Broadcasting and transmission coordination for ad hoc and sensor networks.

A Thesis

Submitted to the Faculty

of

Drexel University

by

Ananth V. Kini

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy in Computer Engineering

November 2008

Dedication

To my parents,
Geeta Kini and Venkatesh Kini,
and to my sister, Aparna.

Acknowledgements

First and foremost I thank my advisor Dr. Steven Weber. He took a huge leap of faith by accepting me as his first graduate student, for this, I have and will always be in his debt. Without his guidance, perseverance, faith and insight this dissertation would not have seen the light of day. I have learnt a lot in the time I have spent working with him. He has taught me the meaning of perseverance, that there is no substitute for hard work, that one needs to be constantly plugging away at things, and that failure is nothing but an opportunity to learn something new, and try something different.

I owe a great debt to my various committee members, each of which provided me with invaluable insight, motivation and inspiration. I would like to thank Dr. Harish Sethu for his guidance through my dissertation proposal and defense, he was extremely candid with his advice and his input has greatly improved the quality of this dissertation. I would also like to thank Dr. Nagarajan Kandasamy for always encouraging me and for finding the time to provide me with invaluable insight and feedback. From him I learnt that success has three key elements: hard work, persistence, and enforcing the old adage "all work and no play makes Jack a dull boy". I would also like to thank Dr.'s John Walsh and Athina Petropulu for their continuing feedback and encouragement which has greatly improved the quality of this dissertation. Finally I would like to thank Dr. Ali Shokoufandeh for his invaluable advice, support, and positive reinforcement every step of the way. His passion for research and teaching are inspirational and I am terribly proud on having been able to collaborate with him during my years at Drexel. It is impossible to quantify in words the deep sense of appreciation I have for being fortunate enough to have met him, and learnt all that I have from him.

In addition, I would like to extend a special thanks to Dr.'s Jaudelice C. de Oliveira and Nihat Bilgutay. Dr. de Oliveira was a supreme help when I started at Drexel, in fact she is the reason I met my advisor. Her guidance, feedback, and advice has helped me immensely. Dr. Bilgutay was the Department Head when I joined Drexel, he was extremely helpful in every which way, and put the students first, always. I am certain no one will disagree when I state that he will be terribly missed by everyone who was fortunate enough to have met him. My heart goes out to his family. I would also like to thank Dr.'s Som Tyagi and T.S. Venkataraman for their help during my first year at Drexel.

I have been fortunate to meet several people that have made my years at Drexel enjoyable and memorable. In order to ensure that I do not leave out anyone, I shall list them in chronological order. They are to name a few: Ganesh, Sukrit, Vilas, Anbu, Regina, Sparky, Abhishek, Bryan, Nick, Ryan, Danielle, Alfred, Zhen, John, Sagar, Nikhil, Khalid, Dave. These and many others didn't just make my years at Drexel bearable, but thoroughly enjoyable.

A special thanks to Sukrit and Vilas for the laughs that we shared when the going was (relatively) good, and the not so fun times when we would collectively vent our frustration. The venting was key in keeping all of us (relatively) sane, or at least no less sane than when we first started at Drexel.

I also must thank Stacey, Tanita and Delores. I have shared many a fun moment (well more fun for me than them anyway) with these awesome ladies, and have bothered them endlessly. Kudos to your patience. A special thanks to Chad as well, for all his help this past year.

Lastly, and most importantly, I would like to thank my parents and my sister. The credit for all my achievements goes solely to them.

Table of Contents

List of Tables

List of Figures

Abstract

Broadcasting and transmission coordination for ad hoc and sensor networks
Ananth V. Kini
Steven Weber, Ph.D.

This thesis studies the performance benefit of coordination in wireless sensor networks (WSNs) and ad hoc networks (AHNs). WSNs are often comprised of a large number of inexpensive nodes with short battery life and limited communication and processing capabilities. AHNs are wireless networks operating without the benefits of network infrastructure (basestations) or centralized control.

WSNs often require control messages be broadcast to the entire network. We study the performance of a class of randomized broadcast protocols that employ coordination to reduce the transmission of redundant messages and to reduce the occurrence of message collisions. Specifically, information coordination entails a potential transmitter employ local information to infer whether or not its potential receivers would be interested in its message, while communication coordination entails a potential transmitter employ local information to infer whether or not its transmission would interrupt other ongoing transmissions. The individual and joint benefits of these two forms of coordination are assessed through their impact on a variety of natural performance indicators.

AHNs working with limited spectrum perform best when simultaneous transmissions are coordinated to avoid collisions. Optimal transmission coordination is a combinatorial optimization problem that is, in general, intractable for large size networks, even with global information and central control. Constraints on simultaneous transmissions may arise from both transceiver limitations (e.g., half-duplex designs) and from requirements on the signal to interference ratio. We study the transmission coordination optimization problem under a variety of natural communication constraints. Our work identifies particular instances where the problem may be solved

by greedy algorithms, and studies the performance of several natural heuristic solutions.

1. Introduction

Wireless communications pose numerous challenges, some of which are very different from those encountered in the wireline context. One challenge is the proper choice of the medium access control (MAC) protocol. The MAC protocol for ad hoc networks, sensor networks, and in general any kind of wireless platform has a major effect on performance and QoS that can be achieved. Although research into MAC protocols for wireless networks dates back several decades (*e.g.,* Aloha [1]) there are still many important protocol design and performance questions to be answered. One of the reasons for this is that different applications and scenarios may require distinct solutions. A MAC protocol that might work for an ad hoc network may not work as well for a sensor network.

In essence, a MAC protocol is a distributed resource allocation algorithm seeking to coordinate transmission attempts so as to minimize the occurrence of message collisions. Depending upon the network scenario, achieving this distributed coordination may be easy or hard, and moreover, coordination may or may not yield a significant performance increase above an unsophisticated protocol (*e.g.,* Aloha). Assessing the performance benefit of transmission coordination is therefore an important question which we address in this thesis. In particular, this thesis studies the use of transmission coordination in *i*) broadcast communication protocols for wireless sensor networks (WSNs) and *ii*) point to point communication protocols for ad hoc networks (AHNs).

1.1 Broadcasting protocols for wireless sensor networks (WSNs)

Recent advancements in wireless communications technology, digital electronics, and miniaturization technology have resulted in the development of new wireless platforms and applications. WSNs are a prime example [2]. Sensor nodes are low cost, low power devices that are small in size and possess the functionality to operate in a variety of environmental conditions. Their multi-functional capabilities make them attractive for applications in military, medicine, environmental study, and security fields to name a few. In military operations it is quite normal to encounter situations where one might need to monitor an environment or location that is unsafe for personnel. Sensor networks provide the means to monitor such environments without putting personnel in harm's way. They can be used to monitor biological and/or chemical attacks, command and control, surveillance, communication equipment, targeting systems, and damage assessment to name a few possible military applications. Some environmental applications of sensor networks are monitoring migratory patterns of animals, agricultural monitoring, meteorological research, and pollution monitoring. In healthcare some uses of sensor networks include diagnostics, tracking/monitoring doctors and/or patients in a hospital. Although sensor networks are as yet not prevalent in home and commercial environments, the near future holds promise for applications in these as well.

A WSN consists of nodes, densely deployed in the area of interest. They can be deployed in large numbers due to their low per unit cost. Their low cost is in turn due to their limited power resources and low processing capabilities. Since these nodes are limited in power and computational ability, anything other than the simplest of communication protocols might be difficult to implement.

Control of sensor networks often requires multihop message dissemination to the entire network. For example, a surveillance WSN may require that, upon detecting a target, the detecting node broadcast a message to that effect to the network. Alternately, a control node may need to broadcast some instruction to the network,*e.g.,* a new sleep schedule, or perhaps a new instruction on how to sense or use power. An obvious distributed protocol for multihop message broadcast is flooding: each node, upon receiving a message for the first time, transmits that message. The problem with the flooding protocol is its inefficiency: if a node has k "neighbors" with which it can communicate, then under the flooding protocol each node will receive each message k times. Only the first reception is of use to the node, and the remaining $k-1$ message receptions are redundant, and therefore waste energy. This observation suggests the use of *randomized* broadcast, or gossip, protocols.

Under a gossip protocol, each node upon receiving the message, transmits the message with some probability p (known as the gossip probability). Under gossip each node operates independently, there is no coordination at all. The focus of our research on WSNs is to gauge the effect of coordination on the performance of randomized broadcast protocols. This coordination is achieved through the exchange of local state information. We propose an improvement of gossip through the use of information coordination (IC) and communication coordination (CC). IC is used to reduce the number of redundant transmissions, ensuring that only useful messages are transmitted, whereas CC is used to reduce the number of colliding transmissions. Our primary figure of merit to gauge the impact of IC and CC is the broadcast capacity. The *broadcast capacity* of a WSN is defined to be the maximum rate at which the network may generate messages intended for distribution to the entire network when subject to certain QoS conditions. We compare the achievable broadcast capacities for (i) gossip, (ii) gossip + IC, (iii) gossip + CC, (iv) gossip + IC + CC. In addition, we contrast the relative benefits and costs associated with these protocols under a realistic communication scenario for different topologies ((i) a uniformly distributed

WSN, and (*ii*) a clustered WSN) using various metrics that are pertinent to a WSN. We now give a brief summary of these metrics. First, the *coverage* is the average fraction of nodes in the network that receive a typical message. Second, the *efficiency* is the average ratio of the number of nodes receiving the message over the number of transmissions of that message. Third, the *delay* is the average time a message spends in the system. Fourth, the *collision quotient* is the average number of communication collisions per transmission attempt. Fifth, the *broadcast capacity* defined earlier. Sixth, the *energy consumption* is the energy consumed by the nodes in the network. Our study includes several real world phenomena that affect wireless communication: channel variations, interference, and random node failures due to battery depletion. The study of how these protocols perform under our various metrics is conducted via simulation. Our key findings are:

- The use of IC or CC will have little or no effect on the broadcast capacity when the network is either *i*) underloaded (as there is no need for coordination), or *ii*) overloaded (as no amount of coordination can help). The use of IC and/or CC can have a significant effect on the broadcast capacity in the intermediate loading regime.

- As network load increases, the use of CC has several effects: *i*) delay increases, *ii*) efficiency increases, *iii*) coverage is higher than under non-CC protocols. Delay increases because nodes must wait an increasingly long time for viable transmission opportunities, thereby increasing the fraction of messages that are dropped due to violation of their delay constraint. Efficiency (the number of successful receptions over the number of transmission attempts) increases since there are fewer transmission attempts. Coverage is higher than under non-CC protocols since without CC messages die out very early due to high collision rate.

- As network load increases, the use of IC has several effects: *i*) the collision rate increases (much more dramatically than with CC), *ii*) the efficiency decreases, *iii*) the delay decreases. Uncoordinated communication attempts are increasingly likely to collide as the network load increases, and this increased collision rate decreases the efficiency. As load increases there are more messages that die out prematurely, and this decreases the average time a message spends in the system, decreasing the delay metric. Finally, in a low-loaded network IC suffers a higher delay than non-IC protocols: transmitting only when a sufficiently high number of neighbors do not yet have the message results in messages propagating more slowly than when this requirement is absent.

- Increasing the degree of coordination required for transmission has both positive and negative effects. Increasing the IC requirement leads to increased efficiency, but if the requirement is too stringent then eventually too many nodes will elect not to transmit and the messages will die out early, thereby reducing coverage. Similarly, increasing the CC requirement leads to reduced collisions, but if the requirement is too stringent then eventually too many nodes wait too long for a suitable transmission time, thereby increasing queueing delay, leading to message time out, and again, to reduced coverage.

- As network load increases, the use of IC reduces the number of redundant transmissions, resulting in an improvement in performance over protocols that employ only CC. Beyond a certain network load however, IC alone is unable to sustain coverage due to the lack of any mechanism to reduce colliding transmissions. As as we move to the over-loaded regime, the use of CC provides an improvement in coverage.

- The fraction of energy spent in querying local state information is high in the underloaded regime where congestion is low and hence delays are short. In the overloaded regime, almost all energy is consumed in keeping the node on because of the long delays required due to high congestion and the relative cost of coordination is very low.

- Message congestion is greater in a clustered WSN as opposed to a uniformly distributed WSN. As a result, even a low degree of IC results in improved performance in comparison to a protocol that does not employ IC. This improvement is clearly evident as we move from the underloaded to the overloaded regime. This is quite different from the uniformly distributed WSN case, where CC alone is able to yield superior performance in the overloaded regime.

- When operating under limited energy constraints it is better to rely purely on CC to sustain coverage in the overloaded regime. The only way to guarantee better coverage when employing both **IC** and **CC** is to either limit the reliance on CC, or allocate a large energy budget.

1.2 Transmission coordination in ad hoc networks (AHNs)

An ad hoc network is a wireless network that operates without the benefit of network infrastructure or centralized control. AHNs working with limited spectrum perform best when simultaneous transmissions are coordinated to avoid collisions. We focus on the problem of transmission coordination under a set of communication constraints. The constraints are a set of rules that define permissible concurrent transmissions, imposed as a result of hardware limitations of the transceiver units or the operational mode of the network. Transmission coordination is an important part of scheduling which involves arranging simultaneous transmissions in time or frequency. This constrained coordination problem has natural parallels in graph theory, such as the graph coloring problem, and the maximum weighted independent set problem. We focus on the presence or absence of the following communication constraints and their combinations:

1. Half-duplex: A node can not concurrently transmit and receive.

2. Single reception: A node can not concurrently receive from multiple transmitters.

3. Unicast: A node can not concurrently transmit distinct information to multiple receivers.

4. Protocol interference: A node can not receive if there is any interfering transmission in its vicinity.

Each transmission is represented as an edge directed from transmitter to receiver. The constrained coordination problem is an optimization problem which requires an integer linear program (ILP) formulation. The objective of the optimization problem is to maximize the weighted sum of the selected edges. We show that some of the communication constraint problems are totally unimodular (TUM). TUM allows them to be solved via linear program (LP) relaxation. A subset of these TUM problems are also matroids. This means that their solution can be obtained via use of a greedy algorithm. Some of the problems that aren't matroids have optimal solutions that may be well approximated by natural heuristics. The heuristics we consider select edges on the basis of their lengths and/or the maximum number of edges that can be activated for a transmitter (receiver). Finally some of the problems have solutions that are *not* well approximated by these heuristics. Our key contributions are:

- We characterize the various problems by way of their Primal-Dual LP formulations.

- We prove that only the single receiver and unicast problems individually yield matroids.

- We prove that neither half-duplex nor protocol interference problems yield matroids, nor are they TUM.
- Simulations suggest that the combined half-duplex, single receiver and unicast problem (matching problem) can be well approximated by our heuristics. This holds even with the addition of the protocol interference constraint.
- Simulations suggest that our heuristics are able to yield good approximations of the optimal solution for the problems that do not include the interference constraint.
- Simulations suggest that it is more difficult to obtain a good approximation when the edge weights are unity (meaning the objective is to maximize the number of activated edges) than the problem where the objective is to maximize the weighted sum of active edges.
- When operating under only the interference constraint, our heuristics fail to yield good approximations of the optimal.
- Depending on the problem at hand, either both heuristics work very well, or one outperforms the other, or neither works well.

1.3 Thesis outline

The remainder of this thesis has been organized into two chapters.

- In Chapter 2 we propose a suite of randomized broadcast protocols that utilize IC and/or CC, and evaluate their performance when subject to various phenomena that affect wireless communication. The work in this chapter has appeared in [3], [4], [5], and is under review for Elsevier Ad Hoc Networks [6].

- In Chapter 3 we study centralized transmission coordination for ad hoc networks when subject to certain communication constraint sets. These constraints determine compatible simultaneous transmissions. The objective is to find the maximum weighted edge set under the given constraints. We focus on characterizing the structure of the various problems and propose several heuristics that yield good approximations to the optimal solution.

2. Broadcasting protocols for wireless sensor networks (WSNs)

The broadcast capacity of a wireless sensor network (WSN) is defined as the maximum rate at which the network may generate messages intended for distribution to the entire network when subject to certain requirements on coverage and delay. Broadcast capacity is limited by factors such as communication collisions and congestion. Collisions may be reduced through the use of communication coordination (CC), and congestion may be reduced through information coordination (IC), ensuring that only useful messages are transmitted and stored. We study the broadcast capacity of a WSN when subject to various real world phenomena that affect wireless communication, namely channel variations, interference and random node failures. We study the benefits and costs associated with using the IC and CC mechanisms on different topologies through the use of various metrics.

2.1 Introduction

Most of the existing proposed broadcast algorithms for wireless sensor networks are variants of the Gossip protocol[1]. Such protocols are appealing in that they offer a simple and distributed method to disseminate information to most of the nodes in a network using significantly fewer redundant transmissions than flooding. A transmission is redundant if each of the receivers of the transmission has already received the message at an earlier time. Using randomized broadcast to reduce redundant transmissions at the expense of some nodes not receiving a message is an acceptable tradeoff in sensor networks, where energy constraints often trump the desire for message delivery to all nodes. Simple flooding is inefficient in terms of redundant

[1] Gossip protocols, in current parlance, denote randomized broadcast and routing protocols; our use of the term will be restricted to randomized broadcast.

transmissions, and hence, wastes energy.

Gossip and its variants. An often cited gossip protocol is the `GOSSIP1`(p, k) protocol from [7] where *each node, upon first receiving the message, transmits the message in the following time slot with probability* p*, unless the message was received in time slot* $i \leq k$ *in which case the message is transmitted with probability* 1. Transmitting with probability 1 for the first k time slots helps ensure that the message does not die out prematurely. The `GOSSIP1` protocol with $k \to \infty$ or $p = 1$ reduces to flooding. As shown in [7], this protocol offers significant reduction in redundant transmissions relative to flooding by incurring the cost that not all nodes will necessarily receive the message. There is an obvious tradeoff between reducing redundancies and achieving a near-complete message distribution: increasing p will increase the number of nodes that receive the message but also will increase the number of redundant transmissions. **Fig. 2.1** highlights this tradeoff. When p is low efficiency is high, but coverage is low. Increasing p increases coverage but lowers the efficiency. Choosing $p = 0.7$ results in a coverage of nearly 90% while maintaining an efficiency of 1.4. **Fig. 2.2** illustrates a sample realization of the dynamics on a 100×100 lattice network, where upon receiving a message for the first time, nodes transmit to their four cardinal neighbors with probability $p = 0.65$. The four shades illustrate the number of nodes that have received the message at four snapshots in time.

Under gossip each node operates independently, there is no coordination at all. Many researchers have proposed protocols that improve on gossip by making use of various types of local information. Instead of simply gossiping with some gossip probability p, the transmission decision can be made based on local information gathered either passively (through listening) or actively (through issuing query messages to neighbors). In [3] we studied these gossip variants via simulations. We then proceeded to combine and optimize them into a superior protocol, which we called

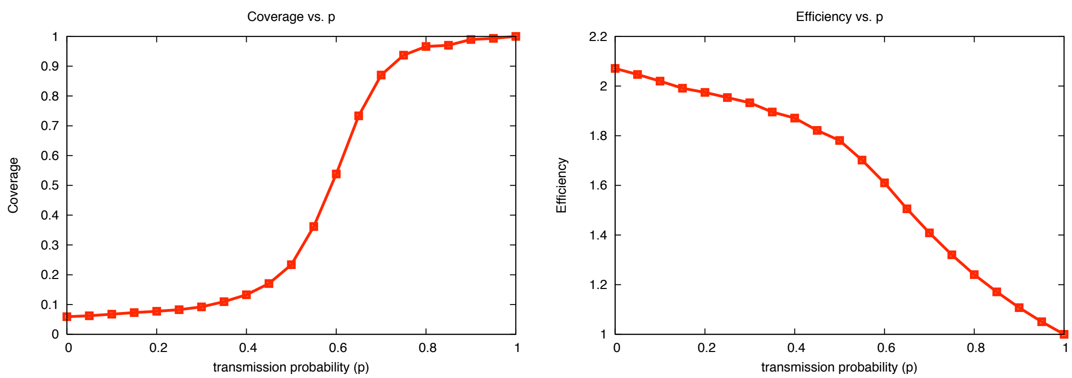


Figure 2.1: Lower p improves efficiency, but penalty is lower coverage

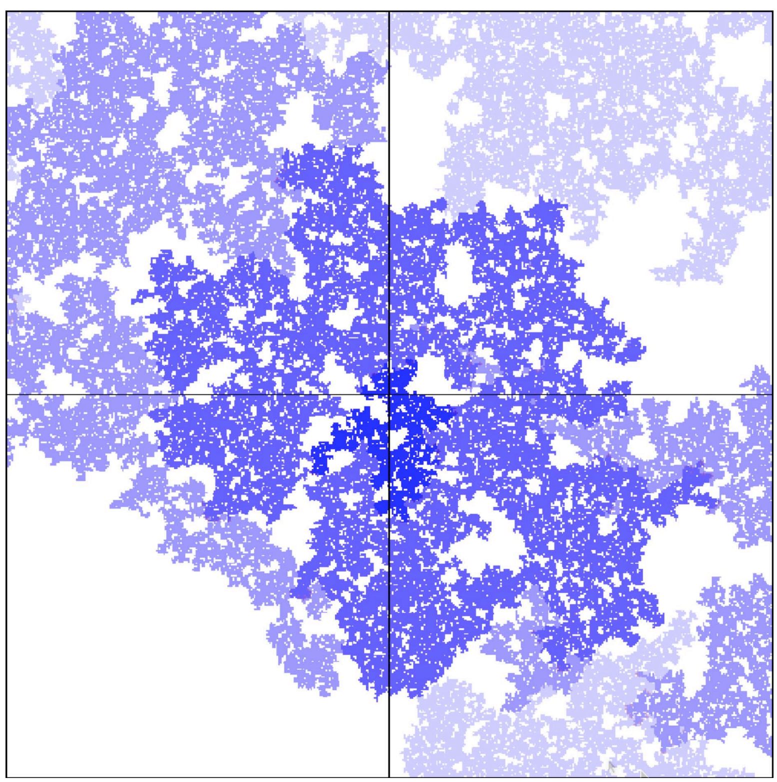


Figure 2.2: Illustration of the gossip protocol with $p = 0.65$ for a 100×100 node network.

`SmartGossip`. This protocol makes use of several improvements: 1) a state vector for recording all available information, 2) "directed" transmissions to reduce latency, 3) a sigmoid based transmission probability function that parametrizes the impact of randomization, 4) confirmation messages which facilitate listening by a node's neighbors, and 5) query-request messages which permit nodes to pull messages from neighbors.

WSN performance metrics. There are several performance metrics relevant to evaluating the performance of any proposed broadcast protocol. We shall formally define these metrics in Section 2.3.2. First, the *coverage* is the average fraction of nodes in the network that receive a typical message. Second, the *efficiency* is the average ratio of the number of nodes receiving the message over the number of transmissions of that message. Third, the *delay* is the average time a message spends in the system. Fourth, the *collision quotient* is the average number of communication collisions per transmission attempt. Fifth, the *broadcast capacity* is the maximum rate at which the network may generate new broadcast messages subject to a constraint on the minimum coverage and maximum delay. Sixth, the *energy consumption* is the energy consumed by the nodes in the network.

Topology of network. A uniform distribution is often used to model the spatial locations of nodes in large size WSNs. This arrangement, although simple and analytically tractable, discounts the fact that the node distribution is not likely to be completely spatially random, *i.e.,* the nodes are generally going to exhibit some degree of clustering. As an example, consider the case where nodes are dropped in a terrain that has small hills and valleys: it is quite possible that sensors will fall more densely in the valleys than on the hills. To characterize the impact of clustering on performance, we will look at two different topologies: a uniform distribution and a cluster process.

Channel model. We use a standard channel model for communication where a message is received provided its instantaneous signal to interference and noise ratio (SINR) exceeds a particular threshold over the duration of the transmission. Transmissions are subject to distance dependent path loss attenuation and distance independent channel variations (in our case, Rayleigh fading).

Coordinations. We focus on two coordination mechanisms: information coordination (IC) and communication coordination (CC). When using IC, whether or not a node transmits depends on the fraction of neighboring nodes not having the message. By deciding to transmit a message only when some minimum number of the node's neighbors do not already possess the message, the number of redundant transmissions can be reduced. When employing CC, nodes only transmit when a certain number of its neighbors are not already receiving from another transmitter. This reduces the number of collisions that may occur. IC improves efficiency, whereas CC reduces collisions. We study the performance of a WSN using IC and/or CC under variations in topology and when nodes die due to energy constraints (battery depletion). We study four broadcast protocols to gauge the individual and joint benefits of IC and CC. These protocols are the combinations of either using or not using IC and CC.

2.2 Related work

The current use of the term gossip is to denote randomized broadcast protocols, although the research on this topic far predates the application of the moniker "gossip". A fundamental insight, apparently first made in 1953 by Landau and Rapoport [8], is that the dynamics of information propagation under randomized broadcast are equivalent to the dynamics of disease spreading among a population (epidemiology), where the interest in both cases is to know who has the message (disease).

Information and communication coordination. As mentioned earlier, the networking community has proposed quite a few gossip variants. Representative work includes SPIN for energy constrained networks in which nodes employ negotiation to reduce redundant transmissions [9]. Directional gossip [10], rumor routing [11], and parametric probabilistic sensor network routing [12] all focus on using gossip as a routing or resource discovery mechanism, *i.e.,* randomized but directed dissemination towards a target node. Two-phase push-pull gossip [13] introduced the idea of a two phase protocol where in the first phase nodes with the message *push* the message, and in the second phase nodes without the message *pull* the message from those nodes with the message. Also [7] looks at gossip with flooding and two-threshold gossip.

The firecracker protocol [14] combines routing with a broadcast based protocol such as [15] to ensure reliable delivery of data to all the nodes in the network. In [16] the authors look at the Deluge protocol. Deluge's density-aware, epidemic properties help achieve reliability in unpredictable wireless environments and robustness when node densities can vary by factors of thousands or more. Although we recognize the importance of these investigations, they are somewhat orthogonal to our focus here, *i.e.,* evaluating the performance improvement for randomized broadcast protocols obtainable through the use of local information and coordination.

Capacity of a multi-hop WSN and wireless multicast. There has been extensive work on characterizing the capacity of a multi-hop wireless ad hoc or sensor network. Representative works include [17], [18], [19], [20]. We emphasize that *the majority of these papers deal with supporting multiple multi-hop point to point connections, not with supporting broadcast communication.* The notion of capacity is very different under the two scenarios: multi-hop connections require $O(N)$ transmissions in a $N \times N$ network, whereas broadcasting requires $O(N^2)$ transmissions. One of the contributions of our work is to take a first step towards a meaningful definition of and investigation into the *broadcast capacity* of a WSN.

Chaporkar *et al.* [21] investigate the problem of finding an optimal transmission policy from the class of stable policies so as to maximize throughput for wireless multicast. In [22] the authors use a stochastic framework to explore the tradeoffs between the QoS parameters such as throughput, stability and loss for multicast transmissions. In particular the paper focuses on optimal transmission strategies to maximize system throughput when subject to stability and loss constraints. Both these papers focus on exploiting the inherently broadcast nature of wireless communication to design an optimal MAC strategy for wireless multicast, and exploring the interplay between the various QoS parameters. We, on the other hand, focus on exploring (via simulation) the effect of communication and/or information coordination on the broadcast capacity. Moreover, we investigate the costs associated with these mechanisms in terms of various performance metrics that are pivotal to WSNs.

2.3 System model

2.3.1 Network and communication model

We consider a WSN whose topology is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N$ nodes and $|\mathcal{E}| = E$ edges. This graph determines the nodes that each node is to coordinate with (but does not dictate which nodes may communicate). The graph is formed by a boolean disk model: given a point process $\{X_i\}$, form the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $(i, j) \in \mathcal{E}$ if $d(X_i, X_j) \leq r$, where $d(X_i, X_j)$ is the distance between X_i, X_j. The parameter r represents the maximum distance between nodes such that two nodes may still successfully communicate in the absence of interference. We assume protocol control packet transmissions are immune from interference while data packet transmissions are subject to interference. That is, protocol control packet success requires a sufficiently high signal to noise ratio (SNR), while data packet transmission success requires a sufficiently high signal to interference and noise ratio

(SINR). Protocol control packet exchanges allow nodes to share their communication status and to specify which packets have already been received at each node. This assumption may be justified by assuming protocol packet transmissions are done out of band, and are sufficiently short to minimize the presence of interference.

Nodes in our protocols communicate asynchronously; hence time is continuous (not slotted). The messages generated by each node for broadcast form independent marked Poisson processes of rate λ/N, denoted by $\Omega^i = \{(T_m^i, V_m^i), m \in \mathbb{N}\}$. Here, $\{T_m^i\}$ are the new message generation times for node i (generated at rate $\frac{\lambda}{N}$), and the independent identically distributed (i.i.d.) marks $\{V_m^i\}$ indicate the transmission duration for each message. Varying V_m^i allows us to capture varying message lengths. Note, the overall message generation rate is $N \cdot \frac{\lambda}{N} = \lambda$. For simplicity, we set the transmission durations, $\{V_m^i\}$, to be independent and exponentially distributed with parameter τ; thus $\mathbb{E}[V] = \tau^{-1}$. Each transmission is preceded by a random pause time. This has the effect of offsetting transmissions. Consider two adjacent transmitters wishing to share a message with a receiver. In the asynchronous setting, the second transmitter would hear the first transmitter and therefore has received information. In particular, the second node knows that *i*) the transmitting node has received the message, and *ii*) all of their mutual neighbors have now received the message (assuming that the received SINR at these nodes is above the threshold). This information may lead the second node to decide not to transmit at all, thereby improving efficiency and/or coverage. The pause times are i.i.d. and exponentially distributed with parameter μ, and hence have a mean of μ^{-1}. Note that each message transmission requires a pause time of mean μ^{-1} followed by a transmission time of mean τ^{-1}; as a result the average service time for a message is $\mu^{-1} + \tau^{-1}$. There is a natural tradeoff for μ. A large μ results in short pause times, thereby resulting in lower delays. A large μ, however, means the system state between pause times is highly correlated, and the overhead required to query the system state is large and has limited benefit. A small μ (longer pause times), on the other hand, results in a low correlation between system state information between pause times, making the information gathered more useful. Each node i maintains an ordered transmission message queue. In addition, each node i also maintains a list of previously transmitted messages. Each message is added to each node's queue at most once, and subsequent receptions of the same message are ignored.

A message transmission by node i is potentially received by any node $j \in \mathcal{V}$. Our communication model allows for channel variations and interference between transmissions. Each link is subject to channel variations implemented as Rayleigh fading: $\psi_{i,j} \sim \text{Exp}(1)$, where $\psi_{i,j}$ denotes the fading experienced by link (i,j). Fading is independent in space (across nodes) and time (across transmissions). We assume the channel coherence time to be on the order of the packet transmission time, thereby implying that a packet sees a constant fading coefficient for the duration of its transmission, and subsequent transmissions by a node see independent fading coefficients. Assuming unit transmission powers, a transmission by node i will result in a potential receiver j seeing a received signal with power of $P^r_{i,j}(t) = \psi_{i,j}(t) d(X_i, X_j)^{-\alpha}$, where $d(X_i, X_j)$ denotes the distance between nodes i and j and α is the pathloss exponent. In general, the instantaneous SINR at node j due to a transmission from i at time t is given by:

$$\text{SINR}_{i,j}(t) = \frac{P^r_{i,j}(t)}{\sum_{k \in \mathcal{T}(t) \setminus i} P^r_{k,j}(t) + \eta}, \tag{2.1}$$

where η is the common noise factor and $\mathcal{T}(t)$ is the set of transmitters at time t. The transmission from i to j is successful provided $\text{SINR}_{i,j}(t) \geq \beta$ over the time during which i is transmitting.

Fig. 2.3 shows the node state model. As can be seen, nodes can only be in one of three states at any given time: *idle, transmit* or *receive* (half-duplex communication).

A node is initially in the *idle* state. When a node (say i) begins to transmit the message at the front of its queue, all nodes in the *idle* state (say j) compute the instantaneous SINR for that transmission. If $\text{SINR}_{i,j}(t) \geq \beta$ then node j moves from the *idle* to the *receiving* state and is locked on to node i's transmission. Every time any other node in the network (say k) begins transmitting the instantaneous SINR is recomputed using Eqn. (2.1). As long as $\text{SINR}_{i,j}(t) \geq \beta$, node j remains in the receiving state (locked on to node i). If, however, this SINR falls below β due to node k's new transmission, the SINR at j from k is computed. If $\text{SINR}_{i,j}(t) < \beta$ and $\text{SINR}_{k,j}(t) \geq \beta$, node j switches allegiance to node k and locks on to its transmission (remaining in the receiving state). Node i's transmission attempt is deemed to have been unsuccessful at j and this is counted as a collided transmission. Alternatively, if $\text{SINR}_{i,j}(t) < \beta$ and $\text{SINR}_{k,j}(t) < \beta$ then node j moves to the *idle* state. Once a node breaks off its allegiance to a transmitting node it can never lock back on to the same transmission.

