ABSTRACT

GUPTA, DIVYA K. Joint Scheduling, Routing and Power Control for Single-Channel Wireless Mesh Networks. (Under the direction of Assistant Professor Rudra Dutta).

Mesh networks is a class of wireless networks consisting of a set of backbone nodes and some client nodes. In this work, we look at the problems of routing, scheduling and power control in such networks, with the ultimate goal of increasing the throughput while satisfying all the traffic requirements. To achieve this, we need intelligent scheduling schemes that take advantage of the possible spatial reuse. We outline two heuristic scheduling algorithms that look at various ways of ordering the links and choosing the ones that might reduce the schedule length, which is a measure of the throughput of the network. Different options have been outlined for choosing the first and subsequent links of every slot in the schedule. This includes the interference score of the link, the magnitude of traffic requirement, the specific interfering links, etc.

We also study in detail how the links in the network affect the schedule length. When certain links are scheduled, the spatial reuse factor is reduced to zero, implying that no other links can be active at the same time as these links. Hence it is of interest to us to study more about these "loner" links as they add to the schedule length. We characterize these links for topologies in square and circular areas. Using simple geometric arguments, we show that links of length $0.579k$ and $0.485d$ in a square area of side k and circular area of diameter d respectively, will always be loners. We also outline a method to analytically find the number of loner links in a given network. This number gives a lower bound on the schedule length.

With the understanding gained from the study of scheduling heuristics, we present a routing algorithm that performs topology control by removing loner links. The traffic on these links is routed using other links in the network. Although this routing scheme does not perform well, it leads to a more intelligent joint routing and scheduling mechanism that succeeds in reducing schedule lengths. The joint mechanism does routing in parallel with the scheduling, by calculating routes over links that fit completely into the current schedule. The power control mechanism for both these schemes is to use the minimum power level needed for communication.

Joint Scheduling, Routing and Power Control for Single-Channel Wireless Mesh Networks

by

Divya K Gupta

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfilment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh

2006

Approved By:

Dr. Mihail Sichitiu

Dr. Harry Perros

Dr. Rudra Dutta
Chair of Advisory Committee

Biography

Divya Gupta was born in Bangalore, India, in 1980. She received her undergraduate degree in Computer Science and Engineering from R V College of Engineering, Bangalore, in 2002. She worked as a software engineer in Cisco Systems India Private Limited for 2 years after graduating. She was part of the Network Security Management Business Unit there and was involved in different projects. Since January 2005, she has been a student of the Computer Science Department in North Carolina State University.

Acknowledgements

Dr. Rudra Dutta has been the strongest guiding force as I worked on this thesis. I would like to thank him for giving direction to my research, for the code he provided me, for his insight into various problems we encountered along the way and most of all, for being a wonderful advisor.

I would like to thank Dr.Perros and Dr.Sichitiu for serving on my advisory committee. I also thank my family, members of my research group and friends for all their help, technical and otherwise. A special thanks to Karthik Tamilmani and Preetham Kajekar for the immense support they gave me at different points in time as I worked on my degree.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless networks differ from traditional wired networks drastically because of the nature of the medium. Channel interference and time varying nature of the medium make it necessary to have different protocols for wireless networks. The IEEE 802.11 has for sometime been the ubiquitous standard for data communications in wireless networks. In particular, the infrastructure mode is very popular. In this setup, end points (computers with wireless capabilities) communicate with access points that are spatially distributed in the network area. Nodes communicate only with the access points through the wireless medium and the access points forward all the data to the destination nodes, through other access points or through routers that are part of the backbone network. Communication from the access points to the backbone network is typically through a wired network, although the standard has provision for wireless communications between the access points (AP). Another popular usage of wireless communication is in the cellular networks. Both GSM and CDMA networks have the end points communicating with a base station. Throughput and delay characteristics of such networks are dominated by the single wireless link from the node to the AP/base station and these have been well studied. [11] calculates the maximum possible throughput in 802.11 networks.

Another class of wireless networks is the ad hoc one, where nodes communicate with other nodes in the network with no infrastructure in place. Mobile nodes themselves handle the data forwarding, creation of the network, maintenance of the network and routing. Such ad hoc networks will find applications in many places that need a network to be created and be functional dynamically. Some examples are sensor networks deployed in military applications or in natural environments like forests and oceans. Ad hoc networks are also possible with computers that run 802.11 as the standard provides for an ad hoc mode where the nodes themselves do the routing and forwarding. Data communications here will typically be multihop wireless since all nodes are connected to their neighbors only through wireless channels.

As wireless networks are becoming increasingly popular, the need for multihop wireless communications is on the rise, where nodes send data to the AP through other nodes. A "mesh" kind of connectivity can exist between nodes in the network, with some of the nodes having direct reach to the AP. Wireless mesh networks can be used to deploy community networks over large areas. This is the class of wireless networks that we focus on, in this work. More specifically, the key characteristics of the wireless mesh networks that we assume are that the backbone nodes are not mobile, that there is some capacity planning that can be done and that the nature of communication quality is not very time-varying.

Some important aspects that affect the throughput of such networks are the routing protocols, scheduling algorithms and the power control mechanisms. A lot of work has been done in these areas individually and as combined problems. As explained in later sections, these problems are strongly interrelated and the solution of one affects the other. Hence we take the approach of working on all three of the problems here, for a joint solution.

Our contribution in this work are as follows. We give two scheduling algorithms - one a baseline simple scheduling algorithm and the other one, a more involved independent sets based algorithm. We also provide a detailed analysis of factors affecting link scheduling like link length and topology. We give different routing heuristics for wireless mesh networks.

The rest of this thesis is organized as follows. Chapter 2 gives an outline of the prior work in this area and Chapter 3 gives a more detailed description of the specific problem that is being dealt with. After that, we look a two scheduling algorithms in Chapter 4. A more detailed study of scheduling and characterization of links is presented in Chapter 5. The chapter also outlines a heuristic routing approach along with a joint routing and scheduling mechanism. The next chapter provides some numerical results.

Chapter 2

Related Work

This chapter gives an outline of prior work done in the areas of power control, routing and scheduling in wireless networks in general or specifically in wireless mesh networks.

2.1 Power control papers

Power control in wireless networks is an important problem. This involves assigning specific transmission powers levels to different nodes. Various factors need to be considered for this assignment, like connectivity of the network, interference levels and traffic load. Following is some of the work which has been done in this area.

One of the seminal papers in this area was [18]. Power control affects many layers of the network stack. The authors of [18] claim that the throughput with all nodes operating at one common power level is nearly optimal, hence a common power level should be used. The claim is based on results from [15] which says that a per node throughput of $O(1/\sqrt{nlogn}$ can be guaranteed with a common power. The common power level to be used is determined by the connectivity of the network. COMPOW protocol is built with the DSDV routing protocol (Destination Sequenced Distance Vector) where each node checks routing tables created with different power levels and chooses lowest level that gives connectivity to as many as what would be got with max power level. It can also be used with other routing protocols.

The COMPOW protocol has its disadvantages and [12] described power control protocols that work well in scenarios where COMPOW fails. [12] describes 3 power control protocols - CLUSTERPOW, tunneled CLUSTERPOW and MINPOW. CLUSTERPOW uses the following logic for routing packets - the source node uses the power level such that no other hop will need to use a power level any higher than this one. This can be suboptimal and hence a recursive lookup mechanism is given which is used in the tunneled version. MINPOW uses the Bellman Ford algorithm with the power requirements as the cost.

[13] shows that throughput and delay in 802.11 MAC based networks can be optimized by using direct transmissions in low and high loads. A per-link-minimality condition is used where nodes use just enough power to transmit to the destination. This protocol will be described more in detail in later sections.

The traffic conditions play an important role in the throughput of the network and it should be considered when the power levels are decided. This idea is used in [16], which outlines power control in demand assigned TDMA networks with a central controller. The protocol uses a traffic matrix and allows for low power transmissions at high load and high power direct transmissions at low load. The paper also describes scheduling under both physical and protocol interference models.

Similarly, load conditions are used to decide power assignments to different sectors in [19]. The PATE (Power Assignment for Throughput Enhancement) algorithm tries to avoid congestion by choosing neighbors that are less loaded. Connectivity of the network is maintained by making each node pick a certain number of neighbors. This number is based on other research which show that if each node has a certain number of neighbors, the network will be connected with high probability. A new cost function is given based on load (attenuated) and interference, which is used to determine the neighbors and powers.

POWMAC [14] is a protocol that uses power control in ad-hoc networks to improve performance. POWMAC uses an Access Window to allow for a series of RTS/CTS exchanges to take place before several data transmissions can take place concurrently. The received signal strength is used to dynamically bind the transmission power of potentially interfering terminals in the vicinity of a receiving terminal. The required transmission power of a data packet is computed at the intended receiver, to allow for some interference tolerance at the receiver. This will allow multiple transmissions to take place in the neighborhood.

2.2 Routing and scheduling papers

The problem of finding routes between nodes in a network is one that has been well studied. It becomes a tougher problem in the wireless network scenario in case there are mobile nodes that cause frequent topology changes and also because of varying channel conditions. The power control scheme used will determine the network topology in a wireless network and the problem of routing will be that of finding alternate paths for nodes that cannot be reached by other nodes directly.

When nodes have data to send to other nodes, there needs to be some coordination between all nodes in the network regarding which transmission occurs when. This can either be done centrally or in a distributed manner and is known as the scheduling problem. Following is some work that has been done on these two problems.

A cross layer design with two phases is presented in [10]. A central controller does the algorithm execution and all nodes have a separate feedback channel to talk to the controller. The algorithm is executed at the beginning of every slot. The first phase determines which nodes can transmit in the current slot (creation of a “valid” scenario) and second phase determines what power levels have to be used to satisfy the SINR (Signal to Interference and Noise Ratio) constraints (creation of an “admissible” scenario). Scheduling policies are briefly described. Power control in ad-hoc networks is compared to that in cellular networks and proved that same can be applied here.

The combined problem of routing, scheduling and power control is discussed in [7]. Scheduling and power control are solved as an optimization problem, optimizing for the data rate and minimizing the overall power consumption and cost. Hierarchical scheduling and power control is done where the links are divided into clusters and top level scheduling is done for the clusters as a whole. Then clusters that are active simultaneously will coordinate mutually to determine interference patterns.

[6] presents a proof that the scheduling problem is NP complete - this is done by reducing the problem to that of finding independent sets. It gives a distributed algorithm for scheduling transmissions although the interference model used is not described in detail, except that nodes that are “two hops away” experience interference a particular node is

transmitting. In this solution, nodes exchange scheduling information with neighbors a few times before filling up their matrices that represent the slots. It is assumed that all nodes have the same amount of broadcast traffic to send.

There has been work done on new routing protocols and new cost metrics for routing. Expected Transmission Count (ETX) [8] is a proposed metric to be used in routing protocols like DSR and DSDV which are very popular routing protocols in wireless ad hoc networks. ETX minimizes the total number of transmissions required for a packet to reach its destination. It incorporates the link loss ratios, asymmetry in the link and interference among successive links along a path. This is in contrast to using simply the hop count as a metric where the routing protocol chooses between multiple paths of the same length, without considering the throughput being offered on the paths. It might happen that a longer path is better than a shorter one that has high interference and therefore might require multiple retransmissions. The ETX of a link is basically the predicted number of data transmissions required to send a packet on that link. It is calculated using the forward and backward delivery ratios of the link, which are measured using dedicated link probe packets.

Another very similar scheme presents a metric called ETT (Estimated Transmission Time) [9] for multi-radio multi-hop wireless networks. ETT of an individual link is a function of the loss rate and the bandwidth of the link. The cost of the path is Weighted Cumulative ETT which takes into consideration the interference along the path. This metric has been tested on networks with multiple radios in each node, which does improve the capacity of the network by allowing simultaneous receiving and transmitting. The new metric is used with a routing protocol called MR-LQSR (Multi Radio Link Quality Source Routing)

Chapter 3

Problem Description

3.1 Background

In any wireless network, many factors affect the throughput of the network. Throughput of the network can be thought of as the amount of data that gets successfully transmitted between all source and destination pairs of the network, in unit time. An equivalent definition is the length of the schedule needed for a given set of traffic requirements. Following are some of the factors of importance.

1. **Physical placement of nodes** - Physical placement of nodes or the topology of the network can be either uniform (density of nodes remains approximately the same anywhere in the area of interest) or clustered (density of nodes is very high in some parts and very low in all other areas). Uniform topologies can be further classified as perfect grid placements and random placements. Random placements of nodes are approximations of grid placements. The physical placement of nodes would determine the characteristics of links in the network and the transmission power required to maintain these links.

2. **Transmission power levels** - For any link in the network to be able to successfully carry data, the transmission power from the sender has to be high enough to have the intended receiver within its communication range. If the receiver is outside this

range, then the communication might not be successfully received. Additionally, the transmission power of one source and destination pair also determines which other transmissions can occur in the network at the same time as this one. The more the number of successful transmissions taking place in parallel, more is the throughput of the network.

3. **Routing and scheduling** - Routing will determine what path the traffic needs to travel over for any given source-destination pair. Whatever routing is chosen, for it to be optimal, a perfect scheduling scheme would be needed.

4. **Traffic matrix** - The pattern of traffic between different source-destination pairs will determine what power level and routing will be optimal for the network. This can represent data in bytes, Mb, Mbps, frames (MAC) or frames per schedule. The most generalized form is the frames per schedule and the others can be reduced to this form with simple transformations. At this point it is important to differentiate between a *traffic* matrix and a *transmission* matrix. The traffic matrix gives the traffic requirements between all node pairs. The transmission matrix is got after these traffic requirements have been routed over the routes for all the node pairs (it is independent of how the routes are obtained). Transmission matrix will give the number of transmissions that need to take place on each link in one schedule length.

All these factors have interdependencies and will together determine the throughput of the network.

3.1.1 Radio Interference

The concept of transmit power range is central to any kind of wireless communication. When a node is transmitting data using a radio, it creates an area around itself in which the signal strength is strong enough for correct reception of the transmitted data. This area is called the transmission range of the node. There is another area, larger than the transmission range, in which the signal of this transmission is not strong enough for correct reception, but is strong enough to disrupt other transmissions. This is known as the interference range. The ratio of the interference range to the transmission or communication range is referred to as the Interference Ratio in this document. Modeling radio interference is tough. Two models that have been considered in prior literature are physical model and the protocol model.

- The protocol interference model - Any transmission is successful if no other node in the interference range is transmitting at the same time.
- Physical interference model - Transmission is successful if the signal strength to noise ratio of the received signal is greater than the threshold value of the signal to noise ratio.

A more extensive description and problem formulations can be found in [17].

3.2 Wireless Mesh Networks

In this section we describe some of the distinguishing characteristics of the wireless mesh networks that are commonly assumed in literature.

1. The network is assumed to have a set of backbone nodes that forward traffic. These backbone nodes are stationary.
2. Each node is assumed to have knowledge about the amount of traffic it would need to send in a given period of time. A traffic estimate matrix is constructed based on the traffic requirements of all the nodes and this matrix is thought of as a fairly stable one.
3. The link quality and the bandwidth as assumed to be unvarying.
4. Minimizing the amount of power utilized by the nodes is not a priority here. It is assumed that the nodes are not power constrained.
5. The nodes are assumed to have good time synchronization with a central controller. This is required for the nodes to be able to send packets in the slots allocated to them.

In this work, we focus mainly on the communication between multiple backbone nodes and not between multiple clients or between clients and backbone nodes. Client nodes may be mobile, but backbone nodes are stationary. Radio interference is the unique characteristic of this environment which must be taken into account.

