Efficient Broadcasting in Tactical Networks: Forwarding vs. Network Coding

Thomas Kunz  
School of Computer Science  
Carleton University  
Ottawa, Canada  
tkunz@sce.carleton.ca

Suranjit Paul  
School of Computer Science  
Carleton University  
Ottawa, Canada  
sjaul2@connect.carleton.ca

Li Li  
Network Systems and Technologies  
Communications Research Centre Canada  
Ottawa, Canada  
li.li@crc.ca

Abstract— Broadcasting (communicating information from one to all or many to all nodes in a network) is an important communication primitive. It is used as a building block in many MANET routing protocols, for example. In addition, broadcasting is a key primitive in tactical networks to support to deliver applications of all-informed voice, group push-to-talk, situational information sharing etc. Supporting one-to-all or all-to-all communication patterns in multihop wireless networks efficiently is therefore important. In this paper, we compare efficient broadcasting protocols based on packet forwarding and based on network-coding techniques. Using a number of network scenarios, we derive lower bounds for the required number of packet retransmissions at the MAC layer to support broadcast with and without applying network coding techniques. We compare these lower bounds with each other, as well as with protocols proposed for each approach. More specifically, we use SMF and PDP as sample forwarding-based broadcast protocols, and a multisource random linear network coding protocol as a representative network coding approach. The results show that network coding has advantages over pure packet forwarding. However, none of the existing solutions comes close to approaching their respective lower bounds, leaving much room for new and improved protocols.

Keywords—MANETs; broadcast; lower bounds; routing; network coding; SMF; PDP

I. INTRODUCTION

Broadcasting (communicating information from one to all or many to all nodes in a network) is an important communication primitive. It is used as a building block in many MANET routing protocols, for example. In addition, broadcasting is a key primitive in multihop tactical networks to support applications of all-informed voice, group push-to-talk, situational information sharing etc. Supporting one-to-all and many-to-all communication patterns in multihop wireless networks efficiently is therefore important. The key metric we are interested in our work is the number of packet transmissions at the MAC layer: if a protocol can deliver data packets to all nodes with fewer packet transmissions at the MAC/PHY layer, this will lower energy and network resource consumption and mitigate the traffic congestion problem in the network. Along this direction, we study the performance of actual protocols, as well as lower bounds on the required number of packet transmissions to identify promising approaches and to identify potential avenues for future research.

A simple approach to broadcasting is flooding: each node, upon receiving a packet for the first time, rebroadcasts it. Such a protocol works well in sparse networks and in networks with high mobility, but is very inefficient, as nodes typically receive the same data packets multiple times. Many more efficient broadcast routing protocols have been proposed to improve on flooding, an early review is [19]. A lower bound for any approach based on packet forwarding exists and is well-known, called the Minimum Connected Dominating Set (MCDS), though determining this lower bound is NP-hard [4]. In this work, we determine this lower bound via a heuristic originally suggested in [3] and compare it to the number of packet retransmissions required by two efficient broadcast protocols, SMF and PDP. The results show that the two broadcast protocols perform similarly, and are not close to the lower bound as the network size increases.

In parallel, network coding has drawn considerable attention recently in the protocol design for MANET, especially to improve the throughput for broadcast and multicast traffic. In particular in the case of broadcasting, it is unclear though whether network coding will, for random networks, provide additional advantages (a further reduction in the number of packet retransmission, called the coding gain). To explore this question, in this work, we determine the lower bound on the number of required packet retransmissions for network coding by formulating this as an integer linear program and solve it for a range of randomly generated multihop wireless networks. We then compare these lower bounds to the MCDS results and also to results we obtain with two broadcast protocols that use network coding proposed in [12]. The results confirm that network coding indeed has the potential to provide significant coding gains (reduced number of transmissions). Similar to the approaches based on packet forwarding, existing network-coding-based broadcast protocols still allow for significant room for improvement.