Fig. 2.4 depicts a sample protocol operation scenario with respect to the state diagram in **Fig. 2.3**. Consider this sample sequence of transmission and receive events. A node i starts transmitting at time t, the resulting instantaneous SINR observed at node j (which is in the *idle* state) is $\text{SINR}_{i,j}(\text{t})$ (calculated using Eqn. (2.1)). Since this SINR exceeds the threshold β, node j locks on to i's transmissions and moves to the *receive* state. At some time t_1 before i's transmission completes, another node k begins transmitting. The new instantaneous $\text{SINR}_{i,j}(t_1)$ is then computed. Since this SINR is below β, i's transmission is lost at j and this is termed a collision. The node j however does not immediately move to the *idle* state. At the same time that (i, j)'s instantaneous SINR is recomputed at t_1, $\text{SINR}_{k,j}(t_1)$ is computed. Since this exceeds β, node j locks on to k's transmission (remaining in the receiving state). If $\text{SINR}_{k,j}(t_1)$ was less than β, then j would move to the idle state. At time t_2, node

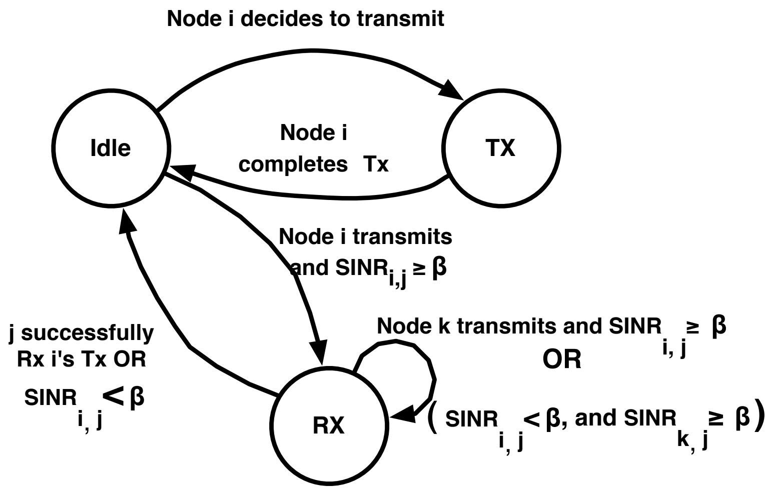


Figure 2.3: Node state transition diagram.

k's transmission completes, and is successfully received by j.

2.3.2 Performance metrics

In the following definitions the term message average denotes an average over those messages that are no longer in the system. Messages can exit the system in one of two ways: either there are no pending transmissions of the message (the message is not in any node's queue), or the message was removed from the system because the message timed out, discussed below. The superscript π above each metric denotes its evaluation for a particular protocol.

1. Coverage ($C^\pi(\lambda)$): The *coverage* of protocol π under message generation rate λ is the average fraction of nodes in the network that receive a message, averaged over all messages. Thus,

$$C^\pi(\lambda) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} C_m^\pi, \tag{2.2}$$

where C_m^π is the coverage for message m, *i.e.,* the fraction of nodes that received message m.

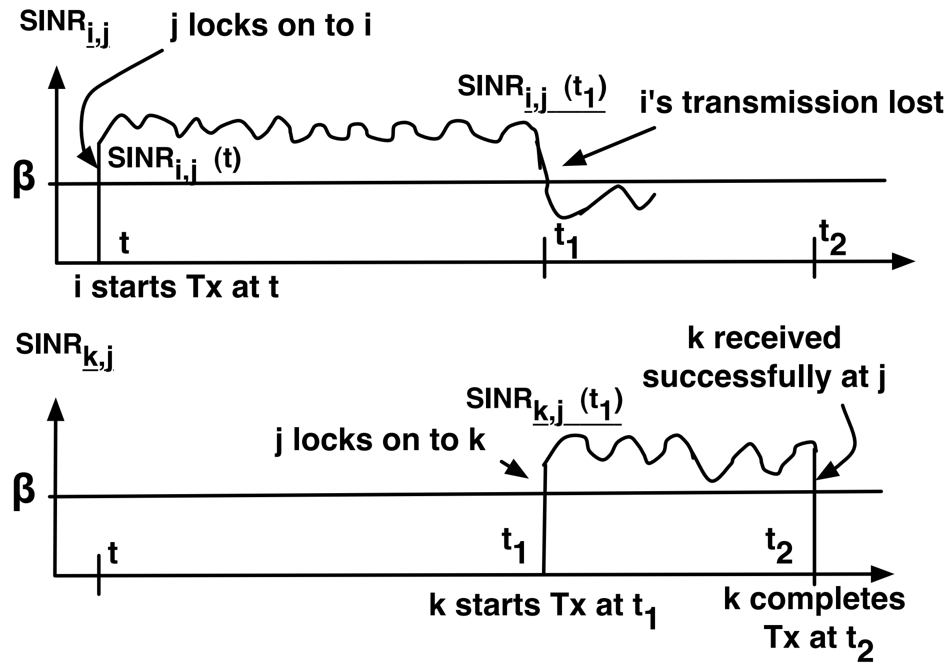


Figure 2.4: Example of an event sequence for a protocol operation scenario.

2. Delay ($D^{\pi}(\lambda)$): The delay for a message m, denoted D^{π}_{m}, is defined as the time from when the message was first introduced into the network (when it was first generated) to the time when that message is no longer present in the queue of any node in the network. Each message is dropped from the system (all the queues containing the message) if more than $D_{\max}$ seconds have elapsed since its initial generation time, T_m. This is included to capture the fact that many broadcast messages are only useful when delivered in a timely fashion. The *delay* of protocol π under message generation rate λ is given by:

$$D^{\pi}(\lambda) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} D^{\pi}_{m}. \tag{2.3}$$

3. Capacity ($\lambda^{\pi}_{\max}(C_{\min})$): The primary goal of a broadcast protocol is to disseminate messages to a large fraction of the nodes in the network. We define the *broadcast capacity* as the maximum new message generation rate, λ, such that the coverage does not fall below a specified minimum, $C_{\min}$. In particular,

$$\lambda^{\pi}_{\max}(C_{\min}) = \sup\{\lambda : C^{\pi}(\lambda) \geq C_{\min}\}. \tag{2.4}$$

As the message generation rate λ increases, one of two things will happen. If the protocol employs communication coordination (defined later), then increasing λ results in increased queueing delays as nodes wait longer for transmission opportunities. Eventually these queueing delays cause the messages to time out, thereby limiting their coverage. If the protocol does not employ CC, then increasing λ results in increased message collisions, causing more messages to die out prematurely, thereby limiting their coverage.

4. Efficiency ($\eta^{\pi}(\lambda)$): Another goal of a broadcast protocol is to minimize the number of redundant transmissions. The *efficiency* of a protocol π under a message generation rate λ is the message average ratio of the number of unique receptions of the message over the number of transmissions for that message:

$$\eta^{\pi}(\lambda) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} \frac{N^{\pi}_{r,m}}{N^{\pi}_{t,m}}, \tag{2.5}$$

where $N^{\pi}_{r,m}$ is the total number of nodes receiving message m and $N^{\pi}_{t,m}$ is the total number of transmissions for that message. The efficiency also captures the average number of nodes receiving the message for the first time for each transmission.

5. Collision quotient ($\nu^{\pi}(\lambda)$): Consider a node j that is locked on to transmitter i (*i.e.*, $\text{SINR}_{i,j}(t) \geq \beta$ at the instant i started transmitting). If another node k commences transmission and the new instantaneous $\text{SINR}_{i,j}(t) < \beta$, then node j will lose i's transmission. In this event a *collision* is said to have occurred (see **Fig. 2.4**). The *collision quotient* of protocol π under new message generation rate λ is defined as the message average ratio of the number of colliding receptions of each message over the number of transmissions of each message:

$$\nu^{\pi}(\lambda) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} \frac{N^{\pi}_{c,m}}{N^{\pi}_{t,m}}, \tag{2.6}$$

where $N^{\pi}_{c,m}$, is the number of colliding receptions for message m.

6. Energy ($\zeta^{\pi}(\lambda)$): WSN nodes generally have limited energy resources, hence it is important to minimize this expenditure. The energy consumed is defined as the average energy consumed over all nodes when operating under a particular protocol. The energy consumed by a node i is a sum of the energy consumption in the *transmitting*, *receiving*, and *idle* states. For the protocols that employ IC and/or CC there is an additional term to account for the energy consumed during the exchange of local information.

$$\zeta^{\pi}(\lambda) = \frac{1}{N} \sum_{i=1}^{N} E^{\pi,i}_{\text{Total}} = \frac{1}{N} \sum_{i=1}^{N} \left(E^{\pi,i}_{\text{T}} + E^{\pi,i}_{\text{R}} + E^{\pi,i}_{\text{I}} + E^{\pi,i}_{\text{Control}} \right), \tag{2.7}$$

where $E^{\pi,i}_{\text{Total}}$ is the energy consumption for node i. The energy consumed by a node in each state is calculated by multiplying the amount of time a node is in each of the states by the power expended while in that particular state. We used data from mote specification sheets [23]. The ratio of transmit power to receive power is 4. We assume that nodes remain in the idle state when not transmitting or receiving, and, based on [23], set the ratio of receive to idle power at 4.5. The energy consumed for control overhead is adjudicated as a one time additional term per query and not as a power consumed over time. This is because the information relayed during a query comprises of the message ID's, which would be only a few bits, and hence would be relayed almost instantaneously. Table 2.1 lists the power consumption under various modes of operation.

Table 2.1: Power consumption under various modes.

P_T	P_R	P_I	E_{Control}
36 (mJ/s)	9 (mJ/s)	2 (mJ/s)	0.001 (mJ)

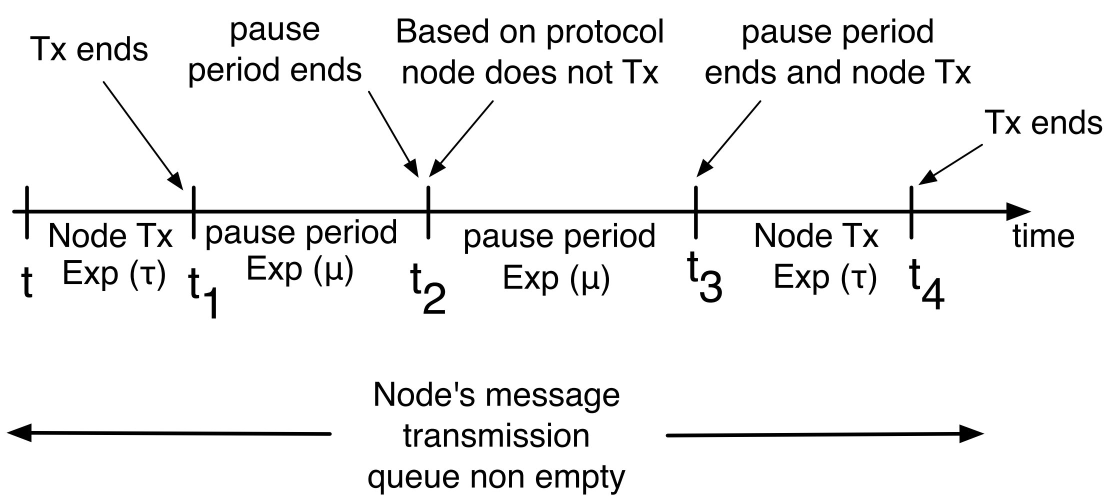


Figure 2.5: The general operation of a protocol for a node.

2.4 Protocol summary

We employ four protocols to study the benefit of using IC and CC. **Fig. 2.5** depicts the general operation of a protocol for a node. The node's message transmissions queue is non-empty and is in the middle of a transmission at time t. A node's transmission lasts for τ^{-1} seconds on average. Upon completion of the transmission (at time t_1), since the queue is non-empty the node will enter a random pause period of average duration μ^{-1}. Upon expiration of the pause period (time t_2) the node decides not to transmit (based on the protocol) and discards the packet. The node will then enter another pause period, at the expiration of which (at t_3) the node decides to transmit the next packet. This transmission then ends at t_4. The process continues as long as the message transmission queue is non-empty.

We consider four protocols for the four possibilities of using or not using IC and CC.

(1) **No IC and No CC.** This is the asynchronous gossip protocol [7] with transmission probability p. Every time (t) the node enters the idle state with a non-empty queue it sets a `transmitTimer` to go off at time $t + \text{Exp}(\mu)$. Upon the expiration of this timer, the node transmits the message at the front of its queue, with probability p, after which the message is moved out of the queue. With probability $1 - p$, the message is removed from the queue without transmission. Subsequent receptions of the same message are ignored. This last statement holds for all the protocols.

(2) **Only IC.** This protocol differs from the **No IC and No CC** protocol in that it uses IC in making transmission decisions. Upon the expiration of node j's `transmitTimer`, it sends a protocol control packet to see what fraction of its neighbors (say δ_j^m) have *not received* that particular message. It transmits the message if this fraction is greater than or equal to the threshold value (δ_I), *i.e.*, if $\delta_j^m \geq \delta_I$:

$$\frac{1}{|\Gamma(j)|} \sum_{i \in \Gamma(j)} \mathbf{1}_{i \text{ has not Rx message m}} \equiv \delta_j^m \begin{cases} \geq \delta_I \rightarrow j \text{ Tx m} \\ < \delta_I \rightarrow j \text{ does not Tx m} \end{cases} . \tag{2.8}$$

$\Gamma(j)$ is defined by $\mathcal{G}$ in Section 2.3.1 (p. 16). Regardless of whether node j transmits the message or not, the message is moved out of the node's queue.

(3) **Only CC.** This protocol works as follows. A node wishing to transmit the message broadcasts a request to send (`RTS`) protocol control packet to its neighbors. Upon hearing this broadcast the nodes in the idle state in the neighborhood of the transmitter respond with a clear to send (`CTS`) protocol control packet. If the node that broadcasted the `RTS` decides to transmit, the idle nodes will move to the *receiving* state (provided of course the received SINR $\geq \beta$). If the fraction of neighbors that

respond with a `CTS` (say υ_j^m) is greater than or equal to some threshold value (υ_C), then the node will transmit. The parameter υ_C specifies the degree of CC:

$$\frac{1}{|\Gamma(j)|}\sum_{i\in\Gamma(j)} \mathbf{1}_{i \text{ is idle}} \equiv \upsilon_j^m \begin{cases} \geq \upsilon_C \rightarrow j \text{ Tx m} \\ < \upsilon_C \rightarrow j \text{ does not Tx m} \end{cases} . \tag{2.9}$$

Messages remain in the node's queues until they are transmitted which can result in increasing message delays. The crucial CC component stems from the *suppression* mechanism. If any other node that received the `CTS` broadcast by the *idle* nodes were to begin a transmission before the issuer of the `RTS` completed its transmission, the `RTS` issuer's transmission could be lost due to collision. In order to prevent this from happening those other nodes that receive the `CTS` will hold off their transmissions for the transmission duration of the original `RTS` issuer.

Fig. 2.6 depicts operation of this protocol. Node j wishes to transmit a message. At the end of the pause duration the node broadcasts a `RTS` (denoted by event 1). Those nodes that are in the idle state (denoted in this case by the nodes i, k) will respond with a `CTS` (denoted by event 2). Node j can then transmit the message assuming $\upsilon_C \leq 2/3$. Nodes i and k then move to the received state. The `CTS` sent by nodes i and k will serve as notice to their neighbors to not transmit until i and k send an `ACK` to j, upon j's completion.

(4) **IC and CC.** This protocol incorporates both CC and IC. The protocol is a simple modification of the **Only CC** protocol described above. The change is as follows: a node upon receiving a `RTS` will respond with a `CTS` if *i*) it is *idle*, and *ii*) it has not already received the message. Thus the CC component comes from the `RTS/CTS` protocol, while the IC component comes from the fact that nodes only send a `CTS` if they want the message. The node issuing the `RTS` will transmit only if the fraction of its neighbors responding with `CTS`'s (γ_j^m) is greater than or equal to some threshold

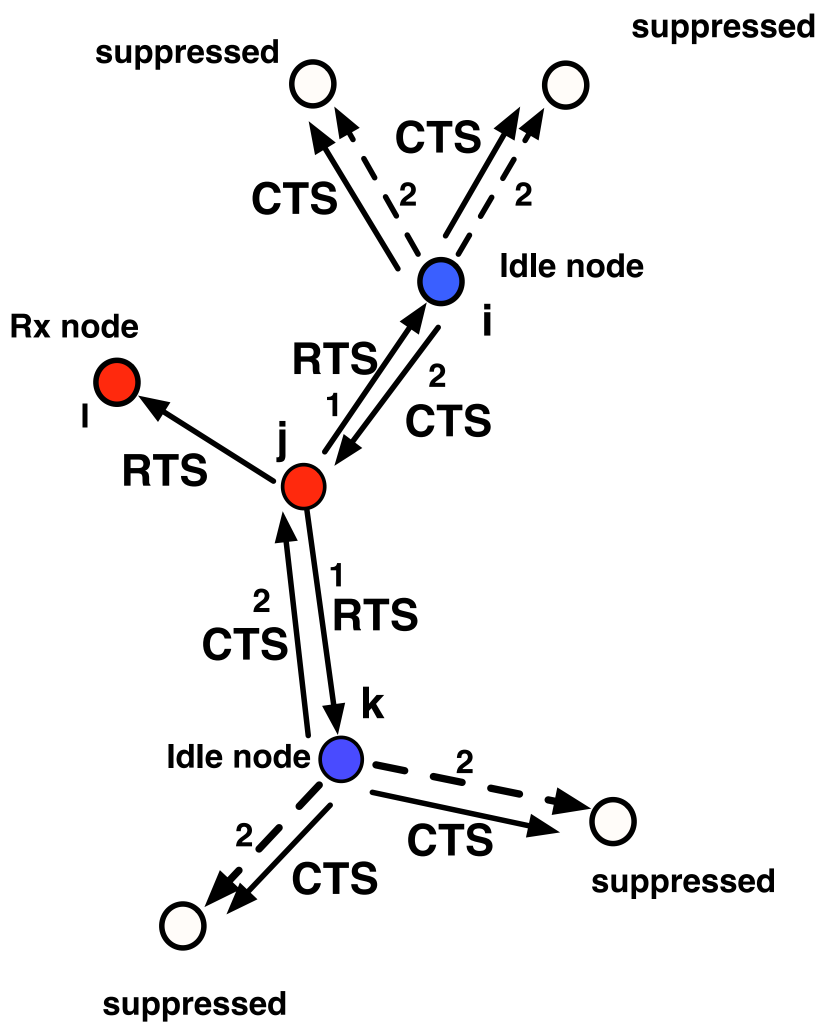


Figure 2.6: General operation of the **Only CC** protocol.

Table 2.2: Categorization of the protocols.

Protocol	Parameter	Transmission criterion
No IC and No CC	$p \in [0,1]$	$u \geq p, u \sim \text{rand}(0,1)$
Only IC	$\delta_I \in [0,1]$	$\delta_j^m \geq \delta_I$
Only CC	$\upsilon_C \in [0,1]$	$\upsilon_j^m \geq \upsilon_C$
IC and CC	$\gamma_{I,C} \in [0,1]$	$\gamma_j^m \geq \gamma_{I,C}$

value, $\gamma_{I,C}$:

$$\frac{1}{|\Gamma(j)|} \sum_{i \in \Gamma(j)} \mathbf{1}_{i \text{ has not Rx message m and is idle}} \equiv \gamma_j^m \begin{cases} \geq \gamma_{I,C} \rightarrow j \text{ Tx m} \\ < \gamma_{I,C} \rightarrow j \text{ does not Tx m} \end{cases} . \quad (2.10)$$

Fig. 2.7 illustrates the **Only IC** and **IC and CC** protocol mechanisms. Table 2.2 summarizes the protocols. For the **Only IC** , **Only CC** , and **IC and CC** protocols, the parameters δ_I, υ_C, and $\gamma_{I,C}$ determine the degree of IC, CC, and IC + CC respectively. For example, as we lower δ_I, the degree of information coordination decreases, thereby redundant transmissions increase, lowering efficiency. Increasing δ_I, on the other hand, reduces the number of redundant transmissions, improving efficiency. Setting $\delta_I = 0$ implies no information coordination, whereas $\delta_I = 1$ implies complete information coordination. Similarly, υ_C determines the degree of communication coordination. Lowering υ_C decreases the degree of coordination, resulting in a possible increase in collisions, whereas increasing υ_C facilitates a higher degree of communication coordination, thereby reducing collisions. Setting $\upsilon_C = 0$ implies no coordination, whereas $\upsilon_C = 1$ implies complete coordination and hence zero collisions. Similarly, $\gamma_{I,C} = 0$ implies neither information nor communication coordination, resulting in poor efficiency and high collisions, whereas $\gamma_{I,C} = 1$ implies maximum efficiency and zero collisions.

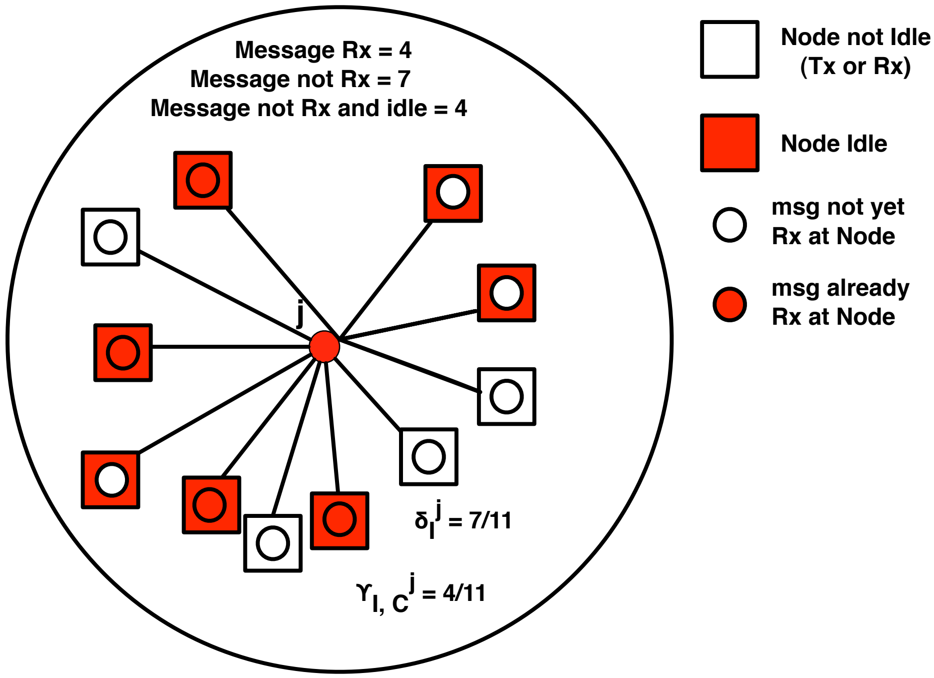


Figure 2.7: **Only IC** and **Both IC and CC** protocol illustration.

2.5 Simulation overview: topologies, choice of network and protocol parameters

We now proceed to study via simulation the performance of the protocols discussed in Section 2.4 via performance metrics defined in Section 2.3.2.

2.5.1 Topologies

We consider two different types of topologies. First we look at a topology where nodes are uniformly distributed in an arena. As mentioned earlier, this distribution does not allow for the fact that the locations of nodes in a WSN will likely exhibit some degree of clustering. With this in mind we also look at a particular clustered topology called a Neyman-Scott cluster process.

Differentiating a uniform distributed from a cluster process.

The Laplacian matrix, also known as the admittance matrix [24], [25], is a matrix representation of the graph. For an undirected graph $G = (V, E)$, let d_i denote the degree of node i, $i = 1..N$. The Laplacian matrix is defined to be the matrix:

$$L(i,j) = \begin{cases} d_i, & \text{if } i = j, \\ -1, & \text{if } i \text{ and } j \text{ are adjacent}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.11}$$

For a graph G with eigenvalues $\lambda_0 \leq \lambda_1 \leq .. \leq \lambda_{n-1}$:

1. L is always positive-semidefinite ($\forall i, \lambda_i \geq 0$).

2. λ_0 is always 0.

The algebraic connectivity of a graph is the second smallest eigenvalue λ_1 of the Laplacian matrix. This eigenvalue is also known as the Fiedler eigenvalue [26] and provides information regarding various important properties for the graph such as clustering, synchronizability, as well its susceptibility to failures. In the context of clustering, a low Fiedler value indicates that the graph has good clustering properties [25], [27].

Table 2.3 shows the Fiedler eigenvalue's of the various realizations of uniformly distributed and clustered WSN topologies (we discuss the clustering process shortly). The average Fiedler eigenvalues for the different realizations of the uniform and clustered WSN topologies are: $\bar{\lambda}_1^{\text{Unif}} = 0.5752$, $\bar{\lambda}_1^{\text{Clust}} = 0.1470$. The lower value of $\bar{\lambda}_1^{\text{Clust}}$ in comparison to $\bar{\lambda}_1^{\text{Unif}}$ implies a higher degree of clustering for this topology.

We define the network radius as the radius of the circular arena within which the nodes are placed. The following simulation parameters remain unchanged for all simulations: noise power, $\eta = 10^{-6}$ W; SINR threshold, $\beta = 4$ (≈ 6 dB); path loss

Table 2.3: Fiedler eigenvalues (λ_1) for different realizations of uniformly distributed and clustered WSN.

λ_1 for diff. realizations of a Uniform WSN	λ_1 for diff. realizations of a Uniform WSN (contd)	λ_1 for different realizations of a Clustered WSN
0.2603	0.6016	0.0942
0.2730	0.6023	0.1178
0.2959	0.6044	0.1288
0.3025	0.6080	0.1305
0.3538	0.6296	0.1333
0.3622	0.6804	0.1424
0.3837	0.6859	0.1468
0.3887	0.7450	0.1481
0.4575	0.7722	0.1499
0.4885	0.7778	0.1514
0.5020	0.7833	0.1528
0.5052	0.7977	0.1579
0.5440	0.7988	0.1781
0.5547	0.8191	0.1784
0.5763	1.1019	0.1932

exponent $\alpha = 4$; delay bound $D_{\max} = 500$ seconds. Our simulator is written in Java and we run simulations to achieve a confidence interval (CI) of 99% with a relative error of 1%. Each data point is the average over 30 different randomly generated topologies and 10 runs for the topology under consideration.

Uniformly distributed WSN

To generate a random uniformly distributed topology the input parameters are (i) the radius of the arena and (ii) the spatial intensity of points ϑ_p. The output is a uniformly distributed point process. The algorithm to generate each realization is given in Algorithm (2.12). **Fig. 2.9(a)** is one such realization of this uniformly distributed WSN. **Fig. 2.10(a)** shows the neighbor coordination graph ($\mathcal{G}$) for this same realization. The results for **Figs. 2.14 - 2.19** are an average over 30 different realizations of a uniformly distributed point process, with results for each topology being an average over 10 runs. We use $R = 57$ m and $\vartheta_p = 0.01$. This results in the average number of points per realization being $\pi R^2 \vartheta_p \sim 100$ nodes.

Uniform distribution

Input: parameters R (radius of the arena), ϑ_p (spatial intensity of the points).

(Default values: $R = 57$ m, $\vartheta_p = 0.01$)

Output: Uniform point process

1. Choose N, the number of points as $\sim$ Poisson($\pi R^2 \vartheta_p$).
2. For each point $j = 1, \ldots, N$:
   - 2.1. choose a random distance $D_j \sim$ Uniform$(0, R)$.
   - 2.2. choose a random orientation $\theta_j \sim$ Uniform$(0, 2\pi)$.
   - 2.3. place point at a location y_j that is at distance D_j from the origin with angle θ_j.
3. Return $y_1, \ldots, y_N$.

(2.12)

Clustered graph

Definitions and notation are adopted from [28]. Neyman-Scott cluster processes [28] are constructed as follows. The parent points form a stationary Poisson process $\psi_p = \{x_1, x_2, ..\}$ of intensity ϑ_p. Each cluster consists of a parent point and its associated child points. The clusters can then be represented by $N^{x_i} = N_i + x_i$ for each $x_i \in \psi_p$, where N_i is a set of translation vectors specifying the location of each child from the parent. For example, **Fig. 2.8** shows child points for $N := \{(1,1), (-1,0)\}$ and $x_i = (0,0)$. The distribution of the child point translation vectors N_i is independent of the parent process. The intensity of the cluster process is $\vartheta = \vartheta_p \bar{c}$,

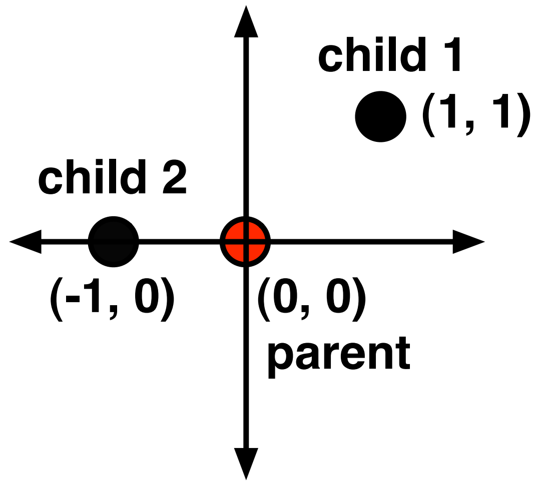


Figure 2.8: Sample cluster process.

where $\bar{c}$ denotes the average number of points in each cluster. The number of points per cluster is Poisson with intensity $\bar{c}$. A *Thomas cluster process* is a particular Neyman-Scott cluster process where the distance of each child point from its parent point is normally distributed with mean 0 and variance σ^2 and the orientation angle is uniform. To generate a realization of this clustered topology the input parameters are (i) the radius of the arena, (ii) the spatial intensity of parent points ϑ_p, (iii) the average number of child points per parent $\bar{c}$, (iv) the variance of the normal distribution σ^2 that specifies the distribution of each child point around the parent. The algorithm to generate each realization is given in Algorithm (2.13). Child points that fall outside the arena are discarded. The results for **Figs. 2.20** and **2.21** are an average over 15 different realizations of a clustered point process, with results for each topology being an average over 10 runs. Our results for the clustered topology are averaged over a fewer number of realizations because of running time issues associated with running simulations on this topology.

Cluster process

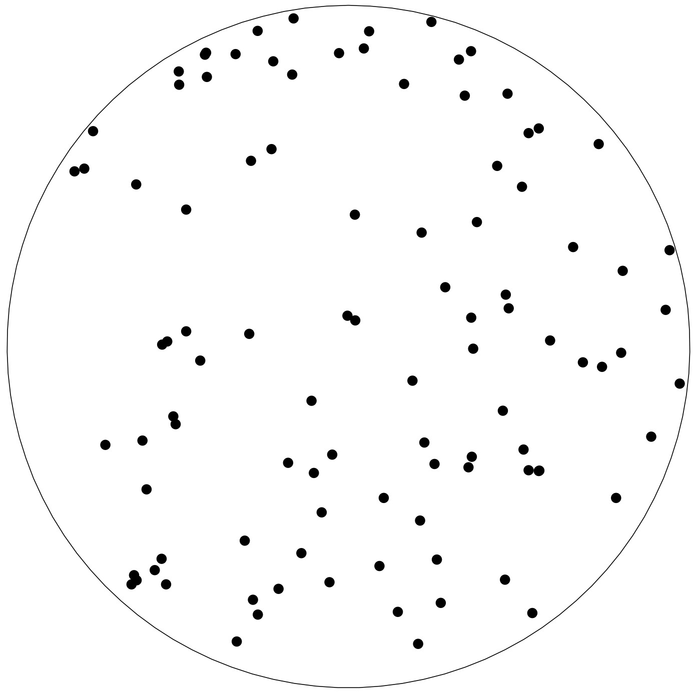
Input: parameters R, ϑ_p, $\bar{c}$, σ^2.

(Default values: $R = 40$ m, $\vartheta_p = 0.002$, $\bar{c} = 12$, $\sigma = 6$)

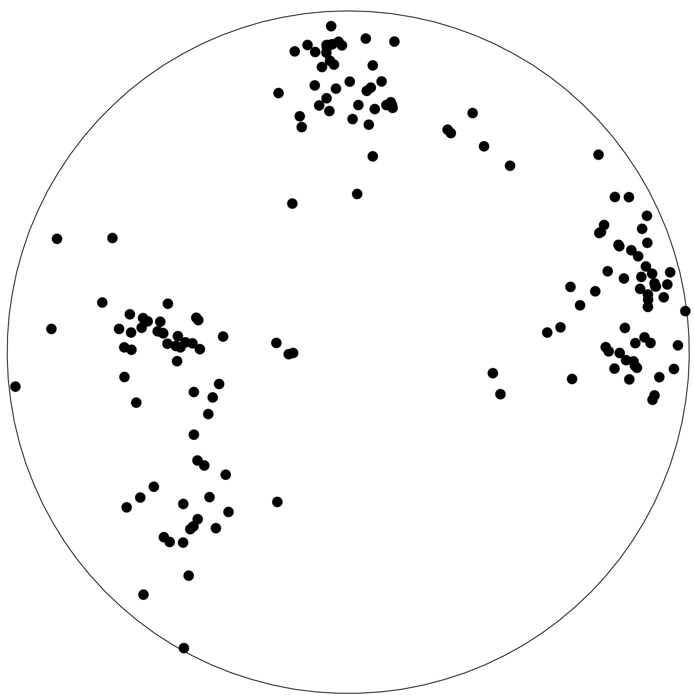
Output: Clustered point process.

1. Choose N, the number of parent points as $\sim$ Poisson$(\pi R^2 \vartheta_p)$.
2. Place the N parent points uniformly at random over $\mathcal{A} = (0, R)$, $\psi_p = \{x_1, \ldots, x_N\}$.
3. For each parent point: $i = 1, \ldots, N$: (2.13)
   - 3a. choose a random number of child points: $M_i \sim$ Poisson$(\bar{c})$.
   - 3b. for each child point $j = 1, \ldots, M_i$.
     - 3.1 choose a random distance $D_{i,j} \sim N(0, \sigma^2)$.
     - 3.2 choose a random orientation $\theta_{i,j} \sim$ Uniform$(0, 2\pi)$.
     - 3.3 place child point i of parent point j at a location y_{ij} that is at distance $|D_{ij}|$ from point x_i with angle $\theta_{i,j}$.
4. Return $\{y_{ij}, j = 1, \ldots, M_i, i = 1, \ldots, N\}$.

