A Comparative Analysis between Forwarding and Network Coding Techniques for Multihop Wireless Networks

Suranjit Paul

spaul2@connect.carleton.ca
Broadband Network Lab, Carleton University

Acknowledgements

First and foremost, I am grateful to my supervisor Dr. Thomas Kunz for his guidance and encouragement throughout my project.

I also express my sincere gratitude to Communication Research Centre Canada for supporting me with their funding.

Table of Contents

List of Tables

List of Figures

CHAPTER 1: INTRODUCTION

In recent years, wireless networks have attracted significant attention due to their potential applications in tactical networks. A wireless network consists of numerous devices that are equipped with processing, memory and wireless communication capabilities, and are linked via short-range ad hoc radio connections. There is no pre-installed infrastructure in this type of network but all communication is supported by multi-hop transmissions, where intermediate nodes relay packets between communicating parties. In the wireless network, each node in the network has limited energy resources. Reducing the number of transmissions required to broadcast messages to the whole network saves energy and reduces spectrum usage. Different routing-based and network coding-based protocols have been proposed to reduce the number of retransmissions. Broadcast communication is an important mechanism to communicate information in wireless networks. In addition, many routing and other network protocols for wireless ad-hoc networks need a broadcast mechanism to update their states and maintain information between nodes.

In the case of a routing-based approach, the optimal broadcast performance would see all nodes in the minimal connected dominating set (MCDS) re-broadcast a packet. Determining such an optimal (i.e., minimal) MCDS is NP-hard, and various approximation heuristics with known approximation ratios have been proposed. On the other hand, for coding-based protocols, a lower number of retransmissions can be obtained based on a linear program.

1.1 Project Objective

Many routing and network coding based techniques have been proposed so far for improved performance of multicast communications in packet networks. Recent research has also applied network coding to mobile ad hoc networks (MANET) for improved network throughput and robustness. How the network coding technique compares with routing-based approaches in a narrowband MANET environment and which approach provides us much better performance for a static wireless network is our main concern for this project. To explore this issue, we conducted a literature survey on routing-based and also network coding-based approaches first. Then we implemented the most suitable approaches (for both routing-based and network coding-based broadcasting) which determine the lower bounds on the number of required packet retransmissions. After that, two different efficient broadcast protocols (based on routing and also network coding) are simulated and the results collected for performance comparison. Throughout this project, we only considered static wireless network.

1.2 Organization of Project Report

The project report has been organized as follows. In Chapter 2, the results of a literature survey for routing-based and network coding-based approaches are summarized. Additionally, the factors for comparison are described in this section. We are interested in protocols that require a low number of packet transmissions at the MAC layer. In Chapter 3, the implemented MCDS algorithm is described, which approximates the lower bound on packet transmissions for any routing-based solution. A linear program that derives the lower bound for network coding is described in Chapter 4. Then in Chapter 5 we described XOR-based network coding techniques for two distributed algorithms, PDP and SMF. NS-2 simulation related parameters are mentioned in Chapter 6. In Chapter 7, we present the performance analysis for routing-based and network coding-based protocols and compare these results with their lower bounds. Finally, in Chapter 8 we summarize our analysis.

CHAPTER 2: BACKGROUND LITERATURE

2.1 Introduction to Mobile Ad Hoc Networks (MANET)

A MANET is an autonomous collection of mobile devices that communicate over relatively bandwidth-constrained wireless links. Since the nodes are potentially mobile, the network topology may change rapidly and unpredictably over time. The dynamic nature of the network topology increases the challenges of the design of ad hoc networks. The network is decentralized; where all network activity including discovering the topology and delivering messages must be executed by the nodes themselves. But in this project, we are only considering static wireless networks. In essence, a mobile network can be seen as a sequence of static snapshots, so this simplifies the analysis significantly.

2.2 Applications of Mobile Ad Hoc Networks (MANET)

The set of applications for MANETs is diverse, ranging from small, static networks to large-scale, mobile, highly dynamic networks and all these networks are constrained by power considerations. The design of network protocols for these networks is a complex issue. Regardless of the application, MANETs need efficient distributed algorithms to determine network organization, link scheduling, and routing. However, determining viable routing paths and delivering messages in a decentralized environment where network topology dynamically varies is not a trivial task. While the shortest path (based on a given cost function) from a source to a destination in a wired network is usually the optimal route, this idea is not easily extended to MANETs. Factors such as variable wireless link quality, propagation path loss, fading, multiuser interference, power expended, recovery from failure, and topological changes, become relevant issues. The network should be able to adaptively alter the routing paths to alleviate any of these effects. Moreover, in a military environment, preservation of security, latency, reliability, defence against intentional jamming, and recovery from failure are significant concerns. Military networks are designed to maintain a low probability of intercept and/or a low probability of detection. Hence, nodes prefer to radiate as little power as necessary and transmit as infrequently as possible, thus decreasing the probability of detection or interception. A lapse in any of these requirements may degrade the performance and dependability of the network. For this reason, determining the appropriate algorithm is most important for this type of wireless network.

2.3 Broadcast Routing Protocols (Packet Forwarding)

In *ad hoc networks*, nodes do not start out familiar with the topology of their networks; instead, they have to discover it. The basic idea is that a new node may announce its presence and should listen for announcements broadcast by its neighbours. Each node learns about nodes nearby and how to reach them, and may announce that it, too, can reach them.

Research on efficient broadcast support in mobile ad hoc networks has proceeded along two main approaches: deterministic and probabilistic. Deterministic approaches predetermine and select the neighboring nodes that forward the broadcast packet. On the other hand, probabilistic or gossiping-based approaches require each node to rebroadcast the packet to its neighbors with a given forwarding probability. The key challenge with these approaches is to tune the forwarding probability: keeping it as low as possible for maximum efficiency while maintaining it high enough so that all the nodes are able to receive the broadcast packets. But if the complete topology is known (feasible for static ad hoc networks), a good centralized approximation algorithm for constructing a small connected dominating set-based approach will yield very few transmissions to reach all nodes; otherwise, pruning-based solutions based on one or two hop topology information have to be considered.

In graph theory, the neighbors of a vertex are all the vertices which are connected to that vertex by a single edge. A dominating set (DS) for a graph is a set of vertices whose neighbors, along with themselves, constitute all the vertices in the graph. A connected dominating set (CDS) of a graph $G = (V,E)$ is a subset of nodes, S such that S is a dominating set of G and the sub-graph of G induced by S is also connected. The minimum Connected Dominating Set (MCDS) problem is to find a connected dominating set of minimum cardinality. Connected dominating sets are useful for routing in mobile ad-hoc networks and other network-related problems. But computing a minimum connected dominating set over a given graph is an *NP-complete* problem [1]. Since there are no polynomial-time algorithms for *NP-complete* problems, approximation algorithms are proposed to obtain near-optimal solutions.

In the following section, we explain several *centralized* algorithms for solving the minimum connected dominating set (MCDS) problem for static wireless network.

2.3.1 Existing MCDS Protocols: Exact Algorithms and Approximations

There are several centralized approximations and *exact* algorithms proposed in the literature to solve the minimum connected dominating set problem. All *exact* algorithms are at best only small improvements of the trivial $O\ (2^n)$ solution. The trivial solution requires checking every possible subset of nodes to determine whether this subset constitutes a minimum connected dominating set. [2] proposes an exact algorithm for the MCDS problem of an arbitrary graph with an improved runtime complexity of $O\ (1.9407^n)$, *relative to* the trivial $\Omega\ (2^n)$ algorithm. The algorithm makes use of some new domination rules and reduction rules and its analysis is based on the Measure and Conquer technique. But this algorithm is not practical for networks as small as even only 100 nodes as it will take a long time to find the minimum connected dominating set.

Guha and Khuller first proposed a two-stage greedy $(\ln \Delta + 3)$-approximation in [3] for MCDS in general graphs, where Δ is the maximum node degree in the graph. In the first step of this algorithm, a CDS is built from one node, then the search space for the next dominator(s) is restricted to the current set of dominatees and the CDS expands until there are no uncovered nodes left. All the possible dominators determined in the first phase are then connected through some intermediate nodes in the second phase.

A new efficient heuristic algorithm for the MCDS problem was proposed in [20]. The algorithm starts with a feasible solution containing all vertices of the graph. Then it reduces the size of the CDS by excluding some vertices using a greedy criterion. This algorithm is especially valuable in situations where setup time is costly because it maintains a feasible solution at any time during the computation. This algorithm provides a better approximation of $H(\Delta) + 2$ than that of Guha and Khuller's. Here, Δ is the maximum degree of the graph and $H(\Delta) = 1 + ½ + ... + 1/n$ is the harmonic function.

In [5], the authors proposed a new approximation algorithm based on Steiner trees, which produces an approximation solution within a factor of 6.8. This approximation algorithm can also be implemented in a distributed manner. This algorithm consists of two steps. In the first step, a maximal independent set is being constructed. In the second step, a 3-approximation for the ST-MSN (Steiner Tree with Minimum Number of Steiner Nodes: Given a unit disk graph G and a subset P of nodes, compute a Steiner tree for P with the minimum number of Steiner nodes) to interconnect the maximal independent set is determined. Note that the size of the optimal solution for the ST-MSN cannot exceed the size of the minimum connected dominating set since the latter can also interconnect the maximal independent set. A Steiner tree is defined as a subset of the vertices of a graph G which is a minimum-weight connected sub-graph of G that includes all the vertices. It is always a tree. Therefore, the ST-MSN has at most 3opt Steiner nodes in the second step. The resulting connected dominating set has a size bounded by 6.8opt.

On the other hand, in [6], another greedy algorithm called S-MIS (Steiner tree with Maximal Independent Set) was proposed with the help of Steiner trees that constructs a CDS within a factor of $5.8 + \ln 4$ from the optimal solution. This is also a two-step algorithm. In the first step, a MIS is constructed and in the second step a greedy approximation for the ST-MSN to interconnect the nodes in the MIS was employed. The resulting CDS has size bounded by $(5.8 + \ln 4)\ opt + 1.2$.

Since NP-hard problems cannot be solved in polynomial time, approximation algorithms are more efficient to use. On the other hand, exact algorithms provide the optimal solution; however their running time is very high even for small problem sizes, which is not practical. And since exact solutions are impractical in this case, the only possibilities left are approximations. In the past several years many approximation algorithms have been proposed for minimum connected dominating set problem. After reviewing all of these algorithms, we found that the algorithm proposed in [4] has better execution time than others. Also the proposed algorithm produces a MCDS of smaller size than others. In addition, this algorithm is less complex than others. Hence this one has been chosen to be implemented. In Chapter 3, we discuss the proposed algorithm in detail.

2.3.2 Simplified Multicast Forwarding (SMF)

Many broadcast algorithms besides simple flooding have been proposed so far for wireless network. These algorithms utilize neighborhood and/or history information to reduce redundant packet transmissions. Here we describe one such algorithm, SMF, which utilizes neighborhood information to reduce the required number of transmissions.

There are two main objectives of SMF. These objectives are:

1. Develop a specification framework for simple IP multicast packet forwarding on MANET interface types, including duplicate packet detection mechanisms.
2. Apply known efficient flooding or relay set mechanisms to SMF for further reducing contention and congestion in wireless multi-hop scenarios.

SMF is described detail in [7] [8]. The duplicate detection mechanism is used to remove and detect duplicate packets from both entering the interface forwarding process and from being delivered to upper layer applications. On the other hand, a basic multicast packet forwarding module, which can be an efficient flooding technique or a relay set mechanism, is flexible in its design and presently supports different flooding design optimizations like Simple flooding, Source-based Multi-Point Relay (S-MPR) flooding, and Non-Source Multi-Point Relay (NS-MPR) flooding.

The concept of "multipoint relaying" is used to reduce the number of *duplicate re-transmissions* while forwarding a broadcast packet. This technique restricts the number of re-transmitters to a small set of neighbor nodes, instead of all neighbors, as would be the case in flooding. This set is kept as small as possible by efficiently selecting the neighbors which cover (in terms of one-hop radio range) the same network region as the complete set of neighbors. This small subset of neighbors is called **multipoint relays** of a given network node. The technique of multipoint relays (or MPRs) provides an adequate solution to reduce the flooding of broadcast messages in the network, while attaining the same goal of transferring the message to every node in the network with high probability.

The information required to calculate the multipoint relays is the set of one-hop neighbors and the two-hop neighbors, i.e. the neighbors of the one-hop neighbors. To obtain the information about one-hop neighbors, most protocols use some form of HELLO messages that are sent locally by each node to declare its presence. In a mobile environment, these messages are sent periodically to get the most updated information. To obtain the information of two-hop neighbors, one solution may be that each node attaches the list of its own neighbors, while sending its HELLO messages. With this information, each node can independently calculate its one-hop and two-hop neighbor sets. Once a node has its one-hop and two-hop neighbor sets, it can select a minimum number of one-hop neighbors which *covers* all its two-hop neighbors. MPRs are dynamically selected by each node and selections are advertised and dynamically updated with hello messages.

Considering S-MPR as the forwarding module in the SMF, it will forward packets if and only if:

a. It receives a unique multicast packet from any of its bi-connected neighbors.
b. The neighbor from which the packet was received has selected the node as an MPR.

In the following we explain a basic algorithm for the S-MPR selection process which is described in [7]. Here, $N(n_0)$ and $N(N(n_0))$ indicate one-hop neighbors and two-hop neighbors of node n_0, respectively.

1. Start with an empty multipoint relay set $MPR(n_0)$
2. First select those one-hop neighbor nodes in $N(n_0)$ as multipoint relays which are the only neighbor of some node in $N(N(n_0))$, and add these one-hop neighbor nodes to the multipoint relay set $MPR(n_0)$
3. While there still exists some node in $N(N(n_0))$ which is not covered by the multipoint relay set $MPR(n_0)$:
    a. For each node in $N(n_0)$ which is not in $MPR(n_0)$, compute the number of nodes that it covers among the uncovered nodes in the set $N(N(n_0))$.
    b. Add that node of $N(n_0)$ in $MPR(n_0)$ for which this number is maximum.

2.3.3 Partial Dominant Pruning (PDP)

Partial Dominant Pruning is another algorithm which also utilizes the neighborhood information for reducing redundant packet transmissions. The dominant pruning (DP) algorithm is one of the promising approaches that utilize two-hop neighborhood information to reduce redundant transmissions. But DP also does not eliminate all redundant transmissions based on two-hop neighborhood information. Two algorithms, total dominant pruning (TDP) and partial dominant pruning (PDP) are proposed in [9]. Both algorithms utilize neighborhood information more effectively. In our study we have chosen PDP for comparison with the centralized MCDS approximation and Simplified Multicast Forwarding.

PDP enhances DP by eliminating the two-hop nodes advertised by a neighbor shared by both the sender and the receiver (forwarder). When a node v receives a packet from another node u, it selects a minimum number of forwarding nodes from the set $N(v)–N(u)$ that can cover all the nodes in the set $U=N(N(v))–N(u)–N(v)–N(N(u)\cap N(v))$. A node can obtain its one-hop and two-hop neighborhood information by periodically sending hello messages. Upon receiving the hello messages, each node updates its neighborhood information. Each forwarding node then again follows the same procedure to select its own forwarding nodes. The forwarding stops when all forwarders have received a packet at least once. The PDP algorithm is described below. Details of this algorithm are described in [9].