3.3 General Assumptions

Following are the other assumptions made in the work presented here:

1. We consider only a single radio channel, which is used for all transmissions and receptions by all nodes.

2. The protocol interference model is used here. In this model if the distance of a receiver from a second sender, to whom the receiver is not intending to listen to, is more than twice the communication range of the second sender, it is assumed that the receiver will be able to receive the data successfully.

3. All links are bidirectional. If node A can reach node B, then node B is also able to communicate with node A.

4. All transmissions are bidirectional, i.e. if a node A is sending a packet to node B, node A is also acting as a receiver in this transmission when node B sends the acknowledgement.

5. We focus on the development of a global scheduling algorithm that can be executed by a central controller - a distributed implementation of the scheduling or issues of signaling to distribute the schedule is outside the scope of this work.

6. Every node is assumed to have an omnidirectional antenna.

7. In reality a fixed set of transmit power ranges are available in a wireless node. Here we assume that a node has the ability to change the transmit power continuously over a wide range, rather than over a set of discrete values, resulting in a network that could be fully connected (a clique).

8. In general the ratio of the interference range to the communication range is assumed to be 2. It is explicitly mentioned in places where a different ratio is used.

9. Throughput is measured in terms of how many slots are needed to complete a schedule that satisfies all traffic requirements. The same schedule will repeat infinitely. The lesser the number of required slots, higher is the throughput.

While assumptions about the interference model, the links and transmissions are widely used and accepted in literature, those about transmit power ranges and throughput measurement are more specific to this work.

3.4 Power control problem

The transmission power used for communication between a source and destination pair determines the communication range and the interference range for this transmission. The transmission power of all the nodes in a network determines the topology of the network and in turn the throughput to a great extent. Higher the transmission power, higher is the connectivity. At the same time, an increase in the power also increases the amount of interference in the network. While higher connectivity would imply increased throughput because of shorter paths, higher interference would imply decrease in the number of active links at any given point of time. Hence controlling the transmission power of the nodes in a network is a very important design goal. Other aspects to the problem are whether power control should be done dynamically based on traffic requirements or statically, right when the network is deployed. It is also interesting to see if it is better to have a single controlling knob for the transmission power of all the nodes in the network or if it better to have finer tuning capabilities where each node's power can be controlled individually. Transmission power of the nodes of a network can be controlled in different ways

1. A single common power level for all nodes that is kept constant over time. This common power level is chosen such that it is the minimum power required for network connectivity (COMPOW) i.e., if the transmission power of all the nodes is reduced to something below this, there will be a few nodes that are not reachable by the rest of the network.

2. A single common power level for all nodes that is constant over time, but high enough for all-to-all connectivity. In the previous scheme the common power level was the minimum required for the network to be just connected. We can also have all the nodes communicating at a power level that is high enough so that every node can reach every other node directly at this power level. We can call this scheme COMPOW-Max.

3. We can also have different power levels for different nodes, that are kept constant over

time. These power levels could, for example, be chosen such that each node is able to reach its closest neighbor.

4. Different power levels can be used for different node pairs, but maintained constant over time. Each node transmits in various different power levels depending on the destination node for this data, but for every given pair of nodes the power level is fixed. When node A needs to send data to node B, it always uses power level P_{ab}. This power level is the minimum required for communication between A and B. This is the philosophy behind the DirectTrans scheme [13].

5. Different power levels can be used for different node pairs, varying over time (changed for each transmission). This would imply that when node A needs to send data to node B, it can sometimes use P_{ab} to send it directly, or it can use some other power level and send it using multiple hops. How to determine this power level would be another problem to be solved. This method provides maximum flexibility.

3.4.1 COMPOW

COMPOW [18] is a power control protocol designed for wireless ad hoc networks. Once the power level has been determined, any proactive routing protocol can be applied on the network to get the actual routes between nodes. COMPOW claims to optimize the capacity of the network and also save on the power consumed. The simple argument for optimization of capacity comes from the results in [15], which says that the throughput got by every node cannot increase beyond $\frac{c}{\sqrt{n}}$. Assuming a random distribution of n nodes in a disc of area A, they give mathematical reasonings which show that the data rate for each node, λ is inversely proportional to r, the communication range of all the nodes. This means that more throughput can be achieved if the nodes use smaller transmission powers. The principle of COMPOW is that all nodes the nodes use the lowest common power level at which the network is connected. The authors provide an architecture of the protocol which uses multiple routing tables to find the optimum power level.

3.4.2 DirectTrans

While COMPOW presented asymptotic results to prove that the network capacity can be optimized by using low power, multi-hop routes, a scheme that we refer to as DirectTrans has been presented in [13], which shows that in practical networks sizes, it is infact better to apply the per-link-minimality condition. This means that every transmission is attempted to be completed in a single hop, through a direct transmission. They show that decreasing the average hop count is favorable to the throughput and delay conditions of the network.

3.5 Routing problem

Given a set of active links, this problem is about which route to choose for every source and destination pair. If routing is fixed first, then it also fixes the minimum power level required for each node to be able to create the links for the routing. We would then need to determine whether these minimum power levels are the optimal power levels for the network. If they are not optimal, we need to find what would be better.

If routing is not fixed initially, then we need to find the optimal routing for the chosen power level (i.e. for the topology resulting from these power levels).

If the connectivity in the network is high, there will be many choices of paths available for the nodes. The choice of routes will then determine how much other traffic each link carries, other than what was originally the requirement for this pair of nodes. Deciding what route to use can be a difficult problem. One way to do it would be to assign costs to each link and then apply some form of path calculation algorithm. The cost can be hop counts or other values [9][8]. Other dynamic distributed routing protocols like AODV (Ad hoc On Demand Distance Vector), DSDV, DSR, etc are also in use.

3.6 Scheduling problem

The amount of traffic on each link determines for how long it needs to be active in one schedule length. The basic function of the scheduling algorithm is to satisfy these transmission requirements. This should be done in a way that when a link is allocated to a particular slot, there are no other transmissions occurring at the same time which could possibly disrupt successful reception of this transmission. While this is the basic requirement of the scheduling algorithm, for the schedule to be optimal, each slot should be packed with as many links as possible. The set of links in a slot should be such they none of the source and destination pairs are in the interference ranges of any of the other nodes that will be active in this slot.

The problem of scheduling links optimally is equivalent to that of creating maximum independent sets. An independent set of a graph is a set of nodes, none of which have links to any other node in the set. A maximum independent set is the largest such set that can be created for this graph. Whereas a maximal independent set is one which is not necessarily the largest, but is such that the addition of any other node to the set will render it a non-independent set.

A connectivity graph is one where the vertices of the graph are the nodes of the network and every link in the network is represented as an edge in the connectivity graph. The conflict graph is somewhat complementary to this graph. Here, every link of the connectivity graph forms a node. There is an edge between two nodes in a conflict graph if the corresponding links in the connectivity graph are interfering. Once the conflict graph is constructed, the problem of scheduling links is reduced to that of finding independent sets in the conflict graph.

3.7 Joint Scheduling, Routing and Power Control

In this section we see how the three problems and their solutions will affect each other.

The power control scheme decides the power levels for each node, which decides the transmission and interference ranges. The set of links formed for a given set of transmission powers decides what routes can be used. Now, if all the traffic requirements are satisfied using this routing, we get a resultant transmission matrix which gives the exact number of transmissions that occur on each link. This transmission matrix will be the input for scheduling. The final schedule got will determine the throughput of the network.

We can also work backwards by saying that, while scheduling, a certain set of links cannot be scheduled in parallel with any other links due to large interference ranges and hence these "unfavorable" links should not be used for routing too much traffic. Given these restrictions on links, we can calculate what paths need to be used for different source destination pairs. Power control mechanism can simply say that for any node pair to transmit, it will simply use the optimal power required to do so. At this point, a relevant question would be as to what these optimal power levels are and whether it is indeed best to use the minimum possible power level for each node pair such that all the links required in the routing are created. Assuming the existence of a perfect scheduling algorithm, it is very obvious that we cannot reduce the power levels below this minimum level as the given routing will not be possible if all required links are not formed. On the other hand, if we increase the power levels beyond the minimum level, it can have two effects:

Addition of interfering link pairs: Two links that were not interfering earlier, can now interfere. As a result of the increase in number of interfering link pairs, the schedule will need to be recalculated. Its length might increase as a pair of links that had been scheduled together might need to be scheduled in different slots now.

Addition of new link: Two nodes that were not in communication range of each other, can now form a new link because of higher power levels. Since we have already fixed the routing, the new link will not be useful in any way, even if it would have given us a more optimal route for some source-destination pair.

Hence, given a routing, it is best to use the minimum possible power levels that satisfy the route requirements. This shows how a given scheduling algorithm can affect routing and how the routing mechanism will determine optimal power levels.

Power control, routing and scheduling are closely related problems and have strong interdependencies. It therefore makes sense to consider solving them as a joint problem, rather than looking at a single piece at a time.

Chapter 4

Scheduling Heuristics

Scheduling wireless transmissions is a difficult problem. It can be described in terms of the maximum independent sets problem, as explained in Section 3.6, which is a well known NP hard problem. Two solutions are presented here.

4.1 Simple scheduler

4.1.1 Description

The following simple scheduling algorithm considers one traffic element (link) at a time and for each of them, tries to find as many other traffic elements that can be scheduled in parallel. All these links are then added to a slot. After the first link for the slot has been chosen, the other elements are got by first finding a set of candidate nodes that are outside the interference range of the current link and then finding the best pair of nodes from this list. The best pair is chosen such that the traffic elements of the links in the slot are of equal or almost equal magnitudes.

An initial ordering of traffic elements is got before starting the scheduling. This ordering can be based on different things -

- Random ordering.
- Magnitude of traffic.

- Link length (distance between source and destination).

Based on the ordering chosen, runtime of the algorithm can vary, as creating the ordering itself would need additional computation. This heuristic works more at the node level, rather than at the link level, when building the slots.

4.1.2 Algorithm

```
N ← Number of nodes in the network
Construct T such that T_ij ← Traffic between nodes i and j
Construct D such that D_ij ← Distance between nodes i and j
Candidates ← 0
while T_ij ≠ 0 i,j = 0, 1, ···, N − 1 do
  Get an ordering of the elements of T
  for all T_ij in the ordering do
    Select the next element T_sd
    Create a slot of length T_slot = T_sd in the schedule for L_sd
    for a = 0 to N do
      for all L_s'd' in current slot do
        if D_as' and D_ad' > INTF_RATIO * D_s'd' then
          Add a to Candidates
        end if
      end for
    end for
    for all (b, c) ∈ Candidates do
      if T_bc > 0 and L_bc does not interfere with any L_mn in current slot then
        if T_bc = T_slot then
          Remember L_bc and break
        else if T_bc has least difference with T_slot then
          Remember L_bc
        end if
      end if
    end for
    if L_bc was found then
```

```
            Schedule $L_{bc}$ in the current slot and update $T_{slot}$
            Go to step 10
        else if no link was found then
            Update $T$ by reducing $T_{s'd'}$ for every $s'd'$ in current slot by $T_{slot}$
        end if
    end for
end while
```

4.1.3 Algorithm Complexity

There are $O(N^2)$ traffic elements in the traffic matrix. Every slot can have a maximum of $N/2$ links in it. The length of the *Candidate* list can be a maximum of N. Based on these points the complexity of the algorithm works out to be $O(N^5)$.

4.2 Interference Score

Every link in the network can be associated with an interference score which is basically the number of other links in the network that this link interferes with. This is an indication of how difficult it would be to find another link that can be scheduled in parallel with this one. Calculation of the score can be altered to reflect this information more accurately by using the traffic matrix. If a link interferes with links that do not have any transmission requirements on them, then these links should not be counted in the interference score calculation. This is how the score is calculated in the scheduling and routing heuristics mentioned in future sections.

4.3 Independent Sets Based Scheduler

4.3.1 Description

In this scheduling algorithm, independent sets are built for each slot. Till all the transmissions in the transmission matrix have been scheduled, the first link of the slot is got by some criteria (explained below) and once selected, all the links that it interferes with are marked as "rejected" for this slot. From the ones that have not yet been rejected, other links are chosen to be scheduled in parallel in this slot. Different ways of selecting the first link of the independent set have been explored. Similarly, different ways of selecting other links of a slot, after the first one has been chosen, have also been tried.

4.3.2 Algorithm

1: $N \leftarrow$ Number of nodes in the network
2: Construct T such that $T_{ij} \leftarrow$ Traffic between nodes i and j
3: Construct D such that $D_{ij} \leftarrow$ Distance between nodes i and j
4: Get an ordering of the elements of T
5: **while** $T_{ij} \neq 0$ i,j $= 0, 1, \ldots, N-1$ **do**
6:     Get the first link, L_{sd} for current slot by *Criterion 1*
7:     $T_{slot} \leftarrow T_{sd}$
8:     Mark $Selected_{ij} \leftarrow 0$ for $i, j = 0, 1, \ldots, N-1$
9:     **while** true **do**
10:         Mark $Selected_{sd} \leftarrow 1$
11:         **if** $T_{sd} < T_{slot}$ **then**
12:             $T_{slot} = T_{sd}$
13:         **end if**
14:         **for** $i = 0, 1, \ldots, N-1$ **do**
15:             **for** $j = 0, 1, \ldots, N-1$ **do**
16:                 **if** L_{sd} interferes with L_{ij} and $Selected_{ij} = 0$ **then**
17:                     Mark $Selected_{ij} = -1$
18:                 **end if**
19:             **end for**
20:         **end for**
21:         Get the next link L_{sd} with $Selected_{sd} = 0$ by *Criterion 2*
22:         **if** no such link L_{sd} exists or $T_{sd} = 0$ **then**
23:             Go to step 25
24:         **end if**
25:     **end while**
26:     Add all L_{ij} such that $Selected_{ij} = 1$ to the current slot

end while

4.3.3 Possibilities for *Criterion 1*

- **Maximum Interference Score** - Links are first ordered in descending order of their interference scores and they are considered for scheduling in that order. The intuition behind this is that if there are a set of very short links that can potentially be scheduled in parallel with many longer links, we want to save these for scheduling later. This will give some options for parallelization of the longer links. If all the short links with low interference scores are scheduled with each other initially, links with higher interference scores will all be loners in the final schedule.

- **Minimum Interference Score** - It can also be argued that links with high interference score can anyway not be scheduled in parallel with any other link and hence we might as well select links in ascending order of interference scores.

- **Random Ordering** - Links are considered in random order, say by node numbering. This would show whether there is any added advantage in calculating interference scores and scheduling according to those values.

4.3.4 Possibilities for *Criterion 2*

- **Maximum Interference Score** - For selecting the next unrejected link for the current slot, links can be ordered in descending order of interference score and the one with the maximum interference score that has not yet been rejected can be selected.

- **Maximum Overlap and Minimum Difference** - When the first link is selected to be scheduled in the current slot, all the other links that interfere with this link get rejected. When selecting the next link for the slot, one way to do it is to select a link such that most of the links that it interferes with have already been rejected *(maximum overlap)* and also, it interferes with very few other links that have not yet been rejected *(minimum difference).* The intuition behind this is that we want to be able to add as many links to the slot as possible, while rejected as few as possible each time. This way, when the next link is being selected for this slot, there is ample choice to choose from. It is a greedy way to maximize the size of the independent set that is got for each slot.

- **Minimum Difference -** The selection criterion can also simply be that the link which causes the rejection of as few unrejected links as possible, is the best one.