The *Thomas cluster* parameters are $\vartheta_p = 0.002$, $\bar{c} = 12$ and $\sigma = 6$. Also we use $R = 40$ m as opposed to $R = 57$ m for the uniformly distributed WSN case. This results in the average number of points per realization of the cluster process being $\pi R^2 \vartheta_p \bar{c} \sim 120$ nodes. When constructing the cluster process we needed to ensure clustering while maintaining connectivity of the graph. Additionally, we had to ensure that the number of nodes in the network wasn't too high (due to running time issues associated with running simulations on this topology). In order to address these points we use a smaller sized arena for the clustered WSN case ($R = 40$ m as opposed to $R = 57$ m used for the uniformly distributed WSN). **Fig. 2.10(b)** shows

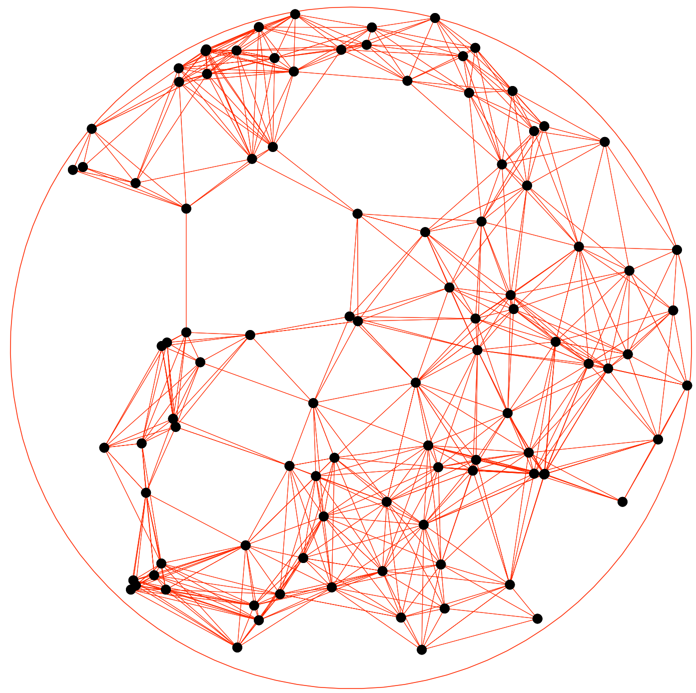


(a) 100 nodes uniformly distributed.

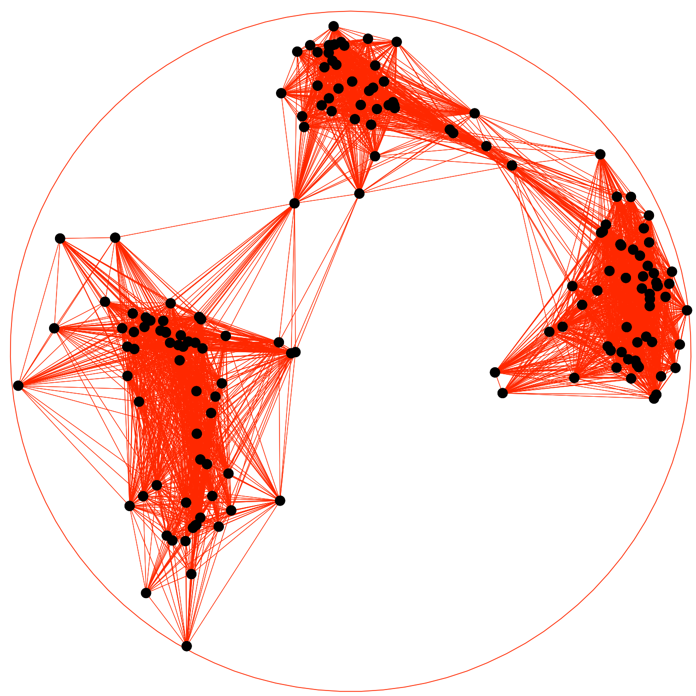


(b) 160 nodes placed using a cluster process.

Figure 2.9: Top: 100 nodes uniformly distributed over a circular arena of radius $R = 57$ m, with $\vartheta_p = 0.01$. Bottom: 160 node *Thomas cluster* process with $\vartheta_p = 0.002, R = 40$ m, $\bar{c} = 12$ and $\sigma = 6$.



(a) Neighbor coordination graph for WSN with 100 nodes uniformly distributed.



(b) Neighbor coordination graph for 160 nodes, placed using a cluster process.

Figure 2.10: Top: Neighbor coordination graph for the uniformly distributed WSN, Bottom: Neighbor coordination graph for the clustered WSN. Here $r : (i, j) \in \mathcal{E} \iff d(X_i, X_j) \le r$ is 22.3 m for both graphs.

the neighbor coordination graph ($\mathcal{G}$) for one such realization of the clustered WSN. Note the higher density of edges in the neighbor coordination graph for the clustered topology.

2.5.2 Choice of network parameters: μ^{-1}, λ

The parameter μ^{-1} specifies a random pause period, thereby ensuring asynchronous operation. After experimenting with a few different values of μ^{-1} we found that as long as (i) the choice of μ^{-1} is large enough so as to ensure asynchronous operation, and (ii) small enough so as to prevent bottlenecks due to large pause durations, the resulting performance metrics depend only on τ^{-1} and not on the choice of μ^{-1}. We also observed that as μ^{-1} approaches 0 seconds (implying synchronous operation), coverage drops, thereby confirming our intuition that an asynchronous mode of operation is favorable for improving performance.

The parameter λ specifies the network wide message generation rate. The choice of λ depends on the choice of μ^{-1}. If the pause duration is large, a large λ will result in messages rapidly congesting the network. Of course the choice of both μ^{-1} and λ depend directly on the size of the network and the density of the nodes. In the case where the average number of hops to traverse the network is large, λ needs to be chosen so as to prevent messages from overwhelming the network. Also, λ for a network in which node density is very high needs to be lower than the λ chosen for a network that has a lower node density. Keeping mind all of the above factors we choose $\mu^{-1} = 1/5$ seconds and $\lambda = 8$ messages per second for the uniformly distributed WSN.

Nodes within a cluster are tightly packed, resulting in a higher density of nodes within clusters as opposed to the uniformly distributed WSN case. As a result, there is a possibility of congestion (due to a message bottleneck) within a cluster if we use the same λ as before. To mitigate this, we use $\lambda = 5$, which is lower than the $\lambda = 8$ we used in the uniformly distributed WSN case. We use $\mu^{-1} = 1/5$ seconds (as before).

2.5.3 Choice of protocol parameters: $\delta_I, \upsilon_C, \gamma_{I,C}$

The parameters δ_I and υ_C help in improving the efficiency and reducing collisions, respectively. The higher the values of δ_I and υ_C, the greater the improvement in performance, however too high a value may negatively impact the coverage. As we increase δ_I the requirements for transmission become more stringent, as a result a node will only transmit if a high fraction of its neighbors do not have the message. Similarly, as we increase υ_C a node will transmit only if a high fraction of its neighbors are in the *idle* state. A high value of υ_C leads to increasing message delays, which in turn results in messages timing out, resulting in reduced coverage. We base our selection of $\delta_I, \upsilon_C, \gamma_{I,C}$ on the coverage (C^π) and efficiency (η^π) metrics. For the **Only IC** and **IC and CC** protocols we select $\delta_I = 0.2$, $\gamma_{I,C} = 0.2$ based on our simulation results, which indicate that these values ensure good coverage in both the underloaded and overloaded regimes. Choosing $\delta_I, \gamma_{I,C} \geq 0.2$ severely limits the achievable coverage in the underloaded regime.

We next present simulation results showing coverage (C^π) and efficiency (η^π) under the three protocols as we vary the average transmission duration (τ^{-1}) for various coordination threshold parameters for the case of the uniformly distributed WSN. **Fig. 2.11** shows Coverage (C^π) vs. mean Tx duration (τ^{-1}) for the (i) **Only IC** and (ii) **IC and CC** protocols for various coordination threshold parameters $\delta_I, \gamma_{I,C}$ for one realization of a uniformly distributed WSN. **Fig. 2.12** shows Efficiency (η^π) vs. mean Tx duration (τ^{-1}) for the (i) **Only IC** and (ii) **Only CC** protocols for the various coordination threshold parameters for one realization of a uniformly distributed WSN.

Fig. 2.11 : Coverage (C^π) vs. mean Tx duration (τ^{-1})

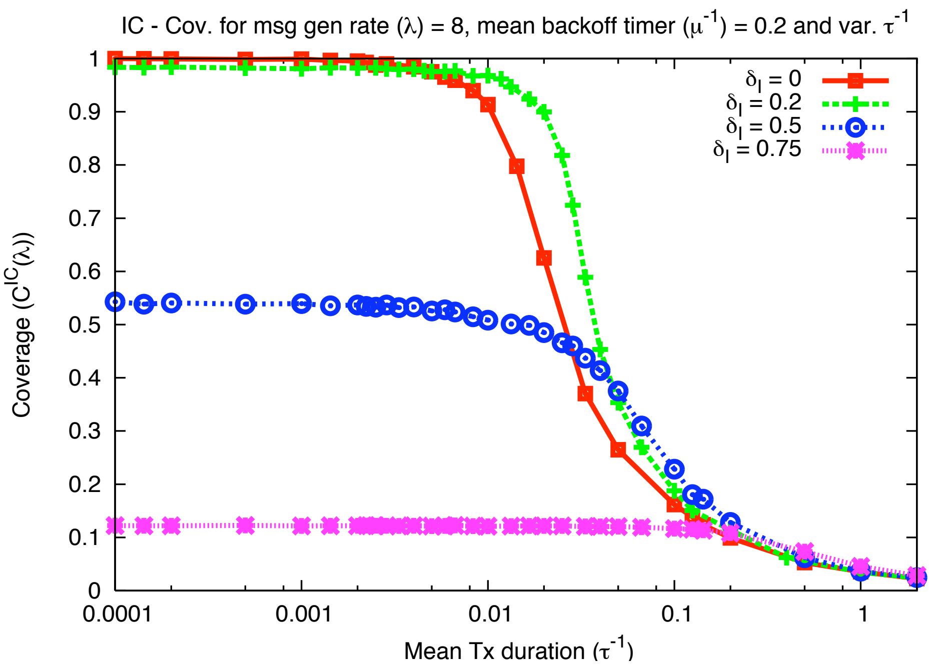
Only IC (Fig. 2.11(a)): There are two clear regimes with respect to τ^{-1}. When τ^{-1} is low, the mean Tx duration is small, as a result collisions are negligible and messages exit the system very quickly. Hence we refer to the low τ^{-1} region as *underloaded*. When τ^{-1} is high, messages congest the network and we call this the *overloaded* region. As δ_I is increased the achievable coverage in the underloaded regime (low τ^{-1}) drops. Choosing $\delta_I \approx 0.2$ ensures near perfect coverage in the underloaded regime and at the same time maintains fair coverage in the overloaded regime. These regimes are highlighted in **Fig. 2.13**.

IC and CC (Fig. 2.11(b)): Observe that when $\gamma_{I,C} < 0.5$ there is no discernible difference in coverage. Also when $\gamma_{I,C} > 0.7$ coverage in the underloaded regime suffers. $\gamma_{I,C} \approx 0.6$ gives the best performance in the underloaded and at the same time maintains competitive coverage in the overloaded regimes.

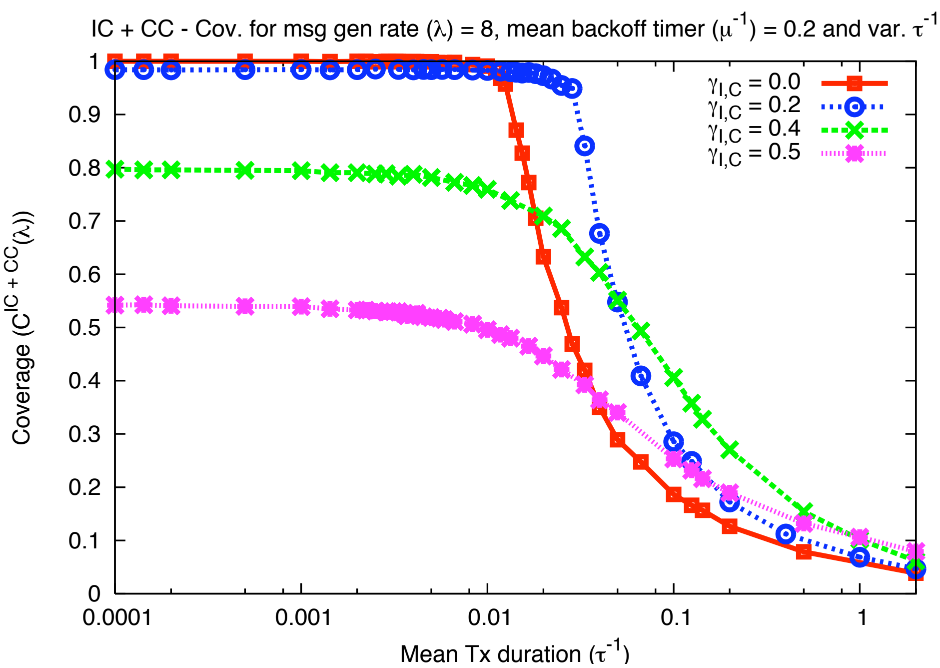
Fig. 2.12 : Efficiency (η^π) vs. mean Tx duration (τ^{-1})

Only IC (Blue lines): As τ^{-1} increases η drops, due to the lack of CC, which results in an increase in collisions thereby resulting in a drop in receptions (and efficiency).

Only CC (Red lines): As τ^{-1} increases the benefit of CC is realized resulting in an increase in η. When τ^{-1} is small collisions are negligible. The function of CC is to coordinate transmissions so as to increase probability of success, *it does not specifically improve efficiency*. Hence for low τ^{-1} efficiency is 1. As τ^{-1} increases, increasing backoffs results in a decrease in transmission attempts due to messages timing out, which leads to reduced coverage. Hence although we observe an improvement in efficiency as τ^{-1} increases, this comes at the cost of reduced coverage.



(a) IC but No CC protocol.



(b) Both IC and CC protocol.

Figure 2.11: Uniformly distributed WSN - Coverage (C^{π}) vs. mean transmission duration (τ^{-1}) for the (a) **Only IC** and (b) **IC and CC** protocols with message generation rate (λ) = 8 and mean backoff duration (μ^{-1}) = 0.2.

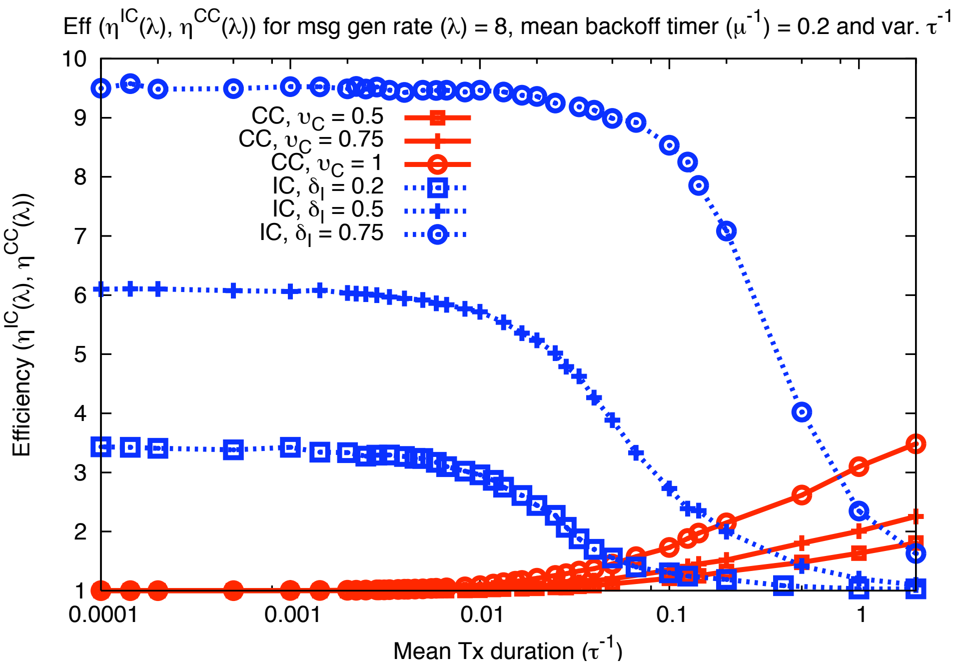


Figure 2.12: Uniformly distributed WSN - Efficiency (η^{π}) vs. mean transmission duration (τ^{-1}) for the **Only IC** (broken lines) and **Only CC** (solid lines) protocols with message generation rate (λ) = 8 and mean backoff duration (μ^{-1}) = 0.2.

2.6 Results for uniformly distributed WSN

Before moving on to discussing the various results we provide a brief summary of the various plots:

1. **Fig. 2.13** shows Capacity ($\lambda^{\pi}_{\max}$) vs. mean Tx duration (τ^{-1}) for the various protocols for one realization of a uniformly distributed WSN.

2. **Fig. 2.14** shows Coverage (C^{π}) vs. mean Tx duration (τ^{-1}) for the various protocols and is an average taken over 30 different realizations of a uniformly distributed WSN.

3. **Fig. 2.15** shows Efficiency (η^{π}) vs. mean Tx duration (τ^{-1}) for the various protocols and is an average taken over 30 different realizations of a uniformly

distributed WSN.

4. **Fig. 2.16** shows Delay (D^π) vs. mean Tx duration (τ^{-1}) for the various protocols and is an average taken over 30 different realizations of a uniformly distributed WSN.

5. **Fig. 2.17** shows Collisions (ν^π) vs. mean Tx duration (τ^{-1}) for the various protocols and is an average taken over 30 different realizations of a uniformly distributed WSN.

6. **Fig. 2.18** shows Energy consumption (ζ^π) vs. mean Tx duration (τ^{-1}) for the various protocols and is an average taken over 30 different realizations of a uniformly distributed WSN.

7. **Fig. 2.19** shows Overhead vs. mean Tx duration (τ^{-1}) for the various protocols and is an average taken over 30 different realizations of a uniformly distributed WSN.

8. **Fig. 2.20** shows Coverage (C^π) vs. mean Tx duration (τ^{-1}) for the various protocols and is an average taken over 15 different realizations of a cluster process.

9. **Fig. 2.21** shows Delay (D^π) vs. mean Tx duration (τ^{-1}) for the various protocols and is an average taken over 15 different realizations of a cluster process.

10. **Fig. 2.22** shows operation of the various protocols when operating under a limited energy budget of $0.5 - 1$ J per node for one realization of the uniformly distributed WSN.

We now compare the various protocols on the basis of the performance metrics discussed in Section 2.3.2 for the uniformly distributed WSN.

Fig. 2.13 : Capacity ($\lambda^{\pi}_{\max}$) vs. mean Tx duration (τ^{-1})

This plot shows the capacity ($\lambda^{\pi}_{\max}$) subject to a coverage constraint of $C_{\min} = 90\%$ versus the mean transmissions duration (τ^{-1}), a proxy for packet length. This plot illustrates the appropriate domain of the two controls we are considering: in the underloaded regime there is no need for control since there are no collisions. In the overloaded regime there is also no need for control since the controls we employ are unable to provide any meaningful capacity improvements under heavy loads due to the high collisions experienced. In between these two regimes, however, we observe the benefits of employing IC and CC. The IC component reduces the number of transmissions, improving efficiency and coverage, whereas the backoff mechanism for CC improves coverage at the cost of increased delay. These effects increase the capacity.

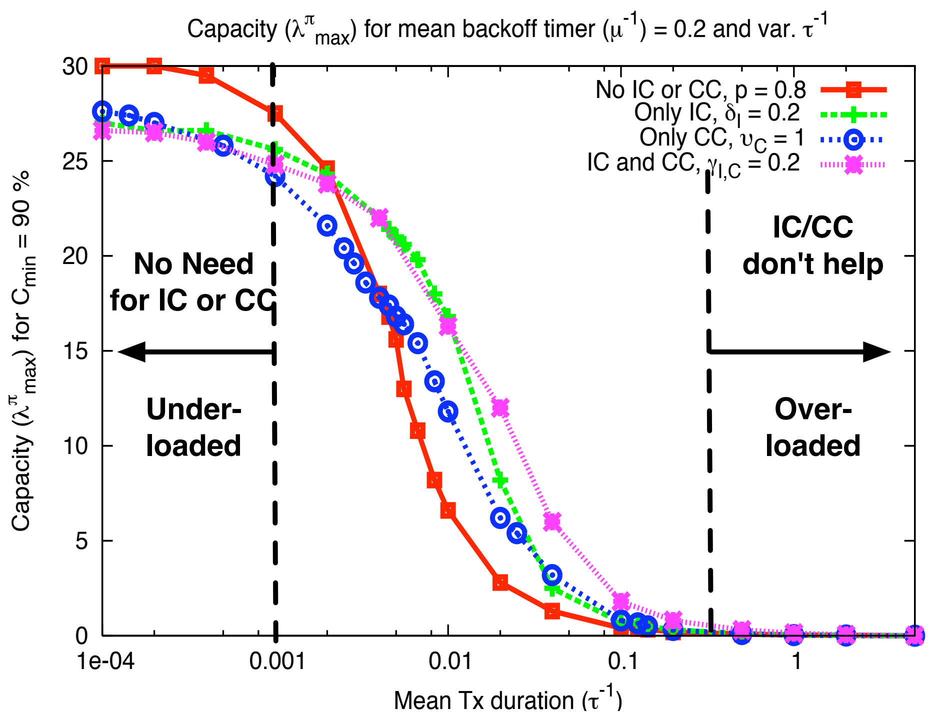


Figure 2.13: Uniformly distributed WSN - Capacity ($\lambda^{\pi}_{\max}$) vs. mean transmission duration (τ^{-1}) for all protocols with minimum coverage ($C_{\min}$) = 90% and mean backoff duration (μ^{-1}) = 0.2.

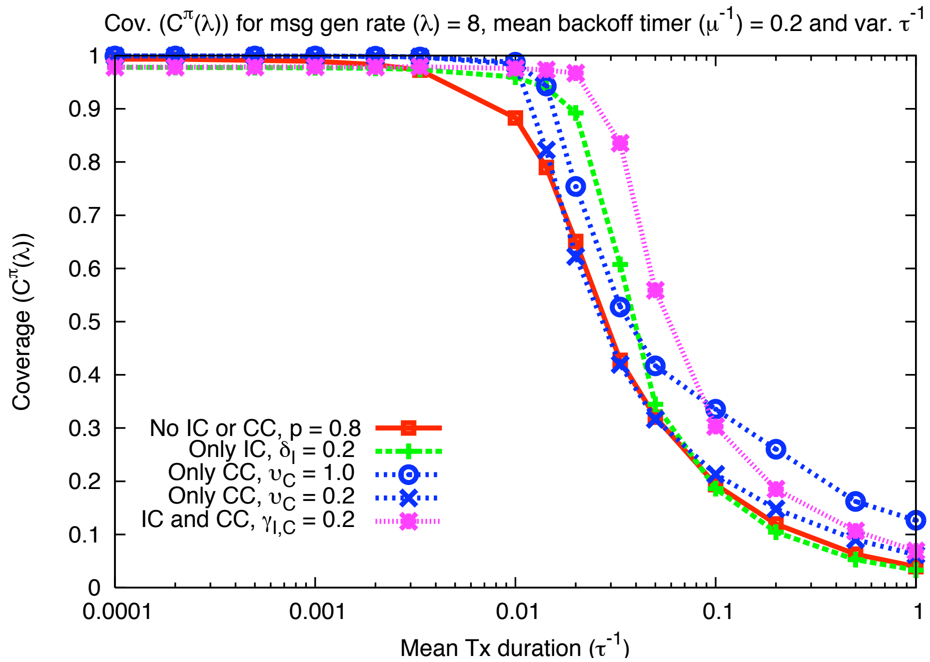


Figure 2.14: Uniformly distributed WSN - Coverage (C^{π}) vs. mean transmission duration (τ^{-1}) for all protocols with message generation rate (λ) = 8 and mean backoff duration (μ^{-1}) = 0.2.

Fig. 2.14: Coverage (C^{π}) vs. mean Tx duration (τ^{-1})

This plot shows coverage vs mean Tx duration (τ^{-1}) for all the protocols. Just like the capacity plot we observe that IC and CC improves the achievable coverage for τ^{-1} roughly between 0.005 and 1. However as τ^{-1} decreases (below 0.005), collisions drop and high coverage is achievable without the use of IC or CC. Also note that in the overloaded regime, CC alone is best for τ^{-1} large (when $\upsilon_C = 1.0$). The `CTS` deferral requests of CC in the overloaded regime have the effect of spacing out transmission attempts in time, achieving a higher coverage than the other protocols, at the expense of increased delay. When operating in the overloaded regime, the **Only IC** suffers from a high number of failed transmissions resulting in reduced coverage. This is a result of the lack of any mechanism to reduce collisions. Adding

CC to this protocol results in improved coverage in the overloaded regime. When all coordination threshold parameters have the same value (0.2) we observe that the **Only CC** protocol performs better than the gossip protocol but not as well as the **Only IC** and **IC and CC** protocols.

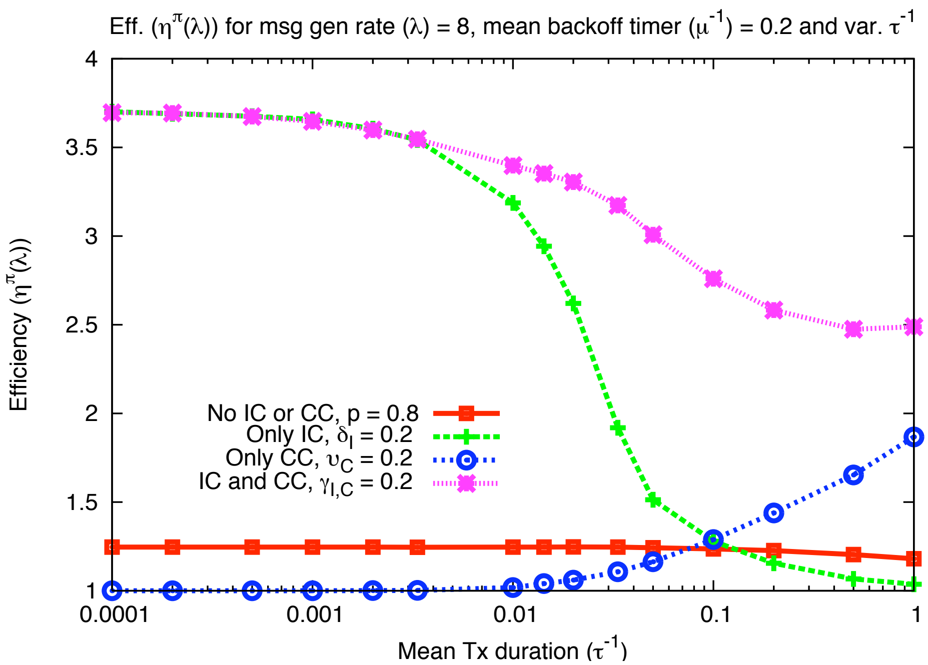


Figure 2.15: Uniformly distributed WSN - Efficiency (η^{π}) vs. mean transmission duration (τ^{-1}) for all protocols with message generation rate (λ) = 8 and mean backoff duration (μ^{-1}) = 0.2.

Fig. 2.15: Efficiency (η^{π}) vs. mean Tx duration (τ^{-1})

Only CC: When τ^{-1} is small, collisions are negligible, as a result backoffs are negligible. Therefore almost all nodes end up transmitting at the first available opportunity. The result is an efficiency of nearly 1 as the ratio of unique receptions per transmission is 1. Hence for very small τ^{-1}, CC is extremely inefficient. When operating under CC, as τ^{-1} increases, backoffs increase, resulting in more timeouts, which in turn results in a fewer number of transmissions, and hence higher efficiency. **Only IC:** IC does not attempt to coordinate transmissions. As a result there is no attempt to postpone transmissions with increasing τ^{-1}. This leads to an increase in collisions, resulting in a decrease in the number of unique receptions. The end result is a drop in η which tends to 1 (as τ^{-1} grows large). **IC and CC:** Here we observe a drop in η with increasing τ^{-1} due to increased collisions. The CC component however mitigates this effect, relative to the **Only IC** protocol. **No IC and No CC:** Efficiency is relatively insensitive to τ^{-1} over the simulated range; it begins to drop first around $\tau^{-1} \approx 0.4$ once the increase in collisions reduces the ratio of successful receptions over transmission attempts.

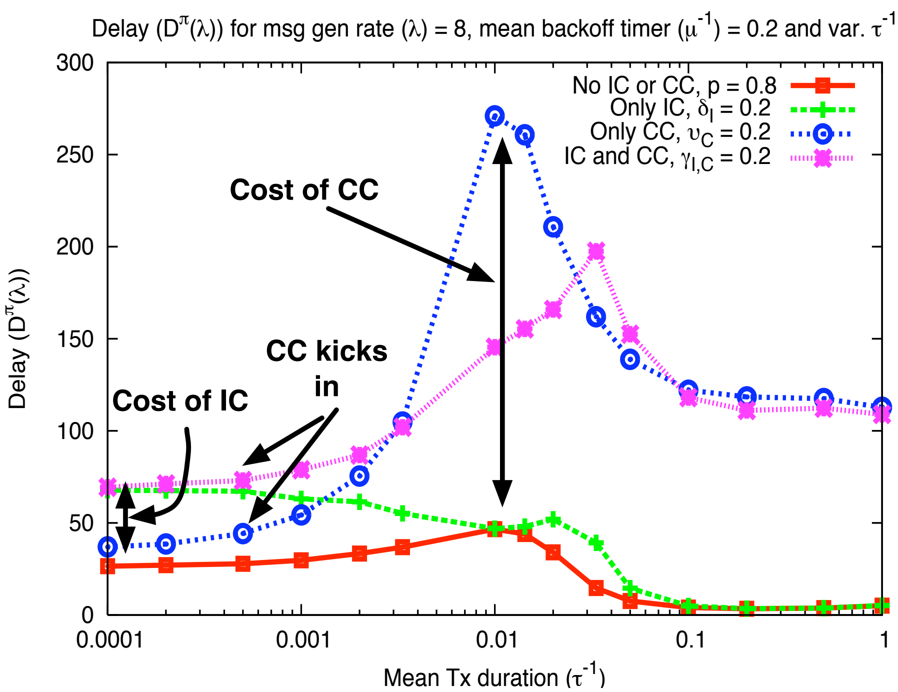


Figure 2.16: Uniformly distributed WSN - Delay (D^{π}) vs. mean transmission duration (τ^{-1}) for all protocols with message generation rate (λ) = 8 and mean backoff duration (μ^{-1}) = 0.2.

Fig. 2.16: Delay (D^π) vs. mean Tx duration (τ^{-1})

Effect of CC: As τ^{-1} increases, message delay increases due to increasing backoffs. We observe that the delay peaks at a certain point, beyond which it begins to drop. As mentioned earlier we set $D_{\max} = 500$ seconds. The reason that the maximum delay observed is not $D_{\max}$ is because not all messages will incur this maximum delay, and prior to congestion, messages exit the system naturally without timeout. Since all metrics are computed as a message average, the lower delays experienced by the earlier messages reduces the average delay to a value that is less than $D_{\max}$. We also observe that once delay peaks, it begins to drop. Increasing τ^{-1} means that message transmissions last longer. Successful reception requires that the SINR at the receiver be sufficiently high for the duration of the transmission. As transmission durations increase the likelihood of a new transmission seeing potential interference increases. The backoff mechanism focuses on providing a potential transmission the best opportunity to be successfully received, successful reception, however, *depends on the SINR*. As τ^{-1} increases there is an increased likelihood of strong interference, which results in a drop in successful receptions, which in turn results in a drop in coverage and hence delay.

Effect of IC: As τ^{-1} increases the lack of CC results in a drop in successful receptions, which results in a drop in delay. Also observe that in the underloaded regime, protocols that employ IC have higher delays than those protocols that do not employ IC. This is because IC reduces the number of transmissions, which means that the message will require more time to traverse the network (as not all nodes that receive the message will transmit it). This improvement in efficiency resulting from a reduction in redundant transmissions comes at the cost of higher delay.

Fig. 2.17: Collisions (ν^{π}) vs. mean Tx duration (τ^{-1})

This plot shows the benefit of CC on collisions. When τ^{-1} is small transmissions are near-instantaneous, as a result collisions are negligible for all the protocols. The rate of collisions increases as τ^{-1} increases. This increase is more rapid for protocols not employing CC. Observe also that adding **IC** to the **Only CC** protocol results in lower number of collisions in comparison to the **Only CC** protocol.

