Efficient Broadcast in MANETs Using Network Coding and Directional Antennas

Shuhui Yang
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180

Jie Wu and Mihaela Cardei
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Abstract—In this paper, we consider the issue of efficient broadcasting in mobile ad hoc networks (MANETs) using network coding and directional antennas. Network coding-based broadcasting focuses on reducing the number of transmissions each forwarding node performs in the multiple source/multiple message broadcast application, where each forwarding node combines some of the received messages for transmission. With the help of network coding, the total number of transmissions can be reduced compared to broadcasting using the same forwarding nodes without coding. We exploit the usage of directional antennas to network coding-based broadcasting to further reduce energy consumption. A node equipped with directional antennas can divide the omnidirectional transmission range into several sectors and turns some of them on for transmission. In the proposed scheme using a directional antenna, forwarding nodes selected locally only need to transmit broadcast messages, original or coded, to restricted sectors. We also study two extensions. The first extension applies network coding to both dynamic and static forwarding node selection approaches. In the second extension, we design two approaches for the single source/single message issue in the network coding-based broadcast application. Performance analysis via simulations on the proposed algorithms using a custom simulator is presented.

I. Introduction

Broadcasting is the most frequently used operation in mobile ad hoc networks (MANETs) for the dissemination of data and control messages in many applications. Usually, a network backbone is constructed for efficient broadcasting to avoid the broadcast storm problem caused by simple blind flooding, where only selected nodes, called forwarding nodes, that form the virtual backbone, forward data to the entire network.

In MANETs, the forwarding node set for broadcast is usually selected in a localized manner, where each node determines its own status of forwarding or non-forwarding based on local information [16], or the status of a node is designated by its neighbors [7]. A smaller-sized forwarding node set is considered to be more efficient due to the reduced number of transmissions in the network, which helps to alleviate the interference and also conserves energy. The connected dominating set (CDS) as a virtual backbone has been widely studied [10], where each node is either a forwarding node or a neighbor to a forwarding node in the set, and the set is connected. Finding a minimum CDS is NP-complete.

In [6], Li et al. exploited network coding in the broadcast application. They applied coding methods to deterministic forwarding node selection approaches to gain a reduction in the number of transmissions, focusing on reducing the number of transmissions each forwarding node performs. Network coding [5] is defined as allowing intermediate nodes to process the incoming information flows. When a forwarding node, decided by a certain approach, needs to forward several messages to all of its neighbors while some neighbors already have some of the messages, this node can combine some of the messages to reduce the number of forwardings, and each neighbor can still get every message via decoding.

For instance, node c gets two messages from nodes a and b respectively. In order to let a and b have each other's message, c needs to forward both the messages as a traditional forwarding node. With network coding, c only needs to forward one coded message containing both original messages through the XOR operation, and a and b can decode the message with the help of their own messages through the XOR operation. Note that the network coding works only when there are multiple messages broadcast at the same time in the network.

In [18], Yang, Wu, and Dai focused on reducing the total number of forwarding directions/sectors of forwarding nodes. Using directional antennas, the omnidirectional transmission range of each node can be divided into several sectors and the transmission can be performed only in selected sectors. Therefore, by reducing the total number of transmission sectors of the forwarding nodes in the network, the interference can be alleviated as well as the energy consumption.

In this paper, we try to combine the efficiency of both network coding and directional antennas to achieve efficiency in broadcasting. We analyze the performance of these two methods and design an algorithm — Efficient Broadcast using Network Coding and Directional Antennas (EBCD), where each node decides its forwarding status using only local information and limited piggybacked broadcast state information. The proposed design is not simply the combination of the two existing methods. We take the advantage of the interactional effects of them to achieve an even better performance. Additionally, we modify the existing directional antenna method to a dynamic mode. As shown in Figure 1 (a), there are four messages, A, B, C, and D generated from nodes a, b, c, and d, respectively. We assume that node e is selected for forwarding using a forwarding node selection method. Therefore, e needs to forward all four messages, costing 4 transmissions totally. In network coding-based broadcasts,
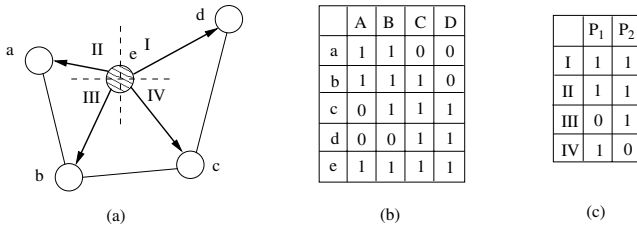
Fig. 1. (a) A sample network, (b) neighbor reception table of node e, and (c) transmission table of node e using coding and directional antennas.

based on 2-hop neighborhood information, e can construct a neighbor reception table as in (b) to record the broadcast state information of the received messages. For instance, when a sends out message A, not only e, but also b gets it. Therefore, b is a "covered" node of message A and there is a "1" in the grid at line b, column A. Based on the table, e then codes these four messages into two combined messages to forward, P_1 ($= A \oplus C$) and P_2 ($= B \oplus D$) ($\oplus$ is the XOR operation) using some network coding algorithms. Obviously, every other node can decode these two combined messages together with the messages it already has in order to gain all four of the original messages. For instance, node b has message A, B, and C. When b receives P_2, it can use $P_2 \oplus B$ to extract message D. a can use $P_1 \oplus A$ and $P_2 \oplus B$ to obtain C and D.

With the help of directional antennas, the omnidirectional transmission range of e can be divided into K sectors (K is 4 in this example), as the dashed lines show in (a). Then e can restrict the transmissions of the two combined messages in only some of the sectors, as shown in table (c). For instance, P_1 only needs to be transmitted in sectors I, II, and IV. If we let the consumption of the transmission of one message in one sector be the unit energy consumption, traditional broadcasting where e transmits all four messages omnidirectionally, costs 16. Broadcasting with network coding costs 8. The broadcast with network coding and directional antennas costs 6. Other than the forwarding nodes, the source nodes can also restrict the transmissions to selected sectors to further reduce the total energy consumption as long as the message can reach a forwarding node.