Chapter 5

Study of Joint Routing and Scheduling

5.1 Effect of length of links on the schedule

Our observations show that in a uniform topology with random traffic simply doing scheduling, however optimally, will not reap great rewards. It needs to be combined with intelligent routing schemes, that take advantage of parallelization in scheduling. The reason for this is that given a square area and a uniform topology, a majority of the links in the network are ones that cannot be scheduled in parallel with any other link in the network. We refer to these links as “loners”.

5.1.1 Characterizing links in a square area

Loners in a square area

The problem under consideration here is, of what length a link should be, for it to be a loner with high probability. One simple reasoning is this - the longest link possible in a square area is one along the diagonal. If we are considering an interference to communication

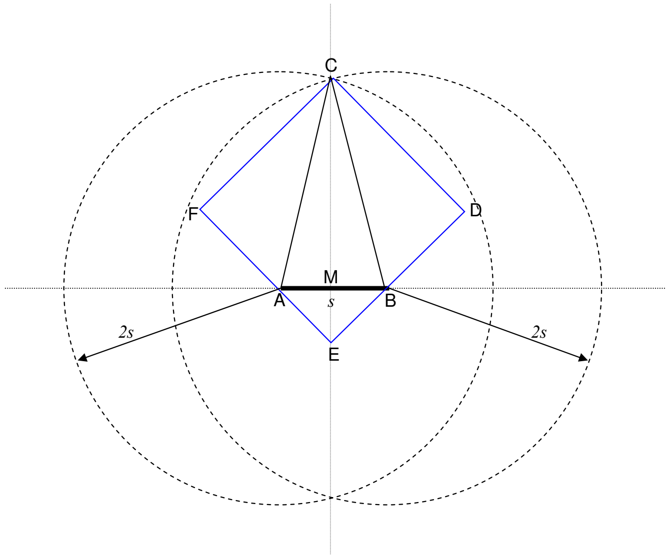


Figure 5.1: Characterizing the length of loner links in a square area

ratio as 2, a link that is half the length of the diagonal will cover the entire square with its interference area, irrespective of where the link is placed. This is an overkill - links that are shorter than the diagonal can be loners depending on where they are placed.

Here is a more accurate reasoning to characterize the length of loners in a square area.

Let AB be a link of length s units. When the link is active, it creates an interference area of circles of radius $2s$ centered around the two nodes as shown in Figure 5.1. We know that any link inside this double circle area cannot be scheduled in parallel with AB. This characterizes the loner length in a double circle area in this particular orientation, but we want a square area. We need to find a square of side k which can be placed inside the double circle in any orientation and at any position such that whenever it contains link AB in it, no part of the square is outside the double circle. This means that, if this square was our area of interest, a link of length s will always be a loner in it, as the entire square is always contained in the interference range of the link and there is no part of the square left out where another link could be placed, such that it can be scheduled in parallel with AB. A square of side s is one such square, but we can find larger ones than that.

The square $CDEF$ in Figure 5.1 is the largest such square. And if we find the

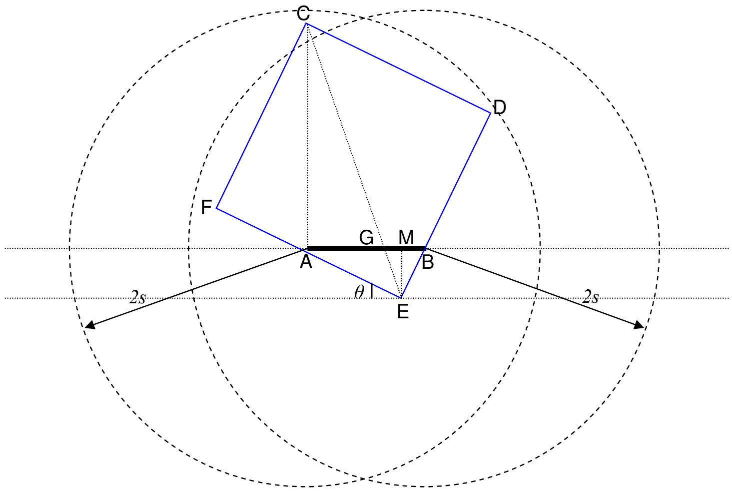


Figure 5.2: Non-diagonal orientation of the square has lesser height

length of the side of this square, the relation between k and s will clearly characterize loner links in a square area. We know that AB is s units, and AC and BC are $2s$ units (radii of the circles). Height of $\triangle ABC$ will then be $1.94s$ using Pythagorean theorem. In $\triangle ABE$, side AE and BE are of equal length and the height of the triangle is $0.5s$. The diagonal of the square is therefore $2.44s$ and the side of the square is $1.725s$. Given a square of side k units, any link of length $0.579k$ or more is always a loner.

It is important to note that links shorter than $0.579k$ can be loners depending on where they are positioned.

Now, the proof for why the diagonal orientation of $CDEF$ is the one we need to consider. We want the largest square such that its height with respect to the link AB is not more than $1.94s$. Let us consider the same square placed in a different orientation as shown in Figure 5.2, making an angle of θ with the horizontal. In $\triangle ABE$, AE is of length $s\cos\theta$ and BE is $s\sin\theta$. In $\triangle MEB$, ME is of length $s\sin\theta\sin(90-\theta)$. In $\triangle GME$, GE is of length $s\frac{\sin\theta\sin(90-\theta)}{\sin(45+\theta)}$. Since we know that the diagonal of the square is $2.44s$ we can calculate the height of $\triangle CAG$ and this works out to be $\frac{2.44}{\sqrt{2}}(\sin\theta+\cos\theta-\sin\theta\cos\theta)$.

A plot of this height against the angle θ is shown in Figure 5.3. As seen, the

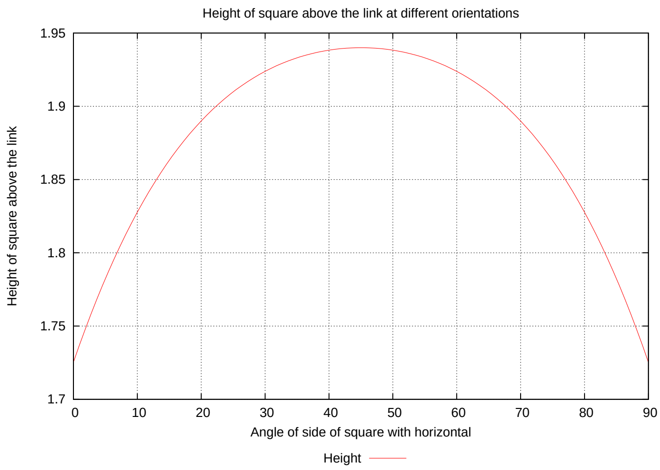


Figure 5.3: Diagonal orientation of the square has maximum height

height of the square above the link AB is maximum when the orientation is with the diagonal perpendicular to the link.

Non-loners in a square area

The next problem to be considered, is to find what length a link should be, for it to *not* be a loner with high probability. For a link not to be a loner, its interference range should not completely cover all the nodes. By similar reasoning as in earlier sections, we require a square area that is too large to fit into the double circle in Figure 5.4. The square $CDEF$ is such that it barely fits in, in the square orientation, which we know has the least height above the link. By turning this square in some angle or by increasing the size of the square even a little, we will have some of its area outside of the double circle. In this non-overlapping area, another link can be placed which can be scheduled in parallel with link AB. The size of this square can be got by solving for x in the $\triangle AGC$. The side of the square works out to be $3.283s$. Hence any link of length $0.304k$ or lesser, inside a square area of side k, will not be a loner.

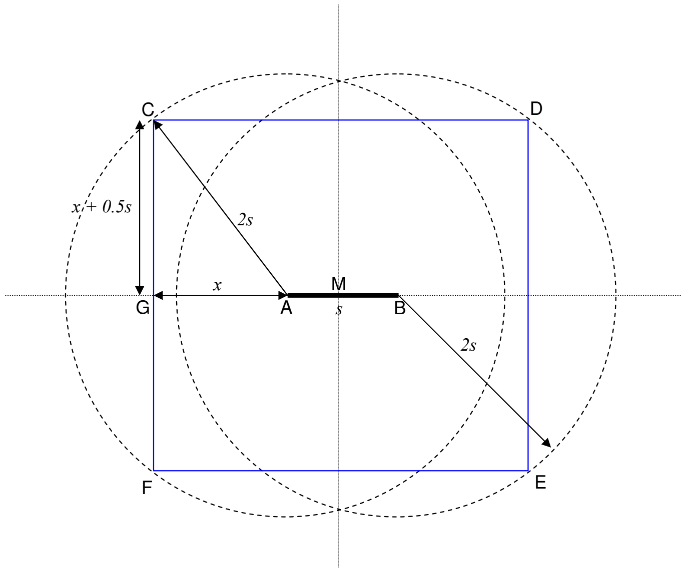


Figure 5.4: Characterizing the length of non-loners in a square area

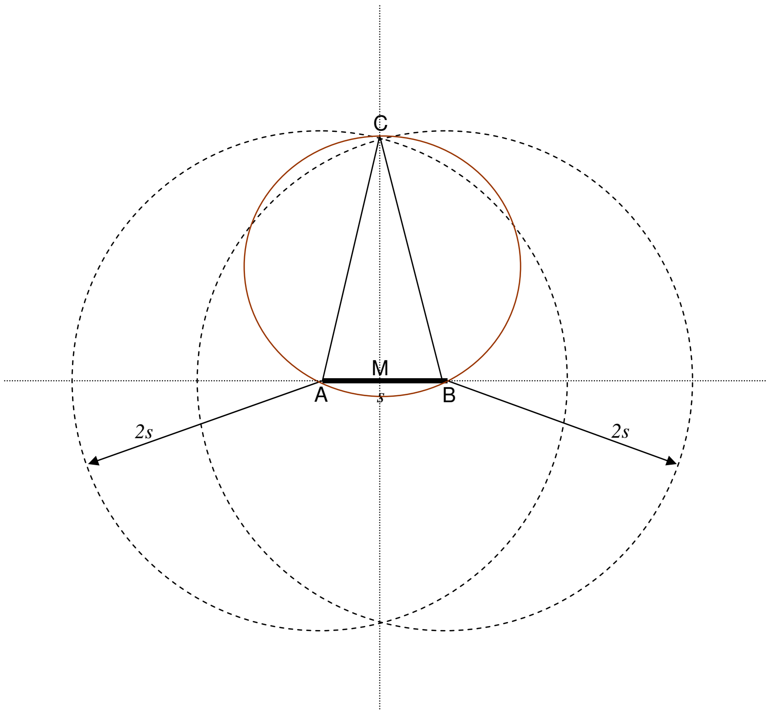


Figure 5.5: Characterizing loner links in a circular area

Table 5.1: Loner and non-loner link lengths with different interference ratios

IR	SQUARE		CIRCLE	
	Non-loner	Loner	Non-loner	Loner
1	0.577k	1.035k	0.5d	0.866d
2	0.304k	0.579k	0.25d	0.485d
3	0.212k	0.408k	0.167d	0.328d

5.1.2 Characterizing links in a circular area

Loners in a circular area

A similar characterization can be done for the loner links in a circular area. We know that the height of $\triangle ABC$ in Figure 5.5 is $1.95s$. The radius of a circumscribed circle for a triangle is given by the formula $r = \frac{a_1 a_2 a_3}{4A}$ where a_i is the side of the triangle and A is the area of the triangle. Using this formula, the diameter of the circle shown in Figure 5.5 works out to be $2.061s$. Based on this result, we can say that in a circular area of diameter k, all links of length $0.485k$ or more are loners.

Non-loners in a circular area

The smallest circle which does not fit into the double circle area is one of radius just over $2s$. Hence in a circular area of diameter d, a link of length $0.25d$ or lesser will not be a loner.

5.1.3 Characterizing links with other Interference Ratios

Similar to the calculations done in the previous sections, we can consider interference ratios other than 2. While the geometrical arguments are omitted, the values got are shown in Table 5.1. The columns labeled ‘Non-loner’ give the length of links in a square and circle, below which no link is a loner. The columns labeled ‘Loner’ give the length, greater than which every link is a loner. These values are for a square of side k and a circle of diameter d.

5.1.4 Finding the number of loner links

If nodes are distributed evenly in a square area, then the number of nodes in a fraction of the area will be proportional to this fraction. From earlier subsection we know that links of length $0.579k$ or more are loners in a square of side k. Given some point P inside this square, we now need to find the number of links of length $0.579k$ or more that can start from P. To do this, we can draw a circle of radius $0.579k$ from P and find the fraction of area of the square that lies outside this circle. Given this area and the density of nodes in the network, it is simple to find the number of links of length $0.579k$ or more that start from P. We can then repeat this process every other point in the square to find the total number of loner links of that length.

Area of overlap between a square and a circle

Given a unit square whose bottom left corner is $(0, 0)$ and top right corner is $(1, 1)$, based on the tool in [4], we can write these equations to find the area of overlap for a circle whose center (h, k) is inside this square and radius is r.

$$A_{ovlp} =$$

$$\min((1-h)(1-k), \frac{1}{2}(r^2 \arcsin\left(\frac{\min(1-h,r)}{r}\right) + \min(1-h,r)\sqrt{r^2 - \min(1-h,r)^2} +$$
$$r^2 \arcsin\left(\frac{\min(1-k,r)}{r}\right) + \min(1-k,r)\sqrt{r^2 - \min(1-k,r)^2}))$$
$$+$$
$$\min((h)(1-k), \frac{1}{2}(r^2 \arcsin\left(\frac{\min(h,r)}{r}\right) + \min(h,r)\sqrt{r^2 - \min(h,r)^2} +$$
$$r^2 \arcsin\left(\frac{\min(1-k,r)}{r}\right) + \min(1-k,r)\sqrt{r^2 - \min(1-k,r)^2}))$$
$$+$$
$$\min((1-h)(k), \frac{1}{2}(r^2 \arcsin\left(\frac{\min(1-h,r)}{r}\right) + \min(1-h,r)\sqrt{r^2 - \min(1-h,r)^2} +$$
$$r^2 \arcsin\left(\frac{\min(k,r)}{r}\right) + \min(k,r)\sqrt{r^2 - \min(k,r)^2}))$$
$$+$$

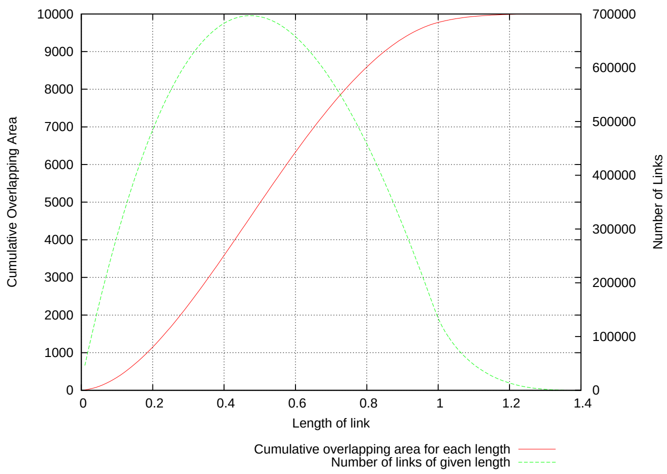


Figure 5.6: Finding number of links of a given length in a 100×100 grid in a unit square

$$\min((h)(k), \frac{1}{2}(r^2 \arcsin\left(\frac{\min(h,r)}{r}\right) + \min(h,r)\sqrt{r^2 - \min(h,r)^2} + r^2 \arcsin\left(\frac{\min(k,r)}{r}\right) + \min(k,r)\sqrt{r^2 - \min(k,r)^2})) \tag{5.1}$$

This four-part equation considers the overlap for each quarter of the circle and sums them up.