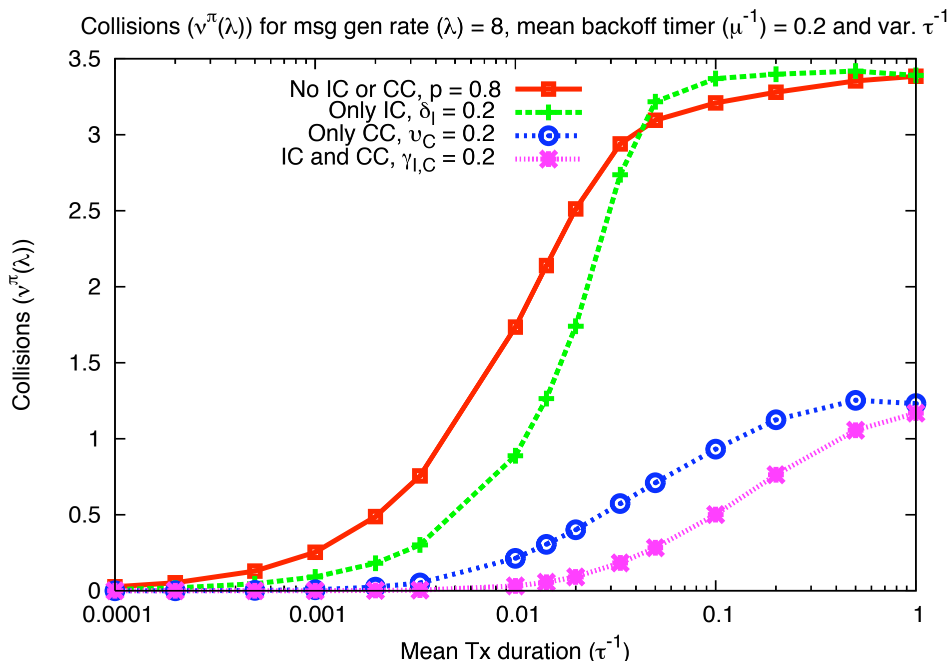


Figure 2.17: Uniformly distributed WSN - Collisions (ν^{π}) vs. mean transmission duration (τ^{-1}) for all protocols with message generation rate (λ) = 8 and mean backoff duration (μ^{-1}) = 0.2.

Fig. 2.18: Energy consumption (ζ^{π}) vs. mean Tx duration (τ^{-1})

In the underloaded regime, we observe that the normalized energy consumption is under 1 (J) for all the protocols. Observe that the energy consumption for the protocols that employ IC is slightly more than those protocols that do not employ IC in this regime. This is the *cost of IC*, and is attributable to the longer message delay under the IC protocols in the underloaded regime. The amount of extra energy is proportional to the degree of IC.

For the protocols employing CC, moving towards the overloaded regime results in increases in both the frequency and duration of backoffs. The increased frequency and duration of backoffs results in increased message delays, which increases the amount of time nodes spend in the idle state, which in turn results in increased energy consumption. There is also an additional energy expenditure due to the increased use of `RTS/CTS` control messages. Observe that between $\tau^{-1} \sim 0.003 - 0.1$ adding **IC** to the **Only CC** protocol results in lower energy consumption (at $\tau^{-1} = 0.02$ this reduction is over 40%).

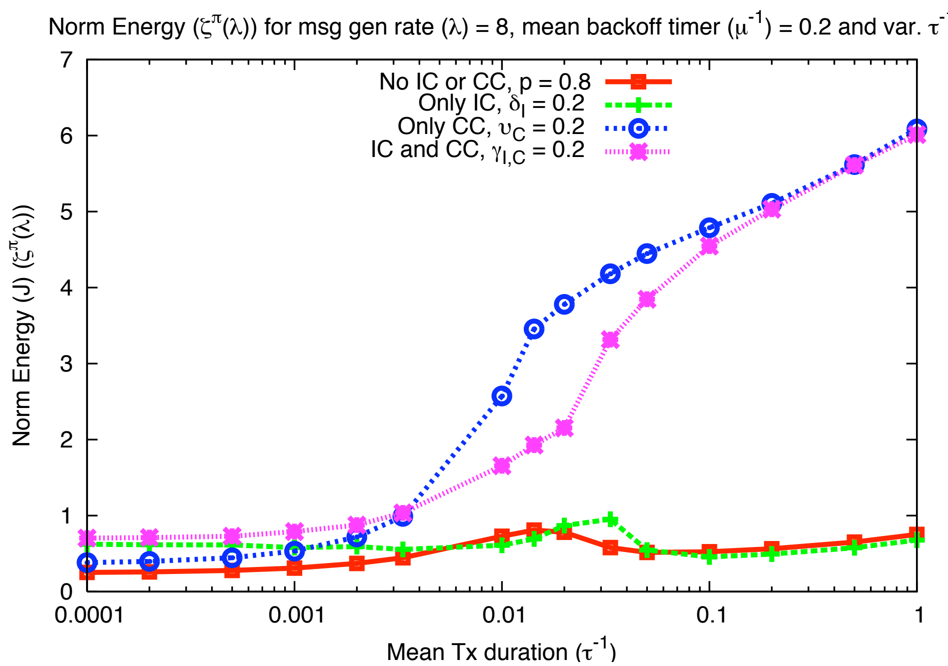


Figure 2.18: Uniformly distributed WSN - Normalized Energy (ζ^{π}) vs. mean transmission duration (τ^{-1}) for all protocols with message generation rate (λ) = 8 and mean backoff duration (μ^{-1}) = 0.2.

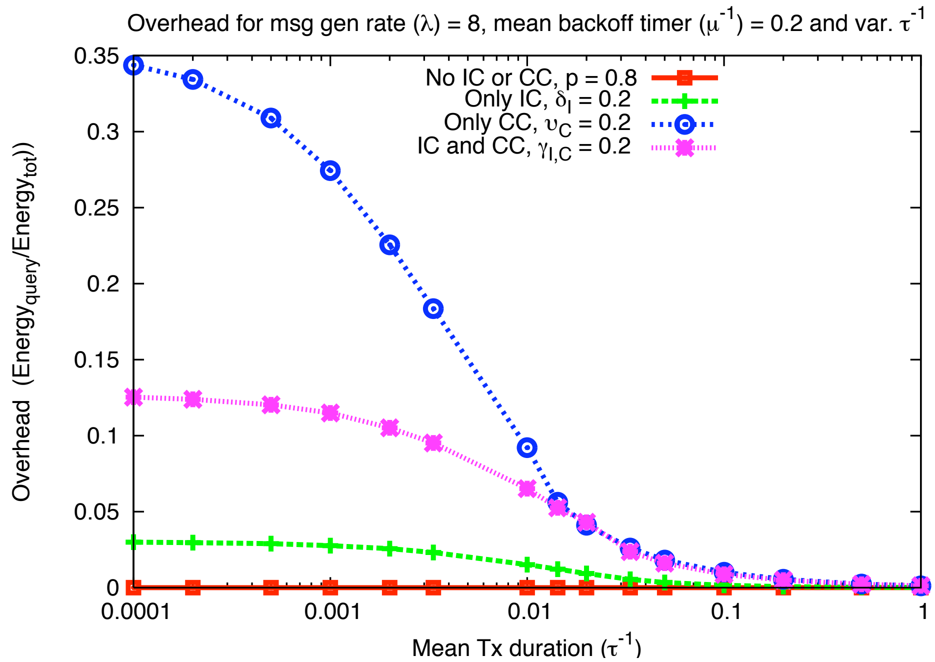


Figure 2.19: Uniformly distributed WSN - Overhead ($\text{Energy}_{query}/\text{Energy}_{tot}$) vs. mean transmission duration (τ^{-1}) for all protocols with message generation rate (λ) = 8 and mean backoff duration (μ^{-1}) = 0.2.

Fig. 2.19: Overhead ($\text{Energy}_{query}/\text{Energy}_{tot}$) vs. mean Tx duration ($\tau^{-1}$)

This plot depicts the fractional overhead measured as the ratio of energy expended in querying to total energy expended.

Underloaded regime. The **No IC and No CC** protocol does not use any local information in making transmission decisions, as a result overhead for this protocol is zero. For the remaining protocols we observe that the fractional overhead is at its maximum value in the underloaded regime and drops as we move towards the overloaded regime. We observe that the **only CC** protocol has the highest overhead. In the underloaded regime every node will end up transmitting every message, querying its neighbors in the process. The **Only IC** protocol employs fewer transmissions and hence has lower overhead than the **Only CC** protocol. **Overloaded regime.**

Only CC: As we move towards the overloaded regime, backoffs increase. Also since transmission durations increase, a large amount of time is spent waiting for the best opportunity to transmit. As a result the fraction of energy expended in querying drops, which is why we observe a drop in overhead for the **Only CC** protocol. **Only IC**: as we move towards the overloaded regime, transmissions durations increase. This results in increasing collisions, leading to lost transmissions, and a drop in query overhead since messages die out. We observe that the overhead is justifiable in the overloaded regime, since the size of queries and responses are small relative to the message size (duration).

2.7 Results for clustered WSN

Fig. 2.20: Coverage (C^{π}) vs. mean Tx duration (τ^{-1})

As observed in the case of the uniformly distributed WSN **Fig. 2.14** the **No IC or CC** protocol is the first protocol that exhibits a drop as we move towards the overloaded regime. **Importance of IC.** Next to exhibit a drop in coverage is the **Only CC** protocol. A major difference between the coverage for the clustered topology vs. coverage for the uniformly distributed case is the difference in coverage between the **Only IC** and **Only CC** protocols. We observe that even with complete CC ($\upsilon_C = 1.0$) the **Only IC** protocol outperforms the **Only CC** protocol over a large range of τ^{-1}. This is a direct consequence of the clustered topology. Clustering leads to increased message congestion, resulting in an increased dependence on **IC** to combat congestion.

Importance of CC. As τ^{-1} increases beyond 0.01 collisions increase and **IC** alone cannot sustain coverage. As a result we observe that the **Only IC** protocol begins to exhibit a drop in coverage. Adding **CC** to the **Only IC** protocol results in the **IC and CC** protocol having the best performance of all protocols as we move towards the overloaded regime. Contrast this to the uniformly distributed case where the **Only CC** protocol with $\upsilon_C = 1.0$ performs better than the **IC and CC** protocol for $\tau^{-1} > 0.1$. This plot highlights the importance of using both IC and CC for a clustered topology.

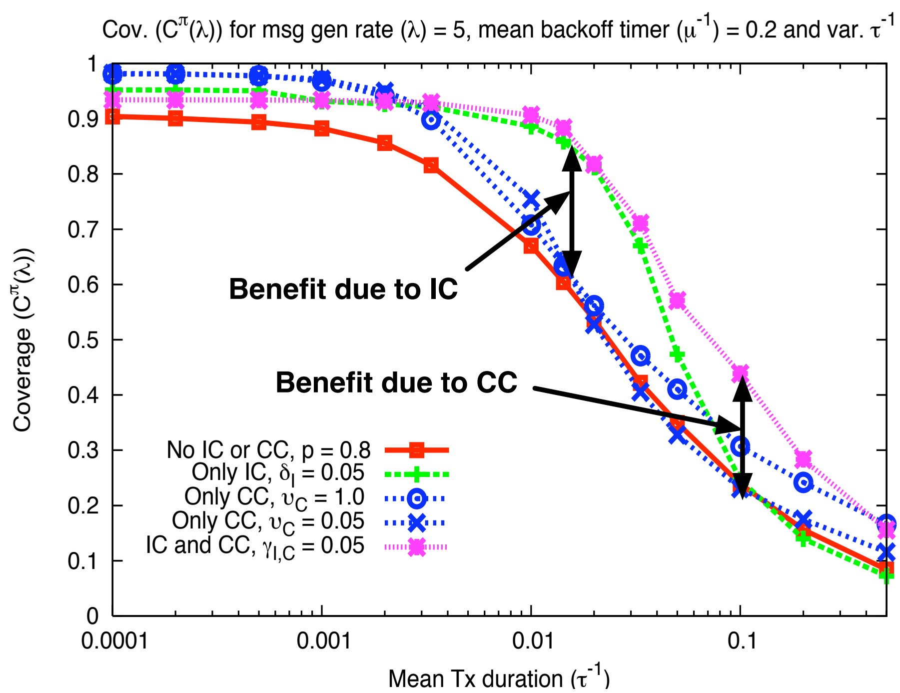


Figure 2.20: Clustered graph - Coverage (C^{π}) vs. mean transmission duration (τ^{-1}) for all protocols with message generation rate (λ) = 5 and mean backoff duration (μ^{-1}) = 0.2.

Fig. 2.21: Delay (D^{π}) vs. mean Tx duration (τ^{-1})

The delay plot for the clustered topology is similar to the delay for the uniformly distributed WSN case. **Effect of CC:** As τ^{-1} increases, message delay increases due to increasing backoffs. Again, once delay peaks, increasing τ^{-1} increases the likelihood of interferers, resulting in a drop in successful receptions, which results in a drop of coverage and hence delay.

Effect of IC: As τ^{-1} increases the lack of CC results in a drop in successful receptions, which results in a drop in delay.

One noticeable difference between this and the delay plot for the uniformly distributed case is the fact that the delay for the **IC and CC** protocol is always less than the delay for the **Only CC** protocol (in the uniformly distributed case they delays are comparable as we move towards the overloaded regime).

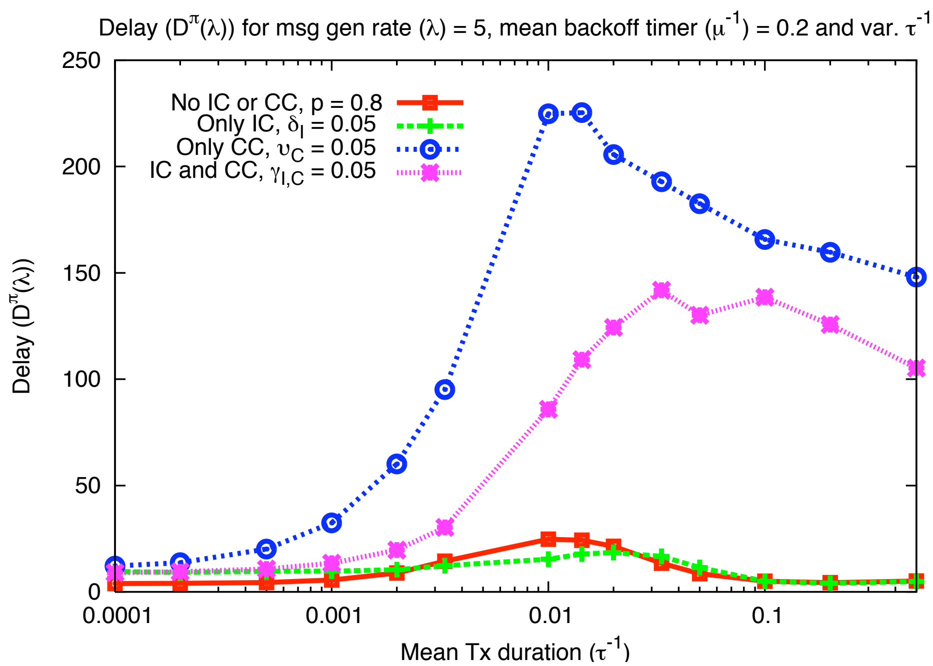


Figure 2.21: Clustered graph - Delay (D^{π}) vs. mean transmission duration (τ^{-1}) for all protocols with message generation rate (λ) = 5 and mean backoff duration (μ^{-1}) = 0.2.

2.8 Results for random node failures

In this section we study the performance of the various protocols for one realization of the uniformly distributed WSN (**Fig. 2.9(a)**) under node failures that result from energy depletion. Each node is assigned a random energy budget between $0.5 - 1$ J under a uniform distribution. Energy consumption for the nodes is then tracked. Once the node's energy budget is depleted, the node fails permanently. Node failures are tracked as a percentage of the total nodes in the network. The simulation is terminated once a specified fraction of the nodes have failed.

Fig. 2.22: Energy budget of $0.5 - 1$ J per node

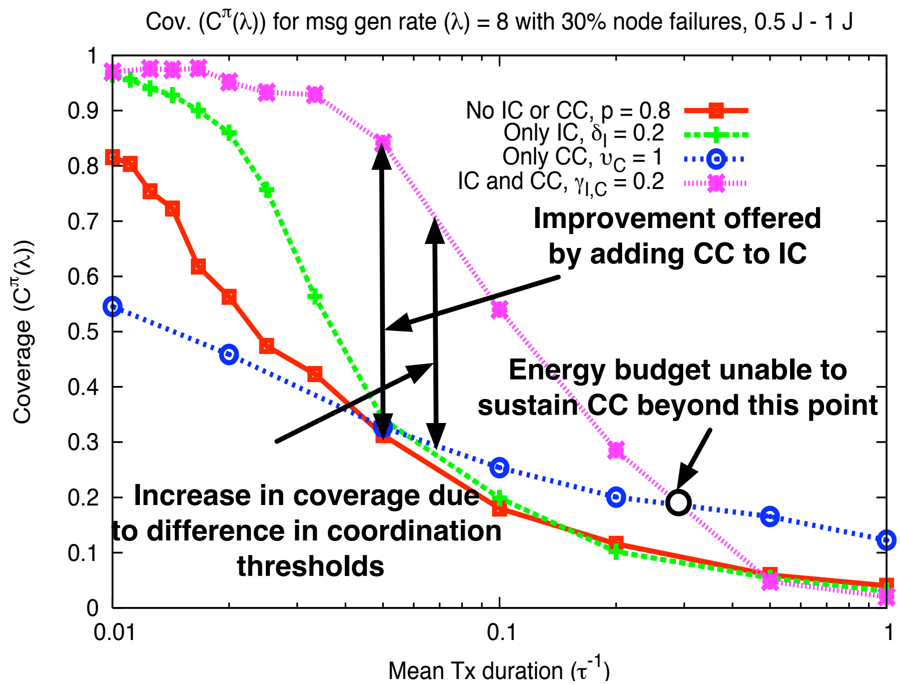
As observed in **Fig. 2.18** the cost of trying to sustain coverage in the overloaded regime is a higher energy budget requirement for the protocols that employ CC. In **Fig. 2.22** we study the effect of energy depletion related node failures on coverage. We study performance for cases where up to 30% and 70% node failures are allowed, before the simulation is terminated.

Fig. 2.22 (top): 30% node failures

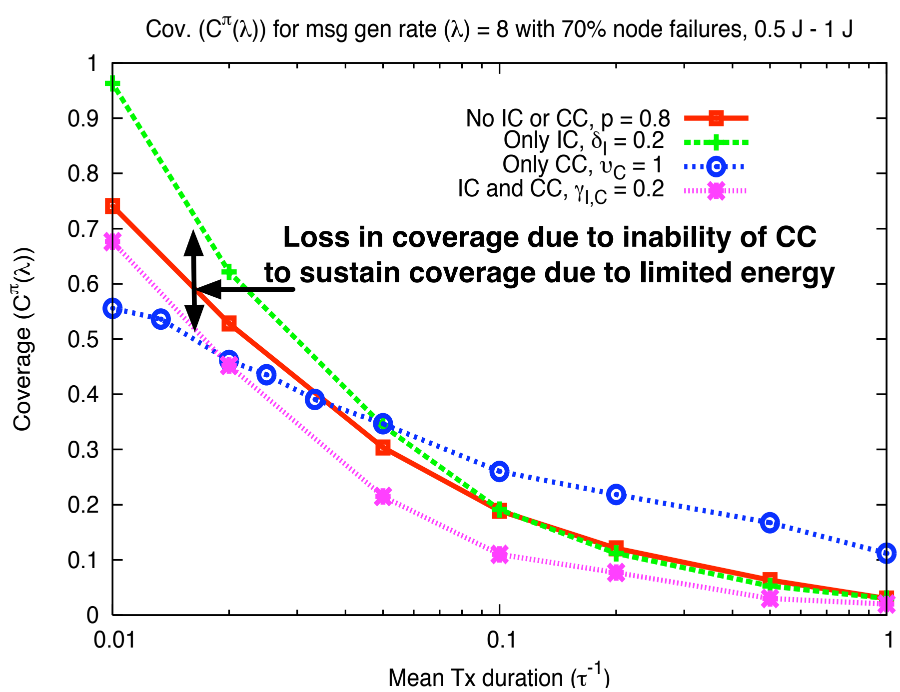
The **IC and CC** protocol performs significantly better than all the other protocols except in the overloaded regime. The increase in coverage of the **IC and CC** protocol over the **Only IC** protocol is attributable to the reduction in collisions achieved by CC, preventing messages from dying out prematurely. The increase in coverage of the **IC and CC** protocol above the **Only CC** protocol is attributable to their different coordination thresholds. With $\upsilon_C = 1.0$, nodes must wait for perfect coordination, which results in message timeouts and hence low coverage. With $\gamma_{I,C} = 0.2$, the CC requirement is much less stringent, resulting in lower delay, less timeout, and increased coverage.

Fig. 2.22 (bottom): 70% node failures

A major difference between this plot and the plot for the 30% case is the coverage for



(a) 30 % node failures, $0.5 - 1$J



(b) 70 % node failures, $0.5 - 1$J

Figure 2.22: Uniformly distributed WSN - Coverage (C) vs. mean transmission duration (τ^{-1}) with message generation rate (λ) $= 8$ and mean backoff duration (μ^{-1}) $= 0.2$, with nodes randomly assigned an energy budget of $0.5 - 1$ J.

the **IC and CC** protocol. We observe that in the 70% case this protocol performs the worst of all protocols over a significant τ^{-1} range. This is because we are now relying more on **CC** to sustain coverage (as specified by the 70% node failures parameter), however the energy budget of $0.5 - 1$ J is insufficient for **CC** to do this, as this budget needs to be divided between two different mechanisms: **IC** and **CC**. Effectively we are expending the complete energy budget of the nodes with no benefit in coverage. When **Only CC** is allowed to use the entire budget we observe that this protocol performs noticeably better than the **IC and CC** protocol.

We summarize the above discussion as follows: *When operating under limited energy constraints it is better to rely purely on CC to sustain coverage in the overloaded regime. The only way to guarantee better coverage for the* **IC and CC** *protocol over the* **Only CC** *protocol is to either limit the reliance on CC, or allocate a large energy budget.*

2.9 Summary

2.9.1 Key findings

- The use of IC or CC will have little or no effect on the broadcast capacity when the network is either *i*) underloaded (as there is no need for coordination), or *ii*) overloaded (as no amount of coordination can help). The use of IC and/or CC can have a significant effect on the broadcast capacity in the intermediate loading regime.

- As network load increases, the use of CC has several effects: *i*) delay increases, *ii*) efficiency increases, *iii*) coverage is higher than under non-CC protocols. Delay increases because nodes must wait an increasingly long time for viable transmission opportunities, thereby increasing the fraction of messages that are dropped due to violation of their delay constraint. Efficiency (the number of successful receptions over the number of transmission attempts) increases since there are fewer transmission attempts. Coverage is higher than under non-CC protocols since without CC messages die out very early due to high collision rate.

- As network load increases, the use of IC has several effects: *i*) the collision rate increases (much more dramatically than with CC), *ii*) the efficiency decreases, *iii*) the delay decreases. Uncoordinated communication attempts are increasingly likely to collide as the network load increases, and this increased collision rate decreases the efficiency. As load increases there are more messages that die out prematurely, and this decreases the average time a message spends in the system, decreasing the delay metric. Finally, in a low-loaded network IC suffers a higher delay than non-IC protocols: transmitting only when a sufficiently high number of neighbors do not yet have the message results in messages propagating more slowly than when this requirement is absent.

- Increasing the degree of coordination required for transmission has both positive and negative effects. Increasing the IC requirement leads to increased efficiency, but if the requirement is too stringent then eventually too many nodes will elect not to transmit and the messages will die out early, thereby reducing coverage. Similarly, increasing the CC requirement leads to reduced collisions, but if the requirement is too stringent then eventually too many nodes wait too long for a suitable transmission time, thereby increasing queueing delay, leading to message time out, and again, to reduced coverage.

- As network load increases, the use of IC reduces the number of redundant transmissions, resulting in an improvement in performance over protocols that employ only CC. Beyond a certain network load however, IC alone is unable to sustain coverage due to the lack of any mechanism to reduce colliding transmissions. As as we move to the over-loaded regime, the use of CC provides an improvement in coverage.

- The fraction of energy spent in querying local state information is high in the underloaded regime where congestion is low and hence delays are short. In the overloaded regime, almost all energy is consumed in keeping the node on because of the long delays required due to high congestion and the relative cost of coordination is very low.

- Message congestion is greater in a clustered WSN as opposed to a uniformly distributed WSN. As a result, even a low degree of IC results in improved performance in comparison to a protocol that does not employ IC. This improvement is clearly evident as we move from the underloaded to the overloaded regime. This is quite different from the uniformly distributed WSN case, where CC alone is able to yield superior performance in the overloaded regime.

- When operating under limited energy constraints it is better to rely purely on CC to sustain coverage in the overloaded regime. The only way to guarantee better coverage when employing both **IC** and **CC** is to either limit the reliance on CC, or allocate a large energy budget.

2.9.2 Design insights

The above findings offer several key design insights. We now demonstrate how these findings can be used at design time to ensure the best performance of the network when operating under various requirements.

- If the operational requirement is to ensure maximum capacity between the underloaded and overloaded regimes then we need to employ both IC and CC. If a design constraint dictates the use of only one of these mechanisms, it is better to use IC as opposed to CC.

- Similarly given the capacity plot for the uniformly distributed WSN we can differentiate the performance of the two mechanisms on the basis of the regime of operation, allowing the designer to decide whether there is any benefit of employing one, both, or neither mechanisms.

- Successful operation may require that coverage never drop below a certain fraction. The designer can use the coverage plot to determine whether the use of a mechanism is justified when operating at a particular τ^{-1}. In the case where the choice of mechanism(s) depends on more than one metric the designer can refer to the various metric plots, using these as a guideline on which to base his selection. For example consider a scenario where delay is a binding constraint. When operating under heavy load, CC will lead to excessive message drops also resulting in low coverage and high node failures due to possible battery depletion. In a situation where we require that coverage be no less than say 50%, the use of CC cannot be justified. This information can be used by the designer to ensure that energy is not wasted.

2.9.3 Future work

There are several promising directions of future work for this section. One of these is to study the performance of the various protocols on a variety of topologies, each of which exhibit unique properties. The results presented here are limited to a uniform distribution of nodes and a clustered topology. Consider two additional topologies: *i*) a small world graph and *ii*) an expander graph. Small world graphs are characterized by their small diameters whereas an expander graph is a sparse graph which has high connectivity. Both topologies are robust to failures. Studying the performance of the protocols on a variety of topologies will yield a better understanding of their performance.

Additionally one may consider implementing the suite of protocols on actual sensor nodes and study their performance for random topologies, and perhaps even some of the topologies discussed in the chapter. Even though an actual implementation might only be possible on a small network (perhaps 10 - 20 nodes), this will offer invaluable insight into possible design and implementation challenges, as well as insight into improvements to the protocols not evident during conceptualization and from the simulation results.

Another direction is to perform a comparison study of the various metrics for the various protocols. Our study of the protocols is currently limited to fixing the network parameters μ^{-1}, λ and studying the relative performance of the various protocols in terms of the defined metrics. An alternate approach would be to study some subset of metrics while keeping all other metrics constant, in effect tuning the coordination threshold parameters. To elaborate, consider the following example: we wish to compare efficiency, delay, and collision quotient for the **Only IC**, **Only CC** and **IC and CC** protocols while ensuring a coverage of 80% for all three protocols, for some preset values of the network parameters. This is done by varying the threshold of coordination for each of these protocols. Similarly one can fix delay for all protocols while comparing the other metrics. This will yield valuable insights about which metrics dominate over others, as well as the relative degrees of coordination required to maintain a certain value of a metric, in effect quantifying the relative dominance of the coordination parameters.

The work presented in this Section has appeared in [3], [4], [5], and is under review for Elsevier Ad Hoc Networks.

3. Transmission coordination for ad hoc networks (AHNs)

3.1 Introduction

Transmission coordination refers to the process of efficiently arranging simultaneous transmissions in space. This forms an important part of scheduling which involves arranging simultaneous transmissions in space, time, and frequency. We assume an ad hoc network (AHN) with limited spectrum, necessitating that concurrent transmissions must be spread out in space to avoid them generating excessive interference for their respective receivers.

Our focus in this chapter is to study optimal transmission coordination under the presence or absence of various communication constraints. These communication constraints can be a result of hardware limitations of the transceiver, or the operational mode of the network. Table 3.1 gives a description of these constraints, and possible means of circumventing them.

The first three constraints, *i.e., half-duplex* (HD), *single reception* (SR), and *unicast* (UC) are termed primary constraints, whereas the *protocol interference* (I) constraint is referred to as a secondary constraint. The HD constraint forbids a node from simultaneously transmitting and receiving, the SR constraint forbids a node from receiving from multiple transmitters, and the UC constraint forbids a node from transmitting distinct information to multiple receivers. The I constraint restricts a node from reception if there is an interfering transmission in its vicinity. A transmission is denoted by a directed edge from the transmitter (tail) to the receiver (head), with edges being weighted. The weights may represent the value of the communication, the quality of the link, *etc.*

The optimization problems in this chapter seek to select a set of concurrent trans-

Table 3.1: Description of, reasons for, and means to circumvent communication constraints.

Constraint	Abbr.	Description
Half-duplex	HD	A node can not concurrently transmit and receive.
Single reception	SR	A node can not concurrently receive from multiple transmitters.
Unicast	UC	A node can not concurrently transmit distinct information to multiple receivers.
Interference	I	A node can not receive if there is any interfering transmission in its vicinity.

Constraint	Reason for constraint	To circumvent constraint
Half-duplex	Transceiver design	Full-duplex transceiver, multiple channels
Single reception	Receiver design	Multi-user detection
Unicast	Source/channel code design	Broadcast channel codes
Interference	Receiver design, lack of CSI, channel conditions	Interference cancellation, error correction, reduced data rates, spread spectrum

missions (edges in the communication graph) that maximize the weighted sum of the selected edges when subject to some subset of the communication constraints. Each possible subset of communication constraints is called a communication constraint set (CCS). We express each of these problems as an integer linear program (ILP). An ILP seeks to maximize a linear objective (say $\mathbf{w}^\mathsf{T}\mathbf{x}$) subject to linear constraints (say $\mathbf{A}\mathbf{x} \leq \mathbf{b}$) over decision variables $\mathbf{x} = (x_1, ..., x_M)$ taking integer values:

$$\max_{\mathbf{x} \in \mathbb{Z}_+^M} \{\mathbf{w}^\mathsf{T}\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\} \tag{3.1}$$

In our case each decision variable is either zero or one ($x_l \in \{0, 1\}$) indicating whether or not the edge is in the selected set.

Two key concepts we will employ for solving these optimization problems are matroids and totally unimodular matrices (TUMs). Matroids are a class of subset systems (used to model constraints) for which greedy algorithms yield optimal solutions. We will classify which CCSs are matroids. TUMs have the property that an ILP with a TUM constraint matrix may be solved as an LP by relaxing the integrality constraint (in our case relaxing each $x_i \in \{0, 1\}$ to each $x_i \in [0, 1]$).

We will also introduce two natural heuristics and study their performance for the various problems. The length heuristic adds edges to the solution set in order of increasing/decreasing edge lengths, while satisfying the constraint criteria. The degree heuristic orders the edges by head or tail degree. If two edges have the same degree we break ties on either edge weight or edge length.

Summary of findings. We characterize the Primal-Dual LP pairs of the problems. We demonstrate that only the UC, SR and the combined UC, SR problems are solvable via LP relaxation. The rest of the problems require ILP formulations which may be computationally intractable for large networks. We also show that the UC and SR problems are matroids and hence solvable via greedy algorithms. We show that at least one (and in some cases both) of our heuristics work very well for all primary constraint problems, however, performance suffers in almost all cases under the inclusion of the secondary I constraint.

The rest of this chapter is organized as follows. Section 3.2 discusses the various communication constraints, construction of the matrices for these constraints, and the formulation of the optimization problem. Related work is discussed in Section 3.3. Section 3.4 presents a brief summary of the results. Section 3.5 lists the Primal-Dual LP pairs for the problems and the impact of TUM on the LP formulations. Section 3.6 investigates which problems yield a matroid structure. Section 3.7 compares the heuristics to the optimal solution. Section 3.8 looks at the impact of the various constraints. Finally, Section 3.9 presents a conclusion and promising future directions.

3.2 Communication constraints and the optimization problem

3.2.1 Communication and Interference graph.

We assume that all nodes employ unit transmission power. The communication graph $G_c = (V_c, E_c)$ (with $|V_c| = N$ and $|E_c| = M$) is directed with an edge from i to j indicating that i is able to successfully transmit to j in the *absence of interference.* Specifically, η is the common noise factor at every node $v \in V_c$. Also, we define the symmetric $N \times N$ channel matrix $\boldsymbol{H}$ with entries $H_{uv} \in \mathbb{R}^+$ that denote the power attenuation factor of the channel from u to v at a particular instant of time. A directed edge (u, v) is added if the SNR from u to v (H_{uv}/η) exceeds the SNR constraint β_c. The interference graph $G_i = (V_i, E_i)$ (with $|V_i| = N$) is constructed in a similar manner with β_i specifying the interference threshold, *i.e.*, $E_i = \{(s,t) : H_{st}/\eta \geq \beta_i\}$. Both graphs G_i, G_c are directed, but edges come in pairs:

$$\begin{aligned} &(u, v) \in E_c(E_i) \text{ iff} \\ &(v, u) \in E_c(E_i) \end{aligned} \tag{3.2}$$

3.2.2 Transmission vector.

In this chapter we focus on unidirectional point to point communication. For the case of *point to point* transmissions, the transmission coordination problem requires identification of an optimal subset of *edges* from the communication graph, $E_S^* \subseteq E_c$. The feasible set is exponentially large in N, *i.e.*, $E_S^* \in \mathcal{P}(E_c)$, the power set of E_c, where $|\mathcal{P}(E_c)| = 2^M$, for $M = |E_c| = O(N^2)$ the size of G_c. The *transmission vector* for a particular point to point schedule, say E_S, is a $\{0, 1\}$-valued M-vector, $\mathbf{x} = (x_1, \ldots, x_M)$, with elements $x_e = 1$ if edge $e \in E_S$ and $x_e = 0$ if edge $e \notin E_S$. The transmission vector is the decision variable for the transmission coordination combinatorial optimization problem.

