In this model, all mesh routers are sending data destined for the gateway. This requires that other mesh routers which are on a pre-defined route to the gateway, retransmit that data in addition to their own data allocation. A dimensionless multiplier, w_u, is applied for each link $u \in L$ (where L is the set of wireless links in the network). Each w_u allows allocations to be defined in terms of T. The amount of data that each link is required to pass on to the next mesh router in the network is determined by the pre-defined route. Interference is modelled through the definition of a *collision domain* for each link $u \in L$, consisting of all other links that would interfere with u if active at the same time. Given the link rate r of each link $u \in L$ in the network, Jun and Sichitiu's work can give the maximum value of T by finding the collision domain that collectively needs to transmit the largest amount of data across its links [2].

Figure 1 shows a network with collision domains outlined for a subset of the links. The maximum value of T is determined by the collision domain for link $(3, 4)$ which must carry traffic of $18T$. Since the maximum data rate that the link can support is r, this gives $T = r/18$ bits per second. This capacity can only be achieved if there is no overhead caused by the contention.

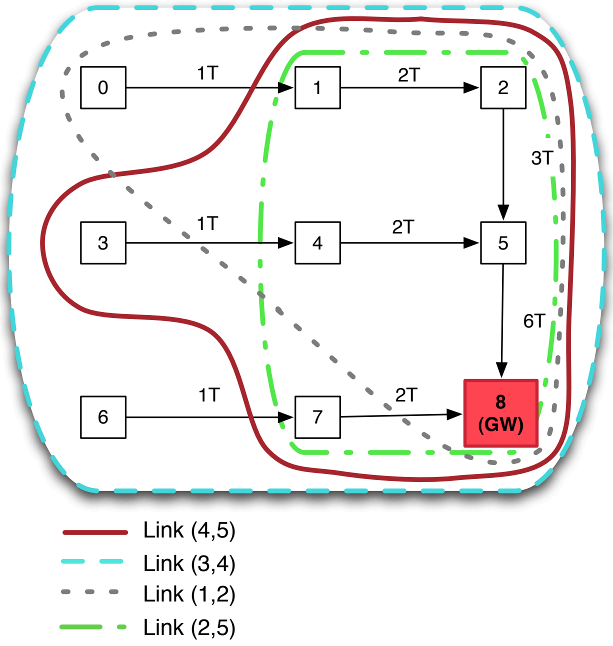

Figure 1: An example network with collision domains as defined in Jun and Sichitiu [2]

Whilst Jun and Sichitiu's method of calculating the nominal throughput is a fast method it does not produce a schedule for link activation, and also it does not provide an optimum solution to the maximum throughput. Link scheduling can lend toward this bound by the careful management of time allocation for link activations. Diverse work on providing link schedules for wireless mesh networks is already in existence, with each method having particular contributions [3, 4, 5, 11].

The aim of link scheduling is to maximise the throughput that can be offered to each node by defining specific times where each link may be active. At each point in time there may be more than one link active as long as no concurrently active links are within interference range of each other. The advantage of link scheduling is that it can increase the potential throughput of the wireless mesh network. Link scheduling is an *NP*-hard problem [12] and accordingly this has led to a range of approaches and heuristics that attempt to provide high performance solutions for network throughput. Frequently this involves applying surrogate objective functions whose optimisation implies desirable characteristics in the resultant schedule [12].

A well cited example is Salem et al. [3], who show a method of obtaining a fair schedule which ensures that the mesh access points are allocated sufficient capacity to service each of their clients in the same manner. The solution presented deals with up and down stream traffic, but in doing so they allocate a separate channel for each direction, thus reducing the problem to two copies of the unidirectional transfer problem that is commonly addressed. A so-called compatibility matrix is used, which is a matrix of all the links that can be transmitting at the same time. From this compatibility matrix, all possible cliques within the equivalent graph are constructed and a 'gain' metric is applied to assess and choose which cliques to use. Effectively this works out the number of slots saved by using the cliques, rather than transmitting the data over one link after another. They show that by using the clique method, the total time for the cycle is reduced from the non-spatial reuse method. A further approach is that of Guan and Zhu [13] who use a weighted vertex colouring technique on a graph $G = (V, E)$ where V is the set of links and E is the set of collision domains. This is formulated as an integer program (IP) with a specific objective function to minimise the sum of the 'maximum weight in each of the colour sets' across all colour sets. Guan and Zhu use the solution to solve a bus network problem.

Malaguti et al. [12] provide a similar solution to Guan and Zhu [13] with the novelty of this solution being applied to scheduling on a batch machine. Their objective function minimises the cost of all colour classes used (effectively the minimum cycle time), where all links are included in at least one colour class. An additional model reduces the solution space by utilising the fact that there are many ways of permuting colours to achieve an equivalently structured solution. Fredrikos et al. [11] extend the problem to include multirate links per time slot. This has the effect of altering the collision domains as well as the throughput of the links. The objective used in this paper is a combination of the throughput and the power usage associated with the different rate powers.

Cicconetti et al. [14] present a power based scheduling model in order to guarantee fair bandwidth, where they use the raw signal-to-noise ratios to generate 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 examined in the literature [15, 16, 17]; this is an well researched alternative to using throughput as a scheduling metric.

There is also considerable related work concerning the use of power and data rates to improve the optimisation of wireless mesh network throughput. The variation of power in the transmissions can alter the range of both the receivable signal and the interference range of the signal. Both Lou et al. [4] and Macedo et al. [18] use the power and rate adaptation to alter the routes available to the network and therefore provide a joint routing and scheduling solution, although Lou et al. state that "multipath optimal routing is not much more efficient than single path optimal routing" [4]. Related work [4, 18, 19, 11, 20] uses a physical model to calculate the interference domains for power controlled scheduling within wireless mesh networks. However we are aware of no existing work that studies a protocol model in this context, although Avallone et al. [19] mention that this could be achievable.

2.1. Our Contribution

The study in this paper makes a number of contributions and tackles some specific issues that have not been previously addressed. In particular:

- We introduce a time-slot model and to improve throughput we assess the effect of optimising the minimum cycle length of slots subject to providing a fair allocation. This is expressed as an integer program assuming a uniform single transmission rate on each link (Sec. 3.2.2). To the best of our knowledge has not been previously presented in the literature and it is shown capable of improving upon the performance of other published techniques;
- We develop a fast heuristic algorithm for the above problem based on ranking links in terms of their need for time slots. This is shown capable of achieving near optimal performance for a range of test cases (Sec. 3.3);
- We extend the single rate model to include the selection of a transmission rate for each link in each time slot, and present this as an integer program (Sec. 4.2). To the best of our knowledge has not been previously tackled in existing literature. We determine the effect of introducing a choice of transmission rate for assignment to each link by comparing it to our single transmission rate model;
- We introduce a fast heuristic algorithm to prioritise link assignment for the above problem and achieve near optimal performance for a range of test cases (Sec. 4.3);

Throughout we adopt a protocol model for transmission and assume that an a-priori routing is determined, enabling any routing approach to be evaluated against its scheduling performance.

3. Single Data Rate Networks

We consider a WMN that has a set of mesh routers connected by wireless links where a single internet gateway is the destination or source for all traffic. Data is passed from one mesh router to another mesh router in a point to point fashion, provided they are within a defined transmission range. We assume that a-priori, a routing algorithm is given that determines to which mesh router data is to be forwarded. The links over which data is transmitted are referred to as the set of links $L = \{l_1, \ldots, l_n\}$. We assume that the rate at which transmissions occur over a link is to be maintained at a constant value. The transmission schedule for the network indicates which links simultaneously transmit and at which times.