With this tool, it is also possible to find the number of links of any given length l (not just all links greater than 0.579k). To find this, we can follow these steps:

1. Find the area of overlap for a circle of radius $l - \triangle l$, which is centered at a point (h, k) inside the square. Let this be $A^{(h,k)}_{l-\triangle l}$.

2. Find the area of overlap for a circle of radius $l + \triangle l$, which is centered at a point (h, k) inside the square. Let this be $A^{(h,k)}_{l+\triangle l}$.

3. Subtract the two values $A^{(h,k)}_{l+\triangle l}$ - $A^{(h,k)}_{l-\triangle l}$ and multiply the result with the density of nodes in the square area to get $N^{(h,k)}_l$, the number of links of length l, with one end at (h, k).

4. Repeat this for all possible values of (h, k) inside the square and add the values to get the total number of links, N_l.

Figure 5.6 shows values got from this procedure for a grid of size 100×100 (10,000

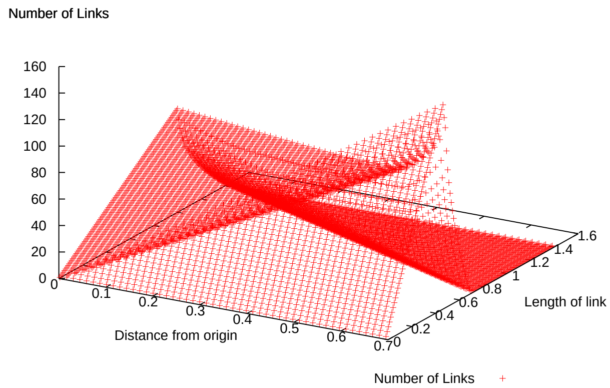


Figure 5.7: Number of links of a given length (100 × 100 grid) at different positions

nodes). The green line gives N_l, the actual number of links of a given length and the red line gives the overlapping area, summed over all possible points as centers (A_l).

This numerical differentiation would give an approximate result. Note that the first four steps perform calculations in the "continuous domain", but the last step moves to the "discrete domain" by multiplying the area of overlap with the density of nodes.

Apart from the length of the links, we are also interested in the positions of the link - shorter links have more chances of being loners if they are placed closer to the center of the square, rather than at the corners. In Figure 5.9, when the link is placed at the corner, the interference range does not cover the entire square, thus allowing for another active link to be present in the non-overlapping area. But when placed at the center, the entire square is covered, making the link a loner. So, the number of loners of a certain length, will increase as we increase the length and also as we move the towards the center of the square. To understand how the number of links of a given length will change as we move towards the center of the square, we can again use the tool to calculate overlapping areas and numerically differentiate these values. Figure 5.7 shows a plot of the number of links of different lengths as we move one end of the link towards the origin (bottom left corner of the unit square). These values are again for a 100 × 100 grid placed inside a unit square.

If we take cross sections of this graph for different values of l, we would get a plot as shown in Figure 5.8. From this we can see that in a square of side k, as we increase the length of the link from $0.1k$ to $0.5k$, the number of links increases towards the center and is

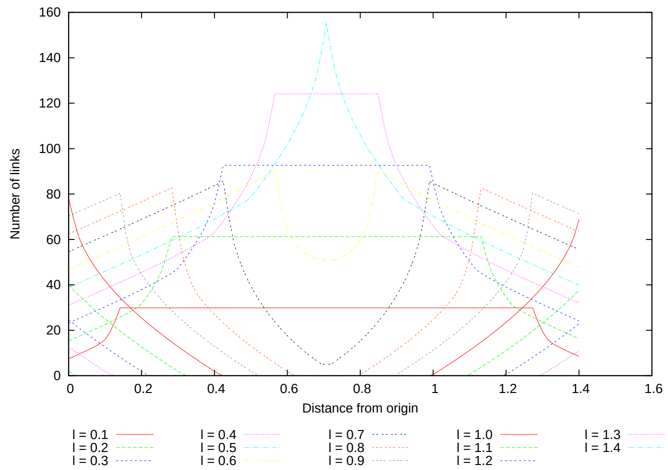


Figure 5.8: Number of links of specific lengths (100 × 100 grid) at different positions

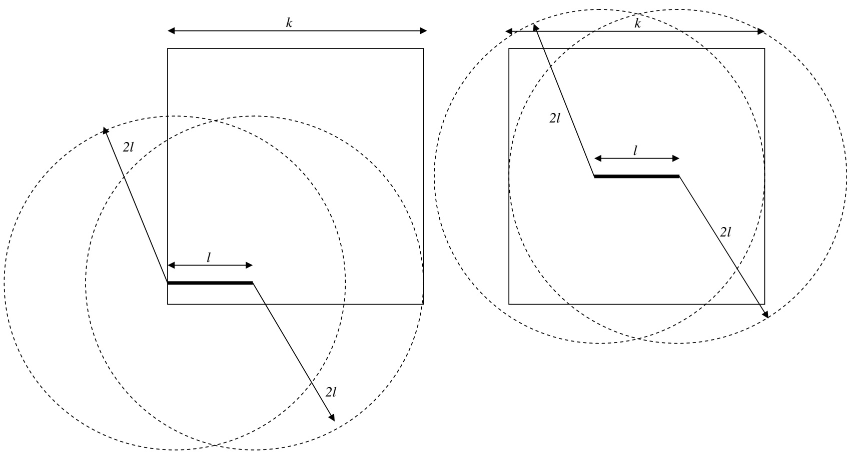


Figure 5.9: Link placed at different positions may or may not be a loner

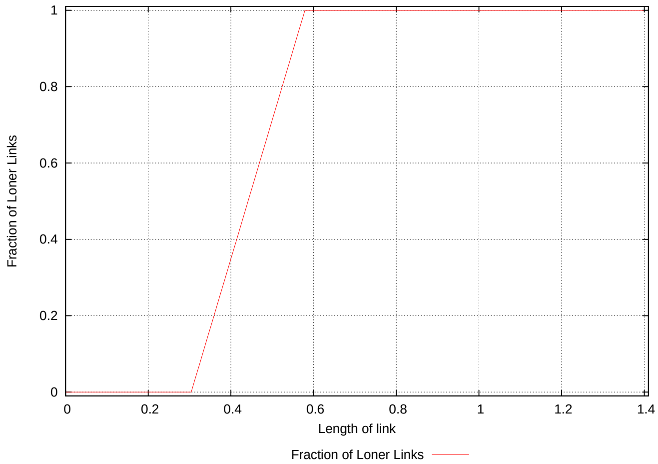


Figure 5.10: Estimating the fraction of links of any length that are loners

low towards the corner. Beyond $0.5k$, the number of links reaches a peak at a point closer to the origin and decrease as we move towards the center.

We already know that all links of length $0.579k$ or more are definitely loners. What about links of length l, $0.304k \leq l \leq 0.579k$? Some of these links will be loners depending on where they are positioned inside the square as seen in Figure 5.9. We can say that as the length of the link increases from $0.304k$ to $0.579k$, larger fractions of links are towards the center of the square and larger fractions are loners. If we plotted a graph of the fraction of links of a particular length that are loners, we know that it would flatten out at 1, for links $0.579k$ and more. We also know that it would remain at 0 till $0.304k$. And between these two values, the fraction would increase, making the curve look like the one in Figure 5.10.

Now that we have estimates of the total number of links and the fraction of them that are loners, for any given length, we can simply multiply these two values for every length and get the total number of loners in the network. Adding up the total number of links gives a value of 49984447 (actual number would be 49995000, using the formula $\frac{N(N-1)}{2}$, implying a 0.02% error in the calculation done using the overlapping area method) and the number of loners is 29497602 (actual number would be 30563246, by programmatically counting the loners). This means that in a 10,000 node grid 59% of the links will be loners.

What we are interested in finally is how these values affect the schedule length. Let us assume that the traffic requirements were such that every link needed to be scheduled for one slot. If 59% of the links are loners, the schedule length cannot be any lesser than

59% of the traffic matrix size. If we assume that the rest of the 41% links can be scheduled on an average with 2 links in a slot, the resultant schedule would be 79.5% of the traffic matrix size.

It has to be noted here, that in the calculation of the number of loners in the network, we have not taken the traffic matrix into account. How this would affect the count, is as follows. Let us take 2 links of length $l, l < 0.579k$, which are placed very close to each other. Their interference ranges would be overlapping in most places, leaving possibly the same links (l') out of the range, that can be scheduled in parallel with these links. Depending on the order in which these two links are scheduled, the first link would get l' in the same slot and the second links would become a loner since the traffic requirements of l' would have been satisfied earlier. Every time this happens, it would result in an increase of schedule length that was not accounted for in the earlier calculations. Therefore, the schedule lengths can be expected to be more than 79.5% in general.

5.2 Routing heuristic using topology control

A lot of prior work has been done in the area of routing in wireless networks in general. The main need for routing is for communication between nodes that are not within the direct communication range of each other. It could also be that the nodes are unaware of exact locations of other nodes. Here, with the simplifications that every node is within transmission range of every other node and that a central controller exists, routing can be used as tool to improve the throughput of the network.

By routing links with high interference scores on links which can be scheduled in parallel with other links. The basic idea is to reduce the number of transmissions that take place on loner links.

From Section 5.1 we know how to characterize some of the loner links in a given square area. Using this knowledge, we can perform some topology control and then route traffic on the new topology. A more detailed algorithm is presented below.

5.2.1 Algorithm

1: $N \leftarrow$ Number of nodes in the network

2: Construct T such that $T_{ij} \leftarrow$ Traffic between nodes i and j
3: Construct D such that $D_{ij} \leftarrow$ Distance between nodes i and j
4: Construct C such that $C_{ij} \leftarrow$ Cost of the link L_{ij}
5: Order L_{ij} in decreasing order of D_{ij}
6: **for all** L_{ij} in the ordering **do**
7:     **if** $D_{ij} > Limit_{TC}$ **then**
8:         Set $C_{ij} \leftarrow \infty$
9:         **if** Network is disconnected **then**
10:             Change C_{ij} back to its original cost
11:             Go to step 15
12:         **end if**
13:     **end if**
14: **end for**
15: **for** $i = 0, 1, \ldots, N-1$ **do**
16:     **for** $j = 0, 1, \ldots, N-1$ **do**
17:         **if** $C_{ij} = \infty$ **then**
18:             Compute the shortest path P_{ij} between nodes i and j
19:             **for all** L_{ab} in P_{ij} **do**
20:                 $T_{ab} \leftarrow T_{ab} + T_{ij}$
21:             **end for**
22:             Set $T_{ij} \leftarrow 0$
23:         **end if**
24:     **end for**
25: **end for**

5.2.2 Network Disconnection Check

1: $Connected \leftarrow \emptyset$
2: Add n_0 to $Connected$
3: $num_{visited} \leftarrow 1$
4: **while** $num_{visited} < N$ **do**
5:     $flag \leftarrow 0$
6:     **for all** $n_i \notin Connected$ **do**

if $C_{ij} < \infty$ for some $n_j \in Connected$ **then**
  Add n_i to $Connected$
  Increment $num_{visited}$
  $flag \leftarrow 1$
end if
end for
if $flag = 0$ **then**
  return $false$
end if
end while
return $true$

5.2.3 Details of the algorithm

- **Cost Matrix** - The cost can be either be a simple hop count or the interference score of the link. If hop counts are used, initially, all costs would be 1 as every node pair has direct links (paths of hop count = 1) between them.

- **Limit for the topology control ($Limit_{TC}$)** - The aim of the topology control phase is to eliminate loner links and the phase should end when all or most of the loner links have been removed. In a square of side k, this limit can be varied between 0 all the way up to k. Specific limits of interest would be:

  1. $Limit_{TC} = 0$ - A limit of 0 is akin to using the COMPOW philosophy of reducing power levels of the nodes until the network is just connected. With this limit, we try to remove all links, irrespective of their length, until network disconnection occurs. Depending on the network disconnection handling (see below), this could result in a topology where every link is required for connectivity.
  2. $Limit_{TC} = 0.304k$ - We know that links shorter than this limit are not going to be loner links. There would be links longer than this limit that are not loners, but we can simply opt to remove any link that has more than a 0 probability of being a loner, based on its length.
  3. $Limit_{TC} = 0.579k$ - Every link longer than this limit is definitely a loner. By choosing this value, we will be removing only such links.

4. $Limit_{TC} = 0.441k$ - The midpoint between the previous two extremes will be of interest to understand how the topology control will affect the routing.
5. $Limit_{TC} = k$ - A limit of k is akin to the DirectTrans philosophy, where every link in the network exists and can be used for communication.

- **Network Disconnection** - As we keep removing potential loner links, by setting the costs to ∞, the network might get disconnected leaving two sets of nodes that have no path between them. This can be handled in two different ways:

  1. Reset the cost of the last link that was removed and stop the topology control.
  2. Reset the cost of the last link that was removed and continue removing other links till $Limit_{TC}$ is reached.

- **Shortest Path Computation** - For the links that were removed during topology control, new paths can be computed using existing shorter links. A well known algorithm like Dijkstra's shortest path algorithm [3] [5] can be used for this purpose, with the cost matrix described earlier.

- **Scheduling** - After the traffic on the removed loner links has been rerouted over other existing links, the final transmission matrix needs to be scheduled. This can be done using one of the heuristic scheduling algorithms given in Chapter 4.

- **Sorting the Links** - In the shown algorithm, the links get sorted in descending order of lengths before the topology control is done. While the length of the links give some idea about how high their interference levels would be, it is not sufficient information. We can instead sort the links by their interference scores before doing topology control. The link costs can be set to infinity till network disconnection occurs or the length of the links reduces below $Limit_{TC}$.

5.3 Joint Routing and Scheduling

In section 5.2, a heuristic routing algorithm was described which utilized the knowledge gained from the analysis of the effect of link lengths on scheduling. The aim was to improve the throughput of the network by reducing the schedule length and to do this, we performed topology control by removing certain links that would have increased the schedule length. We now present a solution that jointly does routing and scheduling to satisfy the traffic requirements.

Let us first recall certain details from the scheduling heuristics of Chapter 4. When a link is chosen to be scheduled in the next slot of the schedule, we try to find other links that can be allocated to the same slot, based on some criterion (size of the traffic element, minimum unrejected interfering links, etc) decided by the scheduling algorithm. We try to find as many links as possible for each slot, until one of the following happens -

1. All remaining links in the network interfere with one or more of the links currently in the slot.
2. The links that do not interfere with any of those already in the slot, but simply do not have any unsatisfied traffic requirements.

It is the latter case that we are interested in. In every slot, we can check if there are links of this nature and create a new "auxiliary" graph with these links. This graph would probably be a disconnected graph with possibly more than one link between the same node pair. Once there are enough links in the auxiliary graph, we can route the traffic requirement of another node pair, using these links. If we select node pairs that would have formed loner links, for every route that we create using auxiliary links, we would be reducing the schedule length.

It has to be noted that the auxiliary path created cannot have two auxiliary links that are were available in the same slot, as these might interfere with each other. It should also be noted that the links in the auxiliary path may not all be of the same "capacity". Here, capacity of the auxiliary link, is basically the length of the slot in which it was available. The slots that were selected, do not have to be in order of the slot numbers, as the schedule repeats infinitely. What this means is as follows - if the auxiliary path created was $n_s \rightarrow n_i \rightarrow n_j \rightarrow n_d$, then the route would still work if the links L_{si}, L_{ij} and L_{jd} were available on slots 30, 20 and 10. Although the link L_{jd} would occur in the schedule before the other two links, as the same schedule repeats, it would be okay to have it this way.