Transmission coordination schemes can be classified as *maximum* or *maximal*. A coordination scheme is *maximum* if its cardinality is at least as large as that of all feasible schedules. A coordination scheme is *maximal* if adding any additional transmissions violates one or more of the constraints. Finding the *maximum* coordination scheme generally requires searching over the entire state space and in some cases can be a NP-Hard problem. *Maximal* coordination schemes on the other hand, are easily found by greedy algorithms.

3.2.3 Performance objective.

The performance objective of interest is to maximize the weighted sum of the selected edges. The weight vector $\mathbf{w} = (w_1, \ldots, w_M)$ has elements $w_m \in \mathbb{R}^+$ denoting the relative value of activating each edge, and the transmission coordination objective is to maximize $\mathbf{w}^\mathsf{T}\mathbf{x}$. Maximizing the number of edges corresponds to $\mathbf{w} = \mathbf{1}$.

3.2.4 Communication constraints

Each of the communication constraints is illustrated in **Fig. 3.1**. The shaded (red) node circles identifies those nodes that violate the constraint(s) under consideration. Under the I constraint each node has an associated list of interfering neighbors. A node may not receive from a transmitter if any of its interfering neighbors are also transmitting. Specifically, define $\Gamma^{\text{in}}(v), \Gamma^{\text{out}}(v)$ as the set of incoming and outgoing neighbors of each $v \in V_i$. The interference constraint states a transmission attempt from u to v fails if any node $t \in \Gamma_i^{\text{in}}(v)$ is transmitting.

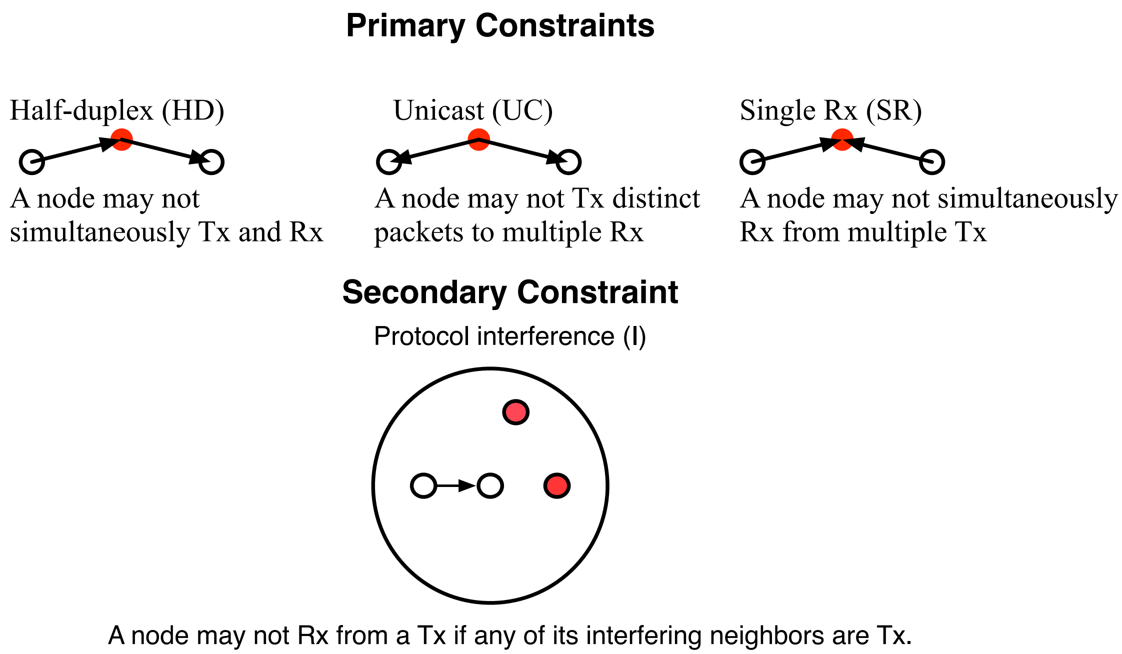


Figure 3.1: Illustration of the primary and secondary constraints. Violating nodes are denoted by red circles.

The constraints are presented in linear form ($\mathbf{Ax} \leq \mathbf{1}$), where $\mathbf{A}$ is $\{0, 1\}$-valued and has M columns. The rows of $\mathbf{A}$ indicate incompatible edges, *i.e.*, the ones in each row in $\mathbf{A}$ represent a set (or pair) of edges that cannot be simultaneously active as they together violate one or more of the governing constraints. The number of rows in $\mathbf{A}$ varies depending on the active constraints. Here $\mathbf{1} = (1, ..., 1)$ is a vector of ones with the same dimension as the number of rows of $\mathbf{A}$. This mathematical formulation of the communication constraints allow us to express the point to point transmission coordination problem as a family of integer linear programming (ILP) problems, shown below. The objective is to maximize $\mathbf{w}^\mathsf{T}\mathbf{x}$ over all possible transmission vectors $\mathbf{x} \in \{0, 1\}^M$, subject to any subset of the three primary constraints, and the protocol interference constraint.

$$\boxed{\begin{array}{rrcl} \text{maximize} & \multicolumn{3}{c}{\mathbf{w}^{\mathsf{T}}\mathbf{x}} \\ \text{over} & \multicolumn{3}{c}{\mathbf{x} \in \{0,1\}^M} \\ \text{subject to any subset of} & & & \\ \text{(primary constraints)} & \mathbf{A}^{\text{HD}}\mathbf{x} & \leq & \mathbf{1} \\ & \mathbf{A}^{\text{SR}}\mathbf{x} & \leq & \mathbf{1} \\ & \mathbf{A}^{\text{UC}}\mathbf{x} & \leq & \mathbf{1} \\ \text{(secondary constraint)} & \mathbf{A}^{\text{I}}\mathbf{x} & \leq & \mathbf{1} \end{array}} \tag{3.3}$$

Primary communication constraints. There are three possible primary communication constraints.

Half-duplex: Given the directed communication graph $G_c = (V_c, E_c)$, define $\mathcal{K}^{\text{HD}}$ to be the set of head-to-tail pairs of edges

$$\mathcal{K}^{\text{HD}} = \{\{(i,j),(j,k)\} : i,j,k \in V_c, (i,j),(j,k) \in E_c\}. \tag{3.4}$$

Let $K^{\text{HD}} = |\mathcal{K}^{\text{HD}}|$ be the number of such edge pairs. Form the $K^{\text{HD}} \times M$ matrix $\mathbf{A}^{\text{HD}}$ with elements:

$$A^{\text{HD}}_{ef,g} = \begin{cases} 1, & g = e \text{ or } f \\ 0, & \text{else} \end{cases}, \tag{3.5}$$

where each row in $\mathbf{A}^{\text{HD}}$ corresponds to a head-to-tail edge pair (e, f) in $\mathcal{K}^{\text{HD}}$. Note that each row of $\mathbf{A}^{\text{HD}}$ has exactly two 1's in the positions of the two edges comprising the pair. The linear constraint $\mathbf{A}^{\text{HD}}\mathbf{x} \leq \mathbf{1}$ prohibits any two edges that form a head-to-tail pair from being simultaneously active.

Single reception: a node can not concurrently receive from multiple transmitters. Define the $N \times M$ matrix $\mathbf{A}^{\text{SR}}$ (with each row representing a node), and an entry:

$$A^{\mathrm{SR}}_{j,e} = 1 \text{ if } e = (i,j) \text{ is an incoming edge of node } j. \tag{3.6}$$

The constraint is $\mathbf{A}^{\mathrm{SR}}\mathbf{x} \leq \mathbf{1}$, *i.e.,* at most one incoming edge for any node can be active at a time.

Unicast: a node can not concurrently transmit distinct information to multiple receivers. Define the $N \times M$ matrix $\mathbf{A}^{\mathrm{UC}}$ (with each row representing a node), and an entry:

$$A^{\mathrm{UC}}_{j,e} = 1 \text{ if } e = (i,j) \text{ is an outgoing edge of node } j. \tag{3.7}$$

The constraint is $\mathbf{A}^{\mathrm{UC}}\mathbf{x} \leq \mathbf{1}$, *i.e.,* at most one outgoing edge of each node can be active at a time.

Interference communication constraint.

Given the directed communication graph $G_c = (V_c, E_c)$ and the interference graph $G_i = (V_i, E_i)$, define $\mathcal{K}^{\mathrm{I}}$ to be the set of pairs of edges that interfere with one another:

$$\mathcal{K}^{\mathrm{I}} = \{\{(i,j),(k,l)\} : i,j,k,l \in V_c,\ (i,j),(k,l) \in E_c,\ (i,l) \in E_i \text{ or } (k,j) \in E_i\}. \tag{3.8}$$

Let $K^{\mathrm{I}} = |\mathcal{K}^{\mathrm{I}}|$ be the number of such edge pairs. Form the $K^{\mathrm{I}} \times M$ matrix $\mathbf{A}^{\mathrm{I}}$ with elements:

$$A^{\mathrm{I}}_{ef,g} = \begin{cases} 1, & g = e \text{ or } f \\ 0, & \text{else} \end{cases}, \tag{3.9}$$

where each row in $\mathbf{A}^{\mathrm{I}}$ corresponds to a pair of conflicting edges in $\mathcal{K}^{\mathrm{I}}$. Note that each row of $\mathbf{A}^{\mathrm{I}}$ has exactly two 1's in the positions of the two edges comprising the pair. The linear constraint $\mathbf{A}^{\mathrm{I}}\mathbf{x} \leq \mathbf{1}$ then prohibits any two edges that interfere with one another in G_i from being simultaneously active. Refer to **Fig. 3.2**, where edges $e, f \in E_c$ interfere if either $(i,l) \in E_i$ or $(k,j) \in E_i$.

Before proceeding we present a table representing some acronyms and notation

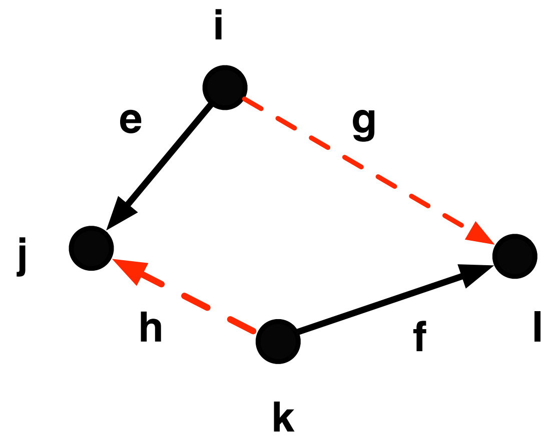


Figure 3.2: Illustration of the protocol interference (I) constraint.

Table 3.2: Acronyms and Notation

Acronyms	**Notation**
LP = Linear program	$G_c = (V_c, E_c)$: Communication graph
ILP = Integer Linear program	$G_i = (V_i, E_i)$: Interference graph
BLP = Binary Linear program	$\Gamma_O(v)$: Outgoing edges of vertex v
HD = Half-duplex	$\Gamma_I(v)$: Incoming edges of vertex v
SR = Single receiver	$h(e)$: head of edge e
UC = Unicast	$t(e)$: tail of edge e
I = Protocol interference	
TUM = Totally unimodular	
CCS = Communication constraint sets	

we will be using throughout the rest of this chapter.

3.3 Related Work

Distributed throughput optimal scheduling. In their seminal paper [17], Tassiulas and Ephremides use a multi-queue system to model a multi-hop wireless network with transmissions being specified via a collection of link activation sets. This paper introduced the concepts of backpressure routing and maximum weight matching and was the first to introduce Lyapunov drift to prove stability in a general multi-hop network. In [29] the authors extend techniques in [17] to a general multi-hop network with time varying channels to derive joint optimal routing and resource allocation algorithms to achieve stability, maximum throughput, and average end-to-end delay guarantees.

In [30] the authors show that maximum throughput algorithms based on max-weight principles yields constant-factor optimality results when the controller implements an algorithm that achieves only a constant factor of the max-weight rule every slot. In [18] the authors use a general interference model based on interference sets to show that greedy maximal scheduling achieves stability when the arrival rates are within a constant factor of the capacity region. Lin and Rasool [19] look at constant time scheduling policies that achieve at least a fraction of the optimal capacity region under the node-exclusive and two-hop interference models. The authors address this problem for both single-hop as well as multi-hop networks.

Recently [31] points out that the scheduling literature reveals no systematic study of the algorithmic and performance impacts of communication constraints. Performance impact refers to the effect of a CCS on the cardinality of the optimal transmission coordination scheme. In this paper the average cardinality of a *maximal* transmission coordination scheme as a function of the network size is studied.

Several of the optimization problems in Eqn. (3.3) can be related to classical problems in graph theory. We now highlight some of the related work for these problems. Additional related work can be found in [31].

Maximum weighted matching (MWM). A matching is a set of edges not sharing a common vertex. Optimization for bidirectional point to point communication subject to all three primary constraints (HD+SR+UC) is an instance of the distance-1 MWM problem. Edmonds [32] proposed an algorithm in 1965 to find a maximum matching (in both bipartite as well as non-bipartite graphs) in polynomial time.

The MWM problem and its variations have been fairly well studied from an algorithmic complexity viewpoint in the context of wireless scheduling. An extension of the MWM is the Maximum weighted K-Valid matching problem (MWKVMs) [33]. Under this problem no two links within K hops of one another can be transmitting simultaneously. The MWM is the MWKVM with $K = 1$. This imposes the restriction that no two active edges can share a common vertex. $K = 2$ implies the distance-2 MWM problem, *i.e.,* no two active edges may be within distance 2 of each other. This problem is commonly known as the induced matching problem. It has been shown that for any $K \geq 2$ the MKVMP problem is NP-Hard [34].

Degree constrained subgraph problem (DCSP). The DCSP is to find an optimal subgraph subject to constraints on the degree of each node in the subgraph. Optimization subject to either the single reception (in-degree at most one) or the unicast (out-degree at most one) constraint is a DCSP problem. DCSPs are solvable as MWMs. In fact we will show that the properties exhibited by the DCSP allow for its solution by greedy algorithms.

3.4 Finding the optimal schedule

The problem of finding the optimum schedule is a combinatorial optimization problem. Linear programming (LP) is a technique for optimization of a linear objective function, subject to linear constraints. An ILP formulation is a LP formulation with the additional requirement that the variables can only take integer values. LP problems can be solved efficiently in most cases. ILP problems on the other hand are, in general NP-Hard. Our problem is a $0-1$ ILP or Binary Integer Programming (BIP) problem. These problems too are classified as being NP-Hard.

All of the problems in Eqn. (3.3) are ILPs. Relaxing the ILP to an LP (replacing $\mathbf{x} \in \{0,1\}^M$ with $\mathbf{x} \in [0,1]^M$) would result in the vector $\mathbf{x}$ having (in general) fractional values, which we would round up or down accordingly to yield a feasible solution to the original ILP. The optimal objective value of this LP relaxation gives an *upper bound* on the optimal objective value of the original ILP (because the relaxation enlarges the search space). This LP relaxation is in general not tight, *i.e.,* we may well have non-integer solutions achieving a strictly higher objective value than any integral solution. In such a case the upper bound of the solution to the LP relaxation may be a poor estimate of the optimal solution to the original ILP. Relaxation is an appealing approach because LP problems can be solved efficiently in most cases.

Although ILP and BIP problems are classified as NP-Hard, there do exist an important subclass of problems that can be solved by LP relaxation. This arises as a result of the constraint matrix $\mathbf{A}$ and the right hand side $\mathbf{b}$ in $\mathbf{Ax} \leq \mathbf{b}$ satisfying certain properties. We now review some relevant terms and theorems.

3.4.1 Unimodularity

An integer matrix of full row rank $\mathbf{A}$ is said to be totally unimodular if every square submatrix of $\mathbf{A}$ has determinant either $0, +1$, or -1 [35]. The following theorem of Hoffman and Kruskal is key to our problem.

Theorem 1 (Hoffman-Kruskal [35]) *Let* $\mathbf{A}$ *be an* m *by* n *integral matrix. Then the polyhedron defined by* $\mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0$ *is integral for every integral vector* $b \in \mathbf{R}^m$ *iff* $\mathbf{A}$ *is TUM.*

For our optimization problems the linear constraints define a convex polyhedron called the feasible region. If a polyhedron is integral (has integer-valued vertices) the relaxation of the ILP is valid, *i.e.,* the LP and ILP have the same solution. This is because the solution of an LP occurs at a vertex and if all vertices are integral then the solution of the LP will be feasible for the ILP. The following is another important theorem we will require.

Theorem 2 (Heller and Tompkins [36]) *Let* $\mathbf{A}$ *be an* m *by* n *matrix whose rows can be partitioned into two disjoints sets* T_1 *and* T_2*, such that* $\mathbf{A}, \mathrm{T}_1$*, and* T_2 *have the following properties:*

- *every entry in* $\mathbf{A}$ *is* $0, +1,$ *or* -1*;*
- *every column contains at most two non-zero entries;*
- *if a column* $\mathbf{A}$ *contains two non-zero entries, and both have the same sign, then their rows are in different sets;*
- *if a column of* $\mathbf{A}$ *contains two non-zero entries, and they have opposite signs, their rows are in the same set.*

Then $\mathbf{A}$ *is TUM.*

3.4.2 Heuristics

We now introduce the various heuristics and provide the intuition behind their choice. We will then study their performance for the various problems.

General heuristic.

We consider a greedy heuristic with edges sorted by several primary and secondary criteria. The general algorithm for the heuristic is shown in Algorithm (3.10).

$$
\boxed{\begin{array}{l}
\text{List edges } E_c \text{ in order of PRIMARY CRITERION, breaking ties} \\
\text{using SECONDARY CRITERION (break remaining ties arbitrarily)} \\
\text{Set } \mathbf{x} = \mathbf{0}; \\
\text{While } E_c \neq \emptyset \\
\quad \text{Remove edge } e \text{ from queue} \\
\quad \text{If } \mathbf{A}(\mathbf{x} + \mathbf{e}_m) \leq \mathbf{b}) \quad \text{then} \quad \text{update} \quad \mathbf{x} := \mathbf{x} + \mathbf{e}_m
\end{array}} \tag{3.10}
$$

Here, $\mathbf{e}_m$ is the zero vector with a one in position m, and $\mathbf{Ax} \leq \mathbf{b}$ represents the set of governing constraints.

The representative primary and secondary sorting criterion we will consider include:

1. Primary = Length (Inc), Secondary = none
2. Primary = Length (Dec), Secondary = none
3. Primary = Degree (Inc), Secondary = Length (Inc)
4. Primary = Degree (Inc), Secondary = Length (Dec)
5. Primary = Degree (Dec), Secondary = Length (Inc)
6. Primary = Degree (Dec), Secondary = Length (Dec)

We will consider two special cases for edge weights: $w_e = 1$ and $w_e = l_{\max}/l_e$, where w_e is the weight of edge e, l_e is the length of edge e, and $l_{\max}$ is the length of the longest edge. As a result of this function, the length sorting criterion is equivalent to an edge weight criterion. Also, we point out that in and out degrees are the same for all nodes, so sorting by in- and out degree yields the same ordering. For every problem we consider all the sorting criteria listed above, choosing the ones that yield the best solution.

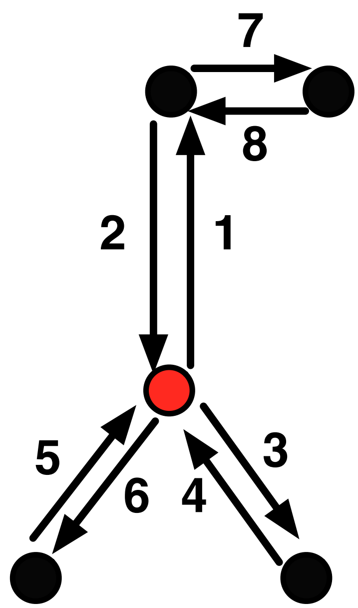


Figure 3.3: **Degree** heuristic vs. **length** heuristic under the HD+SR+UC problem.

We now provide some intuition behind the choice of the **degree** sorting criterion. As an example, consider **Fig. 3.3**. Operating under a **(dec) length** criteria would result in activation of either edge 1 or edge 2. Upon activation of either of these edges no other edge can be activated. If on the other hand we use **degree** as a primary sorting criteria, selecting edges on the basis of increasing node degree would result in activation of two edges, one from the set $\{3, 4, 5, 6\}$, and another from the set $\{7, 8\}$. Hence we see that in certain cases the **degree** heuristic will yield a better solution than the **length** heuristic.

Table 3.3 presents results for the case where the objective is to maximize the schedule cardinality ($w_e = 1$). Table 3.4 lists the results for the case where the objective is to maximize the sum of the active edge weights under the various constraints ($w_e = l_{\max}/l_e$). The first column represents the set of constraints under consideration, the second column specifies whether or not the constraint matrix for the problem is TUM, the third column specifies whether the maximal independent set is also the maximum (optimal) independent set, *i.e.,* whether the problem structure is a matroid. Column 4 and 6 specify the performance when the primary sorting criterion is **edge length**, whereas columns 5 and 7 specify the performance when the primary sorting criterion is **node degree**. When the primary sorting criterion is degree, we specify the secondary criterion that provides the best performance for each problem. Performance is expressed as a percentage of the optimal solution obtained at $N = 100$ (or the highest value of N for which we have an optimal solution).

From Table 3.3 we observe that for the $\beta_c = 0.2$ case (without I), at least one of the heuristics yields a very good approximation of the optimal solution (with the exception of the HD case the heuristics yield an approximation that is 97% or better). For the $\beta_c = 12.0$ case as well, we observe that at least one (and in some cases both) heuristics yield a very good approximation of the optimal solution. Observe also that for the two matroid problems, both heuristics yield the optimal solution (in fact any sorting criterion always yields the optimal solution for these problems). Upon adding the I constraint we observe that the heuristics do not perform as well. Performance of the heuristics for problems that involve the UC constraint improves (albeit marginally) when we go from $\beta_c = 0.2$ to $\beta_C = 12.0$, whereas the approximation for the problems that do not include this constraint worsens. We are unable to explain precisely the reason for this observation. Our intuition is that the addition of the I constraint makes the problem of finding the optimal schedule a hard one as we are trying to maximize the number of transmissions under interference. A greedy heuristic is too simplistic and ignores this constraint, resulting in a poor approximation of the optimal solution. We also observe that in the absence of the I constraint, when utilizing a **length** sorting criterion, sorting by increasing length performs better than sorting by decreasing length. This also holds for the case where the I constraint is added to the primary constraint problems and $\beta_i = 12.0$. However when $\beta_c = 0.2$, we observe that in a majority of cases sorting by decreasing length performs better than sorting by increasing length. Once again we are not precisely sure as to why this occurs.

From Table 3.4 we observe that under a primary sorting criterion that sorts by **length**, sorting by increasing length performs better for all cases. Also when utilizing the **degree** for the primary sorting criterion, we observe that in all cases sorting by increasing degree yields a better solution in comparison to sorting by decreasing degree. Further sorting by increasing length (secondary criterion) yields the best performance. Sorting by increasing degree decreases the possibility of low degree nodes being left out, whereas an increasing length criterion is natural for both cases, since edge weights are a function of edge length (with shorter edges having larger edge weights than longer ones). For the two matroid problems we observe that sorting by increasing lengths (either primary or secondary sorting criteria) yields the optimal solution, thereby showing that a greedy algorithm always yields the optimal solution. Another observation is that in all cases selecting **length** as opposed to **degree** as a primary sorting criterion yields better performance for all problems. Also we observe that for all problems, the heuristics perform better when $\beta_c = 12.0$ as opposed to $\beta_c = 0.2$. This isn't always true for Table 3.3.

Table 3.3: Performance (% of optimal solution for network of $N = 100$ nodes) when objective is to maximize $\mathbf{1}^{\mathsf{T}}\mathbf{x}$.

Problem	TUM	Matroid	$\beta_c = 0.2$	$\beta_c = 0.2$	$\beta_c = 12.0$	$\beta_c = 12.0$
Sorting by			**length**	**degree**	**length**	**degree**
HD	No	No	**86**[1,3]	32[1,4,5]	**83**[3]	82[4]
SR	Yes	Yes	**100**[5]	**100**[5]	**100**[5]	**100**[5]
UC	Yes	Yes	**100**[5]	**100**[5]	**100**[5]	**100**[5]
HD+SR	No	No	68[3]	**97**[4,5]	84[3]	**95**[4]
HD+UC	No	No	68[3]	**97**[4,5]	84[3]	**95**[4]
SR+UC	Yes	No	97[3]	**97**[4,3]	89[3]	**93**[3]
HD+SR+UC	No	No	97[3]	**98**[3,5]	92[3]	**96**[3]
I	No	No	48[2,4]	**50**[2,4]	40[3]	**42**[3]
HD+I	No	No	**75**[2,4]	51[2,4]	35[3]	**39**[3]
SR+I	No	No	52[2,5]	**75**[2,4]	56[3]	**59**[3]
UC+I	No	No	61[2,5]	**88**[2,4]	87[3]	**90**[3]
HD+SR+I	No	No	52[2,4]	**76**[2,4]	35[3]	**39**[3]
HD+UC+I	No	No	60[2,4]	**87**[2,4]	81[3]	**90**[3]
SR+UC+I	No	No	**88**[2,3]	81[2,4]	87[3]	**90**[3]
HD+SR+UC +I	No	No	74[2,4]	**78**[2,4]	81[3]	**90**[3]

[1] Performance for $N = 60$.

[2] Performance for $N = 70$.

[3] Sorting by increasing order for primary (and secondary criteria).

[4] Sorting by decreasing order for primary (and secondary criteria).

[5] No difference between increasing and decreasing sorting for criteria under consideration.

Table 3.4: Performance (% of optimal solution for network of $N = 100$ nodes) when objective is to maximize $\mathbf{w}^\mathsf{T}\mathbf{x}$.

Problem	TUM	Matroid	$\beta_c = 0.2$	$\beta_c = 0.2$	$\beta_c = 12.0$	$\beta_c = 12.0$
Sorting by			**length**	**degree**	**length**	**degree**
HD	No	No	**96[1,2]**	49[1,2]	**93[2]**	75[2]
SR	Yes	Yes	**100[2]**	**100[2]**	**100[2]**	**100[2]**
UC	Yes	Yes	**100[2]**	**100[2]**	**100[2]**	**100[2]**
HD+SR	No	No	**94[2]**	91[2]	**95[2]**	94[2]
HD+UC	No	No	**94[2]**	91[2]	**95[2]**	94[2]
SR+UC	Yes	No	**98[2]**	85[2]	**98[2]**	89[2]
HD+SR+UC	No	No	**99[2]**	80[2]	**98[2]**	88[2]
I	No	No	**39[1,2]**	34[1,2]	**93[2]**	59[2]
HD+I	No	No	**61[1,2]**	40[1,2]	**62[2]**	42[2]
SR+I	No	No	**48[2]**	46[2]	**93[2]**	59[2]
UC+I	No	No	**55[2]**	53[2]	**98[2]**	62[2]
HD+SR+I	No	No	**54[2]**	31[2]	**62[2]**	42[2]
HD+UC+I	No	No	**93[2]**	54[2]	**96[2]**	65[2]
SR+UC+I	No	No	**54[2]**	53[2]	**98[2]**	62[2]
HD+SR+UC +I	No	No	**93[2]**	52[2]	**96[2]**	65[2]

[1] Performance for $N = 60$.

[2] Sorting by increasing order for primary criteria (and secondary criteria when applicable).

3.5 LP vs. ILP and TUM

3.5.1 LP formulations for all problems

We now proceed to characterize the LP formulation for the various problems.

1. HD constraint

We can formulate the LP and its dual for the HD problem as follows:

HD constraint

Primal	**Dual**
Max: $\sum_{e \in E_c} w_e x_e$	**Min**: $\sum_{k \in \mathcal{K}^{\text{HD}}} y_k$
s.t. $x_e + x_{e'} \leq 1, \{e, e'\} \in \mathcal{K}^{\text{HD}}$.	s.t $\sum_{k: e \in \mathcal{K}^{\text{HD}}} y_k \geq w_e,\ e \in E_c$.
$x_e \in [0, 1], e \in E_c$.	$y_k \geq 0,\ k \in \mathcal{K}^{\text{HD}}$.

(3.11)

Recall $\mathcal{K}^{\text{HD}}$ is defined in Eqn. (3.4). Our problem is to activate the set of links that maximize the sum of the edge weights while ensuring that the half-duplex condition is not violated. See figure on left in **Fig. 3.4**. Either we can activate a subset of $\Gamma_I(v)$ or a subset of $\Gamma_O(v)$, for each $v \in V$.

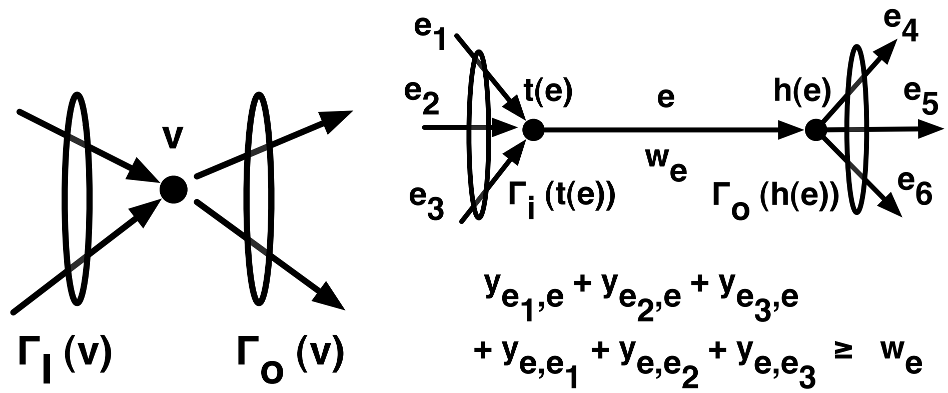


Figure 3.4: Illustration of the primal and dual constraints for the HD problem.

The dual for this problem requires that we minimize the sum of the dual variables $\{y_k\}$ subject to the requirement that for each edge, the sum of the duals pertaining to conflicting edge pairs involving the edge exceeds the weight of the edge. See figure on right in **Fig. 3.4**.

2. SR constraint

The LP and its associated dual for the SR problem is:

SR constraint	
Primal	**Dual**
Max: $\sum_{e \in E_c} w_e x_e$	**Min**: $\sum_{v \in V_c} y_v$
s.t. $\sum_{e \in \Gamma_I(v)} x_e \leq 1, v \in V_c$.	s.t. $y_v \geq \max w_e, e \in \Gamma_I(v), v \in V_c$.
$x_e \in [0, 1], e \in E_c$.	$y_v \geq 0,\ v \in V_c$.

(3.12)

The SR constraint states that at most one incoming edge for each node can be active. The dual problem seeks to minimize the sum of the dual variables such that the dual variable associated with each vertex exceeds the weight of each incoming edge.

3. UC constraint

The LP and associated dual for the UC problem is:

UC constraint

Primal	**Dual**
Max: $\sum_{e\in E_c} w_e x_e$	**Min**: $\sum_{v\in V_c} y_v$
s.t. $\sum_{e\in\Gamma_O(v)} x_e \leq 1, v \in V_c.$	s.t. $y_v \geq \max\, w_e, e \in \Gamma_O(v), \forall v \in V_c.$
$x_e \in [0,1], e \in E_c.$	$y_v \geq 0,\ \forall\ v \in V_c.$

(3.13)

The UC constraint states that at most one outgoing edge for each node can be active. The dual problem seeks to minimize the sum of the dual variables such that the dual variable associated with each vertex exceeds the weight of each outgoing edge.

4. I constraint

The Primal-Dual pair for the I constraint is:

I constraint

Primal	**Dual**
Max: $\sum_{e\in E_c} w_e x_e$	**Min**: $\sum_{k\in\mathcal{K}^{\text{I}}} y_k$
s.t. $x_e + x_{e'} \leq 1, \{e, e'\} \in \mathcal{K}^{\text{I}}.$	s.t $\sum_{k:e\in\mathcal{K}^{\text{I}}} y_k \geq w_e,\ e \in E_c.$
$x_e \in [0,1], e \in E_c.$	$y_k \geq 0,\ k \in \mathcal{K}^{\text{I}}.$

(3.14)

Recall $\mathcal{K}^{\text{I}}$ is defined in Eqn. (3.8). The number of conflicting pairs depends on β_i: a small β_i results in a large number of conflicting edges, whereas a large β_i results a smaller number of conflicting pairs.

The dual for this problem seeks to minimize the sum of the dual variables $\{y_k\}$ subject to the requirement that for each edge the sum of the dual variables associated with each pair of conflicting edges involving the edge exceeds the edge weight.

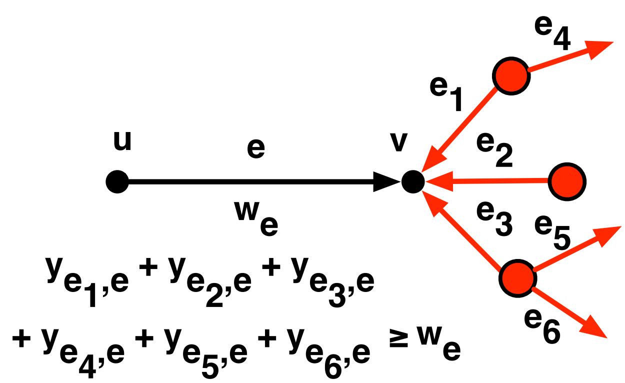


Figure 3.5: Illustration of the dual constraints for the I constraint.