Consistent with previous research in this area (e.g., [5, 3]), the model we use to represent data concerns a buffered average data rate. This implies that the client provides data, at a specified rate, which is then stored in a buffer large enough to ensure that the data is always available for retransmission.

In our model each mesh router is given a *data allocation* which is the rate at which it introduces *new* data to the network. For each mesh router, this data allocation is expressed in multiples of a constant T (which may represent a single client or the aggregation of data from a number of local mobile clients). The aim of our problem is to maximise T, that is, the network throughput is to be maximised while ensuring the traffic offered to each node is scaled identically relative to their requirement.

A mesh router's *traffic allocation* $w_u T$ is the data that it is required to transmit, across link u, to the next mesh router on route to the gateway, where w_u is referred to as the weight of the link u. Without loss of generality, these values are scaled such that $\min_{u \in L} w_u = 1$. The mesh router's traffic allocation is a combination of its data allocation and any received traffic allocation from neighbouring mesh routers in the network.

A link schedule consists of a set of allocated time intervals during each of which a specified subset of links are active in transferring data between mesh routers. A link schedule can be optimised by making the DCF's local choices for link activity such that spatial reuse is taken into account for interference mitigation. Within wireless mesh network research there are two commonly defined approaches for considering an interfering signal, namely the physical and protocol models [21]. The physical interference model, which is used in work such as [22] calculates the required signal power and the power of all other interfering signals to ensure that the signal to noise ratio is above the required threshold. Alternatively the protocol model uses pre-computed transmission and interference ranges to determine whether one link is within interference range of another link [23].

There are many subtle variants of the protocol model; two of the main differences are illustrated by Badia et al. [24] in their 11protocol, to be used in IEEE 802.11 style networks, and 16protocol, to be used in IEEE 802.16 networks. The 11protocol requires that a link $l \in L$ in the network graph is bi-directional such that $(u, v) \in L$ if and only if $(v, u) \in L$. The reason for this is that the 802.11 MAC requires flow control packets to be returned from the receiver to the transmitter. This requirement is relaxed in the 16protocol due to the 802.16 MAC providing an orthogonal channel for the flow control packets to use. Using the protocol model, the incidences of interference within a network can be stored in a boolean matrix to indicate whether there would be a collision between two links if they were to transmit concurrently. In our single data rate solutions we use the 16protocol to model the interference, although our techniques could easily be adopted for the physical model. We also note in the multiple data rate section that simple additions could be adopted to allow our solutions to be used in 802.11 based networks. In this section we present three formal integer programming (IP) models for constant transmission rate link scheduling. These are:

- a vertex colouring IP (denoted *VC*) which is simple to state but does not model all possible valid schedules, Sec. 3.2.1;
- a time slotted IP model with a fixed number of time slots for which the throughput T is maximised directly (denoted *MaxT*), Sec. 3.2.3;
- a time slotted IP model that provides an approximate result which minimises the number of time slots (*MinN*), Sec. 3.2.4.

Due to the number of variables and constraints involved, these integer programming models take a significant time to solve even for modest problems, hence we also present a heuristic algorithm (Sec. 3.3) that provides near optimal solutions in acceptable time.

3.1. Collision Model

The transmission of data by a mesh router m will affect all mesh routers within a specified interference range of m. We say that a link l_t interferes with link l_r if the receiving mesh router from l_r is within interference range of the transmitting mesh router from l_t. The *collision domain* of a link l_i is defined as the set of links that are interfered with by l_i together with the set of links that interfere with l_i. For the single data rate model where we assume all links have the same data rate, we define a collision graph $G = (L, E)$ with a set of vertices $L = \{l_1, \ldots, l_n\}$ representing the links in the network and a set of edges $E = \{e_1, \ldots, e_m\}$ representing potential collisions between two links if they were to concurrently transmit. The collision domain for link u is defined as $\mathcal{N}_G(u)$, which is the neighbourhood of u in the single rate collision graph, as an example from Figure 2, $\mathcal{N}_G(l_0) = \{l_1, l_2\}$. Let r be the data rate that each link can support.

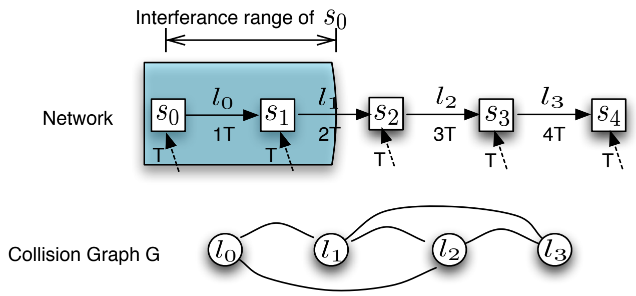

Figure 2: The transposition of a network graph to a single rate collision graph.

3.2. Integer Programming models

In this section we present three integer programming formulations of the link scheduling problem. The first is based on vertex colouring, as used in [13] and [12]. We then show how this model can be improved to give higher throughput by adopting a time-slotted approach. However, since this improvement comes at a higher computational expense, we provide a final simplified IP that gives good approximate solutions.

3.2.1. Vertex Colouring

To avoid collisions, only non-adjacent communication links can transmit simultaneously. Graph colouring allows us to conveniently model concurrent transmissions since all members of an independent set (or colour class) in a single rate collision graph can transmit concurrently.

The aim of the vertex colouring IP model is to find a collection of independent sets of links that maximise the data allocation T (1). This problem is analogous to the weighted vertex colouring problem in a graph, as addressed in [13] and [12]. The model is expressed here using the concept of duty cycles, where the duty cycle d_u of a link $u \in L$ is the proportion of time that u is scheduled to be active. The duty cycle of a link must be at least as large as the ratio of its traffic allocation to its data rate.

The model allocates colours to links and each link in a colour set receives the same duty cycle (whether or not it needs the whole time for transmission). We assign a set of colours $C = \{c_1, \ldots, c_{\Delta+1}\}$ where Δ is the maximum degree in the collision graph. Note that $\Delta+1$ colours are guaranteed to be sufficient by Vizing's theorem [25]. $x_{u,c}$ is a binary decision variable for the assignment of link u to colour c (constraint 7). We define the duty cycle D_c of a colour class c to be the maximum duty cycle over all links assigned colour c (4) and (5), and ensure that these duty cycles are able to be scheduled disjointly (6). Each link is assigned exactly one colour (8), and no two links in the same collision domain can be assigned the same colour (9).

We define the IP model *VC* below:

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

$$\text{subject to: } d_u \geq \frac{w_u}{r} T \qquad \forall u \in L \tag{2}$$

$$d_u \in [0,1] \qquad \forall u \in L \tag{3}$$

$$D_c \geq d_u + x_{u,c} - 1 \qquad \forall c, u \in L \tag{4}$$

$$D_c \in [0,1] \qquad \forall u \in L \tag{5}$$

$$\sum_{c \in C} D_c \leq 1 \tag{6}$$

$$x_{u,c} \in \{0,1\} \tag{7}$$

$$\sum_{c \in C} x_{u,c} = 1 \qquad \forall u \in L \tag{8}$$

$$x_{u,c} + x_{v,c} \leq 1 \qquad \forall c \in C; u \in L, \; v \in \mathcal{N}_G(u) \tag{9}$$

$$c \in C = \{c_1, \ldots, c_{\Delta+1}\} \tag{10}$$

where:

$$x_{u,c} = \begin{cases} 1 & \text{if } u \text{ is assigned colour } c \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Note that a link schedule can be obtained by ordering each of the colour classes sequentially in time, where all links in a given colour class can start transmitting simultaneously.