The following section gives the algorithm for joint routing, built on top of the heuristic independent sets based scheduling algorithm.

5.3.1 Algorithm

1: $N \leftarrow$ Number of nodes in the network
2: Construct T such that $T_{ij} \leftarrow$ Traffic between nodes i and j
3: Construct D such that $D_{ij} \leftarrow$ Distance between nodes i and j
4: $aux_{ij} \leftarrow \emptyset$
5: Get an ordering of the elements of T
6: **while** $T_{ij} \neq 0$ i,j $= 0, 1, \cdots, N-1$ **do**
7: Get the first link, L_{sd} for current slot by *Criterion 1*
8: $T_{slot} \leftarrow T_{sd}$
9: Mark $Selected_{ij} \leftarrow 0$ for $i, j = 0, 1, \cdots, N-1$
10: **while** true **do**
11: Mark $Selected_{sd} \leftarrow 1$
12: **if** $T_{sd} < T_{slot}$ **then**
13: $T_{slot} = T_{sd}$
14: **end if**
15: **for** $i = 0, 1, \cdots, N-1$ **do**
16: **for** $j = 0, 1, \cdots, N-1$ **do**
17: **if** L_{sd} interferes with L_{ij} and $Selected_{ij} = 0$ **then**
18: Mark $Selected_{ij} = -1$
19: **end if**
20: **end for**
21: **end for**
22: Get the next link L_{sd} with $Selected_{sd} = 0$ by *Criterion 2*
23: **if** no such L_{sd} exists **then**
24: Go to step 29
25: **else if** $T_{sd} = 0$ **then**
26: Add num_{slot} to aux_{sd}
27: **end if**
28: **end while**
29: Add all L_{ij} such that $Selected_{ij} = 1$ to the current slot
30: **for** $k = 0, 1, \ldots N-1$ **do**
31: Compute shortest paths P_{kl} from n_k to n_l, $l \neq k$, using links L_{ab} available in

$slot_{ab}^{kl} \in aux_{ab}$, such that $slot_{ab}^{kl} \neq slot_{cd}^{kl} \in aux_{cd}$, $\forall L_{ab}, L_{cd} \in P_{kl}$.
if $D_{kl} > Limit_{TC}$ and $T_{kl} > 0$ **then**
  for all $L_{ab} \in P_{kl}$ **do**
    Add L_{ab} to the slot numbered $slot_{ab}^{kl}$
    Remove $slot_{ab}^{kl}$ from aux_{ab}
    for all $L_{a'b'}$, $a'b' \neq ab$, such that $slot_{ab}^{kl} \in aux_{a'b'}$ **do**
      if $L_{a'b'}$ interferes with L_{ab} **then**
        Remove $slot_{ab}^{kl}$ from $aux_{a'b'}$
      end if
    end for
  end for
end if
end for
Increment num_{slot}
end while

5.3.2 Details of the algorithm

1. $Limit_{TC}$ needs to be decided carefully. If the auxiliary links are used to route the traffic from non-loner links, then the schedule length might not decrease too much.

2. How often the shortest paths are computed in the auxiliary graph is a performance issue. Similarly, how many auxiliary paths are scheduled each time is something that can be varied.

3. In what order do we compute the shortest paths in and assign them to the auxiliary links? If we simply use node numbering to do this, then all the loner links from nodes numbered towards the end will not get rerouted as the auxiliary links would get utilized much before these paths are computed. It is indeed not clear if this would make a difference to the throughput, as rerouting one loner link should be the same as rerouting any other loner link.

4. The most intuitive cost for computing the shortest path is hop count. For each path, we want to use as few auxiliary links as possible, so that more loner links can get rerouted. It is again unclear if using hop counts will maximize this number. It might

be possible that by using a slightly longer path for one loner link, we can get an auxiliary path for another loner link, which would not have been possible had we chosen the shortest path for the earlier link.

5. The independent sets based scheduler uses *Criterion 1* (refer Section 4.3) for link selection in each slot. If we choose the maximum interference score option for *Criterion 1*, then all the loner links (which would have the highest interference scores in the network) will get scheduled first. At this point, no auxiliary graph would have been created and hence we will not be able to reroute the traffic requirements of any of the loner links. Hence using the minimum interference score option would be better for the joint routing and scheduling process. This would schedule shorter links initially, aiding the creation of an auxiliary graph with more links and once their traffic requirements are satisfied, they can be used to route the traffic of other longer links.

6. The shortest path computation can no longer be done by using Dijkstra's algorithm directly. It needs to be modified to keep track of which slot each link of the path is on.

5.4 DiffPow - Differentiated Power Control

DiffPow can be thought of a power control scheme that provides the flexibility to change the transmit power levels of each node individually, at every slot if required, independent of the traffic element currently being transmitted. The power control schemes given by COMPOW and DirectTrans would be subsets of the solutions possible from DiffPow.

5.4.1 The need for DiffPow

The COMPOW protocol is supposed to work well in uniform topologies and the proofs presented in [18] are asymptotic in nature. These results do not hold in real sized networks and traffic requirements. Simple hand-crafted examples to illustrate this are included in Appendix A. COMPOW seizes to work well in clustered or irregular topologies as it forces all nodes to communicate at the same power level. This has also been shown in [12].

5.4.2 Description

The principle behind DiffPow is to use different power levels for different transmissions. While holding on to the per-link-minimality condition of [13], it gives the flexibility of not using single direct transmissions for every traffic element in the matrix. Instead, we can now route the traffic over other nodes, even if the destination node was in fact within direct communication range of the source node. Once the routing has been done, every link in the resultant path is used for transmissions with the per-link-minimality condition.

5.4.3 Routing with DiffPow

Using the understanding gained from the analysis of loner links that affect schedule lengths, we developed routing heuristics that have been described in Sections 5.3 and 5.2. We can apply these routing algorithms and the resultant transmission matrix can be scheduled over the corresponding links, with power control described by DiffPow.

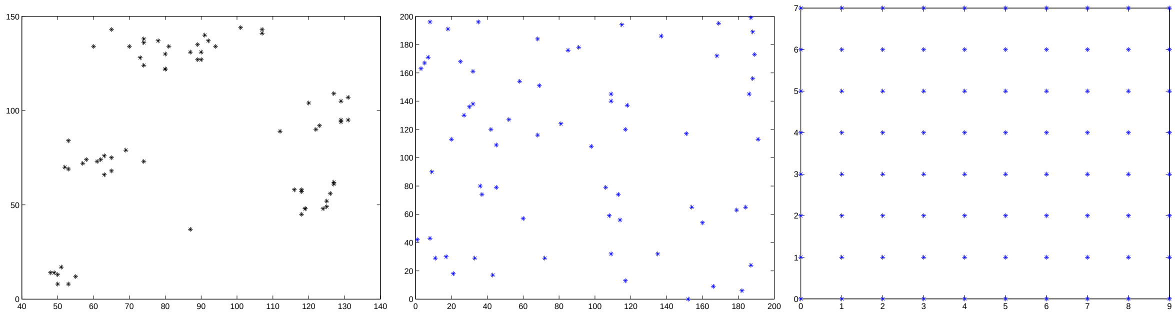
Chapter 6

Numerical Results

6.1 Network and traffic generation

A tool was written to generate different kinds of inputs for the routing and scheduling modules. The network generator takes input parameters from standard input. These parameters include:

- **Area of interest** - The shape of the area (circular, square, rectangular) and the size can be entered.
- **Type of topology** - The topology can either be random, grid or clustered. If a grid topology is chosen, another parameter is required for the space between adjacent nodes. Based on this and the size of the area, the number of nodes in the grid is determined.
- **Number of nodes** - A range of values can be specified and a random value in this range is chosen. If clustered topology is selected, number of clusters can also be specified apart from the total number of nodes.
- **Type of traffic** - The traffic can either be random, all-to-all or clustered. Random traffic implies a random fraction of links is chosen and a value is assigned to it. This value is again a random number whose maximum value can be specified. In clustered traffic, the ratio of intra-cluster traffic to inter-cluster traffic can be specified.



(a) Clustered topology (b) Random Topology (c) Grid Topology

Figure 6.1: Sample Topologies

Uniform random topologies are generated by repeatedly drawing random numbers in the specified X and Y ranges and assigning these coordinates to the nodes. If the chosen shape is a square or a circle, these random numbers can directly be used as coordinates, but if a circular topology is required, then for each point generated, it has to be checked whether or not the point lies within the circle.

For clustered topologies, first a set of pivot nodes are generated randomly based on the number of clusters required. Then the random coordinates are generated as before and then they are moved closer to the closest pivot. The amount by which they are moved closer can also be controlled randomly. This is known as the *Gravity Model*. Sample topologies are shown in Figure 6.1.

The output of the generator will consist of 3 variables - N, the number of nodes, T the traffic matrix, C a list of (x, y) coordinates for the N nodes.

6.2 Scheduling

The following sections show the schedule lengths got for different topologies and traffic patterns using the two different scheduling algorithms presented (along with their different flavors).

6.2.1 Inputs

Topology

- 7x7 Grid - 49 nodes, placed in a grid form with 3 units of distance between consecutive nodes.
- Random Cluster - 50 nodes distributed in clusters, with pivots randomly placed in a square of 50 units.
- Random topology - 50 nodes distributed randomly in a square of 21 units.

Traffic

- All 1s - Every node has 1 unit of traffic to every other node in the network.
- Random traffic - Of all the links in the network, a fraction (0.5) have traffic requirements. Each traffic requirement is a uniform random number in the range 0 to 10.
- Falling traffic - The amount of traffic on a link is inversely proportional to the length of the link.
- Clustered traffic - The traffic between nodes in the same cluster is higher than the that between inter-cluster nodes.

6.2.2 Simple Scheduler

The scheduler was run on different kinds of input topologies and traffic patterns. The graph in Figure 6.2 shows the results got. For every input type, 10 instances were generated. The average of the schedule lengths over all the instances along with the 95% confidence intervals are plotted in the graph, for each type of input. A value of 85 in the graph means that for this kind of topology and traffic, every 100 units of traffic requires 85 slots. As it can be seen, the schedule length for clustered topology with clustered traffic is far lower than for other inputs. This is because of the nature of the input, in that, since more traffic is between nodes that are very close to each other, it becomes possible to schedule more traffic in parallel. Topologies with falling traffic again result in shorter schedules.

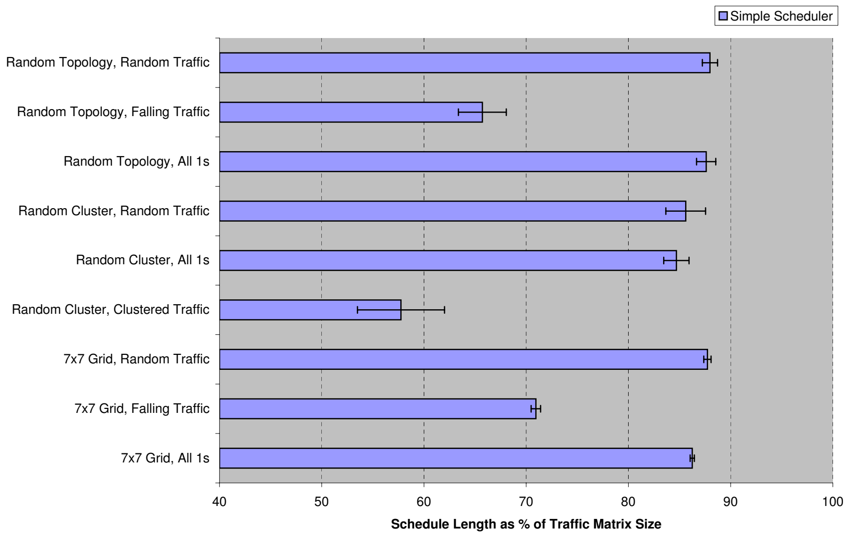


Figure 6.2: Schedule lengths using Simple Scheduler

6.2.3 Independent Sets Based Scheduler

To recall a few details about the independent sets based scheduler, different options for *Criterion 1* and *Criterion 2* can be used to decide on the links to be added to a given slot. The following graphs show a comparison between the performance of these options.

As seen in Figure 6.3, using the maximum interference score to choose the first link of the slot works better in most of the input types. This means that for every slot, the first link is chosen by ordering the links in descending order of their interference scores and choosing the first one that has unsatisfied traffic requirements in it. The other two options are only a little worse than this. Using minimum interference score would imply that the links are ordered in ascending order of interference scores. The third option, called "First Link" in the graph, means that the links are chosen simply by the node numbering, i.e., a link between nodes 1 and 2 is chosen for scheduling before the one between nodes 40 and 45. For the calculation of these values, *Criterion 2* was fixed as the minimum difference option.

Figure 6.4 shows a comparison of how consecutive links in the slot are chosen. The options "Maximum Interference Score" and "Maximum Overlap and Minimum Difference" perform better than "Minimum Difference". For a description of these options, please refer

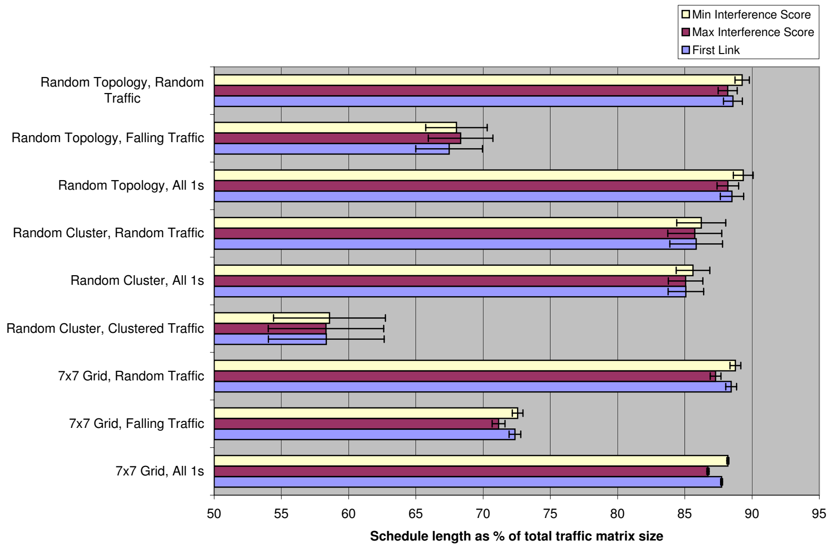


Figure 6.3: Performance of the different options for *Criterion 1*

to section 4.3.4. For these calculations, *Criterion 1* was fixed as the maximum interference score option.

From earlier sections, we know that the interference score is an indicator of the number of links that this link interferes with. It is weighted by the magnitude of traffic requirements on each of those links. As the scheduler is computing the slots, traffic requirements keep getting completed. This would mean that if the interference scores are recomputed after some of the slots were computed, these scores would be different from the ones calculated initially. And these new scores would be a better indication of the current network status. So, logically, we would need to compute the interference scores of the links after every slot is created. Would this really make a difference in the final schedule length? Figures 6.5(a) and 6.5(b) show the schedule lengths got with and without the repeated calculation of interference scores when using maximum and minimum interference scores. It can be seen that the performance gain by repeating the calculation is minimal, although the addition in actual computation in the program is of an order of 4 for each slot.