5. Combined constraints

The Primal-Dual pair for the case when all primary constraints are combined with the secondary constraint is:

I, HD, SR, UC constraints

Primal

$$\begin{aligned}
&\textbf{Max}: \textstyle\sum_{e\in E_c} w_e x_e \\
&\text{s.t. } \textbf{I -}\ x_e + x_{e'} \leq 1, \{e, e'\} \in \mathcal{K}^{\mathrm{I}}, \\
&\quad \textbf{HD -}\ x_f + x_{f'} \leq 1, \{f, f'\} \in \mathcal{K}^{\mathrm{HD}}, \\
&\quad \textbf{SR -}\ \textstyle\sum_{e\in\Gamma_I(v)} x_e \leq 1, v \in V_c, \\
&\quad \textbf{UC -}\ \textstyle\sum_{e\in\Gamma_O(v)} x_e \leq 1, v \in V_c. \\
&\quad x_e \in [0, 1], e \in E_c.
\end{aligned} \tag{3.15}$$

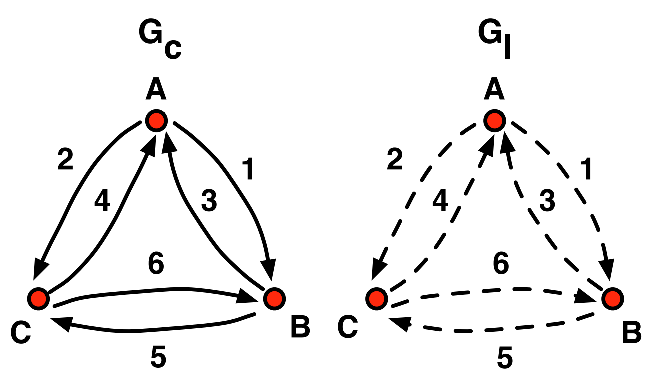
I, HD, SR, UC constraints

Dual

$$
\begin{aligned}
&\textbf{Min}: \textstyle\sum_{p\in K^{\mathrm{I}}} y_p + \sum_{q\in K^{\mathrm{HD}}} y_q + \sum_{r\in V_c} y_r + \sum_{s\in V_c} y_s \\
&\text{s.t } \textbf{I -} \textstyle\sum_{p:e\in\mathcal{K}^{\mathrm{I}}} y_p \geq w_e,\ e\in E_c, \\
&\quad \textbf{HD-} \textstyle\sum_{q:e\in\mathcal{K}^{\mathrm{HD}}} y_q \geq w_e,\ e\in E_c, \\
&\quad \textbf{SR -}\ y_r \geq \max\ w_e, e\in\Gamma_I(r), \forall r\in V_c, \\
&\quad \textbf{UC -}\ y_s \geq \max\ w_e, e\in\Gamma_O(s), \forall s\in V_c. \\
&\quad y_p\geq 0,\ p\in K^{\mathrm{I}},\ y_q\geq 0,\ q\in K^{\mathrm{HD}}, y_r, y_s\geq 0,\ r,s\in V_c.
\end{aligned}
\tag{3.16}
$$

I, HD, UC, SR constraints

Primal	**Dual**
Max: $\sum_{e=1}^{6} w_e x_e$	**Min**: $\sum_{k=1}^{24} y_k$
I $x_2 + x_6 \leq 1,\ x_2 + x_3 \leq 1$	$y_5 + y_6 + y_8 + y_{10} + y_{11} + y_{17} + y_{19} + y_{23} \geq w_1$
I $x_3 + x_6 \leq 1,\ x_4 + x_5 \leq 1$	$y_1 + y_2 + y_9 + y_{12} + y_{13} + y_{14} + y_{19} + y_{24} \geq w_2$
I $x_1 + x_4 \leq 1,\ x_1 + x_6 \leq 1$	$y_2 + y_3 + y_7 + y_{10} + y_{14} + y_{15} + y_{20} + y_{22} \geq w_3$
I $x_3 + x_4 \leq 1,\ x_1 + x_5 \leq 1$	$y_4 + y_5 + y_7 + y_{11} + y_{12} + y_{16} + y_{21} + y_{22} \geq w_4$
I $x_2 + x_5 \leq 1$	$y_4 + y_8 + y_9 + y_{16} + y_{17} + y_{18} + y_{20} + y_{24} \geq w_5$
HD $x_1 + x_3 \leq 1,\ x_1 + x_4 \leq 1$	$y_1 + y_3 + y_6 + y_{13} + y_{15} + y_{18} + y_{21} + y_{23} \geq w_6$
HD $x_1 + x_5 \leq 1,\ x_2 + x_3 \leq 1$	$y_k \geq 0,\ k = 1..24$
HD $x_2 + x_4 \leq 1,\ x_2 + x_6 \leq 1$	
HD $x_3 + x_6 \leq 1,\ x_4 + x_5 \leq 1$	
HD $x_5 + x_6 \leq 1,$	
UC $x_1 + x_2 \leq 1,\ x_3 + x_5 \leq 1$	
UC $x_4 + x_6 \leq 1$	
SR $x_3 + x_4 \leq 1,\ x_1 + x_6 \leq 1$	
SR $x_2 + x_5 \leq 1$	



$x_e \in [0,1], e = 1..6$

(3.17)

Eqn. (3.17) shows the primal-dual formulation for a three node graph with G_c and G_i as shown. For the I and HD constraints each inequality represents a dual variable. Here $y_1 - y_9$ are the dual variables for the I constraint, $y_{10} - y_{18}$ are the dual variables for the HD constraint, y_{19}, y_{20}, y_{21} are the dual variables associated with the UC constraint, and finally y_{22}, y_{23}, y_{24} are the dual variables associated with the SR constraint.

We now proceed to test which of the communication constraint matrices satisfy the TUM criterion, and hence will always yield integral solutions for the corresponding LP relaxation of the original ILP formulations.

3.5.2 TUM of constraint matrices

1. HD constraint. We start by looking at the structure of the half duplex constraint matrix $\mathbf{A}^{\mathrm{HD}}$ defined in Eqn. (3.4). Each row specifies a pair of conflicting edges that cannot be simultaneously active. Also since each row explicitly specifies *a pair* of conflicting edges, some rows are not *independent* of other rows. Consider a network with G_c as shown in **Fig. 3.6**. The constraint matrix ($\mathbf{A}^{\mathrm{HD}}$) for this graph is:

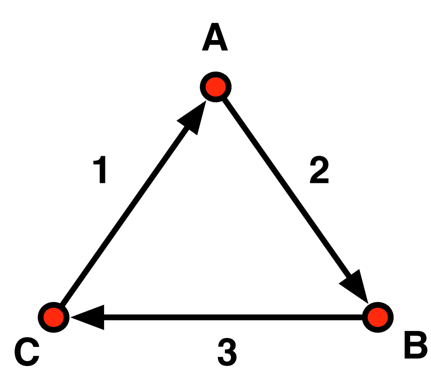


Figure 3.6: Sample communication graph G_c for a three node network.

$$\mathbf{A}^{\mathrm{HD}} = \begin{array}{c} \\ 1,2 \\ 1,3 \\ 2,3 \end{array} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

The determinant is $|\mathbf{A}^{\mathrm{HD}}| = -2 \notin \{-1, 0, +1\}$. As a result the matrix is not TUM, and so the *Hoffman-Kruskal* theorem (**Theorem 1**) does not hold, meaning LP relaxation does not generate a feasible solution to the ILP.

2. SR constraint. We will prove $\mathbf{A}^{\mathrm{SR}}$ is TUM. The matrix $\mathbf{A}^{\mathrm{SR}}$ is defined in Eqn. (3.6). Each column corresponds to an edge, and hence each column has exactly one 1 in row v = h(e). The matrix therefore is TUM by **Theorem 2** (Heller and Tompkins).

3. UC constraint. The same argument holds here.

4. HD+SR/UC constraints. Combining both constraints corresponds to stacking the rows of $\mathbf{A}^{\mathrm{HD}}$ and $\mathbf{A}^{\mathrm{SR}}$. It is obvious that stacking a non-TUM matrix with a TUM matrix yields a non-TUM matrix.

5. SR+UC constraints. For this constraint we stack the rows of the $\mathbf{A}^{\mathrm{SR}}$ and $\mathbf{A}^{\mathrm{UC}}$ matrices. First we modify the constraint so that incoming edges for a node are represented by a -1 instead of 1. The resulting $2N \times M$ matrix is the $\mathbf{A}^{\mathrm{SR+UC}}$ constraint matrix. We see that every entry is $+1$ or -1. Every column contains exactly 2 non-zero entries of opposite sign. The same set condition is trivially satisfied by choosing $T_1 = [2N]$ and $T_2 = \emptyset$. Hence **Theorem 2** (Heller and Tompkins) holds and $\mathbf{A}^{\mathrm{SR+UC}}$ is TUM. Consequently the polygon defined by the LP problem for this constraint set is integral for every integral vector $\mathbf{b}$.

6. HD+SR+UC constraints. See discussion for HD+SR. As discussed in Section 3.3 the condition when all three primary constraints are in effect is equivalent to the unweighted version of the maximum weighted matching (MWM) problem. The only difference in our case is that instead of an undirected edge between two nodes, two communicating nodes have a pair of bidirectional edges. Edmonds [32] proposed an algorithm in 1965 to find a maximum matching (in both bipartite as well as non-bipartite graphs) in polynomial time. Hence, even though the ILP formulation of this problem is not easy to solve, there do exist polynomial time algorithms for the maximum weighted and unweighted matching problems.

7. I constraint. The I constraint forbids interfering transmissions. The structure of the I constraint matrix is defined in Eqn. (3.8). Each row in the I constraint matrix $\mathbf{A}^{\mathrm{I}}$ represents a pair of conflicting edges under the constraint. Consider the 3 node graph in **Fig. 3.7** with G_c and G_i as depicted. We have the following matrix:

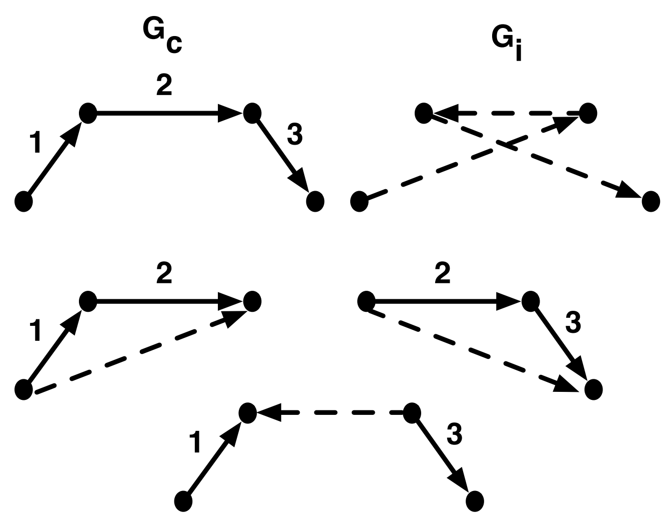


Figure 3.7: G_c and G_i for a four node network and the corresponding interfering edge pairs. Interference precludes any pair of edges from concurrent transmission.

$$\mathbf{A}^{\mathrm{I}} = \begin{array}{c} \\ 1,2 \\ 1,3 \\ 2,3 \end{array} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

As in **Fig. 3.6**, the determinant of this constraint matrix is $|\mathbf{A}^{\mathrm{I}}| = -2 \notin \{-1, 0, +1\}$. As a result the *Hoffman-Kruskal* theorem (**Theorem 1**) does not hold.

3.5.3 Results of LP formulation

We now present some results for the ILP formulation and its LP relaxation. Our results focus on networks ranging from $N = 10$ to $N = 100$ nodes (using steps of $\Delta N = 10$). A data point for any constraint(set) for a given network size N is computed by taking the average of five independent placements of N nodes in the arena, each node placed uniformly at random. All results employ a square arena $\mathcal{A}$ of 100×100 square meters. The communication graph $G_c = (V_c, E_c)$ is constructed using an SNR requirement of $\beta_c = 0.2$ and $\beta_c = 12.0$. The small value of β_c facilitates a larger number of edges, thereby allowing us to illustrate the impact of the constraints on the graph. Our objective function for the various problems (see Eqn. (3.3)) is $\sum w_e x_e$ where w_e is the weight associated with edge x_e. Setting $\mathbf{w} = \mathbf{1}$ results in the objective simply being the cardinality of the schedule (the number of active edges under a given constraint set). We solve the formulations using CPLEX [37], allowing a running time of $T = 1000$ seconds. For Tables 3.5, 3.6, 3.7, $|M|$ denotes the number of edges in the communication graph ($|E_c|$), averaged over the five independent placements of each N. For the LP relaxation we use the following rounding rule:

$$\hat{X}_e = \begin{cases} 0, & X_e < 1/2 \\ 1, & \text{else.} \end{cases} \tag{3.18}$$

It is possible that this particular rounding rule may not yield a good approximation. Instead a rounding rule that is more conservative in rounding up the fractional assignments may provide a better solution.

Our intention here is to highlight that an LP relaxation of the original ILP fails to yield the correct solution for constraint matrices that are not TUM, *i.e.,* the LP relaxation is not tight, and that solving the ILP formulation is much more computationally intensive in comparison to solving the LP formulation. We also show the effect of adding the I constraint to the primary constraint problems.

Table 3.5 shows results of the LP relaxation of the ILP for all possible formulations of the primary constraint problems. From the results we observe the following:

- Under the HD+SR/UC problems the cardinality of a schedule can never be N, as this implies that all nodes are receiving from some node under the HD+SR problem (in the case of HD+UC a cardinality of N means all nodes are transmitting), which in turn implies that some nodes are simultaneously transmitting and receiving, which violates the HD constraint. For the HD+SR+UC problem the cardinality of a schedule can never exceed $N/2$. As a result we see that the LP relaxation is not tight for the (*i*) HD+SR, (*ii*) HD+UC, (*iii*) HD+SR+UC problems. This result is in agreement with our earlier assessment that the HD constraint matrix is not TUM.

- The LP relaxation is tight for the (*i*) SR, (*ii*) UC, and (*iii*) SR+UC problems. This is in agreement with our earlier assessment that these three problems yield TUM matrices.

Table 3.5: Average number of edges vs. network size for all 8 combinations of primary constraints (LP relaxation using rounding rule in Eqn. (3.18)) for $\beta_c = 0.2$.

N	$\|M\|$	HD	SR	UC	HD, SR	HD, UC	SR, UC	HD, SR, UC
10	18.8	9.4	9.0	9.0	7.85	7.85	8.8	4.0
20	71.6	32.5	19.2	19.2	18.62	18.62	19.0	18.5
30	182.0	91.0	29.6	29.6	29.3	29.3	29.6	28.9
40	298.4	148.5	39.8	39.8	39.7	39.7	39.8	39.5
50	450.4	225.2	50.0	50.0	50.0	50.0	50.0	50.0
60	654.8	327.4	60.0	60.0	60.0	60.0	60.0	60.0
70	912.4	456.2	70.0	70.0	70.0	70.0	70.0	70.0
80	1180.8	590.4	80.0	80.0	80.0	80.0	80.0	80.0
90	1557.2	778.6	90.0	90.0	90.0	90.0	90.0	90.0
100	1908.8	954.4	100.0	100.0	100.0	100.0	100	100.0

Table 3.6: Average number of edges vs. network size for all 8 combinations of primary constraints (ILP) for $\beta_c = 0.2$.

N	$\|M\|$	HD	SR	UC	HD,SR	HD, UC	SR, UC	HD, SR, UC
10	18.8	7.4	9.0	9.0	6.0	6.0	8.8	4.0
20	71.6	30.6	19.2	19.2	16.4	16.4	19.0	10.0
30	182.0	61.6	29.6	29.6	24.6	24.6	29.6	14.4
40	298.4	99.0	39.8	39.8	34.4	34.4	39.8	19.8
50	450.4	147.0	50	50.0	44.0	44.0	50.0	24.8
60	654.8	207.0	60	60.0	54.0	54.0	60.0	30.0
70	912.4	279.5	70.0	70.0	64.2	64.2	70.0	35.0
80	1180.8	357.4	80.0	80.0	74.2	74.2	80.0	40.0
90	1557.2	453.6	90.0	90.0	84.2	84.2	90.0	45.0
100	1908.8	537.8	100.0	100.0	94.0	94.0	100.0	50.0

Table 3.7: Average number of edges vs. network size for all 8 combinations of primary constraints with the interference constraint for $\beta_c = 0.2$ and $\beta_i = 4.0$

N	$\|M\|$	I	HD, I	SR, I	UC, I	HD,SR, I	HD,UC, I	SR, UC, I	HD, SR, UC, I
10	18.8	13.4	6.8	9.0	8.2	6.0	5.6	7.8	4.2
20	71.6	36.8	22.2	19.0	16.4	14.6	13.4	15.2	8.4
30	182.0	64.8	48.0	29.6	25.8	24.6	21.4	22.2	14.2
40	298.4	100.2	80.0	39.8	35.2	34.6	30.4	28.0	19.0
50	450.4	133.4	110.4	50.0	43.0	44.0	38.4	33.0	23.6
60	654.8	170.8	148.5	60.0	51.6	54.0	46.6	36.2	28.6
70	912.4	219.5	192.2	70.0	59.6	64.2	55.6	41.6	32.8

Table 3.6 show the results of the ILP formulation for all possible combinations of the primary constraint problems for $\beta_c = 0.2$.

Table 3.7 shows the results of the ILP formulation for all possible combinations of primary constraints with the I constraint. For the I constraint we use $\beta_i = 4.0$, with $\beta_c = 0.2$ as before. Results in this table are limited to $N = 10 - 70$ as the running time of $T = 1000$ seconds is insufficient to find the optimal solution for some problem instances (discussed in Table 3.9). One key observation for this table is the asymmetry of the solutions for the SR and UC problems in combination with the I and HD problems. In the absence of the I constraint the results were symmetrical (Table 3.6). The UC constraint limits the number of transmissions per transmitter, whereas the I constraint limits the number of transmitters. The combination of the UC and I constraints have a greater effect on the solution than the SR+I problem.

3.5.4 Running times

Table 3.8 lists the running times for Table 3.6. We observe that the (*i*) SR, (*ii*) UC, and (*iii*) SR+UC ILP formulations can be solved almost instantaneously. The running time required to solve those problems that require ILP formulations, *i.e.,* those involving the HD constraint can however, be problematic. The increase in running time with increasing network size is clearly visible for the HD problem. The reason for the dramatic difference in running times for the HD problem is due to the lack of any limit on the degree for the nodes under this constraint. For all the other problems there is a limit on either the in-degree (SR), out-degree (UC) or both (in the case of the SR+UC and HD+SR+UC problems). This limit drastically cuts down the solution search space. In the case of the HD problem we need to find the node assignment where the assignment requires deciding whether the node is transmitting or receiving and in each case how many nodes it is transmitting to or receiving from.

This requires searching over a very large search space.

Since the I constraint does not satisfy TUM, addition of this constraint to the primary constraint problems results in these problems requiring ILP formulations. As a result the running time for all these problems becomes an issue as network size N increases. This is plainly visible from the Table 3.9, where even problems involving just the SR and UC constraints (which had running times < 10 seconds) exhibit a significant increase in running time with the addition of the I constraint.

3.6 Matroids and communication constraints

We now explore the use of greedy algorithms in solving the transmission coordination problem. Greedy algorithms yield optimal solutions when the subset system representing the governing communications constraints is a matroid. We now proceed to define matroids and then evaluate which of the various CCSs are matroids.

3.6.1 Matroids and greedy algorithms

Definition: Matroid [38]

A *matroid* is an ordered pair $(\mathcal{E}, \mathcal{I})$ consisting of a finite set $\mathcal{E}$ together with a collection $\mathcal{I}$ of subsets of $\mathcal{E}$ satisfying the following three conditions:

1. $\emptyset \in \mathcal{I}$

2. If $I' \subseteq I \in \mathcal{I}$, then $I' \in \mathcal{I}$

3. If I_1 and I_2 are in $\mathcal{I}$ and $|I_1| < |I_2|$, then there is an element $e \in I_2 \setminus I_1$ such that $I_1 \cup \{e\} \in \mathcal{I}$

If $\mathcal{M}$ is the matroid $(\mathcal{E}, \mathcal{I})$, then $\mathcal{M}$ is called a matroid on $\mathcal{E}$. The members of $\mathcal{I}$ are called the *independent* sets of $\mathcal{M}$, and $\mathcal{E}$ is the *ground* set of $\mathcal{M}$. A subset of $\mathcal{E}$ not in $\mathcal{I}$ is called a *dependent* set. A minimal dependent subset $\mathcal{C}$ of $\mathcal{E}$ is called a circuit. A maximal independent set of $\mathcal{M}$ is called a *base* or a *basis* of $\mathcal{M}$. All bases of a matroid have the same cardinality. A maximal independent subset is the largest subset of $\mathcal{E}$ in $\mathcal{I}$. A minimal dependent subset is the smallest subset $E \subseteq \mathcal{E}$ s.t. $\mathcal{E} \notin \mathcal{I}$. We now look at some examples of matroids.

- Let $\mathcal{F}$ be the set of forests of the graph $G = (V, E)$. Then $\mathcal{M}_G = \{\mathcal{E}, \mathcal{F}\}$ is a *graphic* matroid [39].

- A *matric* matroid [39] $\mathcal{M}_A = \{\mathcal{E}, \mathcal{J}\}$, is the set of linearly independent subsets of $\mathcal{E}$.

- Let $\mathcal{E}$ be a finite set and Π be a partition (a collection of disjoint subsets) of $\mathcal{E}$; $\Pi = \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, ..., \mathcal{E}_p\}$. A subset $\mathcal{I}$ of $\mathcal{E}$ is independent iff no two elements of $\mathcal{I}$ are in the same set of Π, *i.e.,* $|\mathcal{I} \cap \mathcal{E}_j| \leq 1, j = 1, 2, ..., p$. The system $\mathcal{M}_\Pi = \{\mathcal{E}, \mathcal{J}\}$ is a matroid called a *partition matroid* [39]. Later in this section, we will encounter partition matroids.

Matroid Intersection [40]. Given two matroids $\mathcal{M}_1, \mathcal{M}_2$ on the same ground set $\mathcal{E}$, the matroid intersection problem is to find a common independent set of the largest size: $\max\{|I| : I \in \mathcal{I}_1, I \in \mathcal{I}_2\}$. As an example consider the optimal matching problem for a bipartite graph, $G = (V, E)$, where $\{V_1, V_2\}$ is a bipartition of $G = \{V, E\}$, and $\mathcal{I}_i = \{J \subset \mathcal{E}$: each $v \in V_i$ is incident with at most one element of $J\}$. Both $\mathcal{M}_1 = (\mathcal{E}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{E}, \mathcal{I}_2)$ are partition matroids. Effectively the common independent sets are the matchings of G. As a result, the problem of finding a maximum matching for a bipartite graph is the same as the matroid intersection problem where both the matroids are partition matroids.

Matroids are related to the greedy algorithm. Under the greedy algorithm, in each step, we choose any largest weight member of $\mathcal{E}$, not already chosen, which together with the members already chosen forms a subset system while subject to some condition that maintains the property of the set $\mathcal{I}$. For any matroid $\mathcal{M}$ on $\mathcal{E}$,

Table 3.8: Time required to solve the ILP formulation for problems involving only primary constraints. All times are in seconds.

N	HD	SR	UC	HD,SR	HD, UC	SR, UC	HD, SR, UC
10	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
20	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
30	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
40	4.2	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
50	44.2	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
60	465.7	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
70	> 1000.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
80	> 1000.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
90	> 1000.0	< 1.0	< 1.0	1.5	1.5	< 1.0	< 1.0
100	> 1000.0	< 1.0	< 1.0	3.3	3.3	< 1.0	< 1.0

Table 3.9: Time required to solve the ILP formulation for problems involving the interference constraint. All times are in seconds.

N	I	HD, I	SR, I	UC, I	HD,SR, I	SR, UC, I	HD, SR, UC, I
10	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
20	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
30	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0	< 1.0
40	1.2	1.9	< 1.0	< 1.0	< 1.0	0.0	< 1.0
50	12.42	25.6	< 1.0	< 1.0	6.0	< 1.0	< 1.0
60	131.2	153.0	< 1.0	< 1.0	18.0	< 1.0	2.4
70	720.0	740.0	< 10.0	< 10.0	94.0	< 10.0	22.0
80	> 1000.0	> 1000.0	21.0	< 10.0	456.0	< 10.0	66.0
90	> 1000.0	> 1000.0	367.0	42.5	> 1000.0	235.0	633.0
100	> 1000.0	> 1000.0	> 1000.0	230.0	> 1000.0	> 1000	> 1000.0

and for any weighting of $\mathcal{E}$, the greedy algorithm always yields the maximum weight member of the family $\mathcal{I}$ of bases of $\mathcal{M}$ [40], [41].

3.6.2 Communication constraints and matroids

We now proceed to check if the independent sets $\mathcal{I}$ for the transmission coordination problem for the various CCSs are matroids.

Fig. 3.8 depicts all possible independent sets for problems that include the HD constraint for a K_3 graph (complete graph with three nodes) as well as all independent sets for the combined SR and UC problems on the same graph.

1. HD: The condition to be kept in mind while constructing the independent sets under the HD constraint is that no two edges common to any vertex in the independent set can have opposite directionality, *i.e.,* all edges in an independent set that are common to any vertex have to be either all *exclusively incoming* or *exclusively outgoing*, and not a combination of both. Consider the simple counter-example depicted in **Fig. 3.9**. Clearly the union of any of the outgoing edges of i (from $\mathcal{I}_1$) with any of the incoming edges of i (from $\mathcal{I}_2$) violates the HD property. Thus this pair of subsets violates condition 3 in the matroid Definition.

Hence the half-duplex constraint does not yield a matroid, and hence cannot be solved via use of a greedy algorithm. This should come as no surprise, since we have already established that problems involving the HD constraint cannot be solved by LP relaxation.

2. SR: Form the set $\mathcal{I}$ to consist of all distinct unordered subsets of edges in $\mathcal{E}$, with the following restriction: for any $i, j, k \in \mathcal{V}$, any grouping that contains edges $(i, j), (k, j)$ is an invalid grouping. This condition results in grouping of edges where each node in $\mathcal{V}$ is the head of (at most) one edge. Hence for any two groupings $\mathcal{I}_1, \mathcal{I}_2 \in \mathcal{I}$, where $|\mathcal{I}_1| = |\mathcal{I}_2| + 1$, we are guaranteed that $\mathcal{I}_1$ contains at least one edge that has as its head a node that is not the head of any edge in $\mathcal{I}_2$. This edge can then be added to $\mathcal{I}_2$ to form a grouping of cardinality $|\mathcal{I}_2|+1$. Hence the independent subsets $\mathcal{I}$ under the single receiver constraint form a matroid *i.e.,* $\mathcal{M} = (\mathcal{E}, \mathcal{I}^{\text{SR}})$.

3. UC: The proof for the UC constraint is similar to the one for the SR condition above, the only difference is instead of limiting the in-degree of a node to at most 1, we limit the out-degree to be at most 1. Hence the edges that comprise the independent subsets of $\mathcal{E}$ can contain at most one outgoing edge from any node.

4. HD+SR: In this case the independent sets consist of edges such that a node can have either (i) at most one incoming edge (SR constraint/degree constrained subgraph with incoming degree ≤ 1) **OR** (ii) only outgoing edges with no limit on degree (since there is no UC constraint), but not both (i) and (ii). For the K_3 graph we have a simple counter-example. Refer to **Fig. 3.10 (top)**.

5. HD+UC: The same argument as above can be used to show that the HD+UC independent subsets do not form a matroid. Refer to **Fig. 3.10 (middle)**.

6. SR+UC: As seen the UC and SR constraints separately yield matroids. Combining these two however does not yield a matroid. Refer to **Fig. 3.10 (bottom)** for a simple counter-example.

7. HD+SR+UC: Fig. 3.11 provides a simple counter-example.

8. I: We wish to see whether the I constraint yields a matroid. Consider **Fig. 3.12**: G_c denotes the communication graph, and G_i is the interference graph. We choose $\beta_i > \beta_c$, as a result the interference graph is a subset of the communication graph, with the edge between B and C not present in the former. We see that combining (B, C) with edge (C, A) results in interference at A, whereas combining edge (B, C) with edge (A, B) results in interference at C. Hence the independent subsets under the protocol interference constraint fails to yield a matroid. As a a result any problem that includes the I constraint will result in a non-matroid structure.

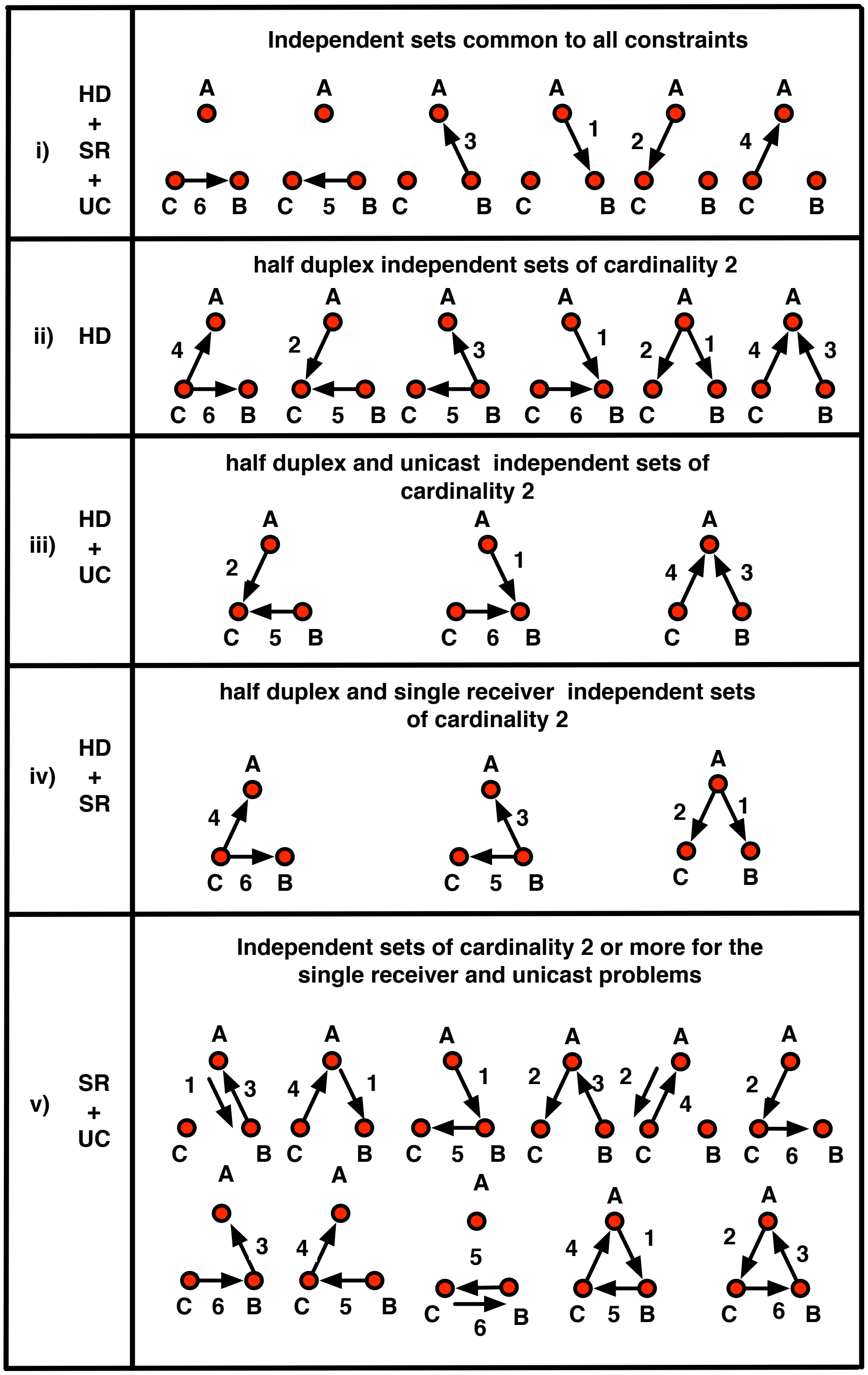


Figure 3.8: Independent sets for the (i) HD+SR+UC, (ii)HD, (iii) HD+UC, (iv) HD+SR, (v) SR+UC problems for a K_3 graph.

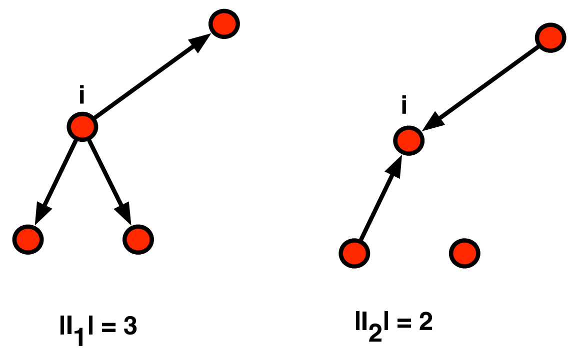


Figure 3.9: Counter-example to show that the HD constraint does not yield a matroid.