3.2.2. Slotted Models

An implicit assumption of the vertex colouring model is that all links in a colour class transmit simultaneously. In practice, it may be possible to achieve higher throughput by considering the duty cycle of the individual links in a colour class rather than just the maximum duty cycle. This is illustrated in Figure 3, where links A and B interfere with each other, but link C has no interference with either of the other links. In the first schedule (a), the vertex colouring model is used; the colour sets are fixed for the duration of the duty cycle, and so the schedule allows link A and C to transmit within the first two-thirds of the cycle, leaving link B to transmit during the final third of the cycle. In Figure 3(b) a slotted model is used, the two slots required contain different colour classes, but both contain link C. This simple example shows that it would be more efficient to divide the time into slots and allocate independent colour sets for each of those slots.

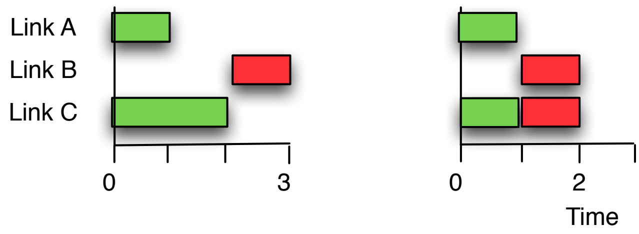

Figure 3: An example schedule: a) Using vertex colouring, b) Using slotted set allocation.

We present two IP formulations for slotted approaches, which differ in their objectives. The first maximises T directly when given a fixed number of slots. This formulation allows us to acquire the maximum (fair) throughput for the network by solving the IP for each possible number of slots. The second IP formulation minimises the number of slots used in the cycle while ensuring that the transmission allocations for each link are satisfied. This method minimises the cycle time in order to maximise the throughput, a technique also used in [12].

3.2.3. Maximise T using fixed slots

In our first slotted model (*MaxT*) [26] the objective is to maximise the data allocation T (12). The model assigns each link u to a number of slots, where $x_{u,i}$ is the decision variable for the assignment of link u to slot i, and N is the number of slots in the cycle. The formulation is as follows:

$$\text{Max } T \tag{12}$$

subject to:

$$x_{u,i} + x_{v,i} \leq 1 \qquad \forall i \in \{1, \ldots, N\}, u \in L, \; v \in \mathcal{N}_G(u) \tag{13}$$

$$\sum_{i=1}^{N} x_{u,i} \geq \frac{N \, w_u \, T}{r} \qquad \forall u \in L \tag{14}$$

$$x_{u,i} \in \{0,1\} \qquad \forall i \in \{1, \ldots, N\}, u \in L \tag{15}$$

The approach maximises the data allocation T assigned to each mesh router whilst ensuring that no colliding links are transmitting at the same time (13) in the same manner as the vertex colouring approach. The value of T is contsrained by (14), an adaptation of constraint (2) from the vertex colouring model *VC* by noting that the duty cycle d_u is equivalent to the ratio of slots allocated to the total number of slots, $\sum_{i=1}^{N} x_{u,i}/N$.

A key feature of this model is that the cycle time (length of the complete schedule) is fixed by the predefined number of slots N. Thus to acquire the optimum throughput, each possible value for the number of slots should be checked. It is worth noting that if the value given for N is too small, there will only be a trivial solution for the IP, with $T = 0$.

3.2.4. Minimum Slots

Obtaining the optimal throughput T using *MaxT* will be impractical for all but the smallest problems, as it requires multiple solutions of an integer program (since if the number of slots were a decision variable, constraint (14) would become non-linear). To alleviate this, the second of our slotted models (*MinN*) aims to obtain near optimal, feasible solutions. That is, where T may not be maximal, but there is guaranteed to be no interference within the schedule. This is achieved by using a different objective (16) to minimise N', the number of slots used. Links are allocated to slots by using constraint (21) 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. The IP *MinN* is given below.

$$\begin{aligned}
&Min\ N' && && (16)\\
\text{subject to: } & y_i \geq x_{u,i} && \forall u \in L, i \in \{1,\ldots,N\} && (17)\\
& y_i \in \{0,1\} && i \in \{1,\ldots,N\} && (18)\\
& w_u \leq \sum_{i=1}^{N} x_{u,i} && \forall u \in L && (19)\\
& N' \geq \sum_{i=1}^{N} y_i && && (20)\\
& x_{u,i} + x_{v,i} \leq 1 && \forall i \in \{1,\ldots,N\}, u \in L, && (21)\\
& && v \in \mathcal{N}_G(u) && \\
& x_{u,i} \in \{0,1\} && && (22)
\end{aligned}$$

N is chosen to be a large upper bound on the number of slots used. Variables y_i indicate the use of slot i (17), hence the actual number of slots used, N', is given by (20). Constraint (19) ensures that each link is allocated at least its required number of slots. The value of T can be computed post optimisation by:

$$T = \min_{u \in L} \left(\frac{r}{w_u} \sum_{i \in N} \frac{x_{u,i}}{N'} \right)$$

3.3. Heuristic Scheduling Algorithm

Solutions to the IPs provide optimum solutions for the given models, but as shown in the results section they are very slow for larger networks. This motivates heuristic approaches to find feasible schedules in acceptable time that are closely approximate to the best solution that can be obtained from *MaxT* or *MinN*. We develop a sequential and constructive heuristic algorithm, SDR1 (shown in Algorithm 1). SDR1 builds up a schedule on a slot by slot basis, by prioritising the links to add to each slot. A link can only be added to the current slot if it can not interfere with previously added links in this slot.

In order to assign a priority to links, the heuristic uses the term *satisfaction* to define a quantifiable value indicating a link's current fitness with respect to the number of slots assigned so far. This *satisfaction* metric is calculated by

$$\frac{|S_u|}{w_u s}$$

Algorithm 1: Single Data Rate Heuristic: HSR(N)

```
s = 0
S_u = {}   ∀u ∈ L

while s < N do
    s++
    # Sort by decreasing need of extra slots
    Create list P of links ordered by increasing |S_u|/(w_u s)
    I(s) = {P[1]}
    foreach j ∈ {2, ..., n} do
        # can link be added?
        if P[j] ∉ ∪_{v∈I(s)} N_G(v) then
            I(s) = I(s) ∪ {P[j]}
    foreach v ∈ I do
        S_v = S_v ∪ {s}
```

where s is the slot that is currently being allocated (starting at slot 1), S_u is the set of slots currently assigned to link u, and w_u, (the link's weight) is the number of slots that the link u requires. Note that $\frac{|S_u|}{s}$ represents the current duty cycle of a link, and so the satisfaction can be thought of as a measure of the duty cycle shared across the traffic load on a link.