Now, we compare the schedule lengths got from the two scheduling heuristics. Figure 6.6 shows the values. For the independent sets scheduler, maximum interference score was used for *Criterion 1* and maximum overlap with minimum difference was used for *Criterion 2*. While the independent sets scheduler outperforms the simple scheduler in

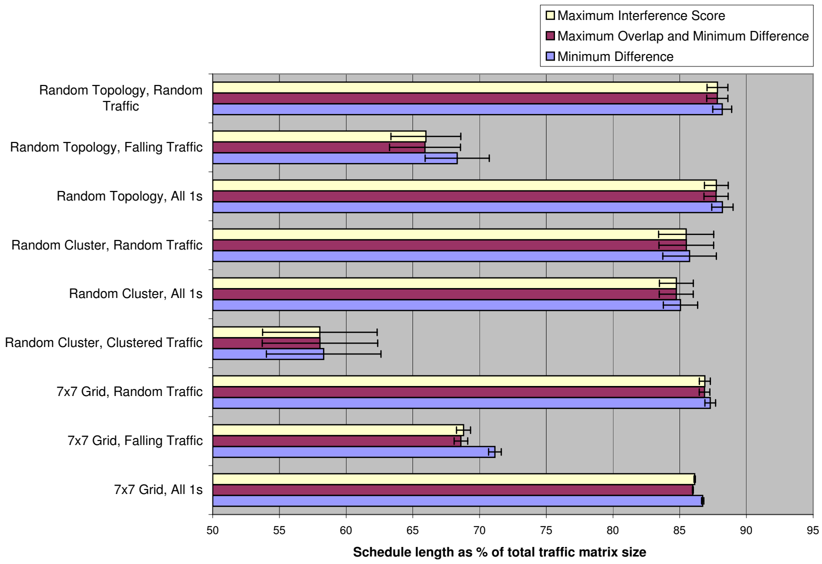
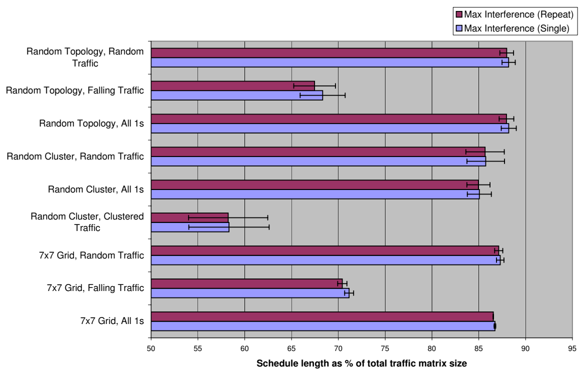
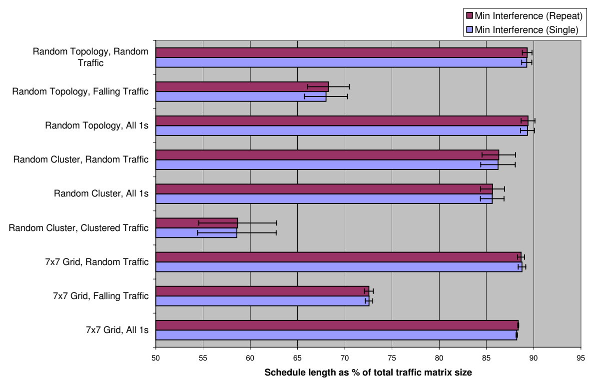


Figure 6.4: Performance of the different options for *Criterion 2*



(a) Maximum Interference Score



(b) Minimum Interference Score

Figure 6.5: Performance difference with repeated interference score computation

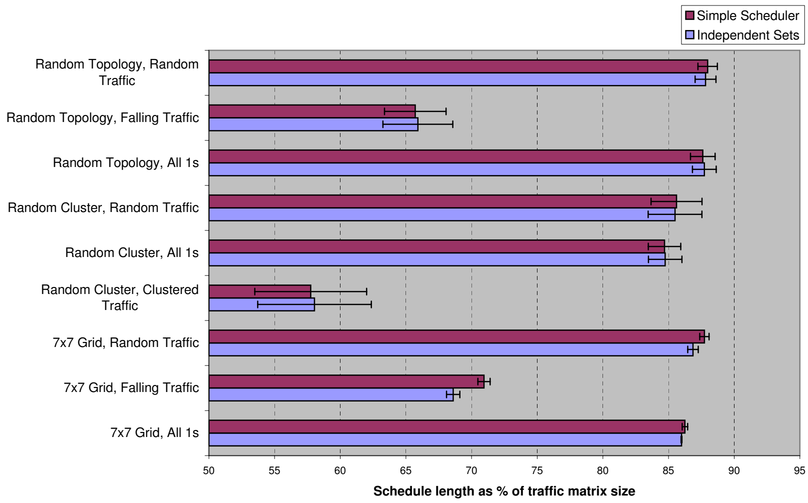


Figure 6.6: Comparison of the two scheduling heuristics

some cases, the difference is again not very large.

Schedule lengths vary with the topology and traffic type, this has been established by the results shown earlier. One of the important characteristics of the traffic pattern is the fraction of links of the network that are carrying traffic. As we increase this fraction, the number of loner links increases, adding to the schedule length. At the same time, the number of very short, non-loner links also increases. Would the increase in schedule length be linear in this case? It indeed is, as shown in figure 6.7. While the actual schedule length increases linearly, the ratio of the schedule length to the size of the traffic matrix remains in the same range. These values are for a 50 node random topology with random traffic.

While the fraction of active links is an important characteristic of the traffic, the density of the nodes is an important characteristic of the topology of the network. If the nodes are randomly distributed, as opposed to a clustered distribution, sparse networks would imply that there are no links that are very short. This would in turn imply that there is lesser parallelization possible in the schedule. Schedule lengths are shown in Figure 6.8.

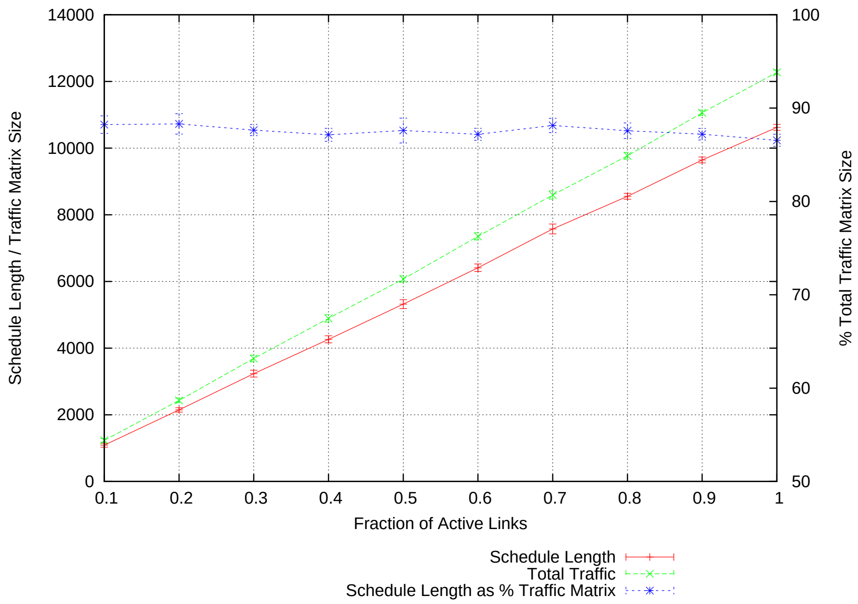


Figure 6.7: Variation of schedule length with fraction of active links

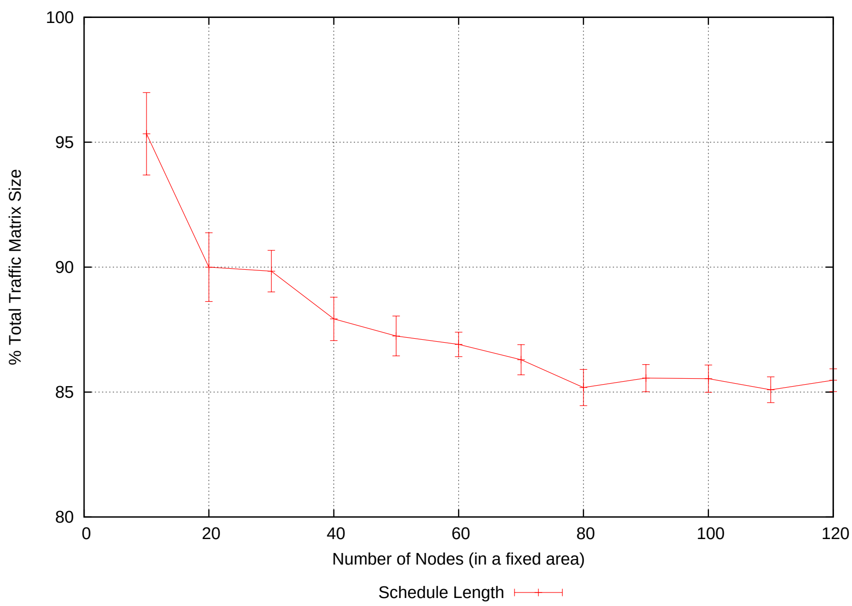
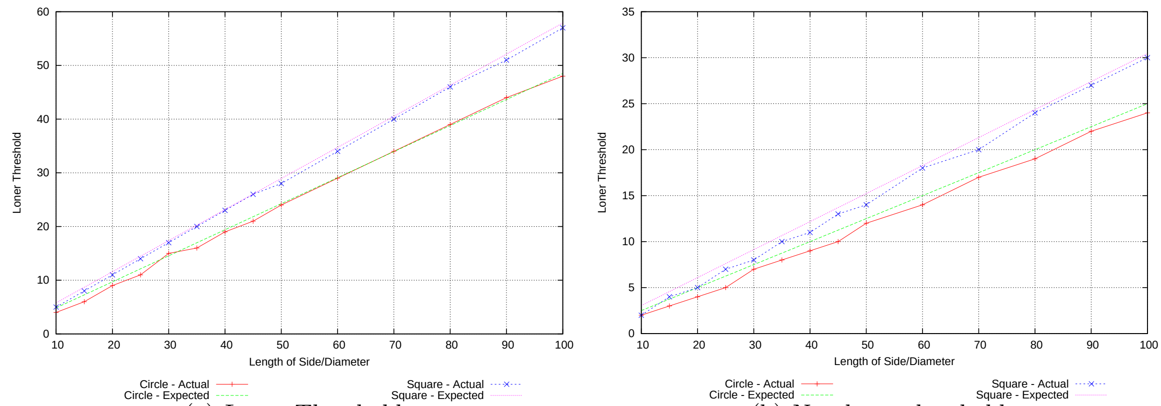


Figure 6.8: Variation of schedule length with density of nodes



(a) Loner Threshold (b) Non-loner threshold

Figure 6.9: Comparing theoretical and actual threshold values for loners and non-loners

6.3 Link Characterization

The lengths of links that are loners and non-loners was calculated by some geometrical arguments in earlier sections. These values for loners and non-loners can be verified by programmatically counting the links in a given topology. This was done for grid topologies of various sizes and plotted. As shown in Figure 6.9(a) and Figure 6.9(b), the values match reasonably well.

In section 5.1.4, we saw how to estimate the number of links of a particular length and also showed the estimated values for a 100×100 grid. Figure 5.10 showed the estimated fraction of these links would be loners. We now see how close these values were to what is got when the links are programmatically counted. Shown in Figure 6.10 are the total number of links of every length, number of loner links of every length and the ratio of the number of loner links to total number of links of every length.

6.4 Routing

6.4.1 Routing heuristic using topology control

Figure 6.11 shows the schedule lengths got for different inputs, using the routing heuristic described in Section 5.2. As in the earlier results, the numbers depict the schedule length as a percentage of the total traffic matrix size. The topology control was done with

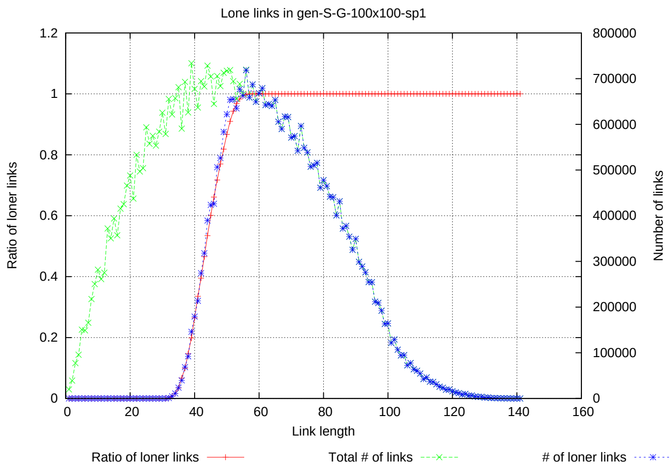


Figure 6.10: Loner and non-loner links in a 100 $\times$ 100 grid

different values of $Limit_{TC}$ and the shortest path algorithm was run with both hops and interference score as cost for the links. As it can be seen, the routing heuristic does not perform well in any of the cases, for any value of $0 \leq Limit_{TC} < 1$. This could mean that the topology control was not removing the right links or the routing algorithm was not choosing the right paths.

To try a different way of doing topology control, the links were then ordered by interference cost. The results are shown in Figure 6.12. Here, deletion of the links was stopped when either network disconnection occurred or the length of the links reduced below the specified $Limit_{TC}$. A few cases gave slightly improved results, although very marginal, for $Limit_{TC} < 1$. This goes on to show that while it is probably necessary to reroute the traffic on high interference links, simply going by a shortest path approach will not yield better throughput. As it is computationally prohibitive to present graphs, we show the results as tables in Figures 6.12 and 6.11.

6.4.2 Joint Routing and Scheduling

Joint routing and scheduling performs better than doing direct transmissions with either of the scheduling heuristics. However, the improvement is not very large. Schedule lengths are shown in Figure 6.13.