Although the forwarding node/edge selection and the further network coding procedures are independent, we show that different underlying forwarding node selection approaches affect the efficiency of network coding significantly. In EBCD, we design the dynamic version of the underlying forwarding node/edge selection approach. We then use a static version without piggybacked information for it to analyze the performance and tradeoffs. We find out that the energy conservation of the dynamic and static versions are comparable, although the dynamic one is slightly better. However, since the static version has less overhead, it is more practical. Also, the network coding-based broadcast approach [6] works only when there are multiple sources with multiple messages in the network. We propose two approaches as another extension to EBCD to deal with the single source with single message

application; the pipeline-based approach (PB) and the spread-out approach (SO). We also discuss the detailed implementation techniques in the proposed EBCD algorithm, such as the timing issue and the neighborhood information discovery issue.

The contributions of this paper are the following:

1) We present the advantages of the combination of the network coding and directional antenna approaches for efficient broadcast and develop the EBCD algorithm.
2) We extend the EBCD algorithm to a static forwarding node/edge selection version to study the performance variation.
3) We propose two approaches for the single source with single broadcast message application.
4) We discuss some implementation techniques in EBCD and conduct performance analysis through simulations on the proposed algorithms.

II. RELATED WORKS AND PRELIMINARIES

A. Broadcast in MANETs

Both probabilistic [15] and deterministic [7], [13], [16] approaches have been proposed for efficient broadcast. Probabilistic approaches use limited neighborhood information (local information) and require relatively high broadcast redundancy to maintain an acceptable delivery ratio. Deterministic approaches select a few forwarding nodes to achieve full delivery and most are localized, where each node determines its status (forwarding or non-forwarding) based on its h-hop neighborhood information (for small values of h, such as 2 or 3). The decision of forwarding nodes can be made under both static and dynamic local views. In the static approaches, only topology information is considered, whereas in dynamic ones, broadcast state information of the neighborhood is also piggybacked.

The CDS concept can be applied for broadcasting. Wu and Li [17] proposed the first localized solution for CDS constuction. Peng and Lu [8] presented a scalable broadcast algorithm where the status of a forwarding node is computed on-the-fly. In [13], Stojmenovic et al. extended [17] to a dynamic version. Sucec and Marsic [14] developed a dynamic approach without using a backoff delay. Lou and Wu [7] devised a total/partial dominant pruning (TDP/PDP) method based on 2-hop topology and broadcast state information. Wu and Dai [16] further proposed a generic CDS formation approach, which can be performed in both dynamic and static modes.

B. Network Coding

Network coding [1], [5] can be used to allow the intermediate nodes to combine packets before forwarding. Therefore, network coding can be used for efficient broadcasting by reducing the total number of transmissions. In [3], Fragouli et al. quantified the energy savings that network coding has the potential to offer in broadcasting. They also proposed an implementable method for performing the network coding, addressing some practical issues such as setting the forwarding

factor and managing generations. In [9], a proactive compensation packet is periodically broadcast, constructed from unforwarded messages using network coding to improve the delivery ratio of the probabilistic broadcast approach.

In [6], Li et al. applied network coding to a deterministic broadcast approach called partial dominating pruning (PDP) [7] in a multiple-source broadcast application. They proved that using only XOR operation, the coding algorithm is NP-complete and developed a greedy XOR-based approach for simplicity. The Reed-Solomon code was exploited to design an optimal Reed-Solomon code-based algorithm.

C. Directional Antennas

The most popular directional antenna model is ideally sectorized, as in [4], where the effective transmission range of each node is equally divided into K non-overlapping sectors, where one or more such sectors can be switched on for transmission or reception. The channel capacity when using directional antennas can be improved, and the interference can be reduced. Some probabilistic approaches for broadcasting using directional antennas are proposed in [4], [11], [12].

In [2], Dai and Wu proposed a localized broadcast protocol using directional antennas, which is source-based. In [18], Yang, Wu, and Dai put forward the directional network backbone for efficient broadcasting using the directional antenna model in a static manner where the backbone is suitable for any source node in the network. They designed the concept of a directional connected dominating set (DCDS) for the construction of a directional network backbone. DCDS extended the CDS approach for broadcast with the help of directional antennas. The minimum DCDS problem is proved to be NP-complete. Using DCDS, not only forwarding nodes but also forwarding edges of each forwarding node are designated. The total energy consumption is reduced, as well as the interference. They developed the node and edge coverage condition for the DCDS problem. All of the above schemes assume an omnidirectional reception mode.

III. BROADCAST WITH NETWORK CODING AND DIRECTIONAL ANTENNAS

In this section, we first extend the approach developed in [18] to construct the directional connected dominating set (DCDS) to a dynamic version, where the constructed DCDS is source-based. We then combine the network coding with the dynamic DCDS to develop the EBCD.

A. Dynamic Directional Connected Dominating Set (DDCDS)

In [18], the concept of using a directional network backbone for efficient broadcast in conjunction with directional antennas was proposed. The omnidirectional transmission range of each node is divided into K sectors and each forwarding node only needs to switch on several sectors for transmission while the entire network gets the broadcast message. The directional connected dominating set (DCDS) is proposed for the construction of a directional network backbone, where each node determines locally not only its status of forwarding or non-forwarding, but also its forwarding outgoing edges if it is a forwarding node. Note that the network is modeled

as a directed graph. Then in a broadcast initiated from any source node, the source uses ominidirectional transmission (or directional transmission if it detects a forwarding node in that direction) to send the message to a neighboring forwarding node. Then forwarding nodes forward the message towards only their corresponding forwarding edges, and the entire network gets the message. The DCDS is a directional network backbone assuming that K is infinite, and each outgoing edge is a transmission sector. When K is finite, the sectors that contain selected forwarding edges are simply switched on for transmission to get a directional network backbone. Note that when K is 1, the DCDS problem turns into the CDS problem.