II. RELATED WORK

The problem of finding the minimal set of forwarding nodes is called the Minimum Connected Dominating Set (MCDS) problem and is known to be NP-hard [4]. The best exact solution to find a Minimum Connected Dominating Set of an arbitrary graph of n nodes we found is described in [9] and solves the problem in $O(1.9407^n)$, a slight improvement of the trivial $\Omega(2^n)$ algorithm. The algorithm makes use of new domination rules, and its analysis is based on the Measure and
Convergent technique [10]. Therefore, the focus has been on heuristics that, in polynomial time, derive good approximations of the MCDS. One group of proposed solutions heuristically derives an MCDS approximation in a distributed fashion, which lends itself to implementation in a network. In 2002, Wan proposed a distributed algorithm [18] which constructs a Connected Dominating Set (CDS) in two phases. These two phases construct a Maximal Independent set (MIS), and a dominating tree, respectively. The MIS constructed in the first phase guarantees that the distance between any pair of its complementary subsets is exactly two hops. In the second phase, the internal nodes of the dominating tree become a CDS. An alternative effective distributed approximation algorithm has been proposed in [10] to construct a minimum connected dominating set for wireless ad hoc networks. This localizes algorithm is based on a maximal independent set (MIS). It first constructs the MIS, where the nodes in MIS are called the dominators. Then, a unique shortest path is selected to connect a pair of dominators whose distance is within three hops. The nodes in these shortest paths are called connectors. All the dominators and connectors form the connected dominating set. This algorithm is experimentally compared only to the previous algorithm [18] and no general evaluation was done. Another distributed MCDS approximation algorithm for wireless sensor networks (WSN) has been proposed in [21]. In this CDS-HG algorithm, a hierarchical graph is used to model the direct connectivity among sensor nodes in a WSN, ignoring connectivity at the same level of the hierarchical graph. After that, nodes at each level compete with their siblings to dominate child nodes at the next level, the winners form the CDS in the end. This algorithm uses two rules, the Essential Node Determination rule and the Competition Priority rule to find the essential nodes which have the highest priority while forming the CDS. Formal analysis shows that this CDS-HG algorithm outperforms existing distributed algorithms in terms of the generated CDS size and message complexity.

If the complete topology is known, centralized heuristics can be applied and provide, in general, a better approximation to the MCDS size. A new efficient centralized heuristic algorithm for the MCDS problem was proposed in [3]. 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. In our work, we know the complete network topologies, and therefore selected this algorithm to derive the lower bound for broadcast solutions based on packet forwarding.

In addition to the MCDS algorithms, other distributed solutions for selecting broadcast forwarding nodes have also been proposed. Within the IETF, for example, a protocol called SMF which is based on the OLSR [6] MPR selection scheme, is under discussion within the MANET work for standardization [16]. An alternative efficient broadcast protocol is PDP [14]. Both protocols rely on periodic HELLO messages to learn the 2-hop neighborhood information. Nodes locally decide which other nodes should rebroadcast packets based on this local information only. In SMF, a node selects a subset of its 1-hop neighbors (called MPR nodes) such that collectively these 1-hop neighbors cover all its 2-hop neighbors. This selection is done upon learning about the neighborhood and detecting changes in it. Once a node selects MPRs, it will signal this information in its Hello message. Once a node broadcasts a packet, all 1-hop neighbors, upon receiving the packet, will check whether it was selected as an MPR by the transmitting node. If that is the case, a node will rebroadcast the packet. PDP will similarly select 1-hop neighbors as packet forwarders. However, in PDP, a node also takes into consideration from which node it received the broadcast packets. Knowing the original nodes’ 1-hop neighborhood information, it can then more aggressively prune the set of 2-hop neighbors that still need to be covered by selecting a subset of its 1-hop neighbors as packet forwarders. As the decision is driven quite a bit by who originally transmitted a packet, PDP defers selecting packet forwarders until a packet is received. If a node sees that it was selected as a forwarder by the packet transmitter, it selects its own forwarder set and adds it to the data packet before rebroadcasting it.