Step 1: Node v uses $N(N(v))$, $N(u)$, and $N(v)$ to obtain

$P = N(N(u)) \cap (N(v)$,

$U = N(N(v)) – N(u) – N(v) – P$, and

$B = N(v) – N(u)$.

Step 2: Node v then calls the selection process to determine the set of forwarding nodes, F.

Selection Process:

Step 1: Let $F(u, v) = [\ ]$ (empty list), $Z = \varnothing$ (empty set), and $K = U S_i$, where $S_i = N(v_i) \cap U$ for $v_i \varepsilon B$.

Step 2: Find the set S_i, whose size is maximum in K. (In case of a tie, the one with the smallest identification i is selected.)

Step 3: $F = F \mid\mid v_k$, $Z = Z \cup S_i$, $K = K - S_i$, and $S_j = S_j – S_i$ for all $S_j \in K$.

Step 4: If $Z = U$, exit; otherwise, goto Step 2.

2.4 Network Coding

Research in information theory discovered that routing alone is not sufficient to achieve maximum throughput in the general model of data networks. Network coding techniques have been proposed for improved performance for broadcast and multicast traffic. Network coding is a technique which looks beyond the traditional store-and-forward approach followed by routers in communication networks. Network coding is a generalization of routing in which nodes can generate output data by encoding previously received input data. Thus, network coding allows information to be "mixed" at a node. Ahlswede et al. in [10] first formally introduced the paradigm of network coding, where they also demonstrated its use in case of single-source multiple-sink network multicast in a wired network. Additional examples of networks are also presented in [10] where it is shown that network coding can improve the overall throughput of the network which can not otherwise be realized by the conventional store-and-forward approach. Network coding has drawn significant interest, especially for broadcast and multicast traffic. However, it is not obvious whether network coding further reduces the number of packet transmissions for random networks in the case of broadcasting.

This section explains several network coding techniques that were proposed for wireless networks. Since we are considering a static network of at most 100 nodes and the connectivity of all nodes are provided, techniques which utilize centralized information are suitable for our purpose. There are many distributed broadcast algorithms which use network coding techniques to deliver packets. To figure out whether network coding is advantageous for random networks, we use a technique, proposed in [11], which determines the lower bound on the number of required packet retransmissions for network coding by formulating this as an integer linear program and solving it for a range of randomly generated multi-hop wireless networks.

2.4.1 Network Coding in MANET

Many network coding techniques have been proposed so far for improved performance of multicast communications in packet networks. Recent research has also applied network coding to MANETs for improved network throughput and robustness. How the network coding technique compares with efficient broadcast in a narrowband MANET environment is not well understood, though much more of a practical concern. To explore this issue, we are conducting this survey to find out an appropriate network coding model which can be used to determine the lower bound on the number of required packet retransmissions for network coding. Later, two different efficient broadcast protocols based on network coding will be described and evaluated. The obtained lower bound will help us to determine how efficiently these protocols are performing.

2.4.2 Typical Network Coding Approaches

Network coding has drawn considerable attention in the protocol design for mobile ad hoc networks to improve the throughput for broadcast and multicast traffic. Although most research explores the performance of network coding using analytical models, there are also a few actual network protocols that use network coding. Some of these works show that network coding can improve throughput, others show the benefit of network coding in terms of packet delay, reliability, or file download times.

The approach proposed in [12] applies network coding to a deterministic broadcast protocol, resulting in a significant reduction in the number of transmissions in the network. To reduce the number of transmissions, two algorithms that rely only on local two-hop topology information and make use of opportunistic listening were proposed. The first algorithm is a simple XOR-based coding algorithm and the second one is a Reed-Solomon based coding algorithm. The simulation results show that the coding-based deterministic approach (nodes pre-select a few neighbors for rebroadcasting) outperforms the coding-based probabilistic approach (each node rebroadcasts a packet with a given probability).

CodeCast, a network coding based ad hoc multicast protocol which is well-suited especially for multimedia applications with low loss and low latency is proposed in [13]. The main component of CodeCast is random network coding, which is used to implement both localized loss recovery and path diversity transparently. The authors demonstrated through simulation that CodeCast achieves a near perfect packet delivery ratio while maintaining lower overhead than conventional multicast.

On the other hand, the authors of [14] present a theorem that unifies and generalizes Edmonds' theorem on routing (i.e. if all nodes other than the source are destinations, the cut bound, which is any cut separating the source from a destination, can be achieved by routing) and Ahlswede *et al.*'s theorem on network coding (i.e. the cut bound can be achieved by performing network coding) by classifying the links in a network into two categories: links entering relay nodes (Steiner edges), and links entering destinations (terminal edges). The authors show that the multicast capacity can be achieved by performing nontrivial network coding (mixing) only at links entering relay nodes. Links entering destinations will only require routing, which leads to a saving in the processing/implementation complexity.

In [15] the authors develop a network coding-based scheme for broadcast traffic in ad hoc networks and compare its performance against simpler solutions, based on flooding and deferred broadcast. They show that network coding is advantageous only in certain cases, such as dense networks, by comparing random linear network coding with two broadcast schemes under a range of scenarios. Their analysis also shows that network coding significantly outperforms other broadcasting schemes in terms of end-to-end packet loss probability and protocol overhead only for large neighbourhood sizes (i.e., more than 12 neighbours) and generation sizes smaller than or equal to three. A generation is defined as a collection of packets that can be allowed to be linearly combined. Dividing packets into generations decreases the decoding complexity and allows to decode data faster (and thus to empty the respective memory).

In [16], random linear network coding for time division duplexing channels for broadcasting is studied. The authors also study the mean time to complete the transmission of a block of packets to all receivers. Numerical results show that the coding scheme proposed in [16] outperforms a Round Robin broadcast scheme in a time division duplexing channel.

Reliability gain as a performance metric for random linear network coding in relay networks is studied in [17] where the authors show the expected number of channel uses per data bit received at the receiver. By analysis they show that random linear network coding provides limited performance gains in comparison to other protocols.

[18] seeks and provides some answers on how efficient broadcasting is over network coding in general and whether it is beneficial to use network coding over routing. It argues that for wireless networks in the 2D space, the asymptotic coding gain for a single-source broadcast is between 1.642 and 1.684 when both the area and the density of the network converge toward infinity. The paper also provides bounds of 1.432 and 2.035 for networks of the Euclidean space of dimension 3.

[19] investigates benefits in terms of energy efficiency that the use of network coding can offer for the problem of broadcasting over ad-hoc wireless networks. [19] also shows that network coding can result in a coding gain of 2 in ring networks and a coding gain of 1.3333 for grid networks and provides protocols that achieve this gain for such specific network topologies, using scenarios where all nodes are sources. Their work also indicates that there is a potential for significant benefits when deploying network coding over a practical wireless ad hoc network environment, especially when we are restricted to use low complexity decentralized algorithms.

Network coding enables more efficient, scalable and reliable wireless networks. After reviewing these papers, we can conclude that the potential advantages of network coding over routing include resource (e.g., bandwidth and power) efficiency, computational efficiency, and robustness to network dynamics. Besides, network coding can increase the possible network throughput and, in the multicast case, it can achieve the maximum data rate theoretically possible. In addition to maximizing throughput, network coding can also maximize the energy efficiency by reducing the number of transmissions required to deliver a message to the whole network. But in the general case it is not known by how much network coding is superior in comparison to routing-based techniques. In this project, we are addressing this issue.

2.5 Performance Comparison Factors for Packet Forwarding and Network Coding

In our study, we are comparing the performance of efficient packet forwarding approaches and network coding protocols to support broadcasting in static multi-hop wireless networks. The comparison is based on both lower bounds derived from analytical models and also the simulation results. More specifically, we compare the following:

- A lower bound for packet forwarding based broadcast protocols generated using the centralized MCDS heuristic proposed in [4].
- The number of PDP forwarders and SMF MPRs, where both PDP and SMF are representative packet forwarding broadcast protocols.
- A lower bound for network coding approaches generated using an integer linear program.
- The number of data packets forwarded by network coding protocols employing XOR coding as representative network-coding based broadcast protocols.

CHAPTER 3: IMPLEMENTED CENTRALIZED MCDS ALGORITHM

The MCDS algorithm that we implemented is proposed in [4] and it uses a heuristic to find the minimum connected dominating set. This algorithm is divided into three phases. In the first phase, a dominating set *D* is constructed, in the second phase a set of connectors are found which can connect nodes in *D*, with the help of a Steiner tree, and in the final phase, pruning is done, where the number of nodes in the MCDS is reduced to make it a near-optimal minimum connected dominating set. A *black node* is a node which is to be present in the Connected Dominating Set or is a Dominator. A *gray node* is a dominatee and a *blue node* is a connector which is to be present in the Connected Dominating Set. In the following section the algorithm for finding an approximation of the minimum connected dominating set is presented.

The algorithm proceeds in three stages.

Stage I

In Stage I, a dominating set is constructed which consists of the minimum number of nodes. This stage consists of the following steps:

(1) An arbitrary unique number say ID is assigned to each Node in the graph *G(V,E),*

(2) Each node is assigned *white color,*

(3) The node *u* with maximum degree is taken from *G(V,E)* and colored as *black*, i.e. it indicates that it is a Dominator,

(4) All the neighbor nodes of the node *u* are colored as *gray*,

(5) Repeat steps 3 and 4 till all the nodes in the graph *G(V,E)* are colored either as *black* or *gray*.

Stage II

In Stage II, a set of connectors *B* is found such that all the nodes in the dominating set are connected. Let a *black node* be a node in *D* and a *blue node* represent a node in *B*. A node in *B* is connected by at most *K* nodes in the graph *G(V, E).* The set of *blue nodes* with given *D* could be found using a Steiner tree. It is a tree interconnecting all the nodes in *D* by adding new nodes between them. The nodes that are in the Steiner tree but not in set *D* are called Steiner nodes. In the minimum connected dominating set, the number of Steiner nodes should be minimal. After this stage a CDS is constructed, which consists of *black* and *blue* nodes.

This involves the following steps:

(1) Select a *gray node* which is connected to the maximum (*K*) number of *black nodes,* set its color as *blue,*

(2) Check whether the Dominating Set *D* is connected or not,

(3) If *D* is connected stop,

(4) Else go to step *1* with *K-1* number of *Black nodes,*

Stage III

Stage III is a pruning stage. In this stage, redundant nodes are deleted from the CDS constructed in Stage II, to obtain the MCDS. Let the CDS constructed in the previous stage be set *F*.

The following steps are used for pruning:

(1) Select a minimum degree node *u* from *F*

(2) Check if *N[u]* is a subset of *N[1]* and *N[2]* and …. *N[n]* where *i* belongs to *F-{u}*

(3) If step 2 returns true then remove node *u* and goto step 1

(4) Otherwise do not remove node *u* and goto step 1

After implementing this algorithm, we developed a revised version where we kept a specific node (node 0) always in the final MCDS. Since node 0 is the source node in our simulation, this revised version allows us to directly compare the results.

Additionally, in the pruning stage for both versions, the original MCDS algorithm does not check the connectivity when a node is removed. We modified this stage by adding a connectivity checking function.

CHAPTER 4: NETWORK CODING LOWER BOUND

We are employing the technique proposed in [11] which is used to determine the lower bound on the number of required packet retransmissions for network coding. Here, the lower bound on the number of required packet retransmissions for network coding is determined by formulating this as an integer linear program and solving it for a range of randomly generated multi-hop wireless networks.

Next we are describing how the linear program to determine the lower bound for Network Coding (NC) is derived. In a first program, the number of packet transmissions per node is minimized, subject to the requirement that all nodes receive at least *N* packets by overhearing transmissions from all their neighbors. The intuition is that if the NC protocol is optimal, working on generations of size *N*, than all packet transmissions will be innovative for all neighbors. So, if each node receives at least *N* packets, with all of them being innovative, they can then decode and therefore receive all native packets in a generation. The resulting integer linear program is relatively straightforward. Let X_i be the number of packet transmissions of node *i* (for a given generation of size *N*), let *N(i)* be the set of one-hop neighbors of node *i*, then the integer linear program is:

$$min \sum X_i$$

Subject to:

$$\forall\, i\colon \sum_{j\,\in N(i)} X_j \;\geq N$$

$$X_0 \geq N$$

$$\forall\, i\colon X_i \;\geq 0, X_i \text{ is integer}$$

The objective function indicates that the interest is in a lower bound on packet transmissions. The first set of constraints ensures that each node *i* receives at least *N* packets by summing up all the packet transmissions of its neighbors, and the second constraint models the fact that in all experiments node *0* is the source node. Additionally, all X_i are greater than or equal to zero and integer variables.

But the resulting lower bound is not realistic since it does not ensure that packets flow from a specific sender. The second linear program corrects this by adding flow constraints: there is a flow from the sender to each receiver; these flows are subject to the typical flow balance constraints. Here typical flow balance constraints mean that the amount of flow on an edge cannot exceed the capacity of the edge and a flow must satisfy the restriction that the amount of flow into a node equals the amount of flow out of it, except when it is a source, which has more outgoing flow, or sink, which has more incoming flow. Flow variables $F_{i,j}(d)$ are introduced which capture the data flow from source *0* over link *i,j* destined to node *d*. In a broadcast scenario, all nodes are receivers, but to simplify the formulation, it is assumed that node *0*, the implied source, trivially receives all packets and therefore $d \neq 0$. The revised second linear program is as follows: The first constraint enforces that the flow over existing links *i,j* does not exceed the number of packets physically transmitted by the head of the link. The second set of constraints captures the flow balance constraints: for source node *0*, it has to generate *N* more packets destined for *d* flowing out of it then it (potentially) receives from its own neighbors. If node *i* is the destination, it consumes *N* packets. Otherwise, a node passes on all received packets over its outgoing links. This does not constraint the solution to a packet forwarding solution: the same physical packet (whose transmission is being modeled by the X_i variables) can be used to pass data on to multiple destinations *d*. The resulting integer linear program is:

$$min \sum X_i$$

Subject to:

$$\forall i, d: j \in N(i):\ F_{i,j}(d) \leq\ X_i$$

$$\forall\ i, d: \sum_{j\ \in N(i)} F_{i,j}(d) - \sum_{l\ \in N(i)} F_{l,i}(d) = \begin{cases} N\ for\ i = 0 \\ -N\ for\ i = d \\ 0\ otherwise \end{cases}$$

$$\forall\ i: X_i\ \geq 0, X_i\ is\ integer$$

This linear program still has one shortcoming. The flow balance constraint is applied on flows. Ensuring that the sum of outgoing flows to a specific destination equals *N* for the source node undercounts the physical transmissions required to achieve this in a broadcast channel. For example, in a ring network, node *0* is modeled as transmitting only *N/2* packets physically, as the transmission of *N/2* packets to its left neighbor and *N/2* packets to its right neighbor for each destination both independently can be satisfied by $X_0 = N/2$. To correctly capture the broadcast nature of the wireless media, a dummy node is introduced where node *i* will transmit its packets to its dummy node ī first, and the dummy node will then forward the packets to all neighbors of *i*. More formally, the final integer linear program is therefore:

$$min \sum X_i$$

Subject to:

$$\forall i, d: \; F_{i,\bar{i}}(d) \leq \; X_i$$

$$\forall \; i, d: F_{i,\bar{i}}(d) - \sum_{j \,\in N(i)} F_{\bar{j},i}(d) = \begin{cases} N \; for \; i = 0 \\ -N \; for \; i = d \\ 0 \; otherwise \end{cases}$$

$$\forall \; i, d: \sum_{j \,\in N(i)} F_{\bar{i},j}(d) - \; F_{i,\bar{i}}(d) = 0$$

$$\forall \; i: X_i \; \geq 0, X_i \; is \; integer$$

Here, the first constraint limits the flows out of a real node *i* (which are all destined to the nodes' dummy node ī) to the number of physical packet transmissions of that node. The second set of constraints imposes the flow balance condition: if the node is the source node, it generates *N* more packets than potentially received from the dummy nodes of its neighbors. If the node is a destination, it consumes *N* more packets than forwarded to its dummy node. For all other nodes, it passes on all received packets. The third set of constraints expresses flow balance constraints on the newly introduced dummy nodes: as these are neither sources nor destinations, their flow balance is always zero. Finally, the last set of constraints enforces that the number of physical packet transmissions by each node are always a positive integer or zero.