A minimum DCDS problem is to find a DCDS with the least forwarding edges which is proved to be NP-complete. If the energy consumption of transmission in any direction is fixed, reducing the number of forwarding edges guarantees the smallest energy consumption in the application of broadcasting using directional antennas.

The *node/edge coverage condition* proposed in [18] constructs a DCDS for a given network locally at each node in a static manner. The constructed DCDS is for any source node in the network. After the exchange of "Hello" messages, each node makes a decision based on only local topology information in the initialization phase before the broadcast application starts. Here, we extend this method to a dynamic version, where each node makes a decision based not only on topology information but also broadcast state information piggybacked in received broadcast messages. It decides its forwarding status and corresponding forwarding edges for each received broadcast message.

In our proposed *dynamic node/edge coverage condition*, each broadcast message piggybacks with it the information of its q most-recently visited nodes (q is a small number such as 2 or 3). A *visited node* for a message is a node that has forwarded the message. Correspondingly, a *visited edge* for a message is an edge that has forwarded the message. Then, when a node applies the coverage condition to determine its status for a received message, it considers the information of visited nodes/edges of this message as well as local topology information. The dynamic version of the node and edge coverage conditions resemble the static ones [18] except that the node and edge priorities are updated based on the piggybacked broadcast state information. Note that the updated new priority is only valid for the corresponding message. Therefore, a node may have a different status (visited or not, forwarding or not) and priorities for different messages. In the following, an unmarked status represents a non-forwarding status. A dominating neighbor means that there is an incoming edge from that neighbor. A absorbant neighbor means that there is an outgoing edge to that neighbor. Note that each node v has a priority $p(v)$ and such a priority is totally ordered within its h-hop neighborhood, which could be the node ID, node degree, or energy level based on different applications.

Dynamic Node Coverage Condition. *Node v is unmarked if, for any two dominating and absorbant neighbors, u and w, a directed replacement path exists connecting u to w such that (1) each intermediate node on the replacement path has*
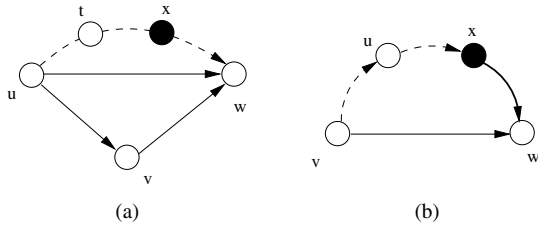
Fig. 2. Directed replacement paths in (a) node coverage, and (b) edge coverage with visited nodes.

Fig. 3. (a) Forward nodes, (b) forwarding nodes and forwarding edges.

a higher priority than v (including visited nodes), and (2) u has a higher priority than v if there is no intermediate node.

Edge Priority Assignment. *For each edge $(v \to w)$, the priority of this edge is $P(v \to w) = (P(v), P(w))$.*

The priority of an edge is a tuple based on the lexigraphic order. The first element is the priority of the start node of this edge and the second one is the priority of the end node. Therefore, there is a total order for all the edges. For example, $P(x \to y) > P(w \to v)$, if and only if, $(P(x) > P(w))$ or $(P(x) = P(w)$ and $P(y) > P(v))$.

Dynamic Edge Coverage Condition. *Edge $(v \to w)$ is unmarked if a directed replacement path exists connecting v to w via several intermediate edges with higher priorities than $(v \to w)$ or visited edges.*

As shown in Figure 2, v is the current node and black nodes are visited ones. Assume the priority is based on the alphabetic order, i.e., $P(a) > P(b)$. (a) shows two types of directed replacement paths from u to w using the node coverage condition. When u is directly connected to w, $P(u) > P(v)$ is necessary. Otherwise, when there are intermediate nodes t and x, then $P(t) > P(v)$ and $P(x) > P(v)$ since x is visited. (b) shows the directed replacement path for edge $(v \to w)$. In this case, both the intermediate edges $((v \to u)$ and $(u \to x))$ have higher priorities than the edge $(v \to w)$. Since edge $(x \to w)$ is visited, the edge $(v \to w)$ can be unmarked. The difference between dynamic and static node/edge coverage conditions is that a visited node/edge has a higher priority node/edge. Note that the dynamic node/edge coverage conditions need h-hop information which means h-hop local topology information and q-hop piggybacked visited node/edge information in each received message. For example, as in Figure 1 (a), if $h = 2$, node a knows all the edges in the network except the edge between nodes c and d.

Theorem 1: Given a directed graph $G = (V, E)$, V' and E' generated by the dynamic node and edge coverage conditions in a broadcast guarantee the full delivery.

The proof of Theorem 1 is in the Appendix. The example in Figure 3 shows a source-based CDS (a) in shaded nodes (s is the source), (b) is the result after applying the dynamic node/edge coverage condition. Nodes b and c are also forwarding nodes. b selects edge $(b \to d)$ as forwarding edge and c selects edges $(c \to e)$ and $(c \to f)$. The edge $(b \to c)$ can be unmarked because a replacement path connecting b to c via s

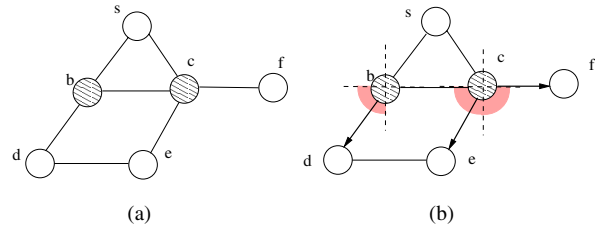exists, and s is a visited node and $(s \to b)$ is a visited edge. Therefore, the priorities of edges $(b \to s)$ and $(s \to c)$ are both higher than that of $(b \to c)$. The same is true for edge $(c \to b)$. Then if K is finite, only the sectors that contain the bold arrows need to be switched on for transmission, much like the grey sectors in (b).