In the heuristic we use the notation $I(s)$ to refer to an independent set of links in the single rate collision graph which are allocated to slot s. For each slot, the satisfaction of the links is computed and the least satisfied link is chosen as $P[1]$ in the list P of links. The links are then examined in order of their satisfaction to find the next link that is not in the collision domain of any of the slot's previously assigned links $P[u] \notin \cup_{v \in I(s)} \mathcal{N}_G(v)$. As suitable links are discovered, they are then greedily added to the slot's link set $I(s)$. This process continues until all the links have been examined. The heuristic continues until N slots have been assigned. The number of slots to be used is then chosen to maximise the throughput T. Formally, let $T(s)$ denote the value of T when the first s slots of this schedule are used:

$$T(s) = \min_{u \in L} \left(\frac{r}{w_u} \cdot \frac{|S_u \cap \{1,\ldots,s\}|}{s} \right)$$

for all $s \in \{1,\ldots,N\}$. Choose $N' \in \{1,\ldots,N\}$ to be the smallest value such that:

$$T(N') = \max_{s \in \{1,\ldots,N\}} T(s)$$

The corresponding link schedule is the defined by the sets

$$S_u \cap \{1,\ldots,N'\}$$

for each $u \in L$.

3.4. Results

To assess the solutions obtained from the integer programs and heuristics, we adopt a range of test problem scenarios and benchmark our findings against three existing models, namely: Jun and Sichitiu's nominal capacity method [2], the heuristic model in Salem and Haubaux's paper [3] and Malaguti's vertex colouring IP [12]. The Jun and Sichitiu's nominal capacity method [2] is included because it shows the results that can be obtained very quickly by deterministic methods (note this value is only a guide and can not necessarily be achieved by a link schedule). Salem and Haubaux's model [3] is included as it uses a heuristic method to find solutions to simple networks, and Malaguti's weighted vertex colouring model [12] is shown as it is the closest comparable model to the vertex colouring IP. These existing models give a broad range of the quality of available solutions and the speed that they can be run. We note that the heuristic used in Salem [3] requires pre-defined independent sets of links, each with a cost function, for which they cite the complexity of the of this clique enumeration problem as being NP-hard; this computation is included in our timings. Since all the models require collision matrices, the calculation of these matrices are omitted from the timings. Throughout all the evaluations we assume a nominal link capacity, consistent with previous literature [2, 3, 12], but accept that this is a simplification due to the affect of control packets and external interference.

We use three problem scenarios, the first being the commonly used "Chain Network" [2, 3], which is a one dimensional line of equidistant mesh routers, with a gateway at one end. We use chains of 5, 10, 15, 20 and 25 mesh routers. In our chain scenarios, the mesh routers are separated by a distance of 100 meters, the transmission range of the mesh routers is set to 110 meters. And Table 1 shows the values for the interference range at the different rates used, calculated using the standard Two Ray Ground model [27]. All non gateway mesh routers are allocated T Mbps of data to be transmitted on the network. We also present results from a second, more realistic grid scenario, similar to the configuration used in Salem and Haubaux's paper [3]. This is illustrated in Figure 4, where the numbers associated with the links in this figure are the required throughput weights. We use grids of 9, 17, 25 and 33 mesh routers. These configurations use the same transmission and interference ranges as in the chain network configurations. In order to examine each model's performance on a less regular network, we present results from a third problem scenario, a collection of randomly generated networks, each consisting of 20 nodes. The random networks are generated by sequentially adding nodes. Randomly located nodes are generated and if they are within transmission range of current nodes in the network they are added, if not they are discarded. This process is repeated until the required number of nodes has been allocated. In all cases traffic is assumed to be routed along a fixed predefined path to the gateway with minimal hops. In the chain and grid networks the link data rate r is set to 54 Mbps, and the resultant throughput T assumes no packet loss or collisions from external networks. For the random networks the data rate used is 18Mb/s. This scenario simplifies the use of the wireless mesh network to a single route, back haul network that use point to point communication only. This will not allow for peer to peer communication within the network. This scenario is consistent with previous literature [2, 12, 3].

The results that we present include the two IP models that are described in Section 3 as well as three existing solutions which are discussed in the related work section. For the *MinN* IP, the input parameter N was set to $|L|^2$. The third model, *MaxT* (Section 3.2.3) depends primarily on the input N that it is given to work with, however, if it is given the number of slots calculated by our *MinN* model, the IP will give the same solution. The IP models have been solved using the CPLEX [28] software suite. In this paper (unless indicated) all the models run on CPLEX have run to completion and therefore give optimum results for the IPs presented.

Table 1: Range and Power values for the different Data Rates using the Two Ray Ground model.

Data Rate Mbps	RxThreshold mW	TxPower mW	Interference range m
18	2.00E-08	5.7704E-04	170.6729663
36	1.00E-07	0.00289205	255.3669777
54	3.16E-07	0.00914546	340.5373378

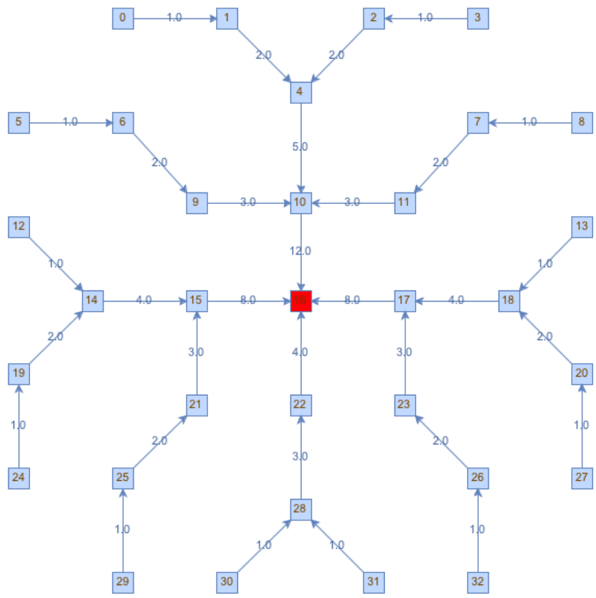

Figure 4: The routing and network configuration of the Grid network.

3.5. IP Models

Table 2: T for single rate IPs on chain and grid networks.

	Jun	Malaguti	VC	MinN
Chain 5	**5.4**	**5.4**	**5.4**	**5.4**
Chain 10	1.286	**1.543**	**1.543**	**1.543**
Chain 15	0.701	**0.9**	**0.9**	**0.9**
Chain 20	0.482	**0.635**	**0.635**	**0.635**
Chain 25	0.367	**0.491**	**0.491**	**0.491**
Grid 9	**4.5**	**4.5**	**4.5**	**4.5**
Grid 17	1.688	**1.8**	**1.8**	**1.8**
Grid 25	0.964	1.149	1.149	**1.174**
Grid 33	0.612	0.794	0.794	**0.831**

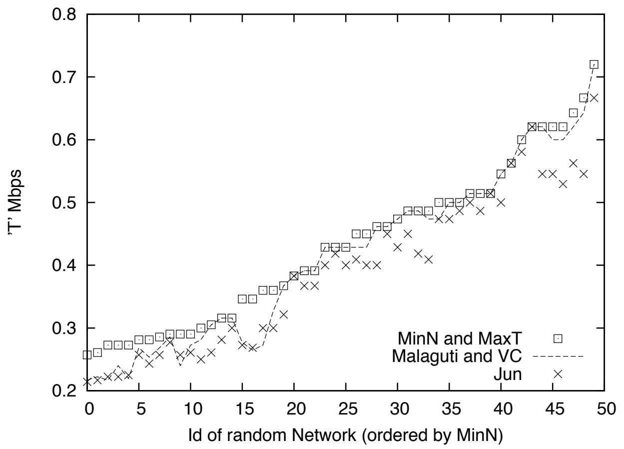

Figure 5: Throughput for IP approaches on a collection of randomly generated networks with 20 nodes.