This routing scheme uses shorter links that have no traffic requirements or whose

Cost	$Limit_{TC}$	7x7 Grid, All 1s	7x7 Grid, Falling traffic	7x7 Grid, Random traffic
	1	86.564626	69.509704	87.788243
Intf score	0.304	109.94898	80.62649	104.952907
	0.579	116.964286	81.23936	119.990257
	0.4415	122.363946	86.125298	123.124391
	0	115.263605	80.949949	121.711595
Hops	0.304	147.02381	102.400409	156.154596
	0.579	127.593537	91.811372	139.395908
	0.4415	140.646259	98.348655	148.749594
	0	161.096939	101.327886	159.483599

Cost	$Limit_{TC}$	Random Cluster, Clustered traffic	Random Cluster, All 1s	Random Cluster, Random traffic
	1	54.066308	85.591837	88.295491
Intf score	0.304	69.424766	213.102041	221.874001
	0.579	59.138566	116.938776	115.76591
	0.4415	69.424766	177.469388	143.716022
	0	69.339756	191.673469	166.549408
Hops	0.304	69.028053	217.306122	232.395267
	0.579	61.88722	128.897959	129.836904
	0.4415	70.331539	183.102041	156.060122
	0	69.821479	195.877551	183.114807

Cost	$Limit_{TC}$	Random topology, All 1s	Random topology, Falling Traffic	Random topology, Random Traffic
	1	88.734694	67.183423	88.284043
Intf score	0.304	153.918367	85.34152	141.761456
	0.579	114	74.750576	114.78713
	0.4415	128.122449	80.752111	125.869353
	0	195.142857	86.600153	151.738707
Hops	0.304	183.632653	96.485035	162.577186
	0.579	137.061224	80.245587	128.664283
	0.4415	157.673469	89.424405	147.318817
	0	207.428571	104.727552	169.938252

Figure 6.11: Sorting the links by length for topology control

Cost	$Limit_{TC}$	7x7 Grid, All 1s	7x7 Grid, Falling traffic	7x7 Grid, Random traffic
	1	86.564626	69.509704	87.788243
Intf score	0.304	126.488095	87.946885	121.338097
	0.579	86.479592	69.305414	87.528418
	0.4415	88.052721	69.084099	89.623254
	0	118.579932	92.747702	122.962001
Hops	0.304	130.569728	98.467824	141.506983
	0.579	86.437075	69.220293	87.528418
	0.4415	90.731293	70.224719	91.149724
	0	132.015306	105.379639	140.02923

Cost	$Limit_{TC}$	Random Cluster, Clustered traffic	Random Cluster, All 1s	Random Cluster, Random traffic
	1	54.066308	85.591837	88.295491
Intf score	0.304	63.077359	176.367347	148.784778
	0.579	53.584585	85.591837	88.34346
	0.4415	54.80306	87.387755	88.34346
	0	63.077359	176.367347	148.784778
Hops	0.304	64.805894	173.795918	158.394627
	0.579	53.952961	86.653061	88.34346
	0.4415	56.134882	87.877551	88.34346
	0	64.805894	173.795918	158.394627

Cost	$Limit_{TC}$	Random topology, All 1s	Random topology, Falling Traffic	Random topology, Random Traffic
	1	88.734694	67.183423	88.284043
Intf score	0.304	95.673469	69.608596	131.475463
	0.579	88.612245	67.075979	88.41404
	0.4415	89.061224	67.168074	88.251544
	0	152.816327	141.596316	161.423464
Hops	0.304	109.061224	76.331543	150.666233
	0.579	89.346939	67.352264	89.015275
	0.4415	91.428571	67.122026	89.632759
	0	160.285714	150.161167	166.103347

Figure 6.12: Sorting the links by interference score for topology control

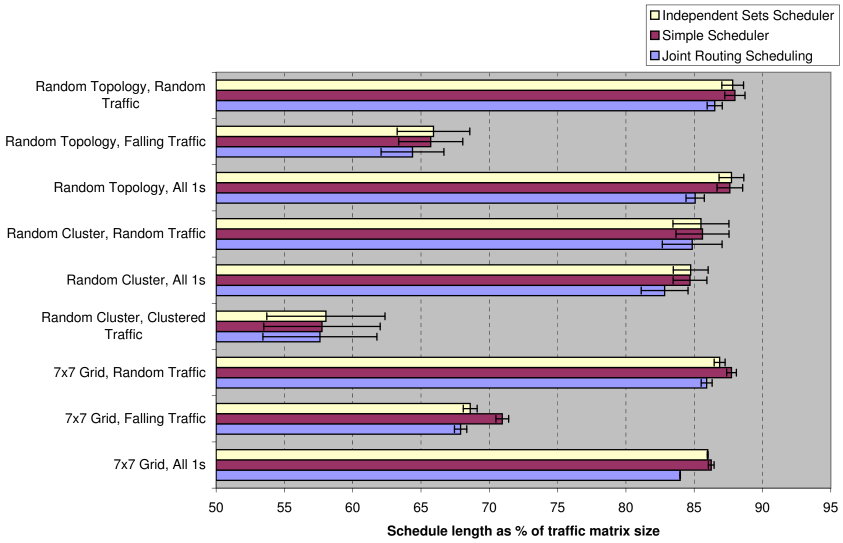


Figure 6.13: Comparing joint routing and scheduling with the other scheduling heuristics

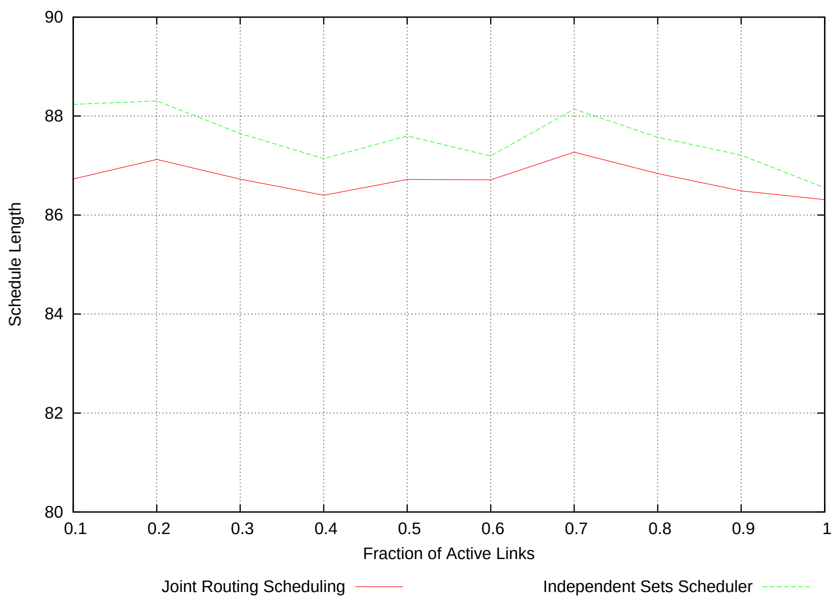


Figure 6.14: Comparing performance of direct transmissions with joint routing scheduling, as fraction of active links increases

requirements have already been satisfied, to route the traffic of longer links. If the traffic requirements were very high, there would be fewer links to use for routing and the improvement in throughput would reduce. This is seen in Figure 6.14 which compares schedule lengths got using the joint scheme and the earlier scheduling heuristic. For lower fractions of active links, the improvement got by using the joint scheme is higher. The improvement reduces as this fraction is increased.

Chapter 7

Conclusion

In this work, we looked at the joint problem problem of routing, scheduling and power control in wireless mesh networks. We presented two scheduling heuristics and studied their performance. We also studied the effect of inter-node distances on schedule lengths and showed that a majority of the links in a network are such that they cannot be scheduled in parallel with any other link. This results in longer schedules, thus reducing the throughput of the network. We also presented a routing heuristic that performs topology control and then applies the shortest path algorithm. While this routing algorithm in general did not give throughput improvements, a more novel combined routing and scheduling scheme did perform better. This joint scheme constructs an auxiliary graph containing links that could have been scheduled in existing slots. By using this graph to route certain traffic requirements, the overall schedule can be shortened. Though this scheme works better than the earlier ones, it needs to be studied better in certain aspects like choosing the right link to schedule using auxiliary graph, choosing a globally optimal path, etc. Concepts from minimum interference routing schemes of wired networks could possibly be used to improve the routing.

Bibliography

[1] http://www.cisco.com/en/uS/products/hw/wireless/ps430/index.html. *Cisco Systems Product Literature.*

[2] http://www.rfcafe.com/references/electrical/path_loss.htm. *R F Cafe*, 2000.

[3] http://en.wikipedia.org/wiki/Dijkstraś_algorithm. *Wikipedia*, 2006.

[4] http://episteme.arstechnica.com/eve/forums/a/tpc/f/6330927813/m/950006129731. *Ars Technica Open Forum*, 2006.

[5] http://www.nist.gov/dads/HTML/dijkstraalgo.html. .

[6] Anthony Ephremides and Thuan V. Truong. Scheduling broadcasts in Multihop Radio Networks. *IEEE Transactions On Communications,*, 38, 1990.

[7] R.L. Cruz and A.V. Santhanam. Optimal routing, link scheduling and power control in multihop wireless networks. *INFOCOM Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, 1:702–711, 2003.

[8] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking*, 2003.

[9] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking*, 2003.

[10] T. ElBatt and A. Ephremides. Joint Scheduling and Power Control for Wireless Ad hoc Networks. *Wireless Communications, IEEE Transactions on*, 3:74–85, 2004.

[11] Jangeun Jun, Pushkin Peddabachagari, and Mihail Sichitiu. Theoretical Maximum Throughput of IEEE 802.11 and its Applications. *Second IEEE International Symposium on Network Computing and Applications*, 2003.

[12] V. Kawadia and P.R. Kumar. Power Control and Clustering in Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 23:76– 88, 2005.

[13] R. Khalaf and I. Rubin. Enhancing the throughput-delay performance of IEEE802.11 based networks through direct transmissions. *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, 4:2912– 2916, 2004.

[14] A. Muqattash and M. Krunz. POWMAC: A Single-channel Power-control Protocol for Throughput Enhancement in Wireless Ad hoc Networks. *IEEE Journal on Special Areas in Communication*, 23:1067– 1084, 2005.

[15] Piyush Gupta and P R Kumar. Capacity of Wireless Networks. *IEEE Transactions On Information Theory*, 46, 2000.

[16] I. Rubin, R. Khalaf, A. Moshfegh, and A. Behzad. Delay-throughput Performance of Load-adaptive Power-controlled Multihop Wireless Networks with Scheduled Transmissions. *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, 5:3060– 3064, 2003.

[17] Shu Huang and Rudra Dutta. Design of Wireless Mesh Networks under the Additive Interference Model. *to appear at International Conference on Computer Communications and Networks*, 2006.

[18] Swetha Narayanaswamy, Vikas Kawadia, R. S. Sreenivas, and P. R. Kumar. Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol. In *European Wireless Conference*, 2002.

[19] Yongqiang Xiong, Qian Zhang, Feng Wang, and Wenwu Zhu. Power assignment for throughput enhancement (PATE): a distributed topology control algorithm to improve throughput in mobile ad-hoc networks. *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, 5:3015– 3019, 2003.

Appendices

Appendix A

Specific Counter Examples for COMPOW

Consider a topology as shown in Figure A.1, with two clusters of nodes at the specified coordinates. We will look at the achievable routing and scheduling for this topology under different traffic patterns.

The power levels considered here are for Cisco Aironet 1200 series got from the data sheet on their web site [1]. There are 5 levels possible for 802.11g - 30mW, 20mW, 10mW, 5mW and 1mW. At the (presumably) maximum power level, we get a range of around 200m outdoors for a data rate of 6Mbps. Based on the Friis Equation [2], the trans-

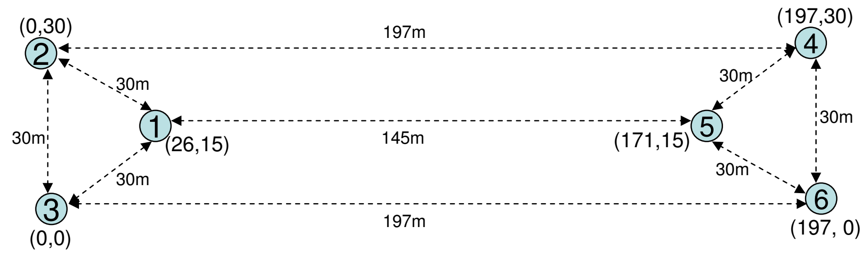


Figure A.1: Network topology

Table A.1: Transmission ranges

Power	Range
30mW	200m
20mW	163.26m
10mW	115.44m
5mW	81.63m
1mW	36.50m

mission ranges at different power levels are got as follows:

The following discussions use these range values.

A.0.3 Uniform Traffic Matrix

With a uniform traffic matrix like the one shown below, all nodes transmit to all the other nodes in the network.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The throughput that can be achieved with this network and traffic under different transmission powers is given here.

Common Power Level (maximum)

When all the nodes transmit at the maximum power level, the topology is fully connected as shown in Figure A.2.

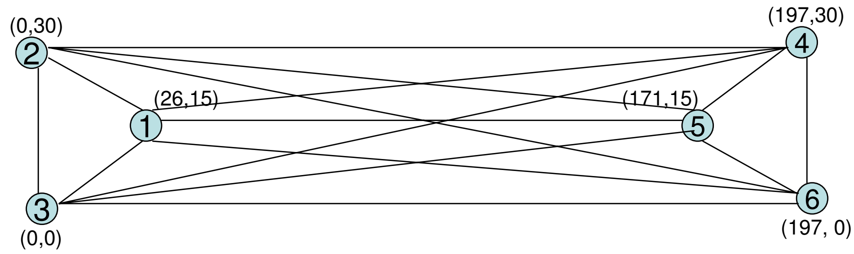


Figure A.2: Topology at maximum power level

- *Interference*

  As all nodes will be able to transmit directly to every other node, this means that every link in the network will interfere with every other link.

- *Routing*

  All source nodes transmit directly to all the destination nodes.

- *Scheduling*

  Since every transmission interferes with every other transmission, at any given time, we can schedule only one single transmission. This means that there is no parallel transmissions happening anywhere in the network and every '1' in the traffic matrix will require a slot of its own.

  As there are 30 traffic elements that need to be scheduled, we will therefore require *30 slots* to have a complete schedule.

Common Power Level (minimum)

When all the nodes transmit at a minimum common power level at which the topology is just connected (COMPOW), the network will look as shown in Figure A.3. In the sample power levels considered, the nodes would need to transmit at 20mW to keep the network connected. The transmission range in this case will be around 160m. There are no links other than the ones shown, as 2-4, 2-5, etc are all greater than 160m.

- *Interference*

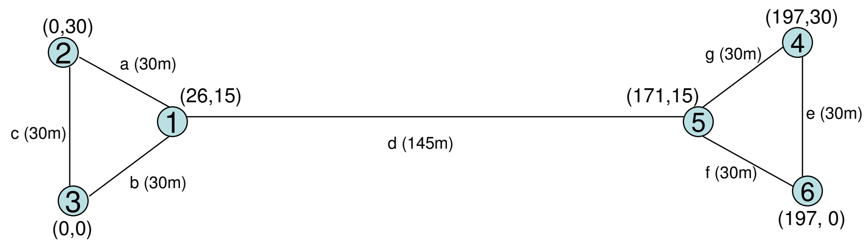


Figure A.3: Topology at common minimum power level

Since the interference range is assumed to be twice the transmission range, every link will interfere with every other link as all nodes are within 320m (twice of 160m) of each other.

- *Routing*

  Wherever possible, it is advantageous for us to have direct transmissions here. If we chose multi-hop transmissions over direct ones, we would be increasing the length of the schedule as only one transmission can be scheduled at any given time. And to complete one traffic element from the matrix, we would then require as many slots as the length of the path in hops.

  Direct Routes: 1-2, 1-3, 1-5, 2-3, 4-5, 4-6, 5-6

  Multihop Routes: 1-5-4, 1-5-6, 2-1-5, 2-1-5-4, 2-1-5-6, 3-1-5-4, 3-1-5, 3-1-5-6

  The multi-hop routes are chosen with the intention of having shortest paths possible between the source-destination pairs for the same reasoning as above. If a longer path is chosen, it would increase the number of slots required for the traffic element to be transmitted.

- *Scheduling*

  With the routing given above, to complete the traffic matrix, 54 transmissions will be required. To satisfy the traffic requirement from 2-3, we need one transmission. For the nodes 2-4, we need 3 transmissions as the route is 2-1-5-4. In this power control scheme, since every transmission requires a slot of its own, we need *54 slots* to complete the schedule.

Differing Power Level - Scheme 1

The next scheme we consider is one where links in the network have power levels associated with it. These power levels are fixed initially and are not changed for each transmission. Let us consider a specific mapping of node pairs and power levels, that results in the topology shown in Figure A.3. The topology here will be same as that for the previous power scheme, but the interference is different. Here, all node pairs except 1-5 operate at 1mW (transmission range of 36m) and 1-5 use 20mW (transmission range of 20mW). This 1-5 link makes the network connected.

- *Interference*

  Based on the chosen power levels and the distances between the nodes in the topology, the following pairs of nodes would interfere if they had transmissions occurring at the same time.
  (a, b) (a, c) (a, d) (b, c) (b, d) (c, d)
  (d, e) (d, f) (d, g) (e, f) (e, g) (f, g)

- *Routing*

  Again, whenever possible direct transmission should be used. If instead of 1-3, we use 1-2-3, we would need two slots to complete this as 1-2 (link a) and 2-3 (link c) form an interfering pair. Within the clusters (nodes [1, 2, 3] and [4, 5, 6]), parallel transmissions cannot be scheduled due to interference. So we must make sure the the lengths of the routes within a single clusters are the shortest ones possible.