B. Efficient Broadcasting Using Network Coding and Directional Antennas (EBCD)

In this subsection, we combine the network coding and directional antenna approaches into the broadcast application, exploiting the advantages of both of them.

Algorithm 1 describes EBCD executed on a node. Before the broadcast starts, each node exchanges "Hello" information with neighbors for h rounds to get the h-hop local topology information. Upon the arrival of the first message, a timer is setup and the piggybacked information in each received message is recorded to update the node priorities. When the timer expires, for each received message, the node/edge coverage conditions are applied based on the topology and broadcast state information (new priorities), and forwarding status and edges of the node are determined. We use the example in Figure 4 to illustrate the procedure. This is the same example as in Figure 1. (a) is the result of DDCDS after step 4 of Algorithm 1. Node e is the forwarding node for messages A, B, C, and D from nodes a, b, c, and d based on the dynamic node coverage condition. Edge $(e \to a)$ is a forwarding edge for messages C and D. Edge $(e \to b)$ is a forwarding edge for message D. Edge $(e \to c)$ is a forwarding edge for message A. Edge $(e \to d)$ is a forwarding edge for message A and B.

In step 5, when the timer expires, node e circumgyrates its directional antennas to let the edge of a sector align to each forwarding edge. There are at most f layouts when the number of selected forwarding edges is f. In each sector of each layout, network coding is applied to determine the final transmissions. The layout with the fewest total transmissions is then selected for use. The node then executes the forwarding. In the algorithm, we assume that steps 4, 5, and 6 can be completed before the arrival of the next message.

In EBCD, network coding is applied in each sector of a layout instead of the entire node as in [6]. We use the XOR-based algorithm from [6]. Assuming $m_1, m_2, \ldots, m_l$ are messages received in order in this sector. $P_1, P_2, \ldots, P_t$ are the final forwarded messages (original or coded). $P_1 = m_1 \oplus \ldots \oplus m_{i_1}$, $P_2 = m_{i_1+1} \oplus \ldots \oplus m_{i_2}$, ..., $P_t = m_{i_t+1} \oplus \ldots \oplus m_l$, where

Algorithm 1 EBCD algorithm at node v.

Before broadcast:
1. Exchange "Hello" messages to update local topology.
Upon reception of the first message (before the timer setup):
2. Setup the timer.
3. Update neighborhood node priorities based on each received message.
4. When timer expires, apply dynamic node/edge coverage conditions for each message.
5. If v is a forwarding node for some messages,
    (1) align the edge of a sector to each forwarding edge,
    (2) determine coded messages in each sector using coding,
    (3) select the position with the fewest total transmissions.
6. Forward coded messages.

Fig. 4. (a) DDCDS, (b) coding, (c) and (d) $K = 2$, (e) and (f) $K = 4$.

each neighbor can decode from P_1 to P_t to get any missing message from m_1 to m_l. A greedy approach can be used. For the received messages in a queue, the algorithm tries to have the maximum number of messages starting from m_1 to create P_1, then to create P_2 and so on. For example, in Figure 4 (b), assuming that the broadcast messages arrive in the order of A, C, B, and D at node e. P_1 is A at first, then e tries to make $P_1 = A \oplus C$. Node a needs message C and nodes d and c need message A. With P_1, all of them can decode. Therefore $P_1 = A \oplus C$ is a correct coding. Then e can try $P_1 = A \oplus C \oplus B$. Since node d needs message B, and it cannot decode P_1 to get B, this is not a correct coding. P_1 remains as $A \oplus C$. Using the same procedure, we can get $P_2 = B \oplus D$.

Figure 4 (b) is the result using only network coding, where e is the forwarding node and forwards the combined message P_1 ($= A \oplus C$) and P_2 ($= B \oplus D$) omnidirectionally. (c) is one layout of EBCD using $K = 2$. Then e needs to transmit C and D in the left sector and A and B in the right sector. (d) is another layout for $K = 2$, where e transmits P_1 and P_2 to the upper sector and P_3 ($= A \oplus D$) to the lower sector. (e) and (f) show the case where K is 4 with different layouts. If we assume that the transmission of one message in a 90° sector costs one unit of energy, the energy consumption in the figures from (b) to (f) are 8, 8, 6, 6, and 5. We can see that the combination of network coding and directional antennas can improve broadcasting performance significantly in terms of energy consumption. Note that the forwarding of node e without network coding or directional antennas costs 16.

The entire procedure can also be illustrated using Figure 5 (a), where m_1 to m_6 are received messages and D_1 to D_6 are the corresponding forwarding nodes/edges decisions for them based on topology and priority information. U means to update the priority information based on the piggybacked information in the received message and D_c is the final transmission decision for several received messages using network coding in a valid timer. Note that the duplicated reception of a processed message is simply discarded such as m_5 received after it has been processed and forwarded. As shown in the figure, the arrival of m_5 after timer 2 expires will not intrigue a new timer or a new updating during the timer 3 period.
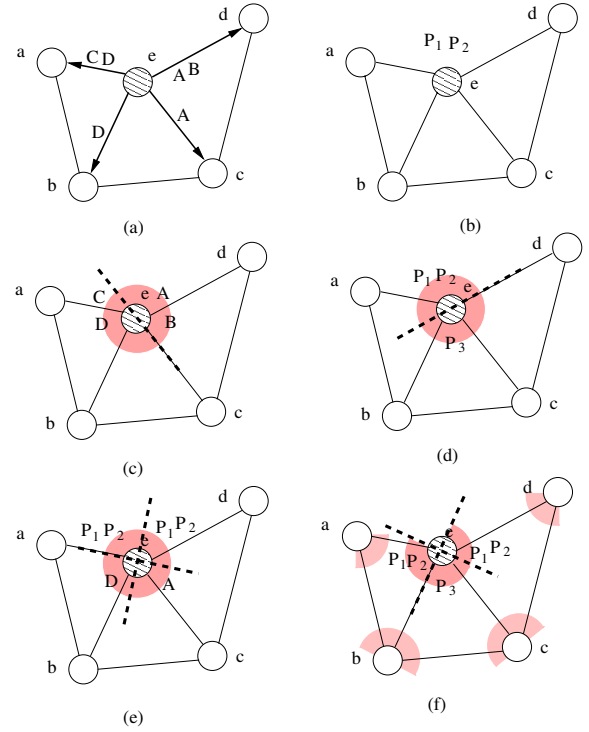
The source nodes in the network can simply use omnidirectional transmission to send out the broadcast messages. In order to further reduce the total energy consumption, source nodes can only switch on sectors in which there are neighbors for transmission. In this case, the message can arrive at at least one forwarding node as well as other non-forwarding neighbors, which helps with the potential network coding conducted later on. As shown in Figure 4 (f), source nodes a, b, c, and d select some sectors to switch on for transmission, shown in the light grey sectors.