We present two sets of results for the single rate IP models: Table 2 shows results for the IP models on the chain and grid networks, together with the approximation given by Jun and Sichitiu's approach; and In Figure 5 the throughputs obtained by the models for the randomly generated networks are presented. From these results we can make three observations.

From Table 2 we can see that the IPs show little difference for chain networks, this is probably due to the regularity of the networks. The grid results however show that when the network's layout becomes larger and more complex the models begin to show different results, the *MinN* IP starts to outperform the competing models (seen at Grid25 and above).

In Figure 5 the *MinN* throughput matches the best throughput obtained from applying the *MaxT* IP with *N* between 1 and the value gained from the *MinN* model (see [26] for example results showing the variation of *T* for different *N*). As such, *MinN* and *MaxT* have been plotted together. It can be seen in this figure that whilst our colouring approach does not differ from Malaguti's approach, the irregularity of the networks leads to differences in the *MaxT* approach which is shown to out perform in over 60% of these networks.

An interesting observation to note is that the approximation from Jun and Sichitiu's throughput model is consistently within 25% of the best solution for all these network configurations.

3.6. Heuristics

We present two sets of results for the single data rate heuristic model: Table 3 compares the performance of the heuristic algorithm *HSR* (with $N = 5|L|^2$) to an implementation of Salem's heuristic algorithm, this table includes the results of the *MinN* IP for easy reference; and Figure 6 shows the throughputs obtained by the models for the randomly generated networks. From these results three further observations can be made.

It can be observed in Table 3 that the *HSR* algorithm closely matches the results obtained by the *MinN* IP and as the complexity of the networks increases the differences decrease to being within 99.9% (seen at Chain20 and Grid17 and above)

Table 3: *T* for single rate heuristic on chain and grid networks. † no result obtained within 72 hours. Note MaxT results are identical to MinN.

	MinN	Salem	HSR
Chain 5	**5.4**	**5.4**	5.324
Chain 10	**1.543**	**1.543**	1.539
Chain 15	**0.9**	**0.9**	0.899
Chain 20	**0.635**	**0.635**	**0.635**
Chain 25	**0.491**	**0.491**	**0.491**
Grid 9	**4.5**	**4.5**	4.486
Grid 17	**1.8**	**1.8**	1.799
Grid 25	**1.174**	1.125	1.173
Grid 33	**0.831**	—†	0.830

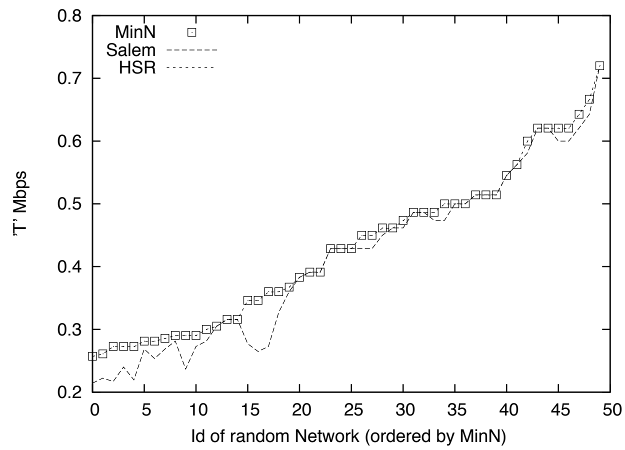

Figure 6: Throughput for Heuristic approaches on a collection of randomly generated networks with 20 nodes.

Comparing Salem's heuristic with our *HSR* it can be observed, in Table 3, that Salem's algorithm outperforms the *HSR* in the smaller networks (Chain5,10,15 and Grid9,17) but, as the networks become larger and more complicated, the *HSR* outperforms Salem's heuristic.

Results for the random networks are shown in Figure 6, showing that *HSR* gives a close approximation (average 99.5%) to the *MinN* IP, and outperforms Salem in 50% of the networks.

Algorithm	5 nodes	15 nodes	25 nodes
VC	17.92	661.13	4239.65
MaxT	14.07	248.7	900.17
MinN	10.09	223.95	2034.67
Salem	1.39	207.5	335770
Malaguti	13.38	133.38	785.43
HSR	1.07	1.92	14.39

Table 4: Run times (ms) for algorithms on various sized networks.

The times taken to run the algorithms are presented in Table 4, the data presented give typical comparisons and are from Chain networks of 5, 15 and 25 nodes. The times include all pre-calculations for each algorithm. Additionally average timings for 10 runs using larger networks have been undertaken with the HSR Algorithm; these show that a 50 node random network takes on average 206ms and a 100 node random network takes on average 2.34s to solve. The data shows a significant improvement in the time taken to run the HSR algorithm, which, as expected, is more evident for the larger sized networks

4. Rate and Schedule Assignment Models

In this section we show how throughput gains can be achieved by allowing the link schedule to set the data rate for a link in each time slot. As in Section 3 and unlike previous research [11, 4], we maintain a particular fixed network routing and seek to maximise the throughput by optimising the transmission rate and transmission power (and thus the interference footprint). A benefit of doing this is that it gives us a methodology to assess the "quality" of different possible routing approaches given the need for link scheduling. In addition this approach allows us to examine a novel heuristic approach that has excellent scalability characteristics that are not present in the corresponding IP formulation. For each selectable data rate we assign a power, such that the transmission range (according to the protocol model) remains constant, and only the interference range is altered. For an increased data rate the required power of the signal at the receiver is also increased; this is the receive threshold. This increase in required power at the receiver requires an increase in the transmission power, which in turn increases the interference range of the signal.

Figure 7 shows a simple network that can take advantage of multiple transmission rates. In this figure, each mesh router is separated by 100m and the interference ranges for mesh routers s_0 and s_4 are indicated by the shaded areas. We show two collision graphs for the network, one for rate 0 (36Mbps) and one for rate 1 (54Mbps), both using a transmission distance of 110m and interference ranges of 260 m and 340 m respectively. As can be seen from the collision graphs, the higher data rate leads to extra collisions. This means that l_0 and l_3 can transmit concurrently when both using the lower rate.

Examples of single and multiple rate schedules are given in Figure 8 for the network in Figure 7. The row labelled "Slot" indexes the slot number and the rows labelled "Multi" and "Single" show the link assignments to slots and their rates for the multi rate schedule and the single rate schedule.

In the single rate case, links l_1 and l_2 are required to send traffic of $3T$ at 54 Mbps during 3 out of 10 slots, giving $T = \frac{54 \cdot 3}{3 \cdot 10} = 5.4 Mbps$. Considering l_0 and l_3 also gives $T = \frac{54 \cdot 2}{2 \cdot 10} = 5.4 Mbps$. When multiple data rates are used, links l_1 and l_2 are required to send traffic of $3T$ at 54 Mbps during 3 out of 9 slots, giving $T = \frac{54 \cdot 3}{3 \cdot 9} = 6 Mbps$. l_0 and l_3 now send traffic (simultaneously due to reduced interference) of $2T$ at 36 Mbps during 3 out of 9 slots, giving $T = \frac{36 \cdot 3}{2 \cdot 9} = 6 Mbps$.