CHAPTER 5: NETWORK CODING PROTOCOLS

This chapter describes the XOR-based coding approach [22] that we used in PDP and SMF to compare their performances with the lower bound for networking coding.

Consider the network in Figure 1, where source S_1 wants to deliver the stream of messages a_i to both R_1 and R_2, and source S_2 wants to send the stream of messages b_i to the same two receivers. Assume all links have a capacity of one message per unit of time. If routers only forward the messages they receive, the middle link will be a bottleneck, which for every time unit, can either deliver a_i to R_1 or b_i to R_2. In contrast, if the router feeding the middle link XORs the two messages and sends $a_i \oplus b_i$ (or any linear combination of a^i and b_i), as shown in the figure, both receivers obtain two messages in every time unit. Thus network coding, i.e., allowing the routers to mix the bits in forwarded messages, can increase network throughput.

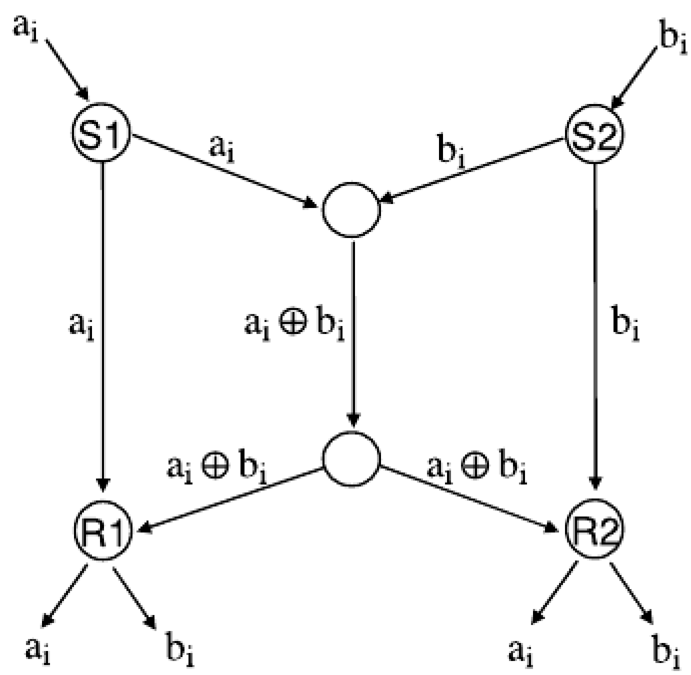


Figure 1: A simple scenario showing how network coding improves throughput. All links have a capacity of one message per unit of time. By sending the XOR of a_i and b_i on the middle link, we can deliver two messages per unit of time to both receivers.

We use XOR-based network coding approach in our PDP and SMF approximations and evaluate performances for both fixed size and fixed density network.

For XOR-based network coding approach, at first we consider only one source but we did not get enough coding opportunities. For one source, we achieved only 4 to 5% coding opportunities which are not enough. Later we considered four different sources and generate simulation results for all scenarios. In that case, we achieve at most 15% coding opportunities. In our study, we always kept four nodes; 0, 1, 2, and 3 as our sources for XOR-based network coding. Since all scenarios have been generated randomly, it does not matter which nodes are sources.

The linear program for network coding assumes a single sender. To compare XOR-based network coding with the lower bound, we calculate the required number of forwarders per source and then compare them with analytically derived lower bound.

CHAPTER 6: SIMULATION SETUP

All parameters that have been used for simulation are briefly described in this section. These are the parameters we used to generate simulation results to evaluate both the forwarding and also the network coding techniques.

Implementation Environment

Operating System: Linux Ubuntu 9.04

NS2 Version

Ns-allinone-2.29

Area: We used both fixed area and fixed density network in our analysis for both packet forwarding and network coding. For the fixed area network, we consider an area of 500x500 square-meters and then increase the number of nodes. On the other hand, for fixed density networks, we generated scenarios for different nodes but kept the density equal, that is, the ratio between the network areas and the number of nodes is kept equal. Below is the table (Table 1) that shows the number of nodes and corresponding areas.

No. of Nodes	Network Area
10	346x346
20	490x490
30	600x600
40	693x693
50	774x774
60	848x848
70	916x916
80	979x979
90	1039x1039
100	1096x1096

Table 1: Number of nodes and corresponding network size for fixed density network

Number of Nodes: at most 100 nodes (Starting from 10 nodes, increased by 10). Here, we are considering a small static wireless network of at most 100 nodes.

Simulation Duration: 100s.

For our simulation, 100 seconds is enough to broadcast all packets into the whole network. We start sending data packets in the 51^{st} second and in total we are sending 10 packets, two packets per second.

Number of Sources: 1 (for Packet Forwarding), 4 (for Network Coding).

We consider only one source, node ‘0’ for simulating packet forwarding algorithms (PDP and SMF). On the other hand, in our simulation for network coding approaches (PDP/XOR and SMF/XOR), we consider four sources, node ‘0’, ‘1’, ‘2’, and ‘3’. Here, we consider four sources because having only one or two sources for network coding does not generate enough coding opportunities. Also, increasing the number of sources further (to six or eight, for example) does not give us better opportunities too. For this reason, we choose four sources. Besides, since the scenario generation is done randomly, we can choose any four sources from the total number of sources. In our study, we choose nodes ‘0’, ‘1’, ‘2’, and ‘3’. Afterwards, we calculate the required number of forwarders per source to compare with the lower bound.

Protocol Control Parameter:

Hello packets are sent out to discover and maintain neighbor relationships. So, we will periodically send Hello Packets.

Hello packets sending Interval: 5 seconds;

Hello packets are used to find the MPRs or Forwarding nodes. We need to send a few Hello packets (>2) to first learn about the one-hop and two-hop neighborhood, then determine the MPR or Forwarder set, and, where necessary, propagate this information to the selected neighbors.

Radio Range: 250 meters.

For wireless simulation, a radio range of 250 meters is the default radio transmission range and are based on IEEE 802.11 Wifi transmission ranges. The default values for RXThresh_ and CSThresh_ have been used in our simulation. RXThresh_ is the reception threshold. If the received signal strength is greater than this threshold, the packet can be successfully received. And CSThresh_ is the carrier sensing threshold. If the received signal strength is greater than this threshold, the packet transmission can be sensed. However, the packet cannot be decoded unless the signal strength is greater than RXThresh_.

Size of Data Packets: 256 Bytes.

A packet size of 256B will be used for our simulation. Since delay is not our concern in this study, size of the data packets does not impact our results.

Number of Data Packets: 10 packets.

10 packets send from the source at the rate of 2 packets per second. Since a single packet could get lost, we send 10 packets in total.

Packet Delivery Ratio (PDR):

The simulation results are all presumably for scenarios where nearly all nodes receive nearly all packets, i.e., the Packet Delivery Ratio are close to 100%.

Time to start sending Data Packets: 51 seconds.

The source node begins sending data after approximately a 51 second scenario start-up time. Start-up time can be defined as the time overhead required before the actual simulation can start in ns. Since Hello packets will be periodically sent each 5 seconds, 50 seconds allows for 10 rounds of hello message exchange, allowing nodes to obtain a stable and complete view of its 1- and 2-hop neighborhood.

Jitter: 20ms.

A Jitter value of 20ms is used in our simulations. In order to prevent nodes in a wireless network from simultaneous transmission, whilst retaining the wireless network characteristic of maximum node autonomy, a randomization of the transmission time of packets by nodes, named as jitter, is employed.

Data Rate: 2.048 kbps.

CBR data rate is 2.048kbps. But it is not used for CBR packet generation (Commented out in the packet generation file, *interval* value is used instead).

CHAPTER 7: RESULTS AND COMPARATIVE ANALYSIS

In this chapter, we discuss the forwarding and network coding lower bounds along with simulation results of two distributed approximations, SMF and PDP. We statistically analyzed these results. Using T-tests, we evaluate which technique performs better for what scenarios.

We analyzed and compared results for two different types of networks; one with fixed network size and the other with fixed density. We compare the performance of our considered algorithms for both cases. We start with 10 nodes in the network and increased by 10 until we reach 100 nodes.

In the following section we compare the performance of different packet forwarding techniques for fixed network size. In this project, our performance criterion is the total number of transmissions required to send data to all nodes in the network. A lower, ideally minimal number of transmissions consume fewer resources like bandwidth or energy.

7.1 Performance Comparison of Packet Forwarding Techniques for Fixed Size Network

We generated results for our implemented centralized MCDS algorithm, and also obtained results for two distributed algorithms after simulating them in NS2. Results generated for different sized networks start from 10 nodes to at most 100 nodes. For each case, 50 different scenarios have been generated and evaluated. Moreover, we generated scenarios for a sparse and also a dense network and evaluated the protocol performance.

We implemented two different versions of our centralized MCDS algorithm. In the first version, the algorithm chooses its MCDS without any constrained. In the second version we have modified the code so that node ‘0’ is always in the MCDS. We have a fixed node (node ‘0’) as a sender in our NS2 simulations for distributed algorithms, SMF and PDP. Here, we are considering only one source. In our constrained version of MCDS algorithm, we always kept node ‘0’ in the MCDS so that both of the distributed approximations, PDP and SMF, can be compared directly.

7.1.1 Comparison among Average Number of Forwarding Nodes

In total, we have generated 500 scenarios to compare both distributed algorithms to the centralized approximation. We generated results for different sized networks starting from 10 nodes to at most 100 nodes. Based on the results collected for all 500 scenarios (50 scenarios for each case), the average number of *forwarding nodes* required to cover all nodes in the network are shown in the following table (Table 2) for both the *unconstrained* and *constrained* MCDS version along with PDP and SMF.

Table 2: Average number of *forwarding nodes* for all algorithms (50 scenarios for each network size)

No. of Nodes in Network	Average # of nodes in MCDS (*unconstrained*)	Average # of nodes in MCDS (*constrained*)	Average # of Active *Forwarders* for PDP	Average # of Active *MPRs* for SMF
10	2.52	3.14	3.30	3.30
20	2.88	3.52	3.82	3.82
30	3.26	3.68	4.54	4.44
40	3.70	3.88	4.60	4.54
50	3.90	3.96	4.78	4.72
60	4.04	4.12	4.82	4.86
70	4.16	4.40	5.20	5.20
80	3.90	4.38	4.94	4.88
90	4.32	4.40	5.12	5.10
100	4.38	4.64	5.48	5.62

If we observe Table 2, we can see that the *unconstrained* MCDS algorithm provides the lower bound for all cases. The *constrained* version of our MCDS algorithm, where we ensure that *node ‘0’* is always in the resulting MCDS, also provides a smaller number of re-broadcasting nodes compared to the two distributed algorithms. In the following section we will test whether the differences between these approaches are statistically significant or not.

To test whether the differences in the average means for different algorithms are statistically significant or not, we conducted parametric test. The parametric test called *t-test* is used for conducting statistically significant tests in the testing of hypotheses. It is based on Student's t statistic.

7.1.2 T-test Analysis between Unconstrained MCDS and Constrained MCDS

Following is a snapshot of the result that we have generated after running the analysis tool *t-test: two-sample assuming unequal variances* in Microsoft Excel for a network with 10 nodes.

Table 3: t-test Result for *unconstrained* and *constrained* versions of centralized MCDS approximation for the network with 10 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	U. MCDS	*C. MCDS*
Mean	2.52	3.14
Variance	0.70367347	0.735102
Observations	50	50
Hypothesized Mean Difference	0	
df	98	
t Stat	-3.65493934	
P(T<=t) one-tail	0.00020786	
t Critical one-tail	1.66055122	
P(T<=t) two-tail	0.00041572	
t Critical two-tail	1.98446745	

From the above table we can find the *t-statistics* value, *p* value along with other values. We will consider the *P(T<=t) two-tail* value to conclude whether the difference in this case is statistically significant or not.

In this case, the calculated *p* value (0.00041572) is less than the *level of significance (0.05)*. This indicates that, for a network with 10 nodes, there is evidence of statistically significant difference between the two means of *unconstrained MCDS* and *constrained MCDS.* More specifically, the *unconstrained* version will determine an MCDS size that is smaller than the *constrained* version, and the difference in performance is not due to random effects.

As for another example, in Table 4, where t-test results are shown for a network with 60 nodes, the calculated *p* value (0.528562131) is greater than the *level of significance (0.05)*. So, there is no evidence of statistically significant difference between the two versions of the MCDS algorithms which means that the different MCDS sizes determined by the *unconstrained* and *constrained* versions are not statistically significant for a network with 60 nodes.

Table 4: t-test Result for *unconstrained* and *constrained* versions of centralized MCDS algorithm for the network with 60 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	U. MCDS	*C. MCDS*
Mean	4.04	4.12
Variance	0.406530612	0.393469388
Observations	50	50
Hypothesized Mean Difference	0	
df	98	
t Stat	-0.632455532	
P(T<=t) one-tail	0.264281066	
t Critical one-tail	1.660551217	
P(T<=t) two-tail	0.528562131	
t Critical two-tail	1.984467455	

We performed t-tests to find out whether the difference between the means for both versions of the MCDS algorithms is statistically significant or not.

After our experiments and statistical analysis on 50 scenarios for each of the different network sizes (starting from 10 nodes to 100 nodes), we find when there is a small number of nodes in the network (in our case, 10 to 30 nodes), there is evidence of statistically significant difference between the means of these two versions of MCDS algorithm, where the *unconstrained* version always has a smaller MCDS size than the *constrained* version.