IV. EXTENSIONS OF EBCD

A. Static vs. Dynamic Forward Node Selection

As mentioned above, in [6], Li et al. applied network coding to a dynamic forwarding node selection approach, the PDP-based approach, and stated that the coding can be directly applied to other localized deterministic approaches for broadcasting. The previously proposed EBCD also uses a dynamic forwarding status approach. Here, we extend the proposed EBCD to a static forwarding node selection approach to analyze their overall performance.

In the static version of EBCD, we apply the coding to the static forwarding node/edge selection, the node/edge coverage conditions in [18], as shown in Algorithm 2. We will compare the performance and tradeoffs of these two algorithms in the simulation section. As in Algorithm 2, in the initialization phase before the broadcast starts, local information is collected via the exchange of "Hello" messages. Then the node determines its forwarding status. This status is for all of the following received messages. Then a timer is setup when the
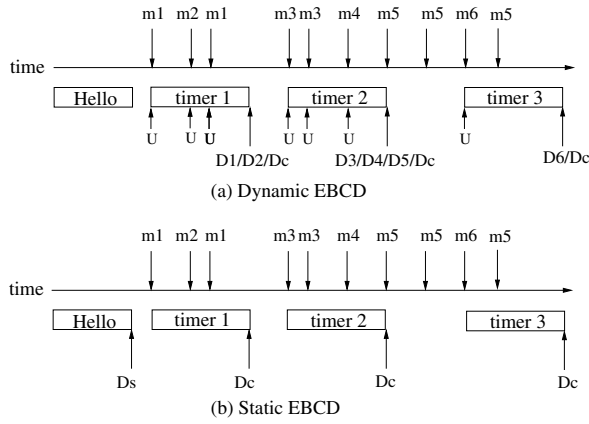
Fig. 5. Illustration of execution procedure of dynamic/static EBCD.

first message arrives. When the timer expires, network coding is applied in each sector of each layout, and the best one is selected for use.

The entire procedure is illustrated in Figure 5 (b). After the exchange of "Hello" messages, D_s determines the status of the node and also the selected forwarding edges if it is a forwarding node for all the following received messages. Then upon the reception of the first message, a timer is setup. When the timer expires, coding is applied to all received messages to determine the final coded messages for transmission.

The size of the forwarding node set selected using a dynamic manner is smaller than or equal to the one by a static manner, since in the former one the information of visited nodes helps to increase the probability of the node being an non-forwarding node. However, redundant transmissions by the extra forwarding nodes in the static manner may help to increase the potential network coding in the later phase and hence the overall performance, which will be verified via simulation. The obvious advantage of the static EBCD is less overhead. The broadcast messages do not need to piggyback the broadcast state information. Also, as shown in Figure 5 (b), fewer forwarding nodes/edges decisions need to be made.

B. Single Source/Single Message Broadcast

As mentioned above, broadcasting using the network coding method [6] is designed for the application of multiple sources with multiple broadcast messages, where a forwarding node has the potential of combining some of the messages to reduce the number of transmissions. In the application of a single source, only when the rate of the generation of messages is large enough, the network coding may work in nodes relatively far away from the source node.

We design two approaches for the single source/single message broadcast issue. We assume a single source in the broadcast and the rate of the generation of messages is large enough that it can be viewed as single message application. The basic idea is to divide the single message into several segments and treat each segment as a single broadcast message.

1) Pipeline-Based Approach (PB): When there is only one source node in the network broadcasting one message, the source node divides the message into k segments and sends each segment as a single broadcast message, and broadcasts them one-by-one in the pipelined manner. In this way, the single source/single message problem turns into single source with multiple message broadcast. In the area near the source node, the effect of network coding is not significant since all the segments tend to come from one direction. However, in the farther area, the effect is expected to be significant. As shown in Figure 6 (a), s divides the broadcast message into k segments and sends them out via k broadcasting. Therefore, the neighbors of source node s get the first broadcast message S_1, then the second one S_2 in order from s.

2) Spread-Out Approach (SO): In order to enhance the effect of network coding in the single source/multiple message broadcast using the message segmentation method, we can further apply the message spread method to first spread the segments out into the entire network. After the source node divides the outgoing message into k segments, it uses random walk to spread the $k-1$ of these segments. Some kind of TTL control can be used to make sure the segments randomly scatter out into the network. Upon arriving at a destination, a segment is broadcast by the destination node. The source itself keeps a segment for later broadcast. As shown in Figure 6 (b), the $k-1$ segments are spread in the entire network. In this way, the application turns into the multiple source with multiple messages broadcast. Although the unicast in the preprocessing phase costs extra overhead, the transmission reduction earned from network coding in the entire network is expected to be more significant. Note that the nodes on the unicast routes can mark themselves as visited nodes for the bypassing segments, which helps to potentially reduce transmission.

V. IMPLEMENTATION ISSUES

In this section, we discuss some implementation techniques in the above proposed algorithms.