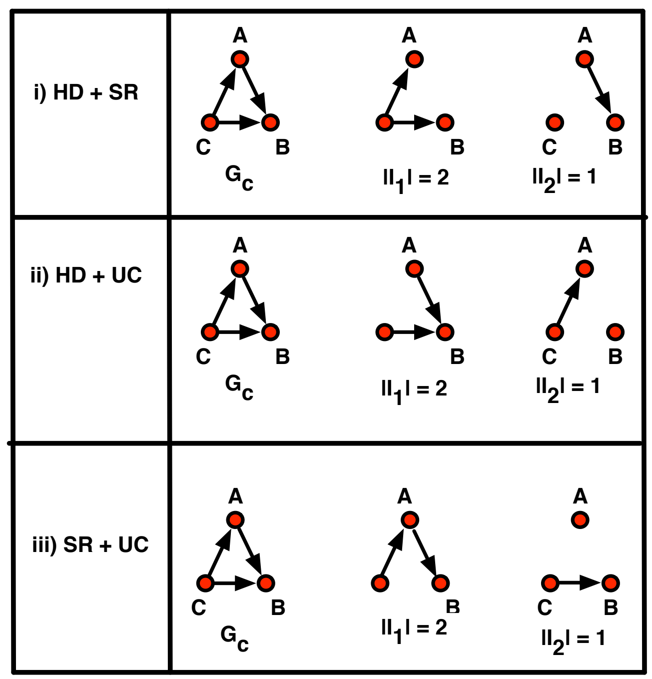


Figure 3.10: Counter-examples for (i) HD+SR, (ii) HD+UC, (iii) SR+UC.

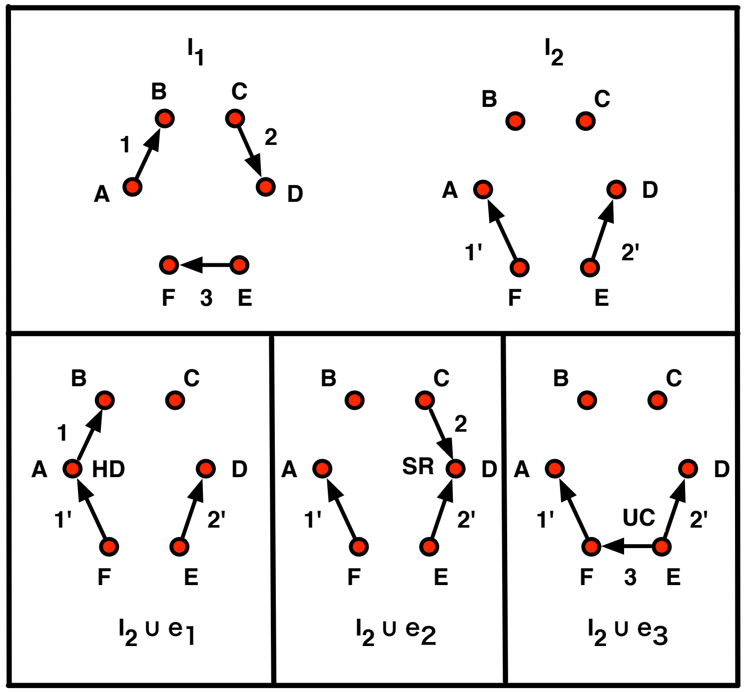


Figure 3.11: Counter-example demonstrating that the problem under all three primary constraints fails to yield a matroid. Adding any edge from $\mathcal{I}_1$ to $\mathcal{I}_2$ violates one of the primary constraints.

3.6.3 Importance of matroids in network design

From a design standpoint finding the optimal transmission coordination scheme implies solving a global optimization problem, which requires centralized control. In the best case this solution may be found in polynomial time (as in the case of the MWM problem). For most problems however, as network size increases, the explosion in search space makes centralized policies difficult and expensive. As a result one needs to make use of decentralized (distributed) policies that are able to obtain good approximations of the optimal solution in constant or polynomial times.

Matroids drastically reduce the complexity of of the optimization problem. This reduction in complexity, obviates the need of a centralized policy. As a result, the optimal transmission coordination scheme can now be found using decentralized poli-

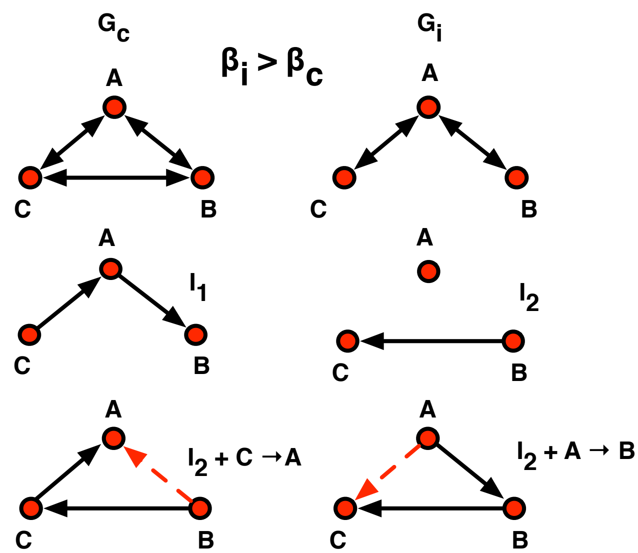


Figure 3.12: Counter-example demonstrating that the I constraint fails to yield a matroid. Adding either edge from $\mathcal{I}_1$ to $\mathcal{I}_2$ gives a pair of edges that interfere with one another.

cies, by making use of a simple greedy heuristic. This makes them readily scalable. As an example consider the SR and UC problems (which we have shown to be matroids). The optimal transmission coordination scheme for these problems can be found in linear time ($O(M)$ where M is the number of edges in graph).

3.7 Heuristics vs. optimal solution

We now compare the solutions provided by the **edge length** and **degree** heuristics to the optimal solution. We show that depending on the constraint set, either, both, or neither heuristic yields a good approximation to the optimal solution.

We will evaluate two different edge weights: $w_e = 1$ and $w_e = l_{\max}/l_e$ (p. 74). We evaluate the eight combinations of the primary constraints without the I constraint, and then the eight combinations with the I constraint. We present results for $\beta_c = 0.2$ and $\beta_c = 12.0$; this allows to us to see how the heuristics compare when the number of edges is large ($\beta_c = 0.2$) and small ($\beta_c = 12.0$). For $\beta_c = 0.2$, and $N = 100$, we set $|M| = 1908$, whereas for $\beta_c = 12.0$, and $N = 100$, we set $|M| = 284$. Below is a list of figures for this section.

A. Only Primary

1. $\beta_c = 0.2$
   - **w** $= 1$ - **Fig. 3.13**
   - **w** $\neq 1$ - **Fig. 3.14**
2. $\beta_c = 12.0$
   - **w** $= 1$ - **Fig. 3.15**
   - **w** $\neq 1$ - **Fig. 3.16**

B. Primary and Interference

1. $\beta_c = 0.2$
   - **w** $= 1$ - **Fig. 3.17**
   - **w** $\neq 1$ - **Fig. 3.18**
2. $\beta_c = 12.0$
   - **w** $= 1$ - **Fig. 3.19**
   - **w** $\neq 1$ - **Fig. 3.20**

3.7.1 Primary constraint problems

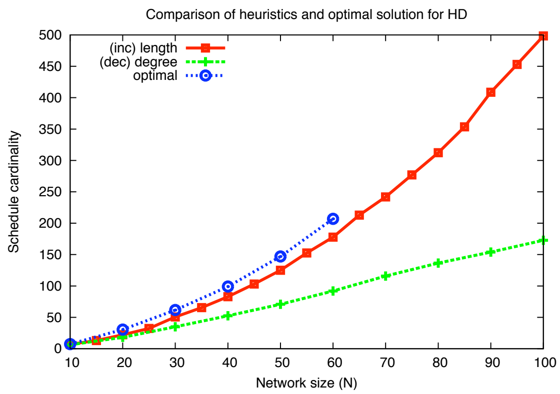
3.7.1.1 SNR threshold $\beta_c = 0.2$.

The communication graph is constructed using an SNR requirement of $\beta_c = 0.2$. The **length** sorting algorithm sorts edges by increasing length.

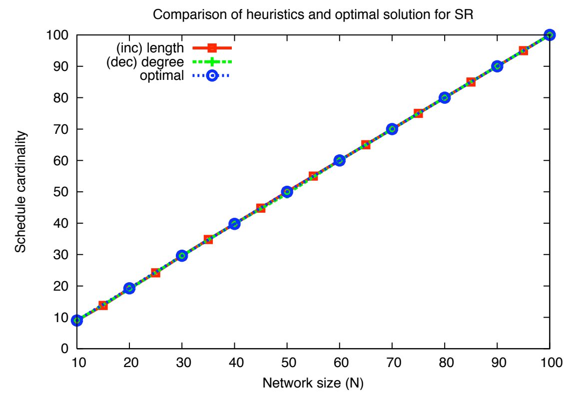
A. Finding optimal cardinality of schedule (Maximizing $\mathbf{1}^\mathsf{T}\mathbf{x}$) (Fig. 3.13).

1. **HD (Fig. 3.13(a)).** We observe that the **length** heuristic provides a good approximation to the optimal solution. We are unsure as to the precise reason for this performance.

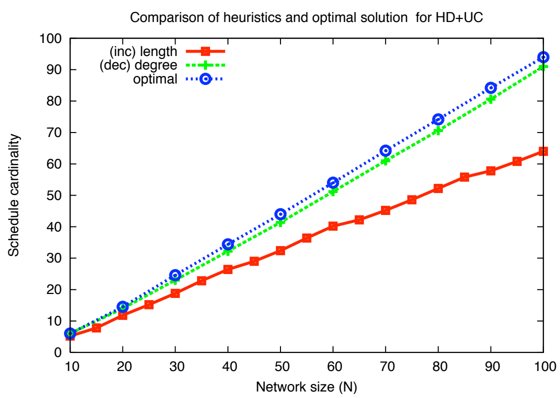
2. **SR/UC (Fig. 3.13(b)).** This plot is for SR, but the UC plot is identical. This plot demonstrates that the solution obtained using both heuristics matches the optimal solution. For SR under each heuristic we add one incoming edge per node. As long as every node has an incoming edge all nodes will end up with one and only one incoming edge. When $\mathbf{w} = 1$ the optimal solution in



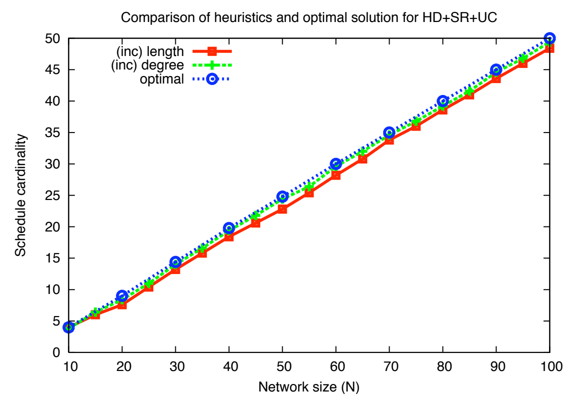
(a) HD



(b) SR, UC



(c) HD+SR, HD+UC



(d) HD+SR+UC

Figure 3.13: Heuristics vs. optimal solution under various primary constraints with $\beta_c = 0.2$. Objective is to maximize cardinality of the schedule.

not unique: any algorithm that goes through all the edges will find an optimal solution. A similar argument also holds for the UC constraint problem.

3. **HD+SR / HD+UC (Fig. 3.13(c)).** This plot is for HD+SR, but the HD+UC plot is identical. This plot demonstrates that the **length** heuristic performs quite poorly but the **degree** heuristic is near optimal. In the case of the HD+SR constraint a good strategy is to activate as many outgoing edges as possible for each node. For the HD+UC case a good strategy is to activate as many incoming edges as possible for each node. This is done by making all nodes receivers while minimizing the number of transmitters for the HD+SR case. For the HD+UC problem this is done by making all nodes transmitters while minimizing the number of receivers. The **length** heuristic in this case fails to yield a good approximation due to the fact that we are trying to make a decision on an edge by edge basis, when what is required is to make a decision on per node basis.

4. **HD+SR+UC (Fig. 3.13(d)).** This problem strives to find the maximum matching, such that each node has exactly one edge (either incoming or outgoing). This plot shows that both heuristics provide a good approximation to the optimal solution. The **degree** heuristic provides a slightly better solution compared to the **length** heuristic. This can be attributed to the fact that by picking nodes in order of increasing degree, we minimize the probability that lower degree nodes might be unmatched. When nodes are picked on the basis of distance a low degree node might be left out, meaning that this node is not part of the matching.

B. Maximizing $\mathbf{w}^{\mathsf{T}}\mathbf{x}$ (Fig. 3.14).

1. **HD (Fig. 3.14(a)).** We observe that the **length** heuristic is near optimal while the **degree** heuristic is severely sub-optimal. The time required to find the optimal solution for $N > 60$ exceeded the $T = 1000$ seconds timeout period set for CPLEX.

2. **SR/UC (Fig. 3.14(b)).** Any greedy heuristic will yield the optimal solution.

3. **HD+SR / HD+UC (Fig. 3.14(c)).** Both heuristics are near optimal here, in contrast with the $\mathbf{w} = 1$ case (**Fig. 3.13(c)**) where the **degree** heuristic was near optimal but the **length** heuristic was sub-optimal. This is a consequence of the edges having different weights and using a sorting policy that gives preference to the edge with the largest weight under both **length** and **degree** heuristics.

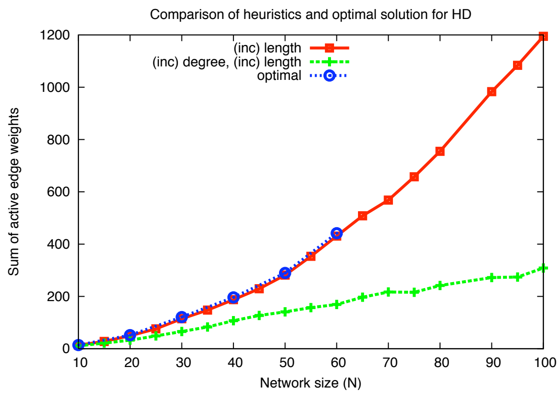
4. **HD+SR+UC (Fig. 3.14(d)).** The **length** heuristic is near optimal whereas the **degree** heuristic is sub-optimal, in contrast with the $\mathbf{w} = 1$ case (**Fig. 3.13(d)**) where both heuristics were nearly optimal. This is because selecting edges purely on edge weight guarantees the selection of the edge with the largest weight, whereas selecting on the basis of degree under the HD+SR+UC constraint does not.

3.7.1.2 SNR threshold $\beta_c = 12.0$.

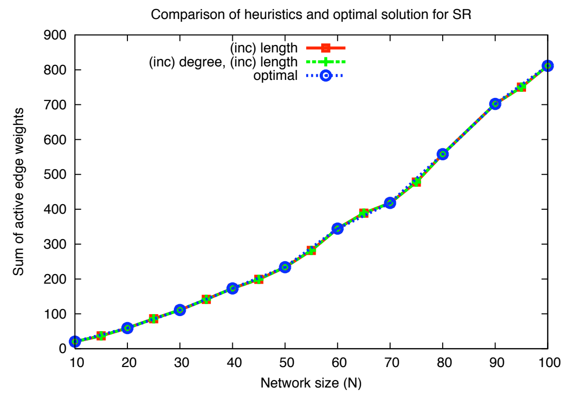
The communication graph in this case is constructed using an SNR requirement of $\beta_c = 12.0$.

A. Finding optimal cardinality of schedule (Maximizing $\mathbf{1}^{\mathsf{T}}\mathbf{x}$) (Fig. 3.15).

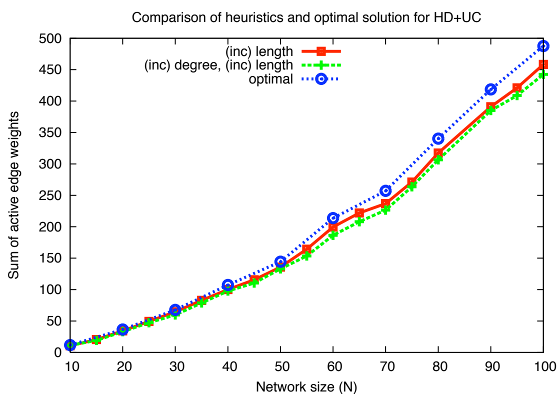
1. **HD (3.15(a))**. Both heuristics are optimal for N up to around 60 nodes, while for $N > 60$ both heuristics are increasingly sub-optimal, but the performance



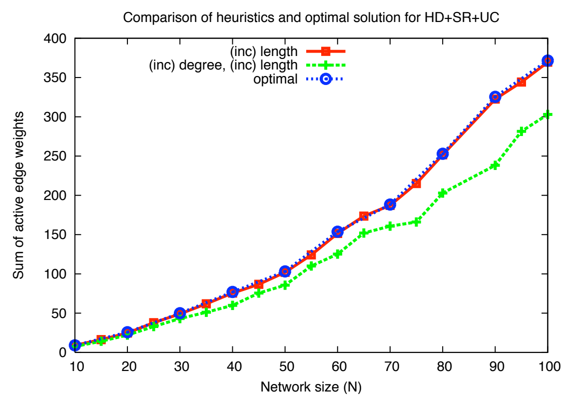
(a) HD



(b) SR, UC

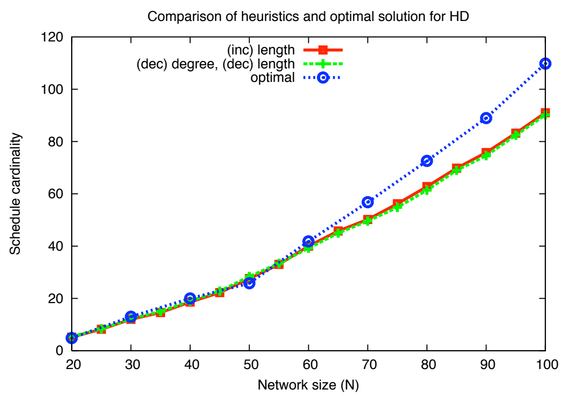


(c) HD+SR, HD+UC

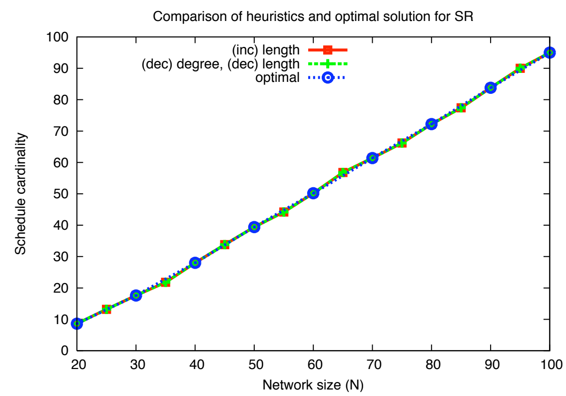


(d) HD+SR+UC

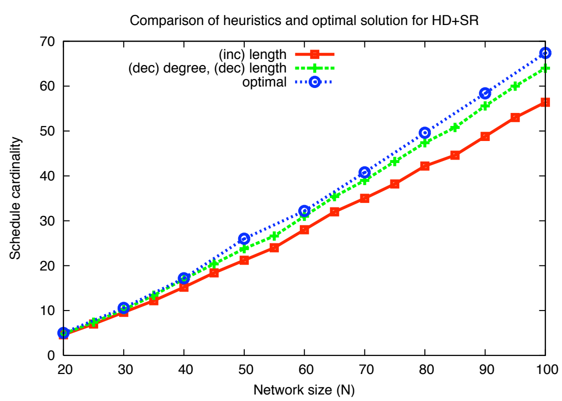
Figure 3.14: Heuristics vs. optimal solution under various primary constraints with $\beta_c = 0.2$. Objective is to maximize $\mathbf{w}^\mathsf{T}\mathbf{x}$.



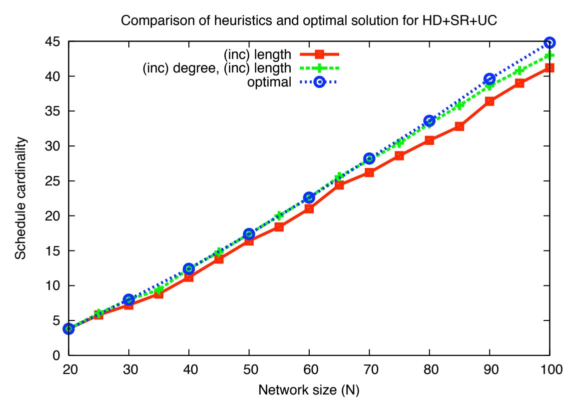
(a) HD



(b) SR, UC



(c) HD+SR, HD+UC



(d) HD+SR+UC

Figure 3.15: Heuristics vs. optimal solution under various primary constraints with $\beta_c = 12.0$. Objective is to maximize cardinality of the schedule.

under the two heuristics is identical. The **degree** heuristic works better for $\beta_c = 12.0$ than for $\beta_c = 0.2$, at least for the available data up to $N = 60$ (see **Fig. 3.13(a)**).

2. **SR/UC (3.15(b))**. This plot is for SR, but the UC plot is identical. The optimal solution is not unique and any greedy heuristic will find an optimal solution.

3. **HD+SR (3.15(c))**. This plot is for HD+SR, but the HD+UC plot is identical. We observe the **degree** heuristic is near-optimal while the **length** heuristic is less optimal. The **length** heuristic performs better for $\beta_c = 12.0$ than for $\beta_c = 0.2$ (see **Fig. 3.13(c)**).

4. **HD+SR+UC (3.15(d))**. The **degree** heuristic reduces the possibility of low degree nodes being unmatched, relative to the **length** heuristic. The **length** heuristic does not do this. Although not shown here, using a **degree** heuristic that selects edges in order of decreasing node degree yields a solution that is worse than the solution obtained using the **length** heuristic.

B. Maximizing $\mathbf{w}^\mathsf{T}\mathbf{x}$ (Fig. 3.16).

1. **HD (Fig. 3.16(a)).** As in the case of **Fig. 3.14(a)**, the **length** heuristic outperforms the **degree** heuristic, yielding a very good approximation of the optimal solution when N is small. We also observe better performance for the **degree** heuristic for $\beta_c = 12.0$ than for $\beta_c = 0.2$ (see **Fig. 3.14(a)**). This is a result of the the fact that as the number of edges in the graph drop, the penalty of sorting on a per node basis decreases.

2. **SR/UC (Fig. 3.16(b)).** As discussed in **Fig. 3.14(b)** any greedy heuristic will obtain the optimal solution.

3. **HD+SR / HD+UC (Fig. 3.16(c)) and HD+SR+UC (Fig. 3.16(d))** Both heuristics yield nearly identical results for **Fig. 3.16(c)**. Performance of the **degree** heuristic improves for $\beta_c = 12.0$ (**Fig. 3.16(d)**) than for $\beta_c = 0.2$ (**Fig. 3.14(d)**) for the same reason as in **Fig. 3.16(a)**.

3.7.2 Interference constraint problems

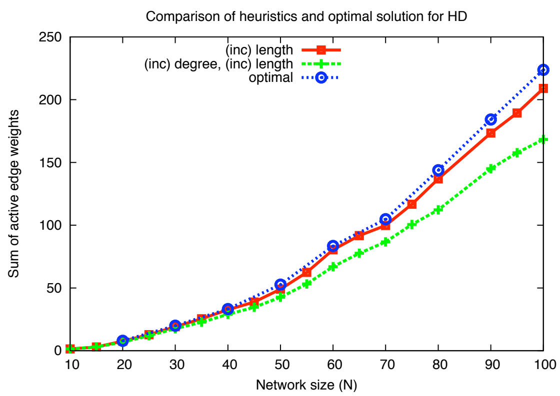
3.7.2.1 Primary constraint problems with I constraint for $\beta_c = 0.2$.

A. Finding optimal cardinality of schedule (Maximizing $\mathbf{1}^\mathsf{T}\mathbf{x}$) (Fig. 3.17).

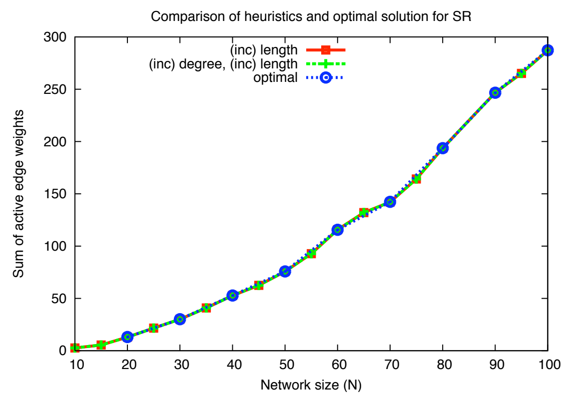
1. **I (Fig. 3.17(a))**. Under the I constraint both heuristics exhibit similar performance. The two heuristics are increasingly sub-optimal with increasing N.

2. **HD+I (Fig. 3.17(b))**. Under the HD+I constraint the **(dec) length** heuristic performs better than the **degree** heuristic with increasing N. Although not shown here, the **(inc) length** heuristic yields a solution that is severely sub-optimal (the maximum schedule cardinality under HD+PR is only 24 at $N = 70$). We are not certain as to why the **(dec) length** heuristic performs better than the **(inc) length** heuristic.

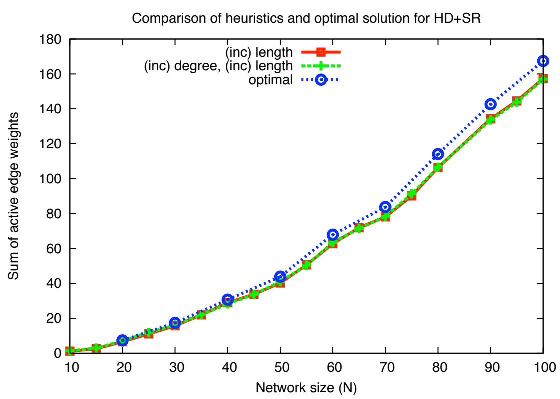
3. **SR+I, UC+I (Fig. 3.17(c))**. We observe that under the SR+I constraint the **degree** heuristic performs better than the **length** heuristic, however both are sub-optimal. Under the **degree** heuristic edges are added on a per node basis. Under the **length** heuristic an attempt to add two successive edges may result in one of them violating the I constraint since they may belong to two different originating nodes, which would cause interference to a receiver in range. Adding edges on a per node basis prevents this. A similar argument holds for the UC+I constraint problem. Observe that the optimal cardinality for the UC+I problem is less than that of the SR+I problem. This is a direct



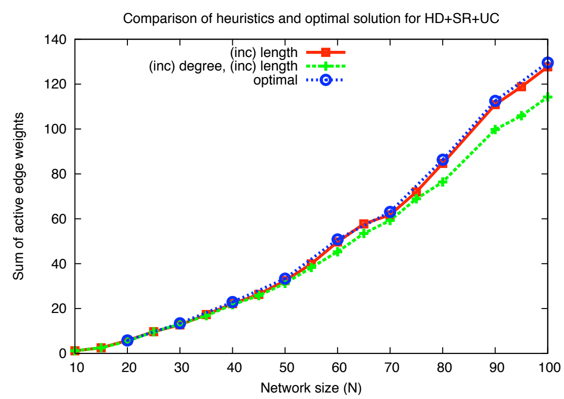
(a) HD



(b) SR, UC

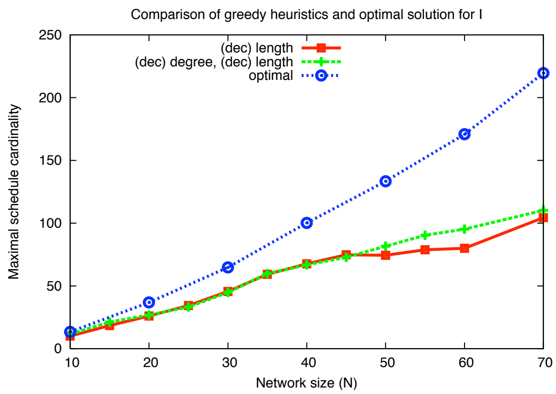


(c) HD+SR, HD+UC

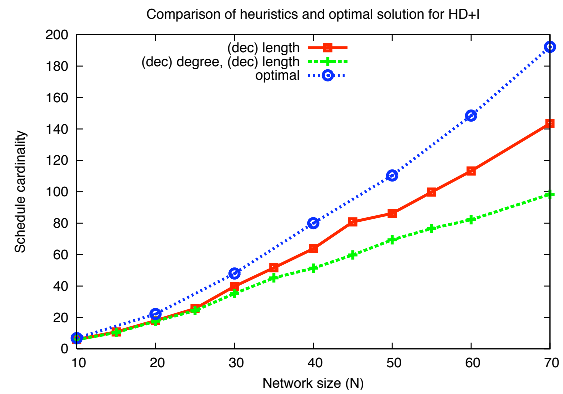


(d) HD+SR+UC

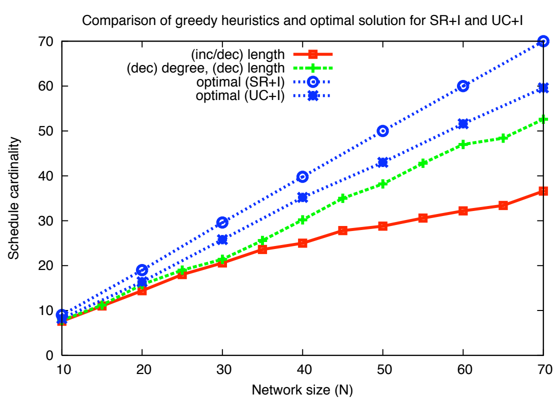
Figure 3.16: Heuristics vs. optimal solution under various primary constraints with $\beta_c = 12.0$. Objective is to maximize $\mathbf{w}^\mathsf{T}\mathbf{x}$.



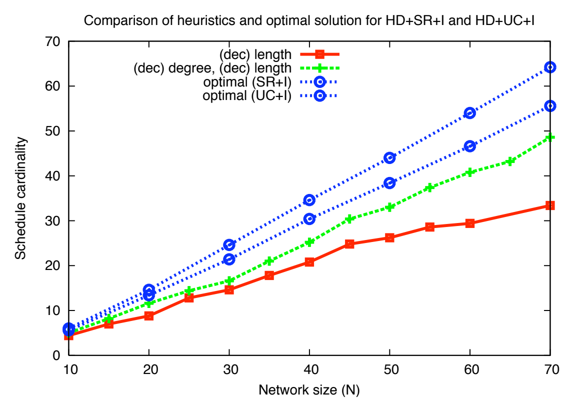
(a) I



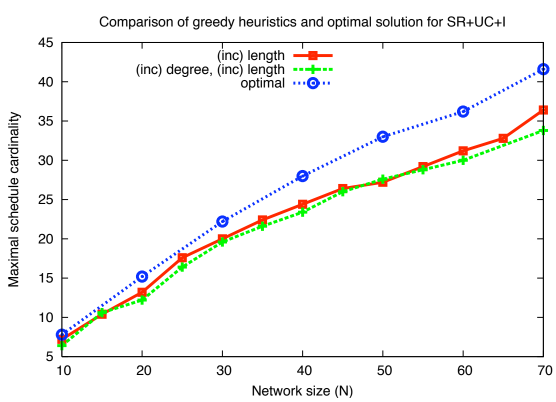
(b) HD+I



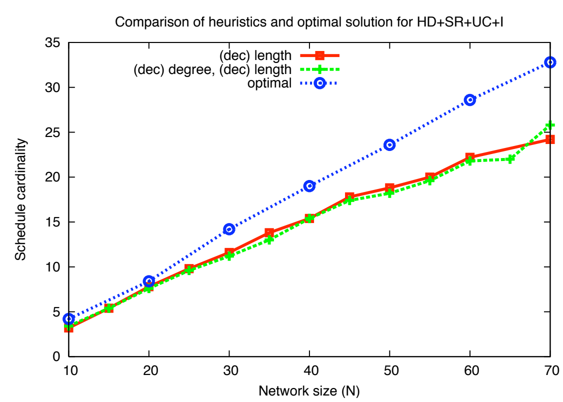
(c) SR+I, UC+I



(d) HD+SR+I, HD+UC+I



(e) SR+UC+I



(f) HD+SR+UC+I

Figure 3.17: Heuristics vs. optimal solution for various primary constraints with $\beta_c = 0.2$ and I constraint with $\beta_i = 4.0$. Objective is to maximize cardinality of the schedule.

result of the I constraint which limits the number of transmitters that may be simultaneously transmitting.

4. **HD+SR+I, HD+UC+I (Fig. 3.17(d))**. As was observed in **Fig. 3.13(c)** the **degree** heuristic performs better than the **length** heuristic, the performance of the **degree** heuristic however, drops in comparison to **Fig. 3.13(c)**. As in the case of **Fig. 3.17(c)** we observe an asymmetry in the solutions of HD+SR+I and HD+UC+I.

5. **SR+UC+I (Fig. 3.17(e)) and HD+SR+UC+I (Fig. 3.17(f))**. Both heuristics yield similar solutions, which are increasingly sub-optimal with increasing N.