In parallel, network coding has drawn considerable attention recently in the protocol design for MANET, especially to improve the throughput for broadcast and multicast traffic. The majority of research explores the performance of network coding using analytical models, showing that network coding can improve throughput [13], reliability [11], packet delay [2], and file download times [15], for example. However, few actual network protocols that exploit network coding have been proposed in the literature, nor have they been thoroughly evaluated for the network coding promises through extensive simulations. Where this has been done, the results have been mixed. For example, [17] (which shares an author with [2]) describes CodeCast, a random linear network coding-based multicast protocol. The reported simulation results show that network coding experiences significantly higher packet latencies, compared to an alternate packet-forwarding-based multicast protocol, contrary to the results in [2].

Also, relatively little is known about whether network coding is advantageous for broadcasting in general wireless multihop networks. In the case of a single source broadcast in wired networks, network coding provides no benefits [1, 20]. At the same time, [1] 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 (based on a geometric argument). Similarly, [7] 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 very specific network topologies, using scenarios where all nodes are sources. Using simulation, [5] compares random linear network coding with two broadcast schemes under a range of scenarios. In their analysis, network coding is advantageous only in certain cases, such as dense networks.

In this work, we are comparing the performance of efficient packet forwarding and network coding protocols to support broadcasting in static multihop wireless networks. The comparison is based on both lower bounds derived from analytical models/heuristics and the actual performance of protocols that implement a specific approach. More specifically, we compare the following:
• A lower bound for packet forwarding based broadcast protocols generated using the centralized MCDS heuristic in [3].

• 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 RS coding and XOR coding [12] as representative network-coding based broadcast protocols.

III. SIMULATION SETUP

In this paper, we only consider static network scenarios. Assuming that the overall spatial distribution of nodes does not change as a function of the node mobility model (which is not true for all mobility models), mobile scenarios can be considered as a sequence of static snapshots, and the results averaged over many different static networks hold for them as well. Also, as we are interested in lower bounds, we will create traffic in our simulations that will never saturate the network. Using the NS2 setdest tool, we generated 25 static scenarios with N nodes, N ranging from 10 to 100 in steps of 10. The nodes are randomly deployed in a square area, based on a uniform distribution. The square area grows with the number of nodes such that the average node density is constant and ranges from 346 * 346 m² for the 10 node network to 1096 * 1096 m² for the 100 node network. Due to the edge effects, the average number of neighbors varies from 6 for the 10 node network to 15 for the 100 node network, based on the default 250 m wireless transmission range in NS2.

We implemented the MCDS heuristic in [3] as a C program that reads in the NS2 network topology and determines the MCDS set. Two different versions of that heuristic were implemented: in the unconstrained version, the algorithm is free to pick any set of nodes as MCDS, in the constraint version we force the algorithm to always include node 0. This allows us to compare the MCDS results directly with simulation scenarios in which node 0 is fixed as a source node.

To solve the linear program that derives lower bounds for network coding solutions, we similarly implemented a C program that reads in the NS2 network topology and generates the appropriate linear program, which is then solved with the GNU Linear programming kit, GLPK, version 4.42.

For the simulation results, we implemented SMF, PDP, and the XOR and Reed-Solomon coding-based broadcast protocols as described in [12,14,16]. All protocols share quite a bit of functionality, for example they all collect 2-hop neighborhood information via periodic Hello message broadcasts. Also, PDP and SMF propose similar heuristics in determining forwarders and MPR nodes, though based on potentially different subsets of the 1-hop and 2-hop neighbor information. To enable coding in the protocols in [12], nodes have to be aware of their one-hop neighbors and keep track of the packets they believe neighbors already received. This is easily derived from the known one-hop neighborhood information of the packet transmitter, which is available from the Hello messages. All protocols were implemented in NS2.29. In all scenarios, a single source, node 0, broadcasts packets to all other nodes in the network. To ensure high packet delivery ratios in the simulation results, the source sends data packets at a very low data rate, one packet every 2 seconds, avoiding packet losses due to congestion. Also, since we are dealing with static networks, the Hello message interval was set to a relatively large value of 10 seconds. The source only starts transmitting packets after 100 seconds, allowing nodes to have collected the relevant 1- and 2-hop neighbor information, to determine the MPRs, etc.