A. Neighborhood and Piggybacked Information Collection

Note that no GPS assistance is necessary in the proposed algorithm. In Algorithm 1, each node sends out "Hello" messages K times to the K directions and accomplishes the directional neighborhood discovery. In this case, after h rounds of the message exchange, each node knows its h-hop neighborhood information, which includes both neighbors and in which sectors are these neighbors. According to this information, each node can create the neighbor reception table.
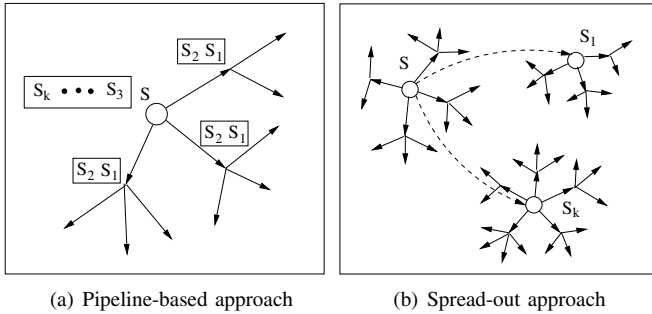
(a) Pipeline-based approach   (b) Spread-out approach

Fig. 6.   Single source/single message broadcast.

(a) Dense network ($d = 18$)   (b) Sparse network ($d = 6$)

Fig. 7.   Comparison of EBCD, Coding, and CDS in number of transmissions.

After a node determines its status together with the forwarding edges, it piggybacks this information in the broadcast message as part of the q most recently visited node information. The node that receives this broadcast message can extract the "visited nodes/edges" information from it.

B. Timer in EBCD

A timer is set for each node to collect several broadcast messages. In the static version of EBCD, it helps with the potential network coding. In the dynamic version, it also helps to collect more broadcast state information piggybacked by these messages to determine the status of the node. The timer selection presents the performance tradeoff between energy consumption and delay. When the timer is set to 0, the effect of network coding almost reduces to 0. When the timer is large enough to counteract the difference of initial time among the broadcast messages in the network, the network coding can be utilized thoroughly. After the forwarding, the timer is reset for the next session. The value of the timer can be set in both a proactive and a reactive way. In the former method, the timer of a node can be set based on the number of neighbors of this node and the diameter of the network. In the latter one, a node can adjust the value of the timer on-the-fly according to the message arrival rate at this node.

VI. SIMULATION

In this section, we evaluate the performance of the proposed EBCD algorithm, as result of the average of 100 simulation trials, by comparing the total energy consumption in terms of the number of message transmissions in the network and also the size of sectors that the message was transmitted. We compare EBCD with two algorithms. (1) the algorithm without network coding or directional antennas (CDS). We simply call it algorithm CDS since the forwarding node set selected by this method is a source-based CDS, which means together with the source node, the forwarding node set forms a CDS for the network. We use a dynamic node coverage condition as used in our EBCD. (2) the algorithm with network coding but without directional antennas (Coding). This is the approach proposed in [6], but in order to make a fair comparison, the underlying forwarding node selection approach we use in Coding is also the dynamic node coverage condition. We also compare EBCD with the proposed static EBCD (S-EBCD) to check the performance variation. Finally, the performance
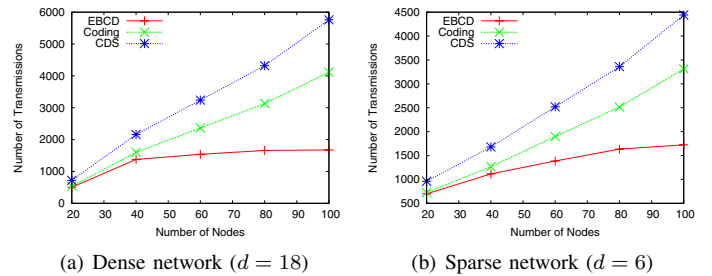
of the two approaches for the single source/single message broadcast application is evaluated, the pipeline-based approach (PB) and the spread-out approach (SO).

A. Simulation Environment

The simulations are conducted on a custom simulator. In the simulation, n nodes are randomly placed in a restricted 100×100 area. The tunable parameters in the simulation are as follows. (1) The number of nodes n. We vary the number of deployed nodes from 20 to 100 to check the scalability of the algorithms. (2) The average node degree d which represents the density of the network. We use 6 and 18 as the values of d to generate sparse and dense networks. (3) The number of sectors of the antenna pattern K. We use 4 and 6 as the values of K. (4) The number of broadcast sessions b, i.e., the number of generated broadcasting messages. b has a fixed value of 20 in the simulation. Therefore, when n is different, we can simulate various data loads in the network. The source nodes are randomly selected. (5) The number of segments k in the PB and SO extensions. k is 10 and 20 in the network. We do not consider node mobility and signal interference in the following simulations.

The following metrics are compared: (1) the number of transmissions in the application. We assume that the transmission of a message (original or coded) following a transmission edge is one transmission, and (2) the average energy consumption for a broadcast message. We assume that one transmission (of a original broadcast message or a coded message) in each sector consumes one unit of energy.

B. Simulation Results

Figure 7 is the comparison of EBCD, Coding, and CDS in the number of transmissions in both dense and sparse networks. (a) is the dense network where the average node degree is 18. We can see that Coding can reduce the number of message transmissions, and the reduction rate is around 1.2. EBCD can further reduce it significantly. When the number of nodes increases, the number of transmissions in EBCD tends to be stable. (b) is when the average node degree is 6. EBCD can still reduce the number of transmissions compared with CDS or Coding. But the reduction rate is lower than that in the dense network.