  Direct Routes: 1-2, 1-3, 1-5, 2-3, 4-5, 4-6, 5-6

  Multihop Routes: 1-5-4, 1-5-6, 2-1-5, 2-1-5-4, 2-1-5-6, 3-1-5-4, 3-1-5, 3-1-5-6

- *Scheduling*

  Based on the chosen routing, the loads on each link of the network is as follows to completely satisfy the traffic matrix.

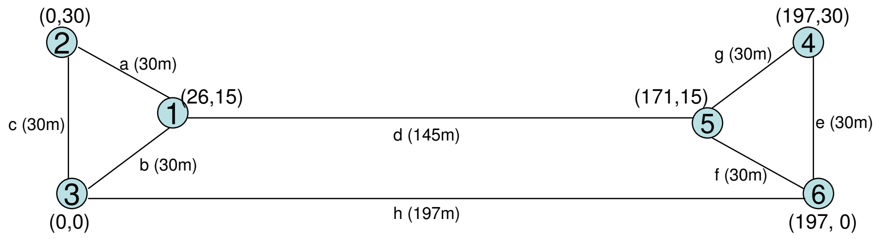


Figure A.4: Topology with power scheme - 2.

a	$-$	8
b	$-$	8
c	$-$	2
d	$-$	18
e	$-$	2
f	$-$	8
g	$-$	8

Here, link a is carrying the traffic elements from 1-2, 2-1, 2-4, 4-2, 2-5, 5-2, 2-6, 6-2. Based on the interference patterns, we can schedule the following links in parallel for the specified units of time to complete the load requirements:

Links	*Slots*
(a, g)	8
(b, f)	8
(c, e)	2
(d)	18

No parallel transmissions can occur other than these, i.e., (a, b) cannot be scheduled together. (a, e) can be scheduled together, but it would not be optimal as link e has only two units of traffic to sent and this would increase the length of the schedule. With this schedule, we require *36 slots* to complete the traffic requirements.

Differing Power Level - Scheme 2

Here we consider the same power control scheme as before, but with a different mapping of power level to the node pairs. We change the power level of 3-6 from 1mW to

20mW to get the topology shown in Figure A.4.

- *Interference*

  Based on the chosen power levels and the distances between the nodes in the topology, the following pairs of nodes would interfere if they had transmissions occurring at the same time.

  (a,b) (a,c) (a,d) (a,h) (b,c) (b,d) (b,h) (c,d) (c,h)
  (d,e) (d,f) (d,g) (d,h) (e,f) (e,g) (e,h) (f,g) (f,h)

- *Routing - 1*

  Since now we have multiple routes available between different node pairs, we will look at a few different route sets. The first one here is simply based on the shortest length.

  Direct Routes: 1-2, 1-3, 1-5, 2-3, 3-6, 4-5, 4-6, 5-6

  Multihop Routes: 1-5-4, 1-5-6. 2-1-5-4, 2-1-5, 2-3-6, 3-6-4, 3-6-5
  (46 transmissions)

- *Scheduling - 1*

  Based on the chosen routing, the loads on each link of the network is as follows to completely satisfy the traffic matrix.

  a – 6
  b – 2
  c – 4
  d – 10
  e – 4
  f – 6
  g – 6
  h – 8

  Based on the interference patterns, we can schedule the following links in parallel for the specified units of time to complete the load requirements:

Links	*Slots*
(a, g)	6
(b, f)	6
(c, e)	4
(d)	10
(h)	8

This results in a schedule of *34 slots*, giving a slight improvement over the previous scheme.

- *Routing - 2*

  Let us now consider another routing scheme which is very close to the routing used for Differing Power Level Scheme 1. The only difference is that 3-6 is a direct link.

  Direct Routes: 1-2, 1-3, 1-5, 2-3, 3-6, 4-5, 4-6, 5-6

  Multihop Routes: 1-5-4, 1-5-6, 2-1-5, 2-1-5-4, 2-1-5-6, 3-1-5-4, 3-1-5
  (50 transmissions)

- *Scheduling - 2*

  Based on the chosen routing, the loads on each link of the network is as follows to completely satisfy the traffic matrix.

  a – 8
  b – 6
  c – 2
  d – 16
  e – 2
  f – 6
  g – 8
  h – 2

  Based on the interference patterns, we can schedule the following links in parallel for the specified units of time to complete the load requirements:

Links	*Slots*
(a, g)	8
(b, f)	6
(c, e)	2
(d)	16
(h)	2

This results in a schedule of *34 slots* again, although the lengths of some of the routes were not the shortest ones possible. We can say that this gain of 2 slots over power scheme 1, is only because of the change of 3-1-5-6 to 3-6.

Here is some further analysis. Let us consider links d and h. These are used only when nodes across clusters need to communicate. There are 18 traffic elements where the source and destination are from different traffic clusters. As long as we use links d and h only for these 18 elements, it does not matter which of the two we use (they are anyway scheduled separately and not combined with any other link).

Coming to the links within the clusters, like argued before, if we use 1-2-3 instead of 1-3, we will be increasing the length of the route within the cluster and hence should be avoided.

In the routing given here, there is a notion of ”counterparts” of links across clusters. Links a and g are counterparts, b and f are counterparts, c and e are counterparts. And in this routing, they are used symmetrically.

- *Routing - 3*

  Direct Routes: 1-2, 1-3, 1-5, 2-3, 3-6, 4-5, 4-6, 5-6

  Multihop Routes: 1-5-4, 1-5-6, 2-1-5, 2-1-5-4, 2-3-6, 3-1-5-4, 3-1-5
  (48 transmissions)

- *Scheduling - 3*

  Loads on each link would be:

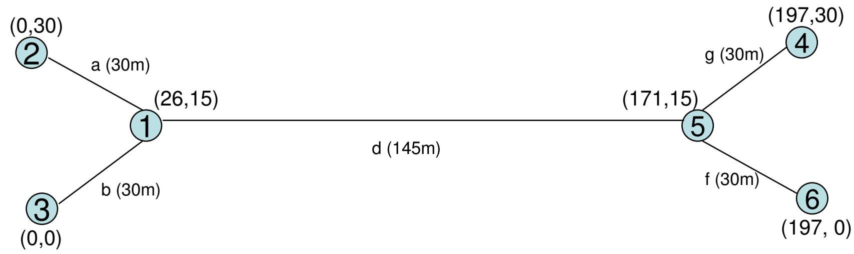


Figure A.5: Topology with power scheme - 3

a	–	6
b	–	6
c	–	4
d	–	14
e	–	2
f	–	4
g	–	8
h	–	4

Based on the interference patterns, we can schedule the following links in parallel for the specified units of time to complete the load requirements:

Links	*Slots*
(a, e)	6
(b, g)	8
(c, f)	4
(d)	14
(h)	4

This results in a schedule of 36 slots.

Differing Power Level - Scheme 3

Here we consider yet another different mapping of power levels to the node pairs. This scheme is using the minimum possible powers in every node, though all the power levels are not the same. As seen in Figure A.5, if any of the power levels are reduced any further, the network would become disconnected.

- *Interference*

  Based on the chosen power levels and the distances between the nodes in the topology, the following pairs of nodes would interfere if they had transmissions occurring at the same time.

  (a, b) (a, d) (b, d) (d, f) (d, g) (f, g)

- *Routing*

  Because of the nature of the topology, there are are no alternate routes for any source destination pairs.

  Direct Routes: 1-2, 1-3, 1-5, 4-5, 5-6

  Multihop Routes: 1-5-4, 1-5-6, 2-1-3, 2-1-5-4, 2-1-5, 2-1-5-6, 3-1-5-4, 3-1-5, 3-1-5-6, 4-5-6

- *Scheduling*

  Based on the chosen routing, the loads on each link of the network is as follows to completely satisfy the traffic matrix.

  a – 10
  b – 10
  d – 24
  f – 10
  g – 10

  Based on the interference patterns, we can schedule the following links in parallel for the specified units of time to complete the load requirements:

Links	*Slots*
(a, g)	10
(b, f)	10
(d)	24

  This results in a schedule of *44 slots*, which is still better than the common minimum scheme.

Direct Transmissions

We can also do away with routing and simply adopt direct transmissions for every traffic element in the traffic matrix, using just enough power to reach the destination in each case. Links (a, g), (b, f) and (c, e) can be scheduled in parallel. So the 30 traffic elements will be scheduled in *24 slots*.

From all these different routing and power control schemes that were tried, it is seen that COMPOW does not work well in non-uniform topologies. There is a need for a differentiated power scheme that also takes the traffic pattern into consideration. In this case, it turns out that the direct transmission scheme with the per-link-minimality condition works best.

A.0.4 Non-Uniform Traffic Matrix

We now consider the same topology with a different traffic matrix.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Common Power Level (minimum)

Power levels will be as in the common minimum power scheme discussed earlier. Hence, the routing and interference will also be the same as earlier. They have been given here again for convenience.

- *Interference*

  Since the interference range is assumed to be twice the transmission range, every link will interfere with every other link as all nodes are within 320m (twice of 160m) of each other.

- *Routing*

  Direct Routes: 1-2, 1-3, 1-5, 2-3, 4-5, 4-6, 5-6

Multihop Routes: 1-5-4, 1-5-6, 2-1-5, 2-1-5-4, 2-1-5-6, 3-1-5-4, 3-1-5, 3-1-5-6

- *Scheduling*

  All the traffic elements in the matrix can be satisfied by direct transmissions. We therefore require *14 slots*, one slot for each element, to complete the schedule.

Common Power Level (maximum)

When the nodes are transmitting at the maximum power levels, the network is fully connected and all links interfere with each other. All the transmissions happen through direct links and no parallel transmissions are possible. We therefore require *14 slots*, one slot for each traffic element, to complete the schedule.

As we can see, for this traffic matrix, both these schemes give the same performance.

Differing Power Level - Scheme 1

The interference and routing will be the same as described for scheme 1 earlier.

- *Interference*

  Based on the chosen power levels and the distances between the nodes in the topology, the following pairs of nodes would interfere if they had transmissions occuring at the same time.

  (a, b) (a, c) (a, d) (b, c) (b, d) (c, d)
  (d, e) (d, f) (d, g) (e, f) (e, g) (f, g)

- *Routing*

  As argued earlier, the best routes would be:

  Direct Routes: 1-2, 1-3, 1-5, 2-3, 4-5, 4-6, 5-6

  Multihop Routes: 1-5-4, 1-5-6, 2-1-5, 2-1-5-4, 2-1-5-6, 3-1-5-4, 3-1-5, 3-1-5-6

- *Scheduling*

  Here, the traffic matrix is such that we require only the direct links in the topology for communication. Each link (a - g) has a load of 2 units. Parallel transmissions can be scheduled on:

Links	*Slots*
(a, g)	2
(b, f)	2
(c, e)	2
(d)	2

A total of *8 slots* is required here to complete the schedule.

Differing Power Level - Scheme 3

Here we consider the mapping of power levels to the node pairs shown in Figure A.5.

- *Interference*

  Based on the chosen power levels and the distances between the nodes in the topology, the following pairs of nodes would interfere if they had transmissions occurring at the same time.

  (a, b) (a, d) (b, d) (d, f) (d, g) (f, g)

- *Routing*

  Because of the nature of the topology, there are are no alternate routes for any source destination pairs.

  Direct Routes: 1-2, 1-3, 1-5, 4-5, 5-6

  Multihop Routes: 1-5-4, 1-5-6, 2-1-3, 2-1-5-4, 2-1-5, 2-1-5-6, 3-1-5-4, 3-1-5, 3-1-5-6, 4-5-6

- *Scheduling*

  Based on the chosen routing, the loads on each link of the network is as follows to completely satisfy the traffic matrix.

  a – 4
  b – 4
  d – 2
  f – 4
  g – 4

Based on the interference patterns, we can schedule the following links in parallel for the specified units of time to complete the load requirements:

Links	*Slots*
(a, g)	4
(b, f)	4
(d)	2

This results in a schedule of *10 slots*, which is still better than the common minimum scheme.

The examples with non-uniform traffic further show that a differentiated power scheme is required and that COMPOW does not perform optimally.

Appendix B

Specific Counter Examples for DirectTrans

Consider the network shown in Figure B and a traffic pattern as given in Figure B.2. Traffic on the links $2-3$, $3-4$, $6-7$, $7-8$, $8-9$ and $9-10$ use 1mW power, $4-9$ and $5-10$ use 10mW, $1-5$ uses 5mW. If we use direct transmissions, a schedule of length 7 slots is required to complete the 9 transmissions, as $2-3$ and $3-4$ can be scheduled in parallel with $6-7$ and $7-8$. If we instead route the traffic on $1-5$ onto the links $1-3$ and $3-5$ using 1mW, as shown in Figure B.3, the schedule length now reduces to 6 slots, as $1-3$ and $3-5$ can be scheduled in parallel with $8-9$ and $9-10$.

In examples shown earlier in Appendix A, having the nodes transmit data directly to the destination yielded the best throughput. This example is an instance where direct transmissions will give sub-optimal throughput.

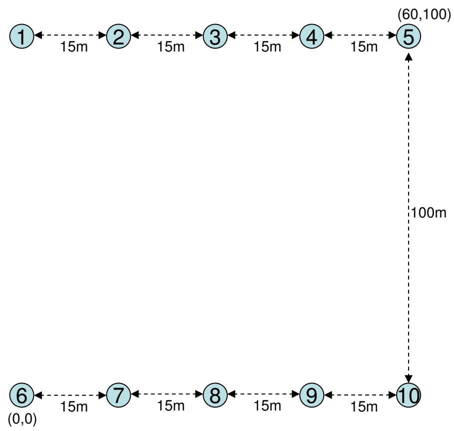


Figure B.1: Network topology

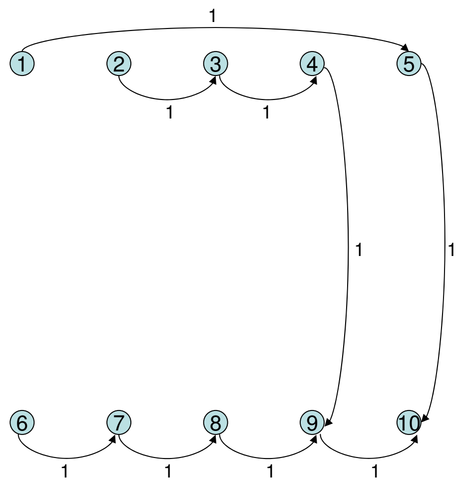


Figure B.2: Traffic pattern

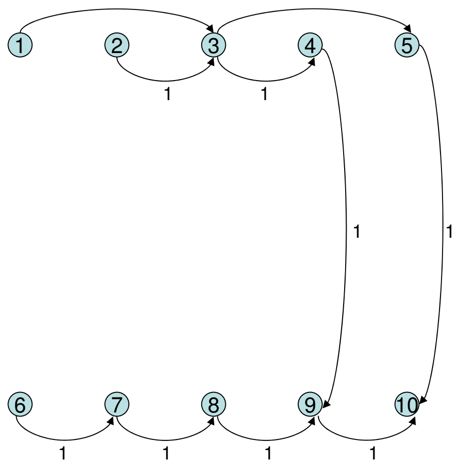


Figure B.3: Routing the traffic