Using our single rate collision model we follow the example of Salem and Sichitiu in allowing control packets to be transmitted on a separate orthogonal channel. This method is acceptable for 802.16 based protocols that uses separate channels to transmit control information. In our multiple data rate collision model this assumption remains the same, although

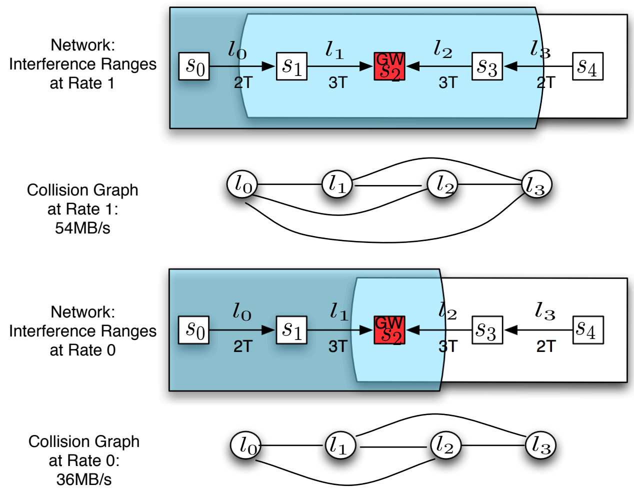

Figure 7: Network for example showing the benefit of multiple rates.

Slot		1	2	3	4	5	6	7	8	9	10
Multi	54Mbps	1	1	1	2	2	2				
	36Mbps							0,3	0,3	0,3	
Single	54Mbps	1	1	1	2	2	2	0	0	3	3

Figure 8: Slot assignments for single and multiple rate schedules.

we note that our model could easily be adapted for use with the 802.11 based wireless protocols by adding additional edges to the collision graph, relating to the reversed direction of each link, to only the lowest data rate graph G_0. These additions would allow the relatively small returned flow control packets to be sent using the lowest data rate and causing the least interference within the network.

4.1. Multiple Rate Collision Model

For the multi-rate model we define a collision graph G_r for each available rate r. Unlike the single rate case, each G_r is a directional graph, because it is no longer possible to say that if link u interferes with link v then both links can't transmit concurrently. That is, it may be possible for link v to transmit at any rate without interfering with link u, whilst link u may only be able to transmit at a lower rate without interfering with link v. In the directional graphs, G_r, an arc $(u \to v)$ represents a link u that would interfere with link v if both transmitted simultaneously. We refer to the set of links that link u interferes with at rate r as its *out-neighbourhood*, $\mathcal{N}^+_{G_r}(u)$. To illustrate this, Figure 9 shows a chain network with two available data rates. The figure also shows the collision graphs for the two rates, G_{r_1} and G_{r_2}. G_{r_2} shows that when transmitting at rate r_1, the link l_3 interferes with link l_0. It can also be seen that the out-neighbourhood of link 3 at rate 0, $\mathcal{N}^+_{G_{r_1}}(l_3)$ is $\{l_1, l_2\}$.

4.2. Integer Programming Model

We provide an IP model to minimise the number of slots in the cycle. To include multiple data rates, the representation includes collision graphs for each of the different rates and decision variables for the rate assignment. Consider a set

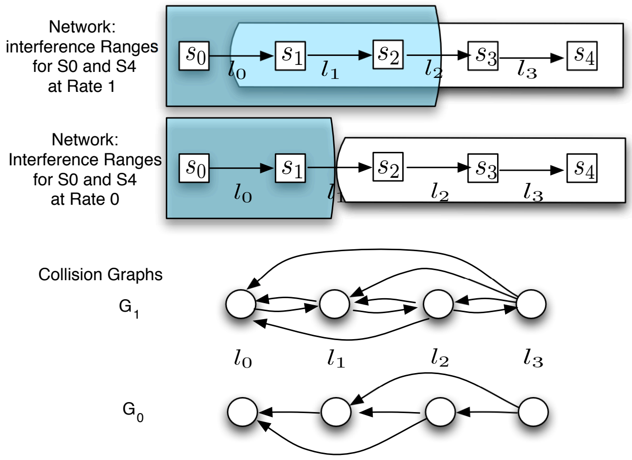

Figure 9: A network graph with interference ranges and the corresponding multi rate collision graphs showing the interference arcs.

R of K possible data rates $R = \{r_1, \ldots, r_K\}$ that could be achieved on each link, defining collision graphs $G_{r_1}, \ldots, G_{r_K}$. The formulation is as follows:

$$Min\ N' \tag{23}$$

$$\text{subject to: } x_{u,i,r} \in \{0,1\} \quad \forall u \in L, i \in \{1,\ldots,N\}, r \in R \tag{24}$$

$$\sum_{r\in R}\sum_{i\in N} \frac{x_{u,i,r}\, r}{r_K} \geq w_u \qquad \forall u \in L \tag{25}$$

$$y_i \geq x_{u,i,r} \quad \forall u \in L, i \in \{1,\ldots,N\}, r \in R \tag{26}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\ldots,N\} \tag{27}$$

$$N' \geq \sum_{i=1}^{N} y_i \tag{28}$$

$$x_{u,i,r} + \sum_{r'\in R} x_{v,i,r'} \leq 1 \qquad \forall u \in L, i \in \{1,\ldots,N\}, \quad r \in R, v \in \mathcal{N}^+_{G_r}(u) \tag{29}$$

$$\sum_{r\in R} x_{u,i,r} \leq 1 \qquad \forall u \in L, i \in \{1,\ldots,N\} \tag{30}$$

In this IP the binary decision variables $x_{u,i,r}$ are used to indicate the selection of link u, time slot i and rate r (24), with only one rate per link per time slot allowed (30). (25) constrains the data carried across all time slots for each link. N is chosen to be an arbitrary upper bound on the number of time slots. The collision domains are enforced by (29) which ensures that for all slots $i \in N$ two links u and v may not be active ($x_{u,i,r} = 1$) if their rates, r and r' respectively, are in the same collision domain, or neighbourhood.

N' is the number of used slots (28), which are denoted by y_i (26), and is minimised to give the shortest cycle time. From the slot assignments and the value for N' the value of T can be computed post optimisation:

$$T = \min_{u\in L}\left(\frac{1}{w_u}\sum_{r\in R}\sum_{i\in N}\frac{x_{u,i,r}\, r}{N'}\right)$$

4.3. Heuristic Approaches

We provide two versions of the heuristic algorithm for the rate assignment model denoted HMR1 and HMR2. As for the single rate case, each uses a measure of satisfaction to prioritise the potential links to add to a slot, but they differ in how they assign rates during consideration of a slot. HMR1 assigns the rate for each link based on currently allocated links, whereas HMR2 additionally allows the previously added link's rate to be dropped if deemed beneficial at the time of the current link allocation. The first heuristic, HMR1, shown in Algorithm 2, adopts the same approach as the single rate heuristic in that for each slot the links are examined in reverse order using a satisfaction measure. In order to consider the different transmission rates, we define a unit of data as being the amount of data transmitted by one link during one slot at the maximum rate. The satisfaction is defined by

$$\frac{D_u}{w_u s}$$

where D_u is the accumulated data transmitted by link u and s is the total number of slots used so far.