Figure 8 is the comparison of EBCD, Coding, and CDS in terms of energy consumption when K is 4 and 6. (a) and (b) are in dense networks. We can see that EBCD can
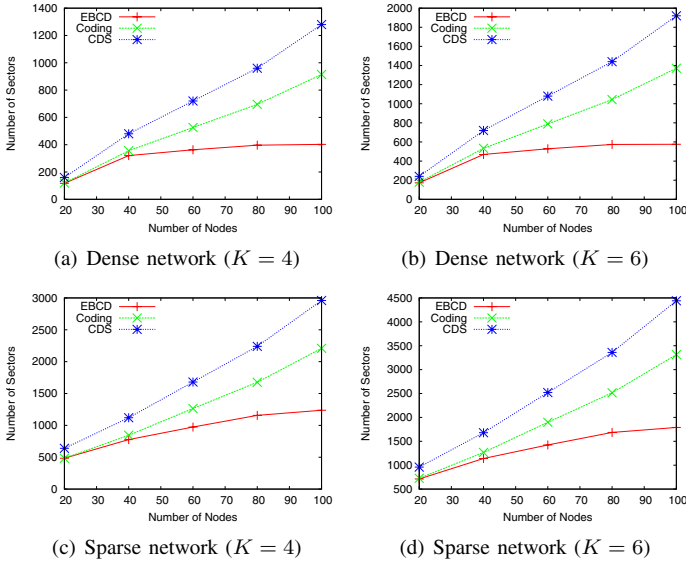
(a) Dense network ($K = 4$) (b) Dense network ($K = 6$)

(c) Sparse network ($K = 4$) (d) Sparse network ($K = 6$)

Fig. 8.   Comparison of EBCD, Coding, and CDS in energy consumption.

(a) Forwarding nodes ($d = 18$) (b) Forwarding nodes ($d = 6$)

(c) Number of transmissions ($d = 18$) (d) Number of transmissions ($d = 6$)

Fig. 9.   Comparison of EBCD and S-EBCD.

further reduce the number of switched on sectors compared with CDS and Coding in which all sectors of a forwarding node need to be switched on for transmission. When K is larger, the reduction rate of EBCD over CDS and Coding is more significant since a larger forwarding area can be pruned. (c) and (d) are in the sparse networks. EBCD also reduces the number of switched on sectors significantly. The larger the value of K, the larger the reduction rate.

Figure 9 is the comparison of EBCD and S-EBCD in terms of the number of forwarding nodes and transmissions in both dense and sparse networks. We can see that although the number of forwarding nodes selected in the static method should be larger than that in the dynamic one, as shown in (a) and (b), the final numbers of transmissions in EBCD and S-EBCD are very close, especially when the network is relatively dense. This is because more forwarding nodes to forward increases the probability of network coding, which makes up for the larger forwarding node set. The forwarding node set of S-EBCD is around 1.3 times larger than that of EBCD while the final number of transmissions is 1.03 times higher. The advantage of S-EBCD is that it only calculates the status of each node once for any broadcast message from any source. It is also unnecessary to piggyback the broadcast state information. Therefore, if the network is dense, S-EBCD is preferred since the overhead of it is smaller while the performance is comparative.

Figure 10 shows the performance evaluations of the two extensions of EBCD, PB and SO with different segment numbers $k = 10, 20$. We can see that in (a) when the network is dense, which means the transmission range is larger, SO has better performance than PB. Smaller k makes the advantage of SO over PB more significant. When k is 20, SO is very close to PB. Because the number of transmissions is larger with larger number of segments in the initial phase of SO. In (b), the results in the sparse network are shown. When k is
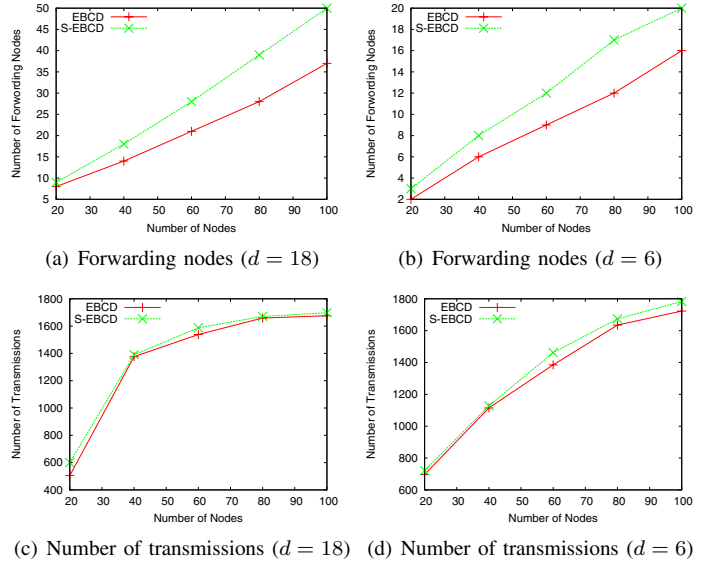
10, SO is better than PB. When k is 20, PB is even better than SO. This is because the initial phase of SO costs a lot of overhead in the case of larger k and smaller transmission range, which leads to more hops to spread out the segments.

The simulation results can be summarized as follows.

1) EBCD has significant performance improvement in terms of the number of transmissions compared with CDS and Coding, especially in relatively dense networks.

2) EBCD has better performance than CDS and Coding in terms of the number of switched on sectors which corresponds to the energy consumption. The larger the value of K, the larger the reduction rate of EBCD over the other two methods.

3) S-EBCD has very close performance to EBCD, especially when the network is relatively dense. Therefore, due to its other advantage, such as less overhead, S-EBCD is another option.

4) SO has better performance than PB when the network is relatively dense and the number of segments a message is divided into is small. When in sparse networks with large k, PB even has a better performance than SO.