IV. BROADCASTING BASED ON PACKET FORWARDING

Figure 1 shows the results we obtained for the approaches based strictly on packet forwarding. The X axis plots the number of nodes in the network, the Y axis shows the number of packet retransmissions at the MAC layer to broadcast a single packet to all nodes in the network, averaged over all 25 scenarios. Constraining the MCDS heuristic to always include node 0 does not increase the MCDS size significantly. SMF and PDP show very similar performance, and are close to the lower bound when the network is small (and therefore almost all nodes are within the 2-hop neighborhood of the source node).

![Figure 1. MCDS size, PDP Forwarders, and SMF MPRs](image_url)

As the network size and area increase, nodes further away from the source rebroadcast packets more often than necessary, due to the incomplete information each node has. For example, a node 2 hop away from the source node will determine forwarders or MPRs without knowing which nodes were already covered by the source node. Somewhat surprisingly, even though PDP is more aggressive in pruning the set of 2-hop neighbors that a node needs to cover when selecting its forwarder set, compared to the MPR selection in SMF, the protocols overall require, on average, almost the same number of packet retransmissions at the MAC layer.
V. Broadcasting Based on Network Coding

The lower bound for Network Coding (NC) is derived using a linear program. In a first program, inspired by [1] and [7], we minimize the number of packet transmissions per node, 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 we ensure that 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:

$$\forall i: \sum_{j \in N(i)} X_j \geq N$$
$$X_0 \geq N$$
$$\forall i: X_i \geq 0, X_i \text{ is integer}$$

The objective function indicates that we are interested 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. Applying this formulation to a ring network of 6 nodes will indeed yield a minimum solution of 3 required packet transmissions per data packet, as shown to be the optimal solution in [7]. However, note that in the limiting case of $N = 1$ (and ignoring the last constraint), this formulation is equivalent to the Dominating Set problem: find a minimal number of nodes that collectively cover all nodes. As such, the resulting lower bound is not realistic, as it does not ensure that packets flow from a specific sender. In fact, a solution that achieves the lower bound in the above case will have nodes 0, 2, and 4 in the network broadcasting a packet to their respective neighbors. Our 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. We introduce flow variables $F_{i,j}(d)$ that 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, we assume that node 0, our implied source, trivially receives all packets and therefore $d \neq 0$. We then revise the first linear program as follows: The first constraint enforces that the flow over existing links $i,j$ does not exceed the number of packets physicially transmitted by the head of the link. The second set of constraints captures the usual 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 neighbours. If node $i$ is the destination, it consumes $N$ packets. Otherwise, a node passes on all received packets over its outgoing links. Note that this does not constraint the solution to a packet forwarding solution: the same physical packet (whose transmission is still 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:

$$\forall i, d: j \in N(i): F_{i,j}(d) \leq X_i$$
$$\forall i, d: \sum_{j \in N(i)} F_{i,j}(d) - \sum_{l \in N(i)} F_{l,i}(d) = \begin{cases} N & \text{for } i = 0 \\ -N & \text{for } i = d \\ 0 & \text{otherwise} \end{cases}$$
$$\forall i: X_i \geq 0, X_i \text{ is integer}$$

Applying this linear program to solve the ring network of 6 nodes again results in a lower bound of 3 packet transmissions per data packet, but this time all nodes are involved in passing packets: half the data packets traverse the circle clockwise, the other half counter-clockwise. This in fact models the operation of the optimal coding protocol proposed in [7], where nodes XOR the data received from their two neighbors before retransmitting them.

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. In the ring network example, 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 $X_0 = N/2$. But in a broadcast channel, both transmissions will occupy the media and therefore should be counted separately. To correctly capture the broadcast nature of the wireless media, we introduce dummy nodes, where node $i$ will transmit its packets to its dummy node $\tilde{i}$ first, and the dummy node will then forward the packet to all neighbors of $i$. More formally, the final integer linear program is therefore:

$$\min \sum X_i$$

Subject to:

$$\forall i, d: F_{i,j}(d) \leq X_i$$
$$\forall i, d: \sum_{j \in N(i)} F_{i,j}(d) - \sum_{l \in N(i)} F_{l,i}(d) = \begin{cases} N & \text{for } i = 0 \\ -N & \text{for } i = d \\ 0 & \text{otherwise} \end{cases}$$
$$\forall i: X_i \geq 0, X_i \text{ is integer}$$
The first constraint limits the flows out of a real node $i$ (which are all destined to the nodes’ dummy node $i_D$) 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, otherwise 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.