But when the number of nodes increase in the network (up to 100 nodes), the difference becomes statistically insignificant, which means both versions have a similar MCDS size on average. Table 5 shows the t-test result for the network with 100 nodes.

Table 5: t-test Result for *unconstrained* and *constrained* versions of centralized MCDS algorithm for the network with 100 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	U. MCDS	*C. MCDS*
Mean	4.38	4.64
Variance	0.607755102	0.520816327
Observations	50	50
Hypothesized Mean Difference	0	
df	97	
t Stat	-1.730588544	
P(T<=t) one-tail	0.043352612	
t Critical one-tail	1.66071461	
P(T<=t) two-tail	0.086705224	
t Critical two-tail	1.984723186	

Hence, in our study, for a network with a small number of nodes (10 - 30), *unconstrained* MCDS achieves a smaller average number of nodes in the MCDS than the *constrained* version, i.e. the difference of these two means is statistically significant. But when there are more nodes in the network (40 to 100), both versions compute almost the same number of nodes in the MCDS; i.e. the difference is statistically insignificant.

7.1.3 T-test Analysis between PDP and SMF

We also analyzed t-test results to find out whether the difference in the means of SMF and PDP is statistically significant or not.

After simulation, we find that for all network sizes; whether we have a small or large numbers of nodes (up to 100) in the network, the difference between the total numbers of active forwarding nodes for both algorithms are statistically insignificant.

In the following tables, we mention t-test results for two networks. Table 6 shows the t-test result for a network with 20 nodes and Table 7 shows the result for 70 nodes.

Table 6: t-test result for PDP and SMF for the network with 20 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	PDP	*SMF*
Mean	3.82	3.82
Variance	1.252653061	1.252653061
Observations	50	50
Hypothesized Mean Difference	0	
df	98	
t Stat	0	
P(T<=t) one-tail	0.5	
t Critical one-tail	1.660551217	
P(T<=t) two-tail	1	
t Critical two-tail	1.984467455	

Table 7: t-test result for PDP and SMF for the network with 70 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	PDP	*SMF*
Mean	5.2	5.2
Variance	1.591836735	1.795918367
Observations	50	50
Hypothesized Mean Difference	0	
df	98	
t Stat	0	
P(T<=t) one-tail	0.5	
t Critical one-tail	1.660551217	
P(T<=t) two-tail	1	
t Critical two-tail	1.984467455	

Here for both cases, the *p* value is greater than the *level of significance*. So we can conclude that the difference between the average numbers of active *forwarding nodes* is statistically insignificant.

We obtain similar results for all networks (10 to 100 nodes) in our tests. In all cases, the difference between their total numbers of *forwarding nodes* is always statistically insignificant.

7.1.4 T-test Analysis between MCDS and PDP

Similarly to the previous t-tests, we conducted t-tests between both versions of MCDS (*unconstrained* and *constrained*) and PDP for all ten different network sizes.

In our study, the t-test results show that, for small and big networks, there is always a statistically significant difference between the means of the *unconstrained* MCDS and PDP, where *unconstrained* MCDS always results in a smaller MCDS size than the number of active forwarders in PDP.

Table 8 shows a snapshot of the t-test result between *unconstrained* MCDS and PDP for a network of 50 nodes.

Table 8: t-test Result for *unconstrained* MCDS and PDP for the network with 50 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	U. MCDS	*PDP*
Mean	3.9	4.78
Variance	0.62244898	2.215918367
Observations	50	50
Hypothesized Mean Difference	0	
df	75	
t Stat	-3.693459773	
P(T<=t) one-tail	0.000208838	
t Critical one-tail	1.665425373	
P(T<=t) two-tail	0.000417676	
t Critical two-tail	1.992102154	

Here, *unconstrained* MCDS always has a statistically significant smaller MCDS size than the number of active forwarders in PDP. This is true for any network with 10 to 100 nodes.

Additionally, when we performed t-tests for *constrained* MCDS and PDP, we found different results for smaller sized networks. When we have 10~20 nodes in the network, t-test results show that there is no statistically significant difference between the means. That is, for any network with 10 or 20 nodes, the *constrained* MCDS will determine an MCDS size that is similar to the number of active forwarding nodes in PDP.

T-test results between *constrained* MCDS and PDP are shown in Table 9 for a network of 20 nodes.

Table 9: t-test result for *constrained* MCDS and PDP for the network with 20 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	C. MCDS	*PDP*
Mean	3.52	3.82
Variance	0.417959184	1.252653061
Observations	50	50
Hypothesized Mean Difference	0	
df	78	
t Stat	-1.641226145	
P(T<=t) one-tail	0.05238847	
t Critical one-tail	1.664624645	
P(T<=t) two-tail	0.104776939	
t Critical two-tail	1.990847069	

On the other hand, when there are more nodes (more than 20) in the network, the t-test results between *constrained* MCDS and PDP indicate that there is a statistically significant difference between the means; which indicates that the *constrained* MCDS will determine an MCDS size that is smaller than the number of active forwarding nodes in PDP, and the difference in performance is not due to random effects.

Table 10 shows t-test results between *constrained* MCDS and PDP for a network with 80 nodes.

Table 10: t-test result for *constrained* MCDS and PDP for the network with 80 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	C. MCDS	*PDP*
Mean	4.38	4.94
Variance	0.444489796	1.771836735
Observations	50	50
Hypothesized Mean Difference	0	
df	72	
t Stat	-2.659843942	
P(T<=t) one-tail	0.004815071	
t Critical one-tail	1.666293696	
P(T<=t) two-tail	0.009630142	
t Critical two-tail	1.993463567	

7.1.5 T-test Analysis between MCDS and SMF

In this sub-section, we discuss the t-test results conducted between MCDS (both versions) and SMF.

As with the comparison between MCDS and PDP, we obtained similar results for all different networks while conducting t-tests between *unconstrained* MCDS and SMF. The difference in average means between these two algorithms is always statistically significant; with the *unconstrained* MCDS always having a smaller MCDS size than the number of active MPRs of SMF.

Table 11 shows the result of the t-test of a network of 50 nodes between *unconstrained* MCDS and SMF.

Table 11: t-test Result for *unconstrained* MCDS and SMF for the network with 50 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	U. MCDS	*SMF*
Mean	3.9	4.72
Variance	0.62244898	2.124081633
Observations	50	50
Hypothesized Mean Difference	0	
df	75	
t Stat	-3.498699417	
P(T<=t) one-tail	0.000395018	
t Critical one-tail	1.665425373	
P(T<=t) two-tail	0.000790036	
t Critical two-tail	1.992102154	

In addition to that, the t-test between *constrained* MCDS and SMF shows similar results as a t-test between *constrained* MCDS and PDP. There is no statistically significant difference between means for networks with 10 or 20 nodes. That is, both of these techniques have a similar number of re-transmitting nodes.

T-test results between *constrained* MCDS and SMF are shown in Table 12 for a network of 20 nodes (no significant difference between means).

Table 12: t-test result for *constrained* MCDS and SMF for the network with 20 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	C. MCDS	*SMF*
Mean	3.52	3.82
Variance	0.417959184	1.252653061
Observations	50	50
Hypothesized Mean Difference	0	
df	78	
t Stat	-1.641226145	
P(T<=t) one-tail	0.05238847	
t Critical one-tail	1.664624645	
P(T<=t) two-tail	0.104776939	
t Critical two-tail	1.990847069	

But when we add more nodes (more than 20) in the network, we find statistically significant differences between the average mean of *constrained* MCDS and the number of active MPRs for SMF, where the average mean of *constrained* MCDS is much smaller than the number of active MPRs.

The t-test results between *constrained* MCDS and SMF for a network of 80 nodes are shown below (statistically significant difference).

Table 13: t-test result for *constrained* MCDS and SMF for the network with 80 nodes

t-Test: Two-Sample Assuming Unequal Variances (50 Scenarios)		
	C. MCDS	*SMF*
Mean	4.38	4.88
Variance	0.444489796	1.413877551
Observations	50	50
Hypothesized Mean Difference	0	
df	77	
t Stat	-2.593517743	
P(T<=t) one-tail	0.00568305	
t Critical one-tail	1.664884537	
P(T<=t) two-tail	0.011366099	
t Critical two-tail	1.991254395	

7.1.6 Comparison of Confidence Intervals

The sample mean is a point estimate of the population mean, i.e. it is a single value which we use to represent the population mean. However, the sample mean varies in repeated samples from the population and thus we need to assess (probabilistically) how close the sample mean is to the population mean. A *confidence interval* is a range of values that describes the uncertainty surrounding an estimate. We indicate a *confidence interval* by its endpoints. A 95% *confidence interval* is the interval that we are 95% certain that it contains the true population value as it might be estimated from a much larger study.

The following figures (Figure 2 to 4) shows the average number of nodes in MCDS (for both versions), *forwarders* (for PDP) and *MPRs* (for SMF) for different sized networks along with their calculated *confidence intervals*. These figures give us an idea about the accuracy of the average number of forwarding nodes for all the concerned algorithms for some networks (Here Figure 2 shows 20 nodes network, Figure 3 and 4 show 60 and 100 nodes networks respectively).

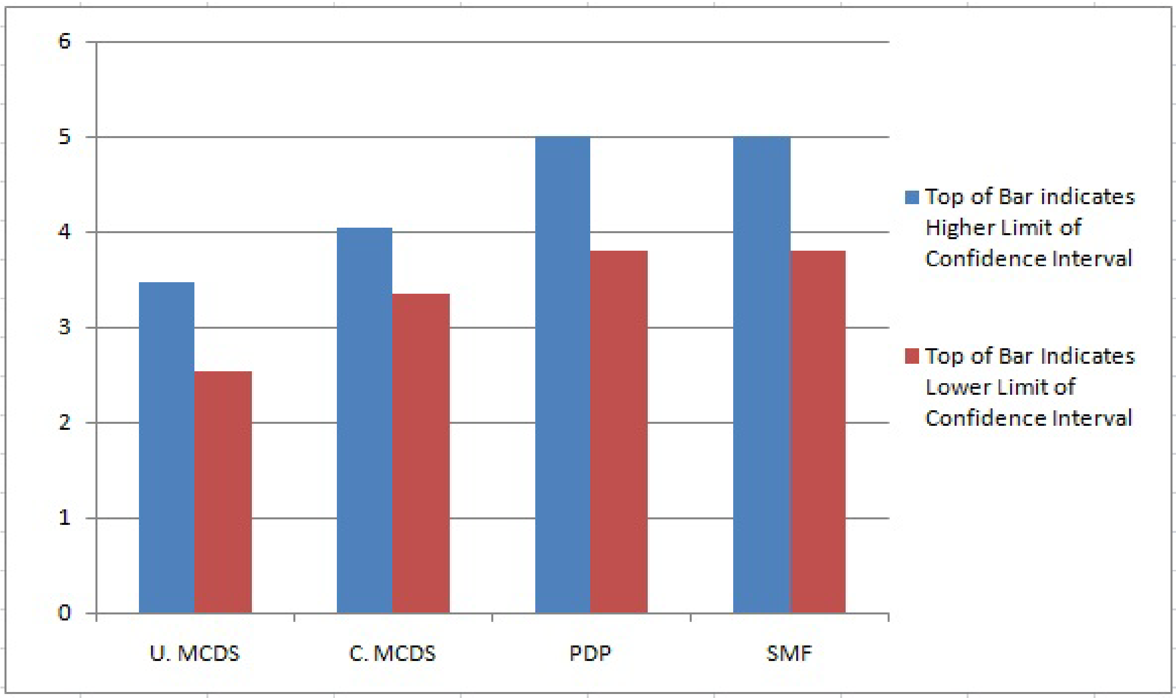


Figure 2: Average MCDS size, forwarders (PDP) and MPRs (SMF) with their *confidence intervals* for a network with 20 nodes

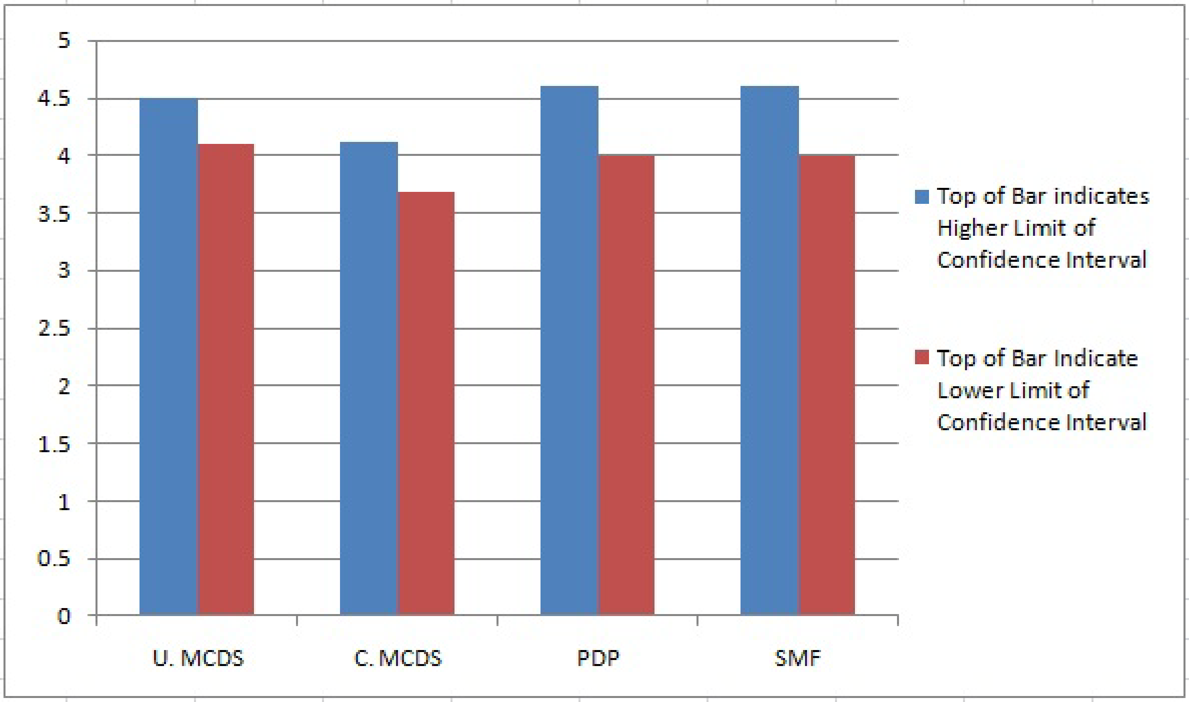


Figure 3: Average MCDS size, forwarders (PDP) and MPRs (SMF) with their *confidence intervals* for the network with 60 nodes