For each selected link in the list P, the heuristic HMR1 checks if the link will be interfered with by any of the currently assigned links transmitting at their given rates. Thus instead of an independent set of links, we consider a set of tuples of links with their assigned rates, $J(s) = \{(u_1, r^1), (u_2, r^2) \ldots, (u_z, r^z)\}$. If possible, each link is added to the current slot at the highest rate that does not induce interference with those previously added. When a selected link is accepted at its maximum allowable rate, it is added to the slot's independent set $J(s)$ and the next link is examined. When all links have been examined, the next slot is assigned. As for the single rate heuristic, HMR1 considers the addition of N slots, and selects the best throughput T that is obtained for a subset of N' of these. When the algorithm has completed, the schedule can be extracted from the set of sets J.

Algorithm 3 shows our second data rate heuristic HMR2; this extends the slot allocation concept in HMR1 by allowing the rate of a link previously added to the current slot to be lowered if this is advantageous. In this heuristic a temporary variable p is used to store a link before it is added to the slot set $I(s)$. p is allocated an initial rate when it is first chosen to be included in the slot, but this rate may be lowered during the selection of the next link.

As with HMR1, during the link allocation for each slot, the links are examined in the reverse order of their satisfaction. However, instead of simply adding a link p at the maximum allowed data rate, the algorithm considers whether dropping the data rate would allow a further link q to be added to this slot. It is the approach of the algorithm to increase the satisfaction of the least satisfied links in the network, therefore such a drop in data rate is only worth applying if the result of the reduced rate transmission increases the satisfaction of p to at least link q's previous satisfaction level. The same termination criteria and selection of the best schedule is applied.

Algorithm 2: Multi Rate Heuristic: HMR1(N)

```
N # upper bound on number of slots
s = 0 # slot index
D_u = 0   ∀u ∈ L # max. traffic allocation

while s ≤ N do
    s++
    Create list P of links ordered by increasing D_u/(w_u s)
    I(s) = {} # Links to be added to slot s
    J(s) = {} # Link, Rate assignments for slot s
    foreach i ∈ {1, 2, ..., n} do
        # Is link interfered by current assignment to slot
        if P[i] ∉ ∪_{(j,r)∈J(s)} N+_{G_r}(j) then
            # Build set of allowed rates for P[i]
            R' = {r ∈ R : N+_{G_r}(P[i]) ∩ I(s) = ∅}

            # Add P[i] to slot at highest possible rate
            if R' ≠ ∅ then
                r = max_{r'∈R'} r'
                u = P[i]
                I(s) = I(s) ∪ {u}
                J(s) = J(s) ∪ {(u, r)}
                D_u = D_u + (r/r_K)
```

4.4. Results

We evaluate the multiple rate model using the same network layouts as for the single rate results. We compare the algorithms for the multiple rate model against the best of the previously evaluated algorithms for the single rate model when run using the highest available data rate (54Mbps). The multiple rate algorithms are given three rate/power combinations to choose from; these adopt the two Ray Ground transmission equation [27] to calculate the transmission and interference ranges (Table 1).

We present two sets of results: Table 5 which shows the scheduled throughputs generated by the models for both the Chain and Grid networks; and Figure 10 which shows the scheduled throughputs generated by the models for a selection of random networks (as used in the single rate evaluation). From these results we can make four observations.

Comparing the solutions from the single rate IP formulation with the multiple rate IP formulation, it can be seen from Table 5 that once the chain network reaches a large enough size (eg. 10 mesh routers) then the multiple rate IP begins to outperform the single rate IP, for the Grid network there is not such an improvement.

In the random networks from Figure 10, it can be seen that the the multiple rate IP outperforms the single rate in 85% of the networks.

The dominance of the multiple rate IP in only some of the network styles can be explained; the multi rate models take advantage of certain network configurations (see Section 4) which are less common in the regular grid configuration.

Algorithm 3: Multi Rate Heuristic: HMR2(N)

```
N # upper bound on number of slots
s = 0 # slot index
D_u = 0   ∀u ∈ L # max. traffic allocation

while s ≤ N do
    s++
    Create list P of links ordered by increasing D_u/(w_u s)
    I(s) = {} # Links to be added to slot s
    J(s) = {} # Link, Rate assignments for slot s
    p = P[1] # Highest priority link to be added
    # Check which other links may also be added
    foreach i ∈ {2, ..., n} do
        # Is link interfered by current assignment to slot
        if P[i] ∉ ∪_{(j,r)∈J(s)} N+_{G_r}(j) then
            # Find rates that allow p and P[i] to be added

            # Build set of allowed rates for p
            R^p = {r ∈ R : N+_{G_r}(p) ∩ (I(s) ∪ {P[i]}) = ∅}

            # Build set of allowed rates for P[i]
            R^{P[i]} = {r ∈ R : N+_{G_r}(P[i]) ∩ (I(s) ∪ {p}) = ∅}

            r^p = max_{r∈R^p} r
            if R^p ≠ ∅ and R^l ≠
            ∅ and (r^p = r_K or 1/(w_p(s+1)) [D_p + r^p/r_K] ≥ D_{P[i]}/(w_{P[i]} s))
            then
                I(s) = I(s) ∪ {p}
                J(s) = J(s) ∪ {(p, r)}
                D_p = D_p + r^p/r_K
                p = P[i]
    R^p = {r ∈ R : N+_{G_r}(p) ∩ I(s) = ∅}
    r^p = max_{r∈R^p} r
    I(s) = I(s) ∪ {p}
    J(s) = J(s) ∪ {(p, r)}
    D_p = D_p + r^p/r_K
```

The performance of HMR1 and HMR2 Heuristics against the multiple rate IP can be seen in Table 5 where both HMR1 and HMR2 give close approximations to MultiMinN. Additionally in Figure 10, the heuristic approaches give close approximations to the multiple rate IP throughputs, with both HMR1 and HMR2 averaging aproximatly 99% of the multiple rate IP results. Neither heuristic consistently outperforms the other, with the throughput from HMR1 being at least that of HMR2 in 47% of cases, and vice-versa in 55%. However, due to their short run times, in practice both could be performed on any network, and the best solution used.

To give an idea of the run time of these algorithms, the multi rate IP takes approximately 20 seconds to solve for the 17 mesh router grid network, compared with 108ms for the HMR1 and 172ms for the HMR2. As the networks get larger (25 mesh routers) the exact solution to the IP model takes several hours, and the heuristic solutions are still solved in less than a second.

Table 5: T for Multi rate models on chain and grid networks. † no result obtained within 72 hours.

	MinN	MultiMinN	HMR1	HMR2
Chain 5	5.4	5.4	5.586	5.4
Chain 10	1.543	1.8	1.765	1.702
Chain 15	0.9	1.038	1.031	0.996
Chain 20	0.635	0.73	0.72	0.7
Chain 25	0.491	0.557	0.549	0.539
Grid 9	4.5	4.5	4.5	4.5
Grid 17	1.8	1.862	1.8	1.895
Grid 25	1.174	1.174	1.173	1.268
Grid 33	0.831	—†	0.830	0.869

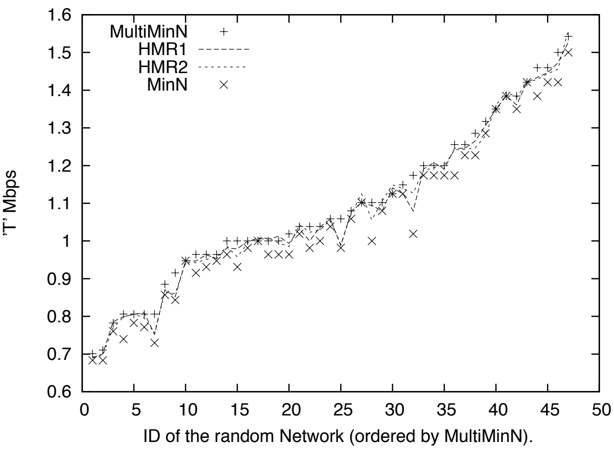

Figure 10: Throughputs for a collection of randomly generated networks with 20 nodes.