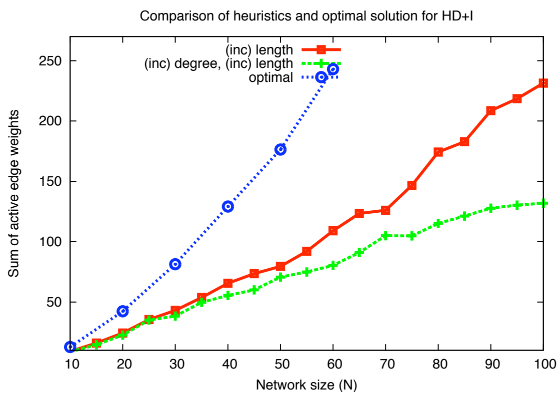
B. Maximizing $\mathbf{w}^{\mathsf{T}}\mathbf{x}$ (Fig. 3.18)

We observe that when both HD and I constraints are in effect the **length** heuristic always outperforms the **degree** heuristic. This is clearly visible for the HD+I (**Fig. 3.18(a)**), HD+SR+I (**Fig. 3.18(e)**) and HD+SR+UC+I (**Fig 3.18(f)**) problems. The two heuristics perform almost identically when the HD constraint is not in effect *i.e.,* for the SR+I (**Fig. 3.18(b)**) , UC+I (**Fig. 3.18(c)**), SR+UC+I (**Fig. 3.18(d)**) cases. Observe that when all constraints are active the **length** heuristic yields a fairly good approximation of the optimal solution. For all other problems however, the heuristics are severely sub-optimal with increasing N.

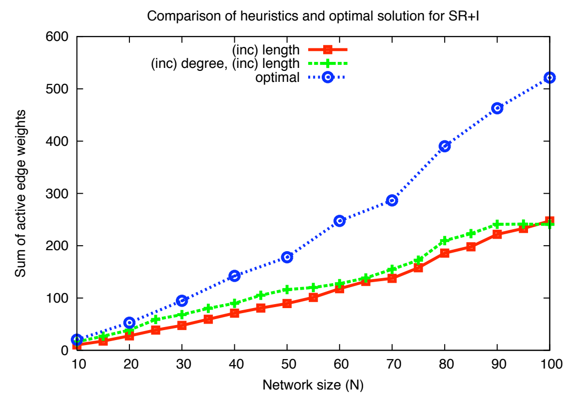
To summarize, we observe that with the addition of the I constraint, the solutions provided by the heuristics become increasingly sub-optimal with increasing N.

3.7.2.2. Primary constraint problems with I constraint for $\beta_c = 12.0$.

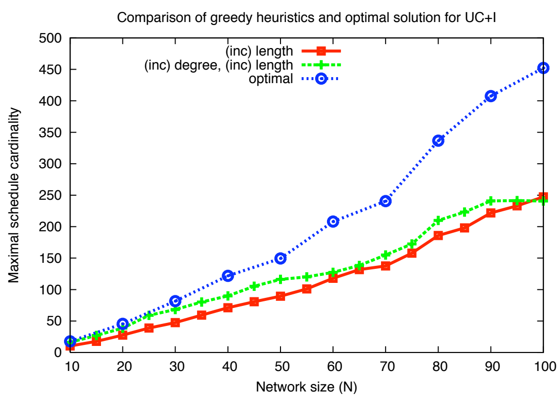
The optimal solutions for the following pairs of problems have the same solution: (*i*) I and SR+I, (*ii*) HD+I and HD+SR+I, (*iii*) UC+I and SR+UC+I, and (*iv*) HD+UC+I and HD+SR+UC+I constraint problems. In effect the SR constraint has



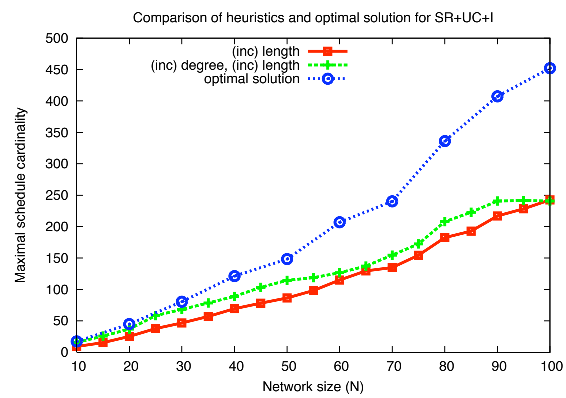
(a) HD+I



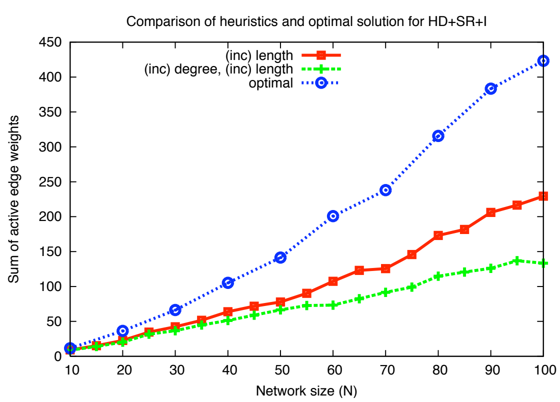
(b) SR+I



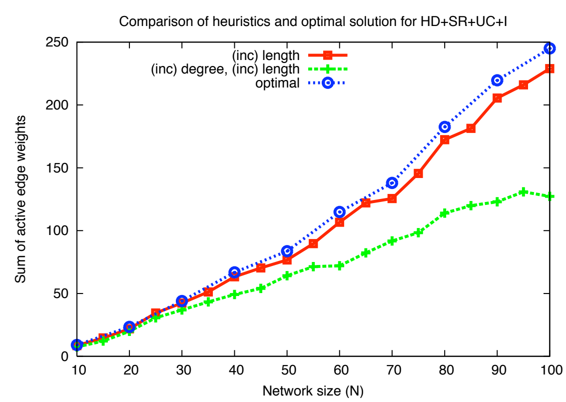
(c) UC+I



(d) SR+UC+I



(e) HD+SR+I



(f) HD+SR+UC+I

Figure 3.18: Heuristics vs. optimal solution for various primary constraints with $\beta_c = 0.2$ and I constraint with $\beta_i = 4.0$. Objective is to maximize $\mathbf{w}^\mathsf{T}\mathbf{x}$.

no impact.

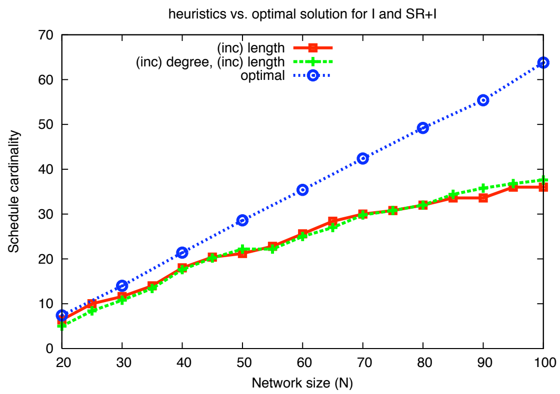
A. Finding optimal cardinality of schedule (Maximizing $\mathbf{1}^\mathsf{T}\mathbf{x}$) (Fig. 3.19).

From the plot of I and SR+I (**Fig. 3.19(a)**) we observe that both heuristics yield nearly similar solutions which become increasingly sub-optimal with increasing N. The same holds for HD+I (**Fig. 3.19(b)**). The I constraint critically impacts and dominates the optimization problem. Two apparently distant transmissions (edges) can interfere. As a result a localized selection policy, such as a greedy heuristic, which provided good performance for the primary constraint problems, now fails.

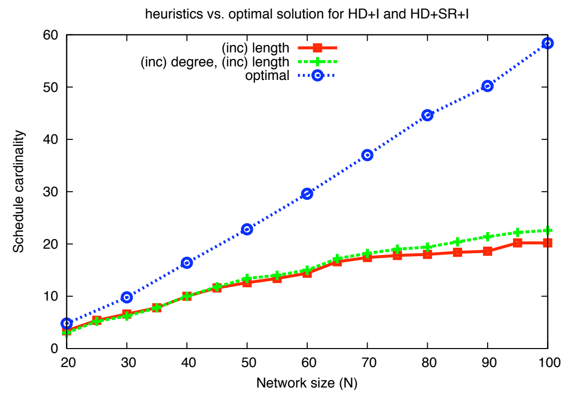
When the UC constraint is in effect (**Fig's. 3.19(c), 3.19(d)**) the heuristics performs better for $\beta_c = 12.0$ than for $\beta_c = 0.2$, for low N. Also when all four constraints are active the **degree** heuristic performs very well for low N.

B. Maximizing $\mathbf{w}^\mathsf{T}\mathbf{x}$ (Fig. 3.20).

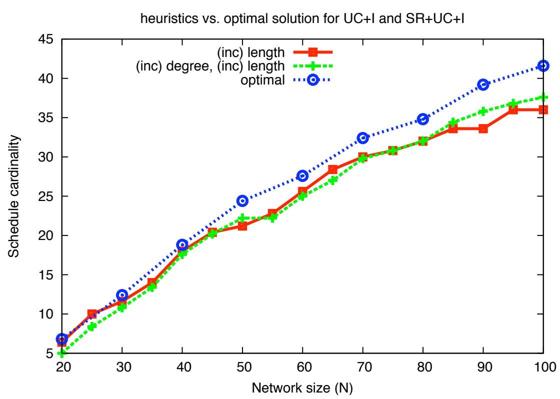
For all the plots we observe that the **degree** heuristic is severely sub-optimal with increasing N. For the UC+I (**Fig. 3.20(c)**) and HD+UC+I (**Fig. 3.20(d)**) plots we observe that the **degree** heuristic yields a fairly good approximation of the optimal. The I constraint by itself makes the problem of finding the optimal schedule a hard one as in effect we are trying to maximize the number of transmissions, with each node possibly being able to transmit to multiple nodes simultaneously. When the UC constraint is in effect, each node can now only transmit to one other node. The combination of the UC and I constraints as well as the fewer number of edges in G_c for $\beta_c = 12.0$ reduces the number of edges that can be simultaneously active. As a result the **length** heuristic can provide a good approximation of the optimal solution. For the UC+I, SR+UC+I, HD+UC+I, and HD+SR+UC+I (**Fig.'s 3.20(c), 3.20(d)**) problems we observe that although the **length** heuristic is increasingly sub-optimal (with increasing N), it still obtains a good approximation of the optimal solution. For the HD+I and HD+SR+I problems (**Fig.3.20(b)**) we observe that the **length**



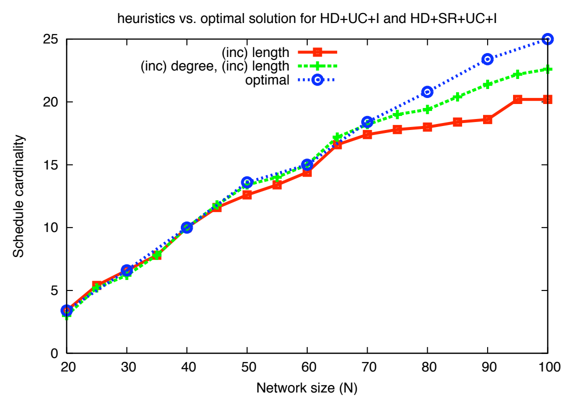
(a) I, I+SR.



(b) HD+I, HD+SR+I.

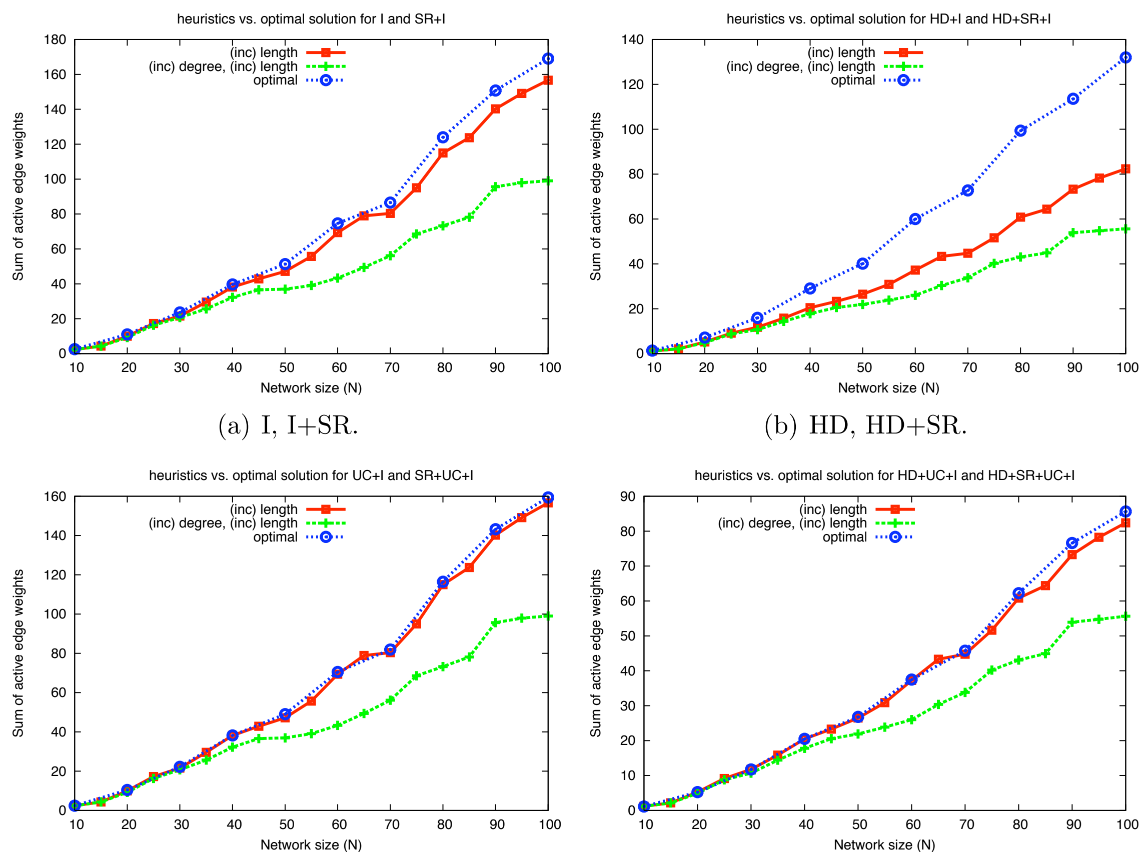


(c) UC+I, UC+SR+I.



(d) HD+UC+I, HD+UC+SR+I.

Figure 3.19: Heuristics vs. optimal solution for various primary constraints with $\beta_c = 12.0$ and I constraint with $\beta_i = 4.0$. Objective is to maximize cardinality of the schedule.



(a) I, I+SR.

(b) HD, HD+SR.

(c) UC+I, UC+SR+I.

(d) HD+UC+I, HD+UC+SR+I.

Figure 3.20: Heuristics vs. optimal solution for various primary constraints with $\beta_c = 12.0$ and I constraint with $\beta_i = 4.0$. Objective is to maximize $\mathbf{w}^\mathsf{T}\mathbf{x}$.

heuristic is severely sub-optimal with increasing N.

We conclude that the heuristics in general fail to achieve good performance in the presence of the I constraint. The only instance where the heuristics provide good approximations of the optimal solutions is for $\beta_c = 12.0$, $\mathbf{w} \neq 1$, *and when the UC constraint is in effect.*

3.8 Impact of the various constraints

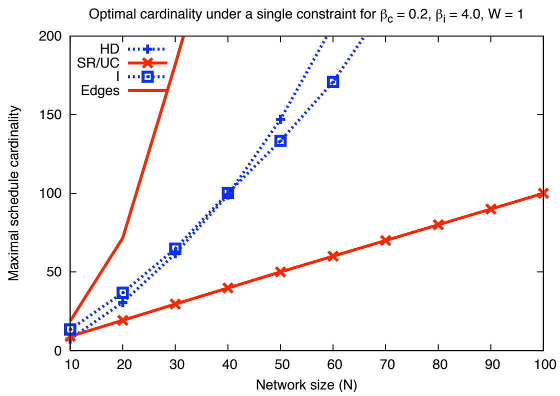
We now proceed to test which constraint(s) are the most limiting **(Fig. 3.21)**.

1. **Operating under a single constraint**. From **Fig. 3.21(a)** we observe that the **SR** and **UC** are the most limiting constraints when $\beta_c = 0.2$, whereas the **HD** constraint is the least limiting. For $\beta_c = 12.0$ (**Fig. 3.21(b)**) however, we observe that the **I** constraint is the most limiting constraint, whereas **HD** is the least limiting for $N \leq 80$. For $N > 80$ both **SR** and **UC** become the most limiting.

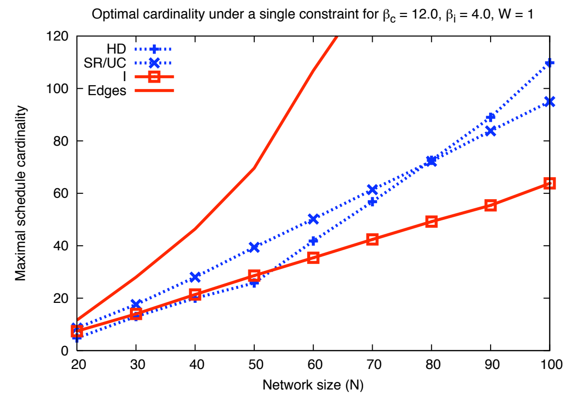
2. **Operating under a pair of constraints**. From **Fig. 3.21(c)** we observe that **HD+I** is clearly the least limiting constraint pair. Also we observe that the **SR+UC** and **SR+I** constraints have the same solution. Also, **HD+SR** and **HD+UC** have the same solution. We also observe that the **UC+I** constraint is the most limiting constraint with increasing N. Of all the constraint pairs, only the solution of the **HD+I** constraint exhibits a quadratic growth in N. For the $\beta_c = 12.0$ case we again observe that the **UC+I** constraint is clearly the most limiting constraint pair with increasing N. Also the **HD+SR/HD+UC** constraints are now the second most limiting pair constraints (in the $\beta_c = 0.2$ case they were clearly the least limiting pairs of constraints). Note also that the solutions of these pairs no longer exhibit a quadratic growth. Finally we point

out that for the $\beta_c = 12.0$ case **SR+UC** is now the least limiting constraint pair.

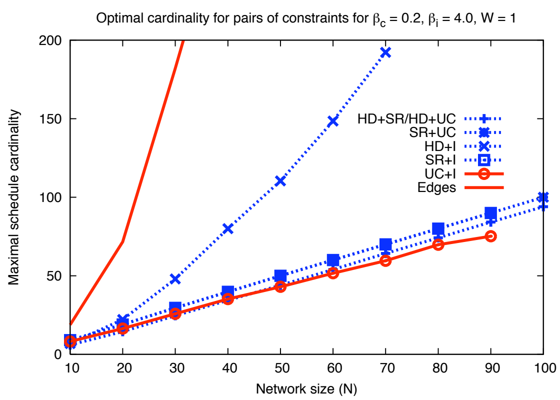
3. **Operating under three constraints**. For the $\beta_c = 0.2$ (**Fig. 3.21(e)**) case we observe that **HD+SR+I** is the least limiting, followed by **HD+UC+I**, and **SR+UC+I**. For N between 10 and 90 **HD+SR+UC** is the most limiting combination, however as N increases beyond 100 we expect that the **SR+UC+I** constraint will be the most limiting combination. For the $\beta_c = 12.0$ (**Fig. 3.21(f)**) case we observe that **HD+SR+UC** is the least limiting combination with increasing N. The **HD+UC+I** combination which was the second least limiting combination in the $\beta_c = 0.2$ case is now the least limiting combination.



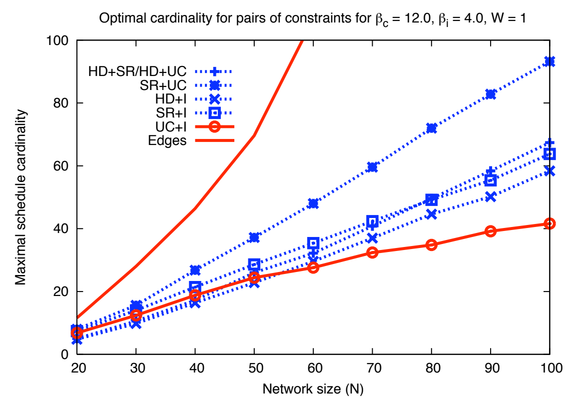
(a) Optimal cardinality under a single constraint for $\beta_c = 0.2$.



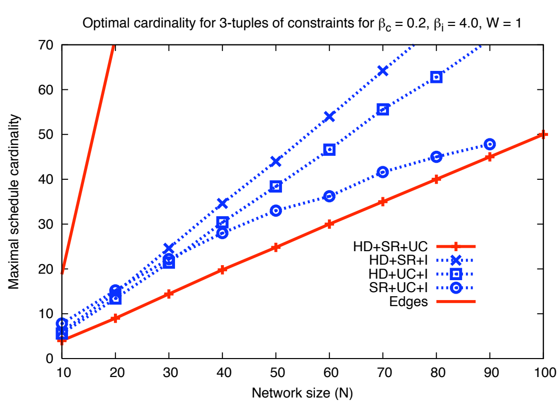
(b) Optimal cardinality under a single constraint for $\beta_c = 12.0$.



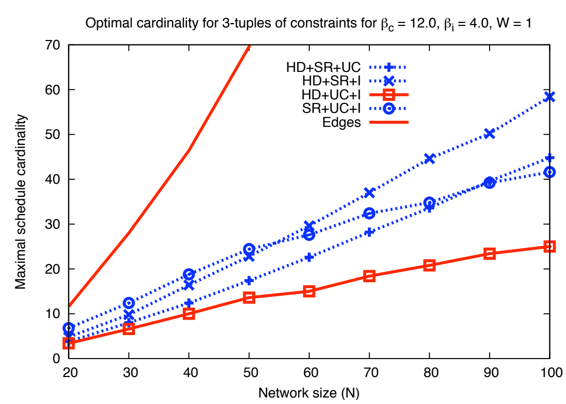
(c) Optimal cardinality under a pair of constraints for $\beta_c = 0.2$.



(d) Optimal cardinality under a pair of constraints for $\beta_c = 12.0$.



(e) Optimal cardinality under 3 constraints for $\beta_c = 0.2$.



(f) Optimal cardinality under 3 constraints for $\beta_c = 12.0$.

Figure 3.21: Optimal solution of various problems with $\beta_c = 0.2, 12.0$ and $\beta_i = 4.0$. Objective is to maximize cardinality of the schedule.

3.9 Summary

3.9.1 Contributions

To summarize:

- We characterize the various problems by way of their Primal-Dual LP formulations.
- We prove that only the single receiver and unicast problems individually yield matroids.
- We prove that neither half-duplex nor interference problems yield matroids, nor are they TUM.
- Simulations suggest that the combined half-duplex, single receiver and unicast problem (matching problem) can be well approximated by our heuristics. This holds even with the addition of the protocol interference problem.
- Simulations suggest that our heuristics are able to yield good approximations of the optimal solution for the problems that do not include the interference constraint.
- Simulations suggest that it is more difficult to obtain a good approximation when the edge weights are unity (meaning the objective is to maximize the number of activated edges) than the problem where the objective is to maximize the weighted sum of active edges.
- When operating under only the interference constraint, our heuristics fail to yield good approximations of the optimal.
- Depending on the problem at hand, either both heuristics work very well, or one outperforms the other, or neither works well.

3.9.2 Future work

There are several promising directions for future work. One such direction looks to address the problems encountered when employing LP relaxation. When employing LP relaxation to solve the original ILP, the tightness of the relaxation depends on the rounding rule employed. Ensuring that the relaxation is tight requires a certain amount of trial and error in choosing the rounding rule. An alternate approach would be to construct a feasible integral solution to the LP (Primal), while making use of a related LP (the Dual) to guide our decisions. The procedure that outlines this constitutes the Primal-Dual algorithm. This is a natural next step to try as we already have the Primal-Dual pairs for each of the problems. Along the same lines one can use semidefinite programming (SDP) techniques to solve our set of linear optimization problems. SDP is generally regarded as an extension of linear programming where the inequalities between vectors is replaced by matrix inequalities. One major difference between LP and SDP is that duality results are weaker for the latter. Nevertheless, SDP can be solved very efficiently in practice. Yet another promising direction is to look at more complex algorithms such as belief propagation techniques which have been successfully adopted in areas like iterative coding and computer vision, that involve graphs with numerous cycles. It is possible that this technique could be successfully adapted to our framework, which in turn may provide better approximations to the optimal solutions specially in the case of problems that involve the protocol interference constraint.

Bibliography

[1] N. Abramson, "The ALOHA system–another alternative for computer communications," in *Proceedings of the American Federation of Information Processing Societies (AFIPS) Records*, Houston, TX, November 1970, vol. 37, pp. 281–285.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[3] A. V. Kini, V. Veeraraghavan, N. Singhal, and S. Weber, "Smartgossip: An improved randomized broadcast protocol for sensor networks," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN)*, Nashville, TN, April 2006, pp. 210–217.

[4] A. V. Kini, N. Singhal, and S. Weber, "Performance of a WSN in the presence of channel variations and interference," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, NV, April 2008, pp. 2899–2904.

[5] A. V. Kini, N. Singhal, and S. Weber, "Broadcast capacity of a WSN employing power control and limited range dissemination," in *Proceedings of the 41th Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, March 2007, pp. 238–243.

[6] A. V. Kini, N. Singhal, and S. Weber, "Broadcast capacity of a WSN under communication and information coordination," in *Submitted to Elsevier Ad Hoc Networks*.

[7] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip–based ad hoc routing," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, New York, NY, June 2002, pp. 1707–1716.

[8] H. G. Landau and A. Rapoport, "Contribution to the mathematical theory of contagion and spread of information: I. spread through a thoroughly mixed population," *Bulletin of Mathematical Biophysics*, vol. 15, pp. 173–183, 1953.

[9] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom)*, Seattle, WA, August 1999, pp. 174–185.

[10] M-J. Lin and K. Marzullo, "Directional gossip: Gossip in a wide area network," in *Proceedings of the European Dependable Computing Conference (EDCC)*, Prague, Czech Republic, September 1999, pp. 364–379.

[11] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *Proceedings of the 1st ACM Workshop on Wireless Sensor Networks and Applications (WSNA)*, Atlanta, GA, September 2002, pp. 22–31.

[12] C. L. Barrett, S. J. Eidenbenz, L. Kroc, M. Marathe, and J. P. Smith, "Parametric probabilistic sensor network routing," in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, San Diego, CA, October 2003, pp. 122–131.

[13] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, "Randomized rumor spreading," in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*, Redondo Beach, CA, November 2000, pp. 565–574.

[14] P. Levis and D. Culler, "The firecracker protocol," in *Proceedings of the 11th ACM SIGOPS European Workshop*, Leuven, Belgium, September 2004.

[15] P. Levis, N.Patel, D. Culler, and Scott Shenker, "Trickle: A self-regulating algorithm for code maintenance and propagation," in *Proceedings of the 1st USENIX/ACM Symposium on Network Systems Design and Implementation (NSDI)*, San Francisco, CA, March 2004, pp. 15–28.

[16] J.W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems SenSys*, Baltimore, Maryland, November 2004, pp. 81–94.

[17] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1936–1949, December 1992.

[18] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in *Proceedings of the 43rd Annual Allerton Conference on Communications, Control and Computing*, Monticello, IL, November 2005, pp. 28–30.

[19] X. Lin and S. B. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC)*, San Diego, CA, December 2006, pp. 1258–1263.

[20] M.J. Neely, E. Modiano, and C. P. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proceedings of the 24th Annual Joint Conference of*

the IEEE Computer and Communications Societies (INFOCOM), Los Angeles, CA, March 2005, pp. 1723–1734.

[21] P. Chaporkar, A. Bhat, and S. Sarkar, "An adaptive strategy for maximizing throughput in MAC layer wireless multicast," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, Roppongi Hills, Tokyo, Japan, May 2004, pp. 256–267, ACM Press.

[22] P. Chaporkar and S. Sarkar, "Stochastic control techniques for throughput optimal wireless multicast," in *Proceedings of the 42nd IEEE Conference on Decision and Control (CDC)*, Maui, Hawaii, USA, December 2003, pp. 1598–1603.

[23] J. Polastre, R. Szewczyk, C. Sharp, and D. Culler, "The mote revolution: Low power wireless sensor network devices," in *Hot Chips 16: A symposium on High Performance Chips*, August 2004.

[24] W. N. Anderson and T. D. Morley, "Eigenvalues of the laplacian of a graph," *Linear and Multilinear Algebra*, vol. 18, no. 2, pp. 141–145, 1985.

[25] D. M. Cvetkovic, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Applications*, Academic Press, 1997.

[26] Miroslav Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[27] D. M. Cvetkovic, "Graph theoretical procedures in clustering discrete data," *Univ. Beograd, Publ. Elektrotehn. Fak.*, vol. 3, pp. 21–26, 1992.

[28] R. K. Ganti and M. Haenggi, "Regularity, interference, and capacity of large ad hoc networks," in *Proceedings of the 40th Asilomar Conference on Signals, Systems, and Computers (Asilomar)*, Pacific Grove, CA, October 2006, pp. 3–7.

[29] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan, "Bounds on average delays and queue size averages and variances ininput-queued cell-based switches," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*, April 2001, pp. 1095–1103.

[30] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[31] S. Weber, "Communication constraints and ad hoc scheduling," in *Telecommunications Modeling, Policy, and Tehcnology*, New York, NY, USA, 2008, vol. 44, pp. 275 – 289, Springer.

[32] J. Edmonds, "Paths, trees, and flowers," in *Canadian Journal of Mathematics*, 1965, vol. 17, pp. 449–267.

[33] G. Sharma, R. R. Mazumdar, and N. B. Shroff, “On the complexity of scheduling in wireless networks,” in *MobiCom ’06: Proceedings of the 12th annual international conference on Mobile computing and networking*, New York, NY, USA, 2006, pp. 227–238, ACM.

[34] L. Stockmeyer and V. Vaziarani, “NP completeness of some generalizations of the maximum matching problem,” *Information processing letters*, vol. 15, no. 1, pp. 14–19, 1982.

[35] A.J. Hoffman and J.B. Kruskal, “Integral boundary points of convex polyhedra,” in *Linear Inequalities and Related Systems; Annals of Mathematical Study (H.W. Kuhn, A.W. Tucker editions)*, Princeton, NJ, USA, 1956, vol. 38, pp. 223 – 246, Princeton University Press.

[36] I. Heller and C.B. Tompkins, “An extension of a theorem of Dantzig’s,” in *Linear Inequalities and Related Systems; Annals of Mathematical Study (H.W. Kuhn, A.W. Tucker editions)*, Princeton, NJ, USA, 1956, vol. 38, pp. 247 – 254, Princeton University Press.

[37] ILOG, “http://www.ilog.com/products/cplex/,” .

[38] D.J.A. Welsh, *Matroid Theory*, Academic Press, 1976.

[39] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, Prentice-Hall, Princeton, NJ, 1982.

[40] W. Cook, W. H. Cunningham, W. R. Pullerybank, and A. Schrijver, *Combinatorial Optimization*, Wiley-Intersciece Series in Discrete Mathematics and Optimization, Malden, MA, 1998.

[41] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algrothims*, MIT Press, 2001.

Appendix A. Publications

- **"Broadcast capacity of a WSN under communication and information coordination"**, *Ananth V. Kini, Nikhil Singhal, Steven Weber*, submitted to Elsevier Ad Hoc Networks.
- **"Performance of a WSN in the presence of channel variations and interference"**, *Ananth V. Kini, Nikhil Singhal, Steven Weber*, in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Las Vegas, NV, April 2008.
- **"Broadcast capacity of a WSN employing power control and limited range dissemination"**, *Ananth V. Kini, Nikhil Singhal, Steven Weber*, in Proceedings of the 41st Conference of Informations Sciences and Systems (CISS), Baltimore, MD, March 2007.
- **"SmartGossip : an improved randomized broadcast protocol for sensor networks"**, *Ananth V. Kini, Vilas Veeraraghavan, Nikhil Singhal, Steven Weber*, in Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN), NashVille, TN, April 2006.
- **"Analysis of gossip performance with copulas"**, *Steven Weber, Vilas Veeraraghavan, Ananth V. Kini, Nikhil Singhal*, in Proceedings of the 40th Conference of Informations Sciences and Systems (CISS), Princeton, NJ, April 2006.
- **"A new approximation for slotted buffered Aloha"**, *Steven Weber, Ananth V. Kini, Athina Petropulu*, in Proceedings of the 42nd Conference of Informations Sciences and Systems (CISS), Princeton, NJ, March 2008.
- **"Achievable throughput and service delay for imperfect cooperative retransmission MAC protocols"**, *Steven Weber, Ananth V. Kini, Athina Petropulu*, in Proceedings of the 3rd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks (PM2HW2N), Vancouver, BC, October 2008.

Vita

Ananth Kini was born in Bombay, Maharashtra, India. He received his B.S. in Electrical Engineering from the University of Bombay, India, in May 1998, and his M.S. in Telecommunications from the Department of Information Sciences at the University of Pittsburgh, Pittsburgh, PA, in May 2002. He joined the Department of Electrical and Computer Engineering at Drexel University in September 2003 and has been affiliated with the Data Networking Modeling Laboratory ever since. His research interests have spanned several areas of Computer Networking, including distributed algorithms for wireless networks, multicasting and broadcasting in wireless networks, transmission coordination for ad hoc networks. He has also been a teaching assistant for a wide variety of graduate and undergraduate level courses such as principles of computer networking, operating systems, modulation and coding, statistical analysis of engineering systems, embedded systems, intelligent systems, micro-controller systems. He has been awarded the outstanding Teaching Assistant award for the year 2003-2004.

His research has led to the publication of six conference papers, and two journal papers that are currently under review. He has served as a reviewer for the Conference of the IEEE Computer and Communications Societies (INFOCOM). He has also served as session chair at the 50th IEEE Global Telecommunications Conference (WCNC 2008) held in Las Vegas, NV, and as a student volunteer for ACM SIGCOMM 2005 held in Philadelphia.