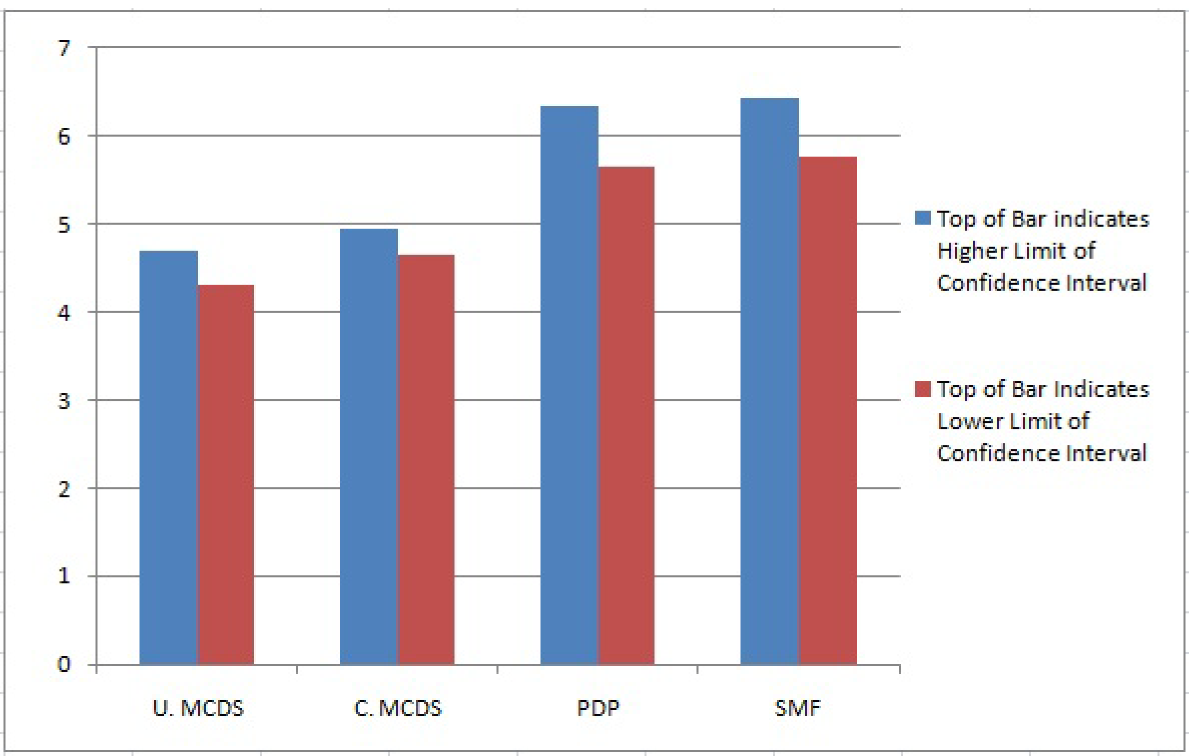


Figure 4: Average MCDS nodes, forwarders (PDP) and MPRs (SMF) with their *confidence intervals* for the network with 100 nodes

7.1.7 Impact of Dense and Sparse Networks

We also analyzed scenarios for some dense and sparse networks. For this purpose, we considered two different network areas. The first one is a dense network, i.e. a network with an area of 350 by 350 square-meters where we will keep the same number of nodes (10 to 100) and analyze the performance for these protocols. For the second scenario, we consider a sparse network, i.e. a network with an area of 750 by 750 square-meters and conduct the same performance analysis. We generated 100 scenarios for both networks and conducted statistical analysis for different network sizes (10 to 100 nodes).

T-test result analysis for a dense network (350 by 350 square-meters):

For all network sizes, we conducted t-tests to evaluate the performances. We always find statistically significant differences between the means of *unconstrained* MCDS compared to *constrained* MCDS, PDP and SMF.

Additionally, we find no evidence of statistically significant difference between the means of PDP and SMF. Their means are similar in this dense network.

On the other hand, when there are 10 to 30 nodes in the network, t-test results for *constrained* MCDS vs. PDP as well as *constrained* MCDS vs. SMF show no evidence of statistically significant difference between the means. But when we added more nodes in the network (in our cases, more than 30 nodes), there exists a statistically significant difference between their means.

T-test result analysis for a sparse network (750 by 750 square-meters):

Similarly, we generated scenarios for the sparse network and performed t-test analysis for different network sizes. Since this is a network with a bigger area, we could not randomly generate a connected scenario with 10 nodes only. But for all other network sizes (20 to 100 nodes), t-test results indicate that there is no evidence of statistically significant differences between the means of *unconstrained* MCDS and *constrained* MCDS; and also between PDP and SMF.

On the other hand, t-test analysis between *unconstrained* MCDS and PDP, *unconstrained* MCDS and SMF, *constrained* MCDS and PDP, and *constrained* MCDS and SMF show that the differences between their means are always statistically significant when the number of nodes in the network is more than 20. But when there are only 20 nodes in the network (this is the smallest network we have tested), we did not find any evidence of statistically significant difference between the means for the t-tests mentioned above.

The following figures (Figures 5 to 8) shows the overall comparison of average MCDS nodes (*unconstrained* and *constrained* versions), forwarders and MPRs for 3 different network size and therefore network densities.

If we observe each figure carefully, we can see that as the network area increases, the number of re-broadcasting nodes increases. But for the distributed algorithms (PDP and SMF), the total number of re-broadcasting nodes increases more rapidly in comparison to the centralized algorithm as the number of nodes in the network increases. That means, for sparse networks, our evaluated distributed algorithms perform worse than the centralized MCDS algorithm.

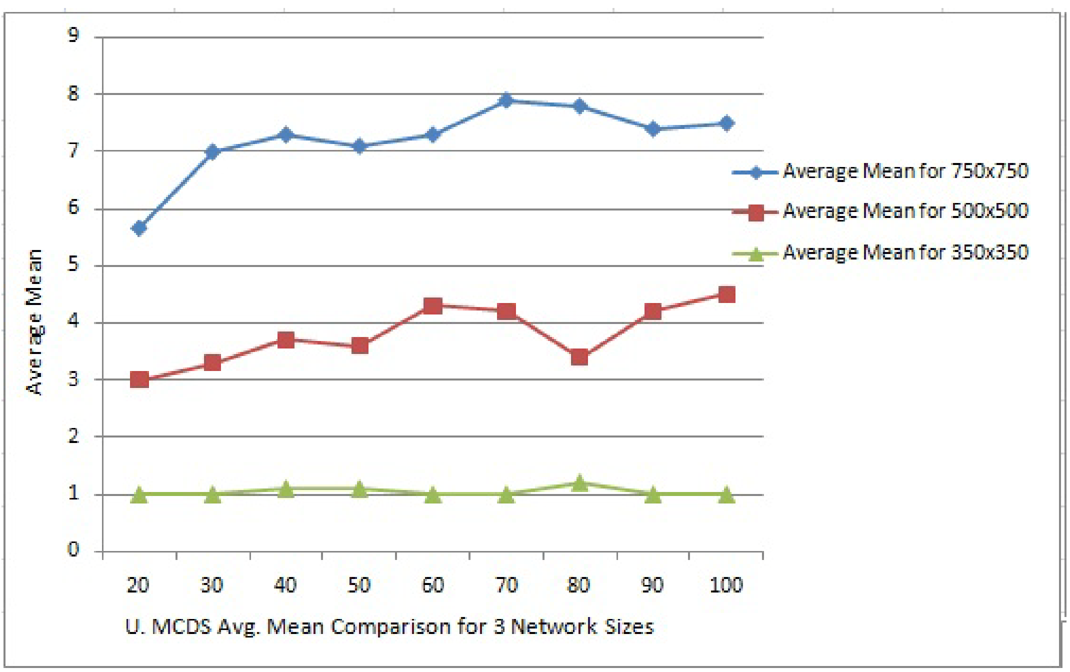


Figure 5: Comparison between MCDS sizes (*unconstrained* version) for 3 different network areas

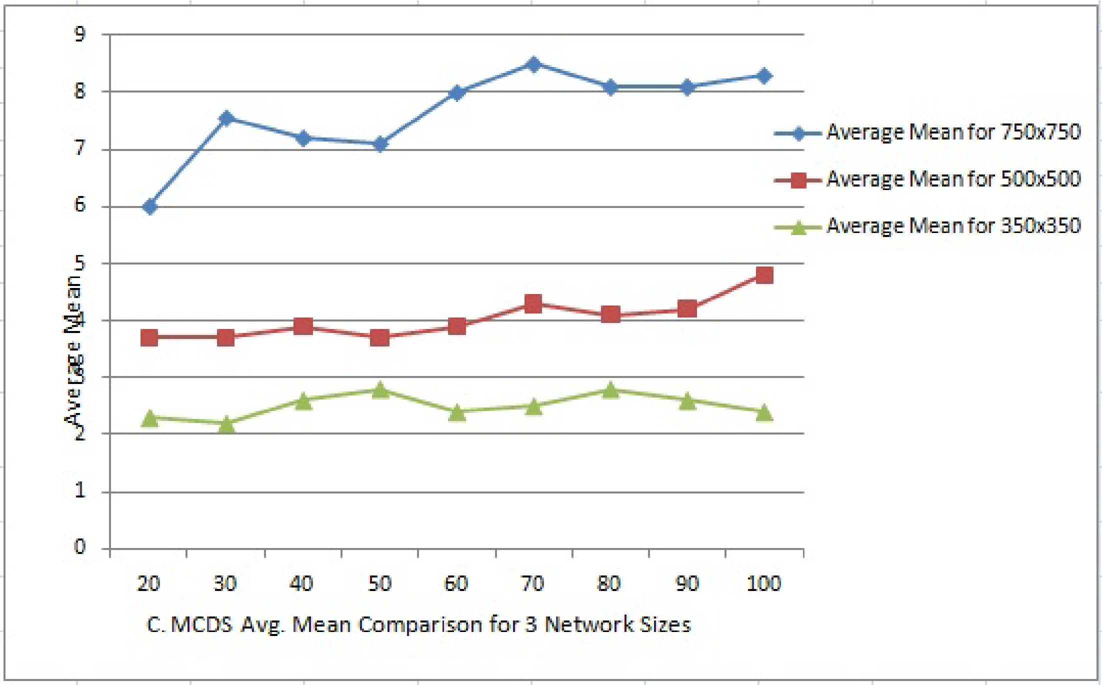


Figure 6: Comparison between MCDS sizes (*constrained* version) for 3 different network areas

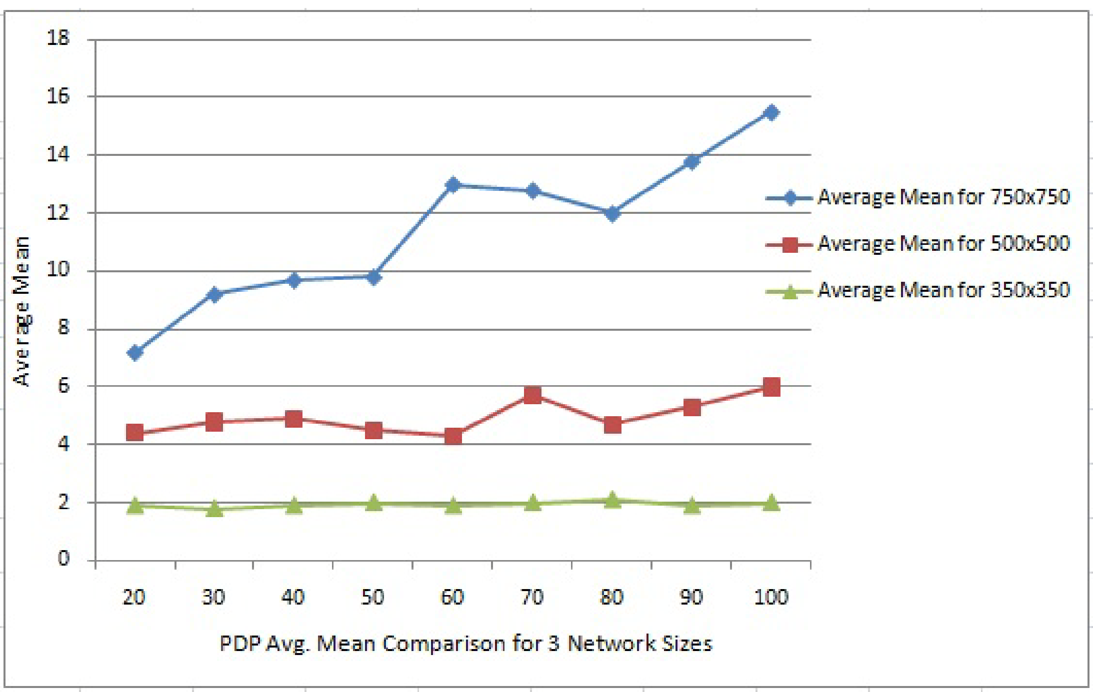


Figure 7: Comparison between forwarders (for PDP) for 3 different network areas

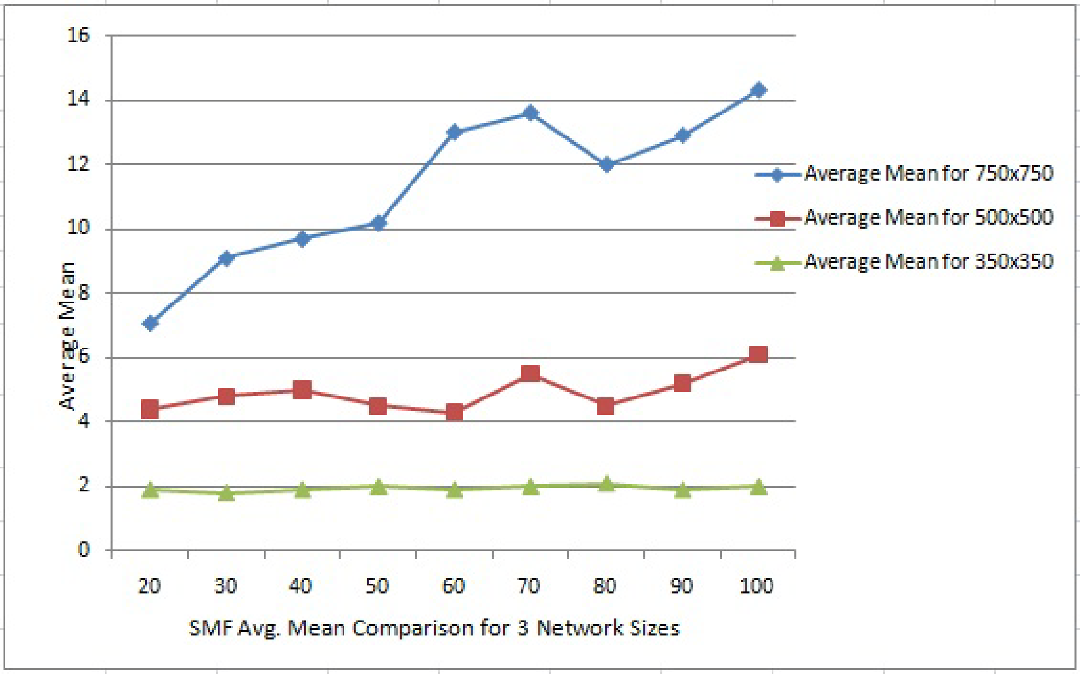


Figure 8: Comparison between MPRs (for SMF) for 3 different network areas

Effect of 200 to 500 nodes in the Fixed Size Network:

For our fixed size network, we conducted simulations for 200 to 500 nodes and observe how that affects the total number of forwarding nodes. We do not find any significant difference when we add 200 or 300 nodes in the network. But when there are 400 or more nodes, the performance deteriorates severely and PDP and SMF both have more forwarding nodes.