These results are obtained using a Macbook Pro running a 2.66GHz Intel Core 2 Duo processor, with 4 GB RAM.

5. Conclusion

We have presented three IP models for the link scheduling of single data rate Wireless Mesh Networks. We have shown that the use of discrete time slots can provide a throughput benefit for large regular networks of over 25 nodes, and also for irregular networks. We also have provided a heuristic approximation model for this time slotted IP model which provides close to optimal solutions in a very small fraction of the run time taken by exact solutions to the IP model. We have taken this idea further and applied it to wireless mesh networks that allow for transmission using multiple data rates, and have used the protocol interference model combined with a power/rate selection to retain a particular routing within a network, whilst allowing the link scheduling to choose between different transmission data rates. We show that this idea can provide an enhanced throughput when compared to the single data rate solutions, although exact solutions to the IP presented is very slow due to combinatorial complexity. To address this problem we have developed two heuristic solutions that can provide close to optimal results in particular network configurations in a very short time frame when compared to exact solutions to the IP model. Additionally when the best heuristic for the multiple data rate formulation is used for comparison, this solution is within an average of 99.2% of the exact solution to the multiple rate IP solution based on sampling 50 random networks. These models can be used to provide a theoretical throughput metric for different networks, and also a throughput metric for different routing configurations for each network. This second use of the models may, in future work, allow networks to be routed based on their suitability for different styles of scheduling algorithms.

References

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

[2] J. Jun, M. Sichitiu, The Nominal Capacity Of Wireless Mesh Networks, IEEE Wireless Communications (2003) 8–14.

[3] N. B. Salem, A Fair Scheduling for Wireless Mesh Networks, in: 18th International Symposium on Personal Indoor and Mobile Radio Communications, Citeseer, IEEE, 2007, pp. 1–5.

[4] J. Luo, C. Rosenberg, A. Girard, Engineering Wireless Mesh Networks: Joint Scheduling, Routing, Power Control, and Rate Adaptation, IEEE/ACM Transactions on Networking 18 (2010) 1387–1400.

[5] M. Alicherry, R. Bhatia, L. Li, Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, in: Proceedings of the 11th annual ACM conference on Mobile Computing, MobiCom, ACM, 2005, p. 58.

[6] P. Djukic, P. Mohapatra, Soft-TDMAC: A Software TDMA-Based MAC over Commodity 802.11 Hardware, in: The 28th Conference on Computer Communications, InfoCom, IEEE, 2009, pp. 1836–1844.

[7] A. Dhekne, N. Uchat, B. Raman, Implementation and Evaluation of a TDMA MAC for WiFi-based Rural Mesh Networks, in: The 3rd Workshop on Networked Systems for Developing Regions, NSDR, ACM, 2009.

[8] Z. Wu, D. Raychaudhuri, D-LSMA: distributed link scheduling multiple access protocol for QoS in ad-hoc networks, IEEE Global Telecommunications Conference, GLOBECOM '04., 2004, 3, 1670–1675.

[9] E. Amaldi, a. Capone, M. Cesana, I. Filippini, F. Malucelli, Optimization models and methods for planning wireless mesh networks, Elsevier, Computer Networks 52 (2008) 2159–2171.

[10] S. Allen, R. Whitaker, S. Hurley, Personalised subscription pricing for optimised wireless mesh network deployment, Elsevier, Computer Networks 52 (2008) 2172–2188.

[11] V. Friderikos, K. Papadaki, D. Wisely, H, Multi-rate power-controlled link scheduling for mesh broadband wireless access networks, IET, Communications 1 (2007) 909–914.

[12] E. Malaguti, M. Monaci, P. Toth, Models and heuristic algorithms for a weighted vertex coloring problem, Springer, Journal of Heuristics 15 (2009) 503–526.

[13] D. J. Guan, X. Zhu, A coloring problem for weighted graphs, Elsevier, Information Processing Letters 61 (1997) 77–81.

[14] C. Cicconetti, I. F. Akyildiz, L. Lenzini, Bandwidth Balancing in Multi-Channel IEEE 802.16 Wireless Mesh Networks, in: International Conference on Computer Communications, InfoCom, IEEE, 2007, pp. 2108–2116.

[15] L. Badia, A. Botta, L. Lenzini, A genetic approach to joint routing and link scheduling for wireless mesh networks, Elsevier, Ad Hoc Networks 7 (2009) 654–664.

[16] H. Shetiya, 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.

[17] M. Sánchez, J. Zander, T. Giles, Combined routing & scheduling for spatial tdma in multihop ad hoc networks, in: Wireless Personal

Multimedia Communications, WPMC, volume 2, IEEE, 2002, pp. (2) 781 – 785.
[18] D. F. Macedo, A. L. dos Santos, L. H. Correia, J. M. Nogueira, G. Pujolle, Transmission power and data rate aware routing on wireless networks, Elsever, Computer Networks 54 (2010) 2979–2990.
[19] S. Avallone, I. Akyildiz, G. Ventre, A channel and rate assignment algorithm and a layer-2.5 forwarding paradigm for multi-radio wireless mesh networks, IEEE/ACM Transactions on Networking 17 (2009) 267–280.
[20] S. Kompella, J. E. Wieselthier, A. Ephremides, H. D. Sherali, G. D. Nguyen, On Optimal SINR-Based Scheduling in Multihop Wireless Networks, IEEE/ACM Transactions on Networking 18 (2010) 1713–1724.
[21] P. Gupta, P. R. Kumar, The capacity of wireless networks, IEEE Transactions on Information Theory 46 (2000) 388–404.
[22] G. Brar, D. M. Blough, P. Santi, T. Cnr, Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Categories and Subject Descriptors, Springer, Wireless Networks (2006) 2–13.
[23] Y. Shi, Y. Hou, J. Liu, How to correctly use the protocol interference model for multi-hop wireless networks, in: The 28th Conference on Computer Communications, IEEE, 2009, pp. 239–248.
[24] L. Badia, A. Erta, L. Lenzini, M. Zorzi, A General Interference-Aware Framework for Joint Routing and Link Scheduling in Wireless Mesh Networks, IEEE Network 22 (2008) 32–38.
[25] V. Vizing, On an Estimate of the Chromatic Class of a -Graph, Diskret. Analiz [in Russian] (1964) 23–30.
[26] I. M. Cooper, S. M. Allen, R. M. Whitaker, Optimised Scheduling for Wireless Mesh Networks using Fixed Cycle Times, in: International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, IEEE, 2011, pp. 1–6.
[27] T. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, 1 edition, 1999.
[28] IBM, Cplex, http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, accessed 9/11/2011.