VII. Conclusions

Network coding has been exploited for efficient broadcasting to further reduce the number of transmissions in the multiple source broadcast application. In this paper, we combine the network coding-based broadcast approach with broadcasting using directional antennas for a more efficient broadcast strategy, developing efficient broadcasting using network coding and directional antenna algorithm (EBCD). We extend existing broadcasting using the directional antenna approach to a dynamic mode. Although the coding-based approach is independent of the underlying forwarding node selection procedure, we show that different forwarding node selections
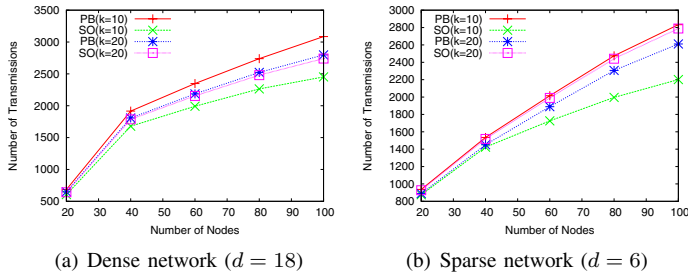
(a) Dense network ($d = 18$)   (b) Sparse network ($d = 6$)

Fig. 10.   Comparison of PB and SO in number of transmissions.

Fig. 11.   Proof of dynamic node/edge coverage conditions.

affect the overall performance significantly. We also discuss the single source/single message application and design two approaches as the extension of the proposed algorithm for it. Performance analysis is conducted through simulations. The proposed EBCD approach has better performance than traditional CDS-based broadcast and the existing network coding-based broadcast in terms of energy consumption. Also, the static version of EBCD has comparative performance in energy conservation with smaller overhead. In the future, we will improve the robustness for mobility and MAC layer interference of the proposed approaches and perform more comprehensive simulation considering a mobile environment.

REFERENCES

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, (4):1204–1216, 2000.
[2] F. Dai and J. Wu. Efficient broadcasting in ad hoc wireless networks using directional antennas. *IEEE Transactions on Parallel and Distributed Systems*, (4):1–13, 2006.
[3] C. Fragouli, J. Widmer, and J.-Y L. Boudec. A network coding approach to energy efficient broadcasting: from theory to practice. In *Proc. of IEEE INFOCOM*, 2006.
[4] C. Hu, Y. Hong, and J. Hou. On mitigating the broadcast storm problem with directional antennas. In *Proc. of IEEE ICC*, 2003.
[5] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard. The importance of being opportunistic: Practical network coding for wireless environments. In *Proc. of ACM SIGCOMM*, 2006.
[6] L. Li, R. Ramjee, M. Buddhikot, and S. Miller. Network coding-based broadcast in mobile ad hoc networks. In *Proc. of IEEE INFOCOM*, 2007.
[7] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, (2):111–122, 2002.
[8] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proc. of ACM MobiHoc*, 2000.
[9] S. Pleisch, M. Balakrishnan, K. Birman, and R. Renesse. MISTRAL: Efficient flooding in mobile ad-hoc networks. In *Proc. of ACM MobiHoc*, 2006.
[10] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. In *Proc. of 35th Hawaii Int'l Conf. on System Sciences (HICSS-35)*, 2002.
[11] C. C. Shen, Z. Huang, and C. Jaikaeo. Directional broadcast for ad hoc networks with percolation theory, *Technical report*, Computer and Information Sciences, University of Delaware. 2004.
[12] D. Simplot-Ryl, J. Cartigny, and I. Stojmenovic. An adaptive localized scheme for energy efficient broadcasting in ad hoc networks with directional antennas. In *Proc. of 9th IFIP PWC*, 2004.
[13] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, (1):14C25, 2002.
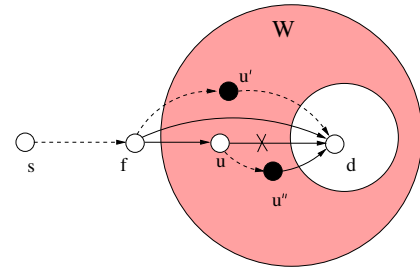[14] J. Sucec and I. Marsic. An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks. In CAIP T*echnical R*eport 248, 2000.
[15] Y. C. Tseng, S. Y. Ni, Y. S. Chen, and J. P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, (2-3):153C167, 2002.
[16] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *IEEE Transactions on Computers*, (10):1343–1354, 2004.
[17] J. Wu and H. Li. On calculating connected dominating sets for efficient routing in ad hoc wireless networks. In *Proc. of ACM DIALM*, 1999.
[18] S. Yang, J. Wu, and F. Dai. Efficient backbone construction methods in MANETs using directional antennas. In *Proc. of IEEE ICDCS*, 2007.

VIII. APPENDIX

Proof of Theorem 1. If we can prove that, given source node s, for any node d in the network, there is a path with all intermediate nodes and edges designated to forward, we prove that a full delivery is achieved. We assume that all of d's neighbors form a ring area as an "outer rim" of node d, like the gray area W in Figure 11. Note that W is not empty. We assume that node u is of the highest priority in area W. If we can prove that either u is a forwarding node and $(u \rightarrow d)$ is a forwarding edge, or one of d's neighbors is a visited node and it forwards the message to d, we contradict the assumption that d cannot be reached from s. We make the two assumptions that either u is not a forwarding node or u is a forwarding node but edge $(u \rightarrow d)$ is not a forwarding edge. We find contradictions for these two cases.

Case 1: u is not a forwarding node. Therefore, for a neighbor f of u, according to the dynamic node coverage condition, either (a) there is a replacement path connecting f to d with at least one intermediate node on it, u', or (b) f directly connects to d, and the priority of f is higher than that of u. u' cannot have a higher priority than u since u is the neighbor of d with the highest priority. If u' is a visited node, d can be covered by u. For b, if f is also neighbor of d, it cannot have higher priority than u.

Case 2: u is a forwarding node but edge $(u \rightarrow d)$ is not a forwarding edge (an "×" is on the edge). A path connecting u to d must exist, with all the edges with higher priorities than $(u \rightarrow d)$ or visited. Since u is the highest priority node, u'' on the path cannot be higher and has to be a visited node with a forwarding outgoing edge connecting to d. In that case, d can be covered.

All of the contradictions above show that d can be reached from the source node s.   □