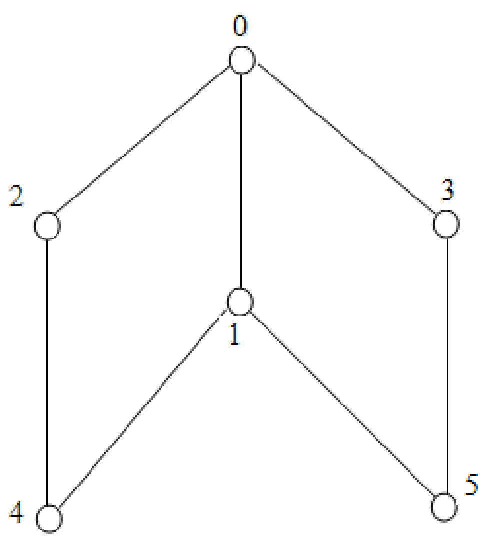
The average number of forwarding nodes (mean values) for these protocols is summarized in the following table:

Table 14: Average number of *forwarding nodes* for all algorithms (200~500 nodes)

Nodes in Network	*unconstrained* MCDS	*constrained* MCDS	Active *Forwarders* (PDP)	Active *MPRs* (SMF)
200	4.9	5.4	6.2	6.5
300	5.0	4.9	6.2	6.7
400	5.4	5.5	11.56	16.43
500	5.2	5.3	17.4	18.67

The reason behind this problem is the congestion in the network. In our study, we found each of the nodes has too many neighbors and when these nodes send hello messages to update their neighborhood information, there are lots of collisions. Hence, most of the nodes do not have proper information about their all neighbors. And because of this incorrect neighborhood information, our distributed protocols (PDP and SMF) select more forwarding nodes. Hence, the number of active forwarding nodes increases.

In the following diagrams (Figures 9, 10, 11, and 12), we describe this issue for a network of 6 nodes. We showed how incorrect neighborhood information leads to a higher numbers of forwarders (for PDP, Figures 9 and 10) and MPRs (for SMF, Figures 11 and 12) when some neighborhood information is missing.

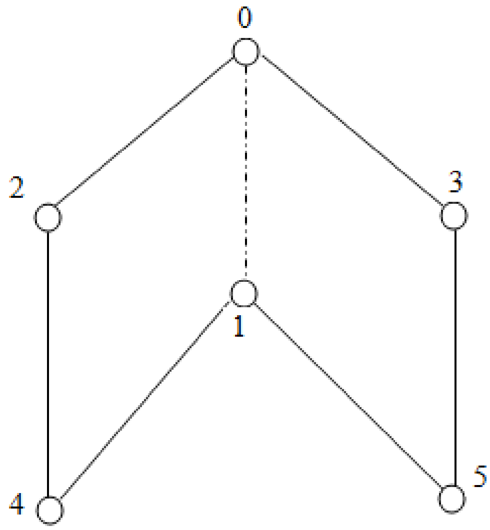


Simplified Multicast Forwarding (SMF) – MPR selection

Node (n_0)	N_1	N_2	MPR (n_0)
0	1,2,3	4,5	1
1	0,4,5	2,3	0
2	0,4	1,3	0
3	0,5	1,2	0
4	1,2	0,5	1
5	1,3	0,4	1

Source: 0, Active MPR: 1

Figure 9: MPR selection for a network with 6 nodes (no missing neighborhood information)



Simplified Multicast Forwarding (SMF) – MPR selection

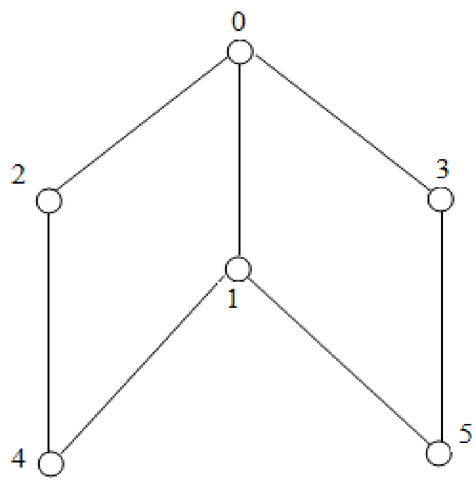
Node (n_0)	N_1	N_2	MPR (n_0)
0	2,3	4,5	2,3
1	4,5	2,3	4,5
2	0,4	1,3	0,4
3	0,5	1,2	0,5
4	1,2	0,5	1,2
5	1,3	0,4	1,3

Source: 0, Active MPRs: 1, 2, 3, 4, 5

Figure 10: MPR selection for a network with 6 nodes (node 0 has no knowledge of node 1)

In the case of SMF, Figure 9 shows that when source 0 has all neighborhood information, there is only one active MPR, which is node 1. But if node 0 does not have any information of node 1 (lost the hello message because of collisions), it will eventually have 5 active MPRs (Figure 10).

Similarly for PDP, we also found that when source 0 has proper neighborhood information, it has only one forwarder (node 1, Figure 11) but when source 0 does not have the information regarding node 1, it calculates in total 5 forwarders (Figure 12).



Partial Dominant Pruning (PDP) – Forwarder selection

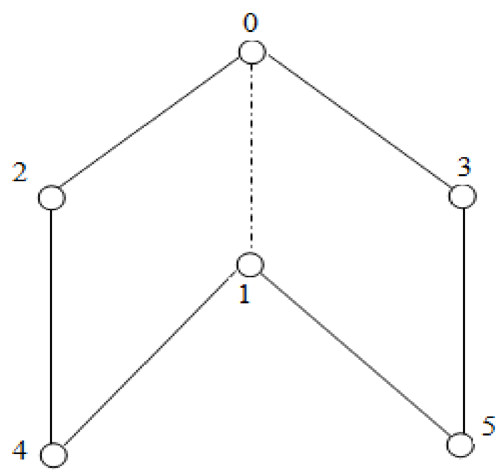
Node (n_0)	N_1	N_2
0	0,1,2,3	0,1,2,3,4,5
1	0,1,4,5	0,1,2,3,4,5
2	0,2,4	0,1,2,3,4
3	0,3,5	0,1,2,3,5
4	1,2,4	0,1,2,4,5
5	1,3,5	0,1,3,4,5

P=N(v) ∩N(u); U=N(N(v))-N(v)-N(u)-P; B=N(v)-N(u), F is the Forwarder set

u	*v*	P	U	B	F
φ	0	Φ	4,5	1,2,3	1
0	1	0,1	Φ	4,5	[]

Source: 0, Active Forwarder: 1

Figure 11: Forwarder selection for a network with 6 nodes (no missing neighborhood information)



Node (n_0)	N_1	N_2
0	0,2,3	0,2,3,4,5
1	1,4,5	1,2,3,4,5
2	0,2,4	0,1,2,3,4
3	0,3,5	0,1,2,3,5
4	1,2,4	0,1,2,4,5
5	1,3,5	0,1,3,4,5

u	*v*	P	U	B	F
Φ	0	Φ	4,5	2,3	2,3
0	2	0,2	1	4	4
0	3	0,3	1	5	5
2	4	2,4	5	1	1
3	5	3,5	4	1	1
4	1	1,4	3	5	5
5	1	1,5	2	4	4

Source: 0, Active Forwarders: 1, 2, 3, 4, 5

Figure 12: Forwarder selection for a network with 6 nodes (node 0 has no knowledge of node 1)

7.1.8 Discussion of Overall Results

We collected results for a centralized MCDS algorithm and two distributed algorithms and analyzed them using statistical tool. Our performance analysis explores a lower number of re-transmissions required to send data to the whole network.

In our analysis, when we do not restrict the MCDS algorithm (i.e., for the *unconstrained* version), it always performs better than PDP and SMF in all cases. Even for the dense and sparse networks, which we have studied here, the *unconstrained* MCDS algorithm is superior to both distributed algorithms except for the sparse network with 20 nodes; there was no significant difference in the means.

On the other hand, when we restrict the MCDS algorithm (i.e., *constrained* version); PDP and SMF perform similar to the *constrained* version for small sized networks (10 to 20/30 nodes). There is no significant difference between them. The same is true for dense and sparse networks. But when we increased the number of nodes in the network (30/40 to 100 nodes), *constrained* MCDS performs better than PDP and SMF.

In the following section we compare the performance of different packet forwarding techniques for fixed density network.

7.2 Performance Comparison of Packet Forwarding Techniques for Fixed Density Network

Here we compare packet forwarding techniques when network density is kept fixed and then increased the number of nodes.

7.2.1 Comparison among Average Number of Forwarding Nodes

We analyze for 100 randomly generated scenarios. The average number of *forwarding nodes* required to cover all nodes in the network are shown in the following table (Table 15) for both the *unconstrained* and *constrained* MCDS version along with PDP and SMF.

Table 15: Average number of *forwarding nodes* for all algorithms (fixed density network)

No. of Nodes in Network	Average # of nodes in MCDS (*unconstrained*)	Average # of nodes in MCDS (*constrained*)	Average # of Active *Forwarders* for PDP	Average # of Active *MPRs* for SMF
10	1.2	1.9	1.8	1.8
20	2.7	3.7	3.69	3.69
30	5.1	5.4	6.3	6.34
40	6.1	6.9	8.77	8.77
50	7.7	8.1	11.76	11.87
60	9.5	9.8	15.14	15.08
70	10.5	11.1	17.37	17.37
80	12	12.5	20.55	20.55
90	13.8	14.4	27.01	26.64
100	15.5	15.8	26.43	26.58

According to Table 15, the *unconstrained* MCDS algorithm provides a lower bound for all cases. The *constrained* version of our MCDS algorithm also provides a smaller number of re-broadcasting nodes compared to the two distributed algorithms. We test whether the differences between these approaches are statistically significant or not in the following section.

We carried out parametric test, called *t-test,* for conducting statistically significant tests in the testing of hypotheses. It is based on Student's t statistic.

7.2.2 T-test Analysis between Unconstrained MCDS and Constrained MCDS

Following is a snapshot of the result that we have generated after running the analysis tool *t-test: two-sample assuming unequal variances* in Microsoft Excel for a network with 10 nodes.

Table 16: t-test Result for *unconstrained* and *constrained* versions of centralized MCDS approximation for the network with 10 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 10)		
	U. MCDS	*C. MCDS*
Mean	1.2	1.9
Variance	0.177778	0.544444
Observations	10	10
Hypothesized Mean Difference	0	
df	14	
t Stat	-2.60473	
P(T<=t) one-tail	0.010393	
t Critical one-tail	1.76131	
P(T<=t) two-tail	0.020785	
t Critical two-tail	2.144787	

In this case, the calculated *p* value (0.020785) is less than the *level of significance (0.05)* which indicates that, for a network with 10 nodes, there is evidence of statistically significant difference between the two means. In other words, the *unconstrained* version will determine an MCDS size that is smaller than the *constrained* version, and the difference in performance is not due to random effects.

On the other hand, when the number of nodes increase in the network (up to 100 nodes), the difference becomes statistically insignificant, which means both versions have a similar MCDS size on average. Table 17 shows the t-test result for the network with 90 nodes.

Table 17: t-test Result for *unconstrained* and *constrained* versions of centralized MCDS algorithm for the network with 90 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 90)		
	U. MCDS	*C. MCDS*
Mean	13.8	14.4
Variance	2.844444	2.266667
Observations	10	10
Hypothesized Mean Difference	0	
df	18	
t Stat	-0.83925	
P(T<=t) one-tail	0.206167	
t Critical one-tail	1.734064	
P(T<=t) two-tail	0.412335	
t Critical two-tail	2.100922	

Hence, in our analysis for fixed density network, we find similar results as in the case of fixed size network. That is, for a small number of nodes (10 to 30), *unconstrained* MCDS achieves a smaller average number of nodes in the MCDS than the *constrained* version, i.e. the difference of these two means is statistically significant. But when there are more nodes in the network (40 to100), both versions compute almost the same number of nodes in the MCDS.

7.2.3 T-test Analysis between PDP and SMF

Below is the t-test analysis for SMF and PDP techniques.

In our simulation, we find no statistically significant difference between the means for PDP and SMF for all networks with same density. These results are the same as those for the fixed size network.

Table 18 shows the t-test result for a network with 10 nodes and Table 19 shows the result for 90 nodes.

Table 18: t-test Result for PDP and SMF for the network with 20 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 10)		
	PDP	*SMF*
Mean	1.8	1.8
Variance	0.4	0.4
Observations	10	10
Hypothesized Mean Difference	0	
df	18	
t Stat	0	
P(T<=t) one-tail	0.5	
t Critical one-tail	1.734064	
P(T<=t) two-tail	1	
t Critical two-tail	2.100922	

Table 19: t-test Result for PDP and SMF for the network with 90 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 90)		
	PDP	*SMF*
Mean	27.01	26.64
Variance	11.42544	8.358222
Observations	10	10
Hypothesized Mean Difference	0	
df	18	
t Stat	0.263056	
P(T<=t) one-tail	0.397747	
t Critical one-tail	1.734064	
P(T<=t) two-tail	0.795495	
t Critical two-tail	2.100922	

Here for both cases, the *p* value is greater than the *level of significance*. So we can conclude that the difference between the average numbers of active *forwarding nodes* is statistically insignificant.

7.2.4 T-test Analysis between MCDS and PDP

In our study, the t-test results show that, for both small and big networks, there is always a statistically significant difference between the means of *unconstrained* MCDS and PDP, where *unconstrained* MCDS always results in a smaller MCDS size than number of active forwarders in PDP. This is true for any network with 10 up to 100 nodes.

Table 20 shows a snapshot of the t-test result between *unconstrained* MCDS and PDP for a network of 50 nodes.

Table 20: t-test Result for *unconstrained* MCDS and PDP for the network with 50 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 50)		
	U. MCDS	*PDP*
Mean	7.7	11.76
Variance	0.677778	1.940444
Observations	10	10
Hypothesized Mean Difference	0	
df	15	
t Stat	-7.93456	
P(T<=t) one-tail	4.76E-07	
t Critical one-tail	1.75305	
P(T<=t) two-tail	9.53E-07	
t Critical two-tail	2.13145	

Additionally, t-tests between *constrained* MCDS and PDP show that for small sized networks, such as 10 to 30 nodes in the network, there is no statistically significant difference between the means.

T-test result between *constrained* MCDS and PDP has shown in Table 21 for a network of 20 nodes.

Table 21: t-test Result for *constrained* MCDS and PDP for the network with 20 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 20)		
	C. MCDS	*PDP*
Mean	3.7	3.69
Variance	0.9	1.161
Observations	10	10
Hypothesized Mean Difference	0	
df	18	
t Stat	0.022027	
P(T<=t) one-tail	0.491334	
t Critical one-tail	1.734064	
P(T<=t) two-tail	0.982669	
t Critical two-tail	2.100922	

But networks with more than 30 nodes results in statistically significant difference between the means of *constrained* MCDS and PDP; which indicate that the *constrained* MCDS will determine an MCDS size that is smaller than the number of active forwarding nodes in PDP, and the difference in performance is not due to random effects.