Applying this final integer linear program to the ring network of 6 nodes, the lower bound for packet transmissions now becomes 3.5 packet transmissions per broadcast packets. Unlike the previous solution, source node 0 now has to transmit all $N$ data packets, which raises the total number of packet transmissions slightly. However, as we grow the number of nodes $K$ in the ring configuration, this one-time additional cost becomes smaller and smaller, asymptotically reaching $K/2$.

Figures 2 and 3 compare our implementation of the broadcast network coding protocols proposed in [12] to the lower bound derived by the final integer linear program. We implemented both XOR and Reed-Salomon coding, using SMF and PDP as underlying packet forwarding mechanisms. The protocols/linear program were applied to the same set of static network scenarios used in the previous section.

In both cases, similar to our findings in the case of broadcasting based on packet forwarding, the proposed broadcast protocols based on network coding are not close to the achievable lower bound. For the scenarios we studied, neither the choice of underlying forwarding protocol (SMF vs. PDP) nor the choice of coding operation (XOR vs. Reed-Salomon) has a noticeable impact on the protocol performance. Comparing Figure 1 to Figures 2 and 3, the required number of MAC broadcasts is reduced, by about 15%, indicating that network coding can be practically made to work, reducing MAC packet transmissions over already quite efficient packet forwarding protocols such as PDP and SMF (which confirms the results in [12]).

![Network Coding over SMF](image1)

![Network Coding over PDP](image2)

![Lower Bound Comparison](image3)

VI. COMPARING PACKET FORWARDING AND NETWORK CODING

The results and discussion in the previous section already indicate that existing network coding solutions can be more efficient (require fewer MAC packet transmissions) than packet forwarding solutions. However, none of the existing protocols are close to achieving the theoretically possible lower bound. In Figure 4 we also compare the lower bounds from the MCDS...
heuristic and the integer linear program. The results show that indeed network coding has the potential to reduce the required number of MAC retransmissions for broadcasting in multihop wireless scenarios. For the larger network, the improvement can be as high as 33%.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we study the question of efficiently supporting broadcasting in multihop wireless networks. In particular, we were interested in protocols that reach all nodes with a small number of MAC packet retransmissions to conserve node energy and reduce the wireless spectrum usage. We explored strategies based on packet forwarding only, as well as broadcasting strategies that utilize network coding. For each strategy, we derived lower bounds on the required number of packet transmissions and compared the performance of proposed broadcast protocols. The results show that actual protocols are still far from achieving the possible lower bound, and that broadcast protocols based on network coding perform better than broadcast protocols based on packet forwarding only. However, for both packet forwarding and network coding, existing protocols are not yet close to achieving their optimal performance. Proposed distributed approximations to the MCDS problem, such as the ones suggested in [10,18,21] may achieve better performance only with significantly higher overheads, limiting their applicability to networks and application scenarios where these costs can be amortized over many broadcast packets.

This work lays the foundation for a number of ongoing efforts. We have not yet studied further the impact of node density on this evaluation. We also are in the process of implementing a random linear network coding protocol in NS2, which we predict to result in better performance than the protocols in [12] once we combine it with an efficient forwarding protocol such as PDP or SMF. In addition, all the results presented here are collected for single-source broadcast scenarios. Supporting multiple sources potentially has a significant impact on the relative performance. The packet forwarding solutions scale linearly with the number of broadcast sources, as packets are all treated independent of each other. Yet the preliminary results we collected with our random linear network coding protocol seem to indicate that efficiently supporting multiple sources, which entails coding over packets originating from different nodes, can further improve the network coding performance. We will explore this further and validate this by appropriately formulating the corresponding lower bound linear program for multi-source network coding. Finally, as we increase network size and network density, we will verify whether the asymptotic coding gains formulated in [1] hold.

VIII. REFERENCES