Table 22 shows a snap shot of the t-test result for a network with 80 nodes.

Table 22: t-test Result for *constrained* MCDS and PDP for the network with 90 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 90)		
	C. MCDS	*PDP*
Mean	14.4	27.01
Variance	2.266667	11.42544
Observations	10	10
Hypothesized Mean Difference	0	
df	12	
t Stat	-10.7766	
P(T<=t) one-tail	7.94E-08	
t Critical one-tail	1.782288	
P(T<=t) two-tail	1.59E-07	
t Critical two-tail	2.178813	

7.2.5 T-test Analysis between MCDS and SMF

As with the comparison between MCDS and PDP, we obtained similar results for all different networks while conducting t-tests between *unconstrained* MCDS and SMF and also between *constrained* MCDS and SMF. In our statistical analysis for fixed density network, we find the difference in average means between *unconstrained* MCDS and SMF is statistically significant for all cases but the difference in means between *constrained* MCDS and SMF is statistically insignificant for smaller networks (10 to 30 nodes) and statistically significant while number of nodes increases.

Table 23 and Table 24 show the results of the t-test of a network of 50 nodes between *unconstrained* MCDS and SMF, and *constrained* MCDS and SMF respectively.

Table 23: t-test Result for *unconstrained* MCDS and SMF for the network with 50 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 50)		
	U. MCDS	*SMF*
Mean	7.7	11.87
Variance	0.677778	2.269
Observations	10	10
Hypothesized Mean Difference	0	
df	14	
t Stat	-7.68179	
P(T<=t) one-tail	1.09E-06	
t Critical one-tail	1.76131	
P(T<=t) two-tail	2.19E-06	
t Critical two-tail	2.144787	

Table 24: t-test Result for *constrained* MCDS and SMF for the network with 50 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 50)		
	C. MCDS	*SMF*
Mean	8.1	11.87
Variance	0.322222	2.269
Observations	10	10
Hypothesized Mean Difference	0	
df	12	
t Stat	-7.40609	
P(T<=t) one-tail	4.1E-06	
t Critical one-tail	1.782288	
P(T<=t) two-tail	8.21E-06	
t Critical two-tail	2.178813	

7.2.6 Discussion of Overall Results

Our performance analysis on fixed density network explores a lower number of re-transmissions required to send data to whole network. In this analysis, when we do not restrict the MCDS algorithm (i.e., for *unconstrained* version), it is always better than PDP and SMF in all cases. On the other hand, when we restrict the MCDS algorithm (i.e., *constrained* version); PDP and SMF perform similar to the *constrained* version for small sized networks (10 to 20/30 nodes). There is no significant difference between them. But as the number of nodes increases, *constrained* MCDS performs better than PDP or SMF.

In our simulation and t-test analysis, we found similar performances for network coding techniques. The following two sections (7.3 and 7.4) discuss the performance evaluation of two network coding techniques, comparing it to the analytically derived lower bound for fixed density network and fixed size network. For network coding, we compare their mean values and conduct different t-tests too.

7.3 Performance Comparison of Network Coding Techniques with Lower Bound for Fixed Size Network

We generated results for different scenarios while keeping the network size constant (for 10 to 100 nodes). For each case, 10 different scenarios have been generated and evaluated.

7.3.1 Comparison among Average Number of Re-broadcast Nodes

In this section we compared the mean values (number of re-broadcasting nodes) for network coding lower bound, Partial Dominant Pruning (PDP/XOR) and Simplified Multicast Forwarding (SMF/XOR). Table 25 shows these results.

Table 25: Average number of *re-broadcasting nodes* among network coding lower bound and other techniques

No. of Nodes in Network	**Average # of *re-broadcasting nodes* in Network Coding Lower Bound**	**Average # of *re-broadcasting nodes* in PDP / XOR**	**Average # of *re-broadcasting nodes* in SMF / XOR**
10	3.35	3	2.9925
20	3.15	3.585	3.5825
30	3.0833	4.3125	4.2825
40	3.3167	4.435	4.4375
50	3.333	4.42	4.395
60	3.375	4.92	4.8775
70	3.5033	4.8075	4.85
80	3.2499	4.705	4.6825
90	3.333	4.715	4.7325
100	3.5	5.515	5.525

In the above table, we can observe that the derived lower bound is indeed always better than PDP/XOR or SMF/XOR. Though the difference among means is not much for smaller networks, it significantly increases when the number of nodes in the network increases. To test whether these differences are statistically significant or not, we conduct t-tests. The following section describes the t-tests results for their means.

7.3.2 T-test Analysis between Network Coding Lower Bound and PDP/XOR

T-test analysis between the means of network coding lower bound and PDP/XOR indicates that for a network with 10 to 20 nodes, there is no statistically significant difference between the means. In contrast, adding more nodes (30 to 100) in the network indicates that the network coding lower bound is significantly lower than the observed performance of PDP/XOR.

Following are two snap shots for t-test results analysis for a network with 20 nodes and 80 nodes (Table 26 and 27, respectively).

Table 26: t-test Result for Network Coding Lower Bound and PDP/XOR for a network with 20 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 20)		
	NCLB	*PDP/XOR*
Mean	3.15	3.585
Variance	0.336111	0.217389
Observations	10	10
Hypothesized Mean Difference	0	
df	17	
t Stat	-1.84897	
P(T<=t) one-tail	0.040965	
t Critical one-tail	1.739607	
P(T<=t) two-tail	0.081929	
t Critical two-tail	2.109816	

Table 27: t-test Result for Network Coding Lower Bound and PDP/XOR for a network with 80 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 80)		
	NCLB	*PDP/XOR*
Mean	3.2499	4.705
Variance	0.106463	0.464278
Observations	10	10
Hypothesized Mean Difference	0	
df	13	
t Stat	-6.09079	
P(T<=t) one-tail	1.92E-05	
t Critical one-tail	1.770933	
P(T<=t) two-tail	3.84E-05	
t Critical two-tail	2.160369	

7.3.3 T-test Analysis between Network Coding Lower Bound and SMF/XOR

We found similar results for t-tests analysis between the means of network coding lower bound and SMF/XOR. That is, for a network with 10 to 20 nodes, there is no difference between their means but for networks with 30 to 100 nodes, lower bound is always better than SMF/XOR.

7.3.4 Discussion of Overall Results

In our analysis, network coding lower bound is lower than the observed performance of the coding protocols (PDP/XOR and SMF/XOR) when there are more nodes in the network. For our analysis, we found this result when number of nodes is 30 or more. But a network with less than 30 nodes will generate a similar number of re-broadcasting nodes for all.

Section 7.4 describes network coding performance comparison analysis for fixed density network.

7.4 Performance Comparison of Network Coding Techniques with Lower Bound for Fixed Density Network

For fixed density network, we also generate simulation results for different scenarios. For each case, 10 different scenarios have been generated and evaluated.

7.4.1 Comparison among Average Number of Re-broadcast Nodes

In this section we compared the mean values (number of re-broadcasting nodes) for network coding lower bound, PDP/XOR and SMF/XOR when network density is always same. Table 28 shows these results.

Table 28: Average number of *re-broadcasting nodes* for all network coding techniques

No. of Nodes in Network	Average # of *re-broadcasting nodes* in Network Coding Lower Bound	Average # of *re-broadcasting nodes* in PDP/XOR	Average # of *re-broadcasting nodes* in SMF/XOR
10	1.8	1.725	1.725
20	2.933	3.23	3.245
30	4.741	5.81	5.845
40	5.3	7.7975	7.935
50	6.5167	10.2575	10.64
60	7.3917	13.155	13.14
70	8.1376	15.6225	15.47
80	9.2658	17.5525	17.7175
90	9.8213	21.505	21.5175
100	10.716	22.955	23.06

According to Table 28, we can see PDP/XOR or SMF/XOR is not close to the lower bound when we have fixed density. The differences among average means are not much for smaller networks, but lower bound is significantly better than both as the number of nodes increases in the network.

We conducted t-tests here too for evaluating the performances of these techniques. Following sections describes the t-tests results between these algorithms.

7.4.2 T-test Analysis between Network Coding Lower Bound and PDP/XOR

Here t-tests show similar results to what we have discussed already for fixed size network. For a network with 10 to 20 nodes, there is no statistically significant difference between the means. But having more nodes (30 to 100) nodes, we see statistically significant difference between the lower bound and PDP/XOR.

Following are two snap shots for t-test results analysis for a network with 20 nodes and 80 nodes (Table 29 and 30, respectively).

Table 29: t-test Result for Network Coding Lower Bound and PDP/XOR for a network with 20 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 20)		
	NCLB	*PDP/XOR*
Mean	2.933	3.23
Variance	0.396001	0.367889
Observations	10	10
Hypothesized Mean Difference	0	
df	18	
t Stat	-1.07459	
P(T<=t) one-tail	0.148377	
t Critical one-tail	1.734064	
P(T<=t) two-tail	0.296755	
t Critical two-tail	2.100922	

Table 30: t-test Result for Network Coding Lower Bound and PDP/XOR for a network with 80 nodes

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 80)		
	NCLB	*PDP/XOR*
Mean	9.2658	17.5525
Variance	0.785446	3.631174
Observations	10	10
Hypothesized Mean Difference	0	
df	13	
t Stat	-12.4691	
P(T<=t) one-tail	6.59E-09	
t Critical one-tail	1.770933	
P(T<=t) two-tail	1.32E-08	
t Critical two-tail	2.160369	

7.4.3 T-test Analysis between Network Coding Lower Bound and SMF/XOR

Our t-tests between network coding lower bound and SMF/XOR for fixed density network also have similar results as fixed size network. That is, for a network with 10 to 20 nodes, there is no difference between their means but for networks with 30 to 100 nodes, lower bound is always better than SMF/XOR.

7.4.4 Discussion of Overall Results

In this study, network coding lower bound always indicates a lower number of re-broadcasting nodes than PDP/XOR and SMF/XOR when there are more than 20 nodes in the network. Otherwise, there is no statistically significant difference between their means.

In the following section we compare both lower bounds, that is, we evaluate performance of packet forwarding lower bound (implemented centralized MCDS algorithm) and network coding lower bound.

7.5 Performance Comparison between MCDS and Network Coding Lower Bounds

In Chapter 3, we have described the centralized MCDS algorithm that we used as our packet forwarding technique to obtain the near-optimal number of required re-transmitting nodes. On the other hand, in Chapter 4, we described the analytically derived lower bound for network coding.

In the following two sub-sections, we discuss and analyze t-test results between these two lower bounds. Section 7.5.1 describes the performance comparison for fixed sized network and Section 7.5.2 contains the analysis for fixed density network.

7.5.1 Performance Comparison for Fixed Size Network

In this section we compare the performances for fixed size network after conducting t-tests between lower bounds (both versions of MCDS algorithm and network coding). Table 31 shows the t-test result for *unconstrained* MCDS and network coding lower bound. It indicates that for a network with 20 nodes, there is no significant difference. But as number nodes increases in the network, network coding has lower number of re-broadcasting than packet forwarding. Table 32 shows the t-test results for 60 nodes. For t-tests between *constrained* version of MCDS and network coding lower bound, we obtained similar results.

Table 31: t-test Result for Packet Forwarding (*unconstrained* MCDS) and Network Coding Lower Bounds for a network with 20 nodes (for fixed size network)

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 20)		
	U. MCDS	*NCLB*
Mean	3	3.15
Variance	0.44444	0.33611
Observations	10	10
Hypothesized Mean Difference	0	
df	18	
t Stat	-0.53689	
P(T<=t) one-tail	0.29896	
t Critical one-tail	1.73406	
P(T<=t) two-tail	0.59791	
t Critical two-tail	2.10092	

Table 32: t-test Result for Packet Forwarding (*unconstrained* MCDS) and Network Coding Lower Bounds for a network with 60 nodes (for fixed size network)

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 60)		
	U. MCDS	*NCLB*
Mean	4.2	3.375
Variance	0.4	0.21181
Observations	10	10
Hypothesized Mean Difference	0	
df	16	
t Stat	3.33539	
P(T<=t) one-tail	0.0021	
t Critical one-tail	1.74588	
P(T<=t) two-tail	0.00419	
t Critical two-tail	2.11991	

Figure 13 shows the line diagram for both MCDS versions and network coding lower bound for fixed size network. It indicates that network coding lower bound is better than both MCDS versions but the difference in networks with 10 to 20 nodes is hardly noticeable.

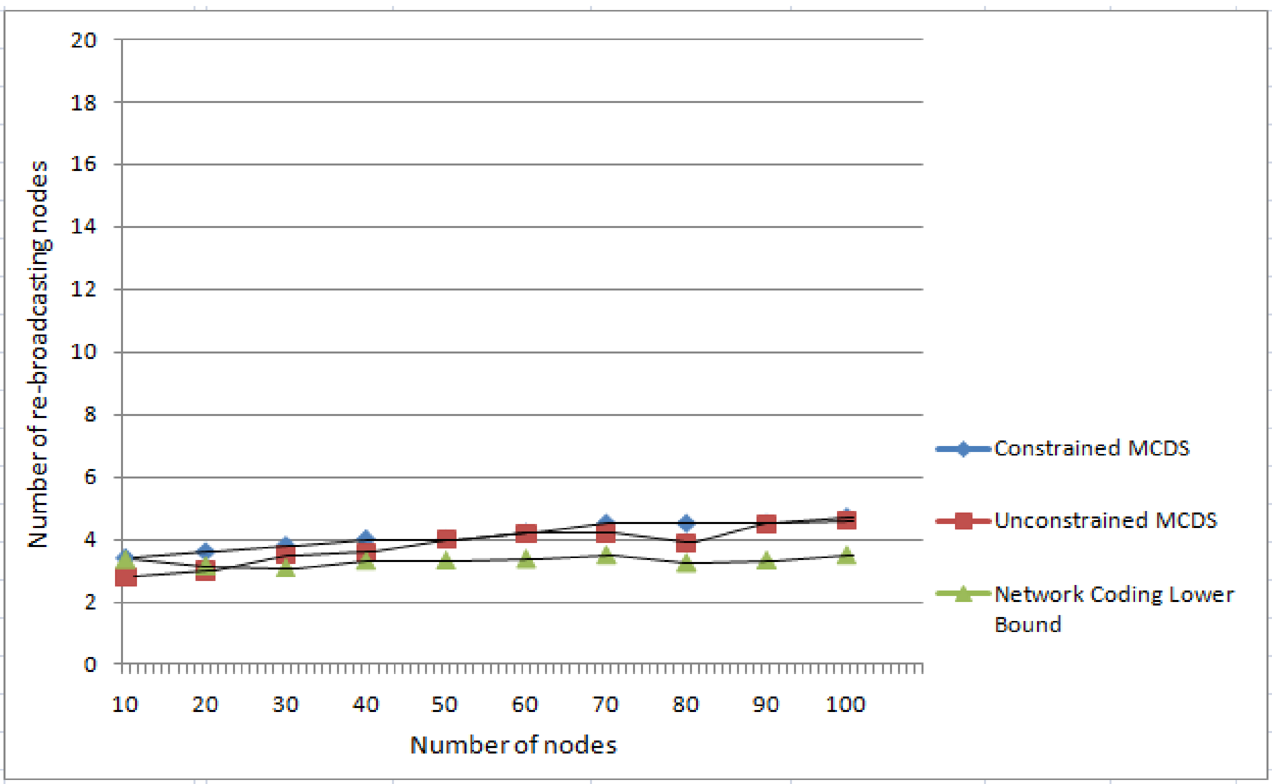


Figure 13: Line diagram shows MCDS and Network Coding Lower Bounds (fixed network size)

7.5.2 Performance Comparison for Fixed Density Network

Here, we conduct t-tests for fixed density network. Table 33 shows the t-test result for *unconstrained* MCDS and network coding lower bound. It shows that for a network with 30 nodes, there is no significant difference. But when number nodes increases in the network (40 or more nodes), network coding has significantly lower number of re-broadcasting than packet forwarding. Table 34 shows t-test result between *unconstrained* MCDS and network coding lower bound for 50 nodes.

Table 33: t-test Result for Packet Forwarding (*unconstrained* MCDS) and Network Coding Lower Bounds for a network with 30 nodes (for fixed size network)

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 30)		
	U. MCDS	*NCLB*
Mean	5.1	4.741
Variance	0.76667	0.18883
Observations	10	10
Hypothesized Mean Difference	0	
df	13	
t Stat	1.16139	
P(T<=t) one-tail	0.13318	
t Critical one-tail	1.77093	
P(T<=t) two-tail	0.26636	
t Critical two-tail	2.16037	

Table 34: t-test Result for Packet Forwarding (*unconstrained* MCDS) and Network Coding Lower Bounds for a network with 50 nodes (for fixed size network)

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 50)		
	U. MCDS	*NCLB*
Mean	7.7	6.5167
Variance	0.67778	0.20646
Observations	10	10
Hypothesized Mean Difference	0	
df	14	
t Stat	3.97934	
P(T<=t) one-tail	0.00069	
t Critical one-tail	1.76131	
P(T<=t) two-tail	0.00137	
t Critical two-tail	2.14479	

On the other hand, for *constrained* version, only when there are 10 nodes in the network, there is no significant difference. Otherwise network coding is always better. Table 35 and 36 show the t-test results between *constrained* MCDS and network coding for 10 and 50 nodes, respectively.

Table 35: t-test Result for Packet Forwarding (*constrained* MCDS) and Network Coding Lower Bounds for a network with 10 nodes (for fixed density network)

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 10)		
	C.MCDS	*NCLB*
Mean	1.9	1.8
Variance	0.54444	0.4
Observations	10	10
Hypothesized Mean Difference	0	
df	18	
t Stat	0.3254	
P(T<=t) one-tail	0.37432	
t Critical one-tail	1.73406	
P(T<=t) two-tail	0.74863	
t Critical two-tail	2.10092	

Table 36: t-test Result for Packet Forwarding (*constrained* MCDS) and Network Coding Lower Bounds for a network with 50 nodes (for fixed density network)

t-Test: Two-Sample Assuming Unequal Variances (# of nodes: 50)		
	C. MCDS	*NCLB*
Mean	8.1	6.5167
Variance	0.32222	0.20646
Observations	10	10
Hypothesized Mean Difference	0	
df	17	
t Stat	6.88601	
P(T<=t) one-tail	1.3E-06	
t Critical one-tail	1.73961	
P(T<=t) two-tail	2.6E-06	
t Critical two-tail	2.10982	

The following figure (Figure 14) shows the line diagram among MCDS algorithm (*unconstrained* and *constrained* versions) and network coding lower bound for fixed density network. It obviously indicates that network coding lower bound is better than both MCDS versions though the difference in a network with smaller number of nodes is hardly noticeable.

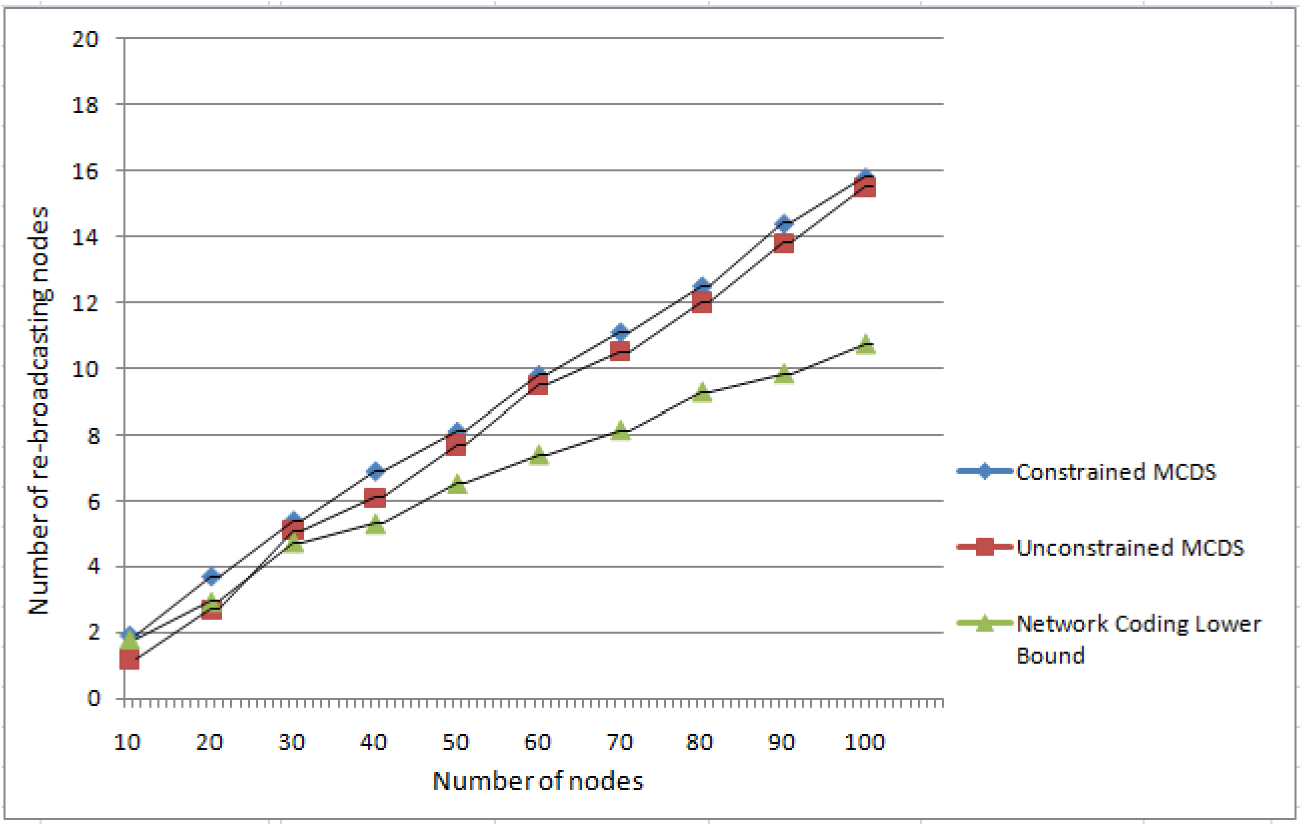


Figure 14: Comparing MCDS and Network Coding Lower Bounds (fixed density)

According to the t-tests between both lower bounds for both, network coding indicates a lower number for most scenarios. Though network coding lower bound is better than MCDS in most of the cases, the difference is not significant for fixed size network. But we observed before that network coding lower bound is better than MCDS algorithms when we consider fixed density network. For fixed density network, network coding provides much lower number of re-broadcasting than packet forwarding.

CHAPTER 8: SUMMARY AND CONCLUSION

In this project, we conducted a comparative analysis between forwarding and network coding approaches for broadcasting in multihop wireless networks. In our study, we consider a static wireless network and statistically evaluated the performance.

In our study, for fixed size and fixed density network, *unconstrained* MCDS version always shows better performances than the *constrained* version since there is no restriction. In the case of packet forwarding techniques, MCDS algorithms outperform both PDP and SMF, specifically when there are more than 20/30 nodes in the network. This is also true for network coding techniques where the number of re-broadcasting nodes in PDP/XOR and SMF/XOR is much higher than the network coding lower bound.

On the other hand, when we compare both lower bounds, that is, packet forwarding lower bound and network coding lower bound, we find that the network coding lower bound seems to require a smaller number of re-broadcasting nodes than the packet forwarding. Though the difference between them is not significant for fixed size network, network coding is much better for the network where the density of the network is constant as we scale up the network.

Hence, we conclude that network coding is more efficient for having lower number of re-broadcasting than any other techniques especially for bigger networks with equal density.

REFERENCES

[1] M.R. Garey, D.S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, Freeman, San Francisco, 1978.

[2] F.V. Fomin, F. Grandoni, and D. Kratsch. *Solving Connected Dominating Set Faster than 2n**. The 26th Conference on Foundations of Software Technology and Theoretical Computer Science, December 13-15, 2006; FSTTCS 2006.

[3] S. Guha, S. Khuller. *Approximation Algorithms for Connected Dominating Sets.* Algorithmica, Vol 20, Issue 4, April 1998.

[4] M. Rai, S. Verma, S. Tapaswi. *A Heuristic for Minimum Connected Dominating Set with Local Repair for Wireless Sensor Networks*. Proceedings of the 2009 Eighth International Conference on Networks, Vol 00, Pages 106-111, IEEE Computer Society.

[5] M. Min, H. Du, X. Jia, C.X. Huang, S.C.-H. Huang, W. Wu. *Improving Construction for Connected Dominating Set with Steiner Tree in Wireless Sensor Networks.* Journal of Global Optimization (2006) 35: 111-119.

[6] Y. Li, M.T. Thai, F. Wang, C.-W. Yi, P.-J. Wan, D.-Z. Du. *On greedy construction of connected dominating sets in wireless networks.* Wireless Communications and Mobile Computing, Vol 5, Issue 8 (December 2005), Pages 927-932.

[7] J. P. Macker, J. Dean, W. Chao. *Simplified Multicast Forwarding in Mobile Ad hoc Networks*. Published in Military Communications Conference, 2004. MILCOM 2004. IEEE.

[8] J. Macker, I. Downard, J. Dean, B. Adamson. *Evaluation of Distributed Cover Set Algorithms in Mobile Ad hoc Network for Simplified Multicast Forwarding*. Mobile Computing and Communications Review, Volume 11, Number 3; July 2007.

[9] W. Lou, J. Wu. *On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks.* IEEE Transaction on Mobile Computing, Vol. 1, Issue 2, April-June 2002.

[10] R. Ahlswede, N. Cai, S.-Y.R. Li and R.W. Yeung, “Network Information Flow”, IEEE-IT, vol. 46, pp. 1204-1216, 2000.

[11] T. Kunz, S. Paul, and Li Li “Efficient Broadcasting in Tactical Networks: Forwarding vs. Network Coding”, Military Communication Conference (MILCOM) 2010-Networking Protocols and Performance (NPP).

[12] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, “Network Coding-Based Broadcast in Mobile Ad hoc Networks”, 26th IEEE International Conference on Computer Communications, IEEE, May 2007, pp. 1739-1747.

[13] J.S. Park, D.S. Lun, Y. Yi, M. Gerla, and M. Medard, “CodeCast: A network-coding-based ad hoc multicast protocol”, IEEE Wireless Communications, Oct. 2006, pp. 76-81.

[14] Y. Wu, K. Jain, and S. Kung, “A unification of network coding and tree-packing (routing) theorems”, IEEE/ACM Trans. Network, June 2006, pp. 2398-2409.

[15] C. Campolo, C. Casetti, C.F. Chiasserini, and S. Tarapiah, "Performance of network coding for ad hoc networks in realistic simulation scenarios", Proc. IEEE ICT '09, pp. 31-36.

[16] D. E. Lucani, M. Medard, and M. Stojanovic, "Broadcasting in time-division duplexing: A random linear network coding approach", Proc. Workshop on Network Coding, Theory, and Applications, June 2009, pp.62-67.

[17] R. Khalili, M. Ghaderi, J. Kurose, and D. Towsley, "On the performance of random linear network coding in relay networks", Proc. IEEE MILCOM 2008, Nov. 2008.

[18] C. Adjih, and S.-Y. Chon, “Wireless broadcast with network coding: Energy efficiency, optimality and coding gain in lossless wireless networks:, Research Report RR-7011, INRIA, France, July 2009.

[19] C. Fragouli, J. Widmer, and J. Le Boudec, “Efficient broadcasting using network coding”, IEEE/ACM Trans. Netw., Apr. 2008, pp. 450-463.

[20] S. Butenko, X. Cheng, C.A.S. Oliveira. *A New Heuristic for the Minimum Connected Dominating Set Problem on Ad Hoc Wireless Networks*, 2004. In S. Butenko, R. Murphey, and P. Pardalos, editors, Recent Developments in Cooperative Control and Optimization, pages 6173. Kluwer Academic Publishers, 2004.

[21] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, “Network Coding-Based Broadcast in Mobile Ad hoc Networks”, 26th IEEE International Conference on Computer Communications, IEEE, May 2007, pp. 1739-1747.

[22] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the air: practical wireless network coding”, Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM’06. Volume 36 Issue 4, October 2006.

Appendix

T-test Assumptions and Procedures

We use t-tests to compare our sample populations and determine if there is a significant difference between their means. The result of the t-test is a *'t' value*; this value is then used to determine the *p-value*. The *p-value* is the probability that *'t'* falls into a certain range. In other words this is the value we use to determine if the difference between the means in our sample populations is significant.

Here *t-test* for independent (unpaired) samples has been considered because our samples are from two different algorithms which are totally independent of each other. Since the number samples are small and variances of both groups (for each test) are not known in our study, we considered the t-test, ***"two-sample assuming unequal variances"***.

To perform the unpaired (independent) two-paired t-test:

1. Start with the hypothesis ($\mathbf{H_0}$) "There is no difference between the populations of measurements from which samples have been drawn". ($\mathbf{H_1}$: There is a difference).

2. Consider the level of significance is 0.05 which means 95% of confidence level.

3. We used the Microsoft Excel's Analysis tool *"t-Test: Two-Sample Assuming Unequal Variances"* to perform our student's t-test. This *t-test* form assumes that the two data sets came from distributions with unequal variances. This tool generates a table which includes the average means, variances, t-statistic value, *p* value (which will be compared with level of significance), degree of freedom.

4. If the calculated *p value is less or equal to the level of significance* (in our study *level of significance i*s 0.05), then we can reject the *Null Hypothesis* (H_0), i.e. there is evidence of a statistically significant difference between the groups of data.

5. Otherwise, if the calculate *p value is greater than the level of significance*, then the *Null Hypothesis* (H_0) is accepted, i.e. there is no evidence of a statistically significant difference between the two groups of data.