Many-to-All Priority-Based Network-Coding Broadcast in Wireless Multihop Networks

Chieh-Hao Chang, Jung-Chun Kao, Fu-Wen Chen, and Shih Hsun Cheng
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan

***Abstract*—This paper addresses the minimum transmission broadcast (MTB) problem for the many-to-all scenario in wireless multihop networks and presents a network-coding broadcast protocol with priority-based deadlock prevention. Our main contributions are as follows: First, we relate the many-to-all-with-network-coding MTB problem to a maximum out-degree problem. The solution of the latter can serve as a lower bound for the number of transmissions. Second, we propose a distributed network-coding broadcast protocol, which constructs efficient broadcast trees and dictates nodes to transmit packets in a network coding manner. Besides, we present the priority-based deadlock prevention mechanism to avoid deadlocks. Simulation results confirm that compared with existing protocols in the literature and the performance bound we present, our proposed network-coding broadcast protocol performs very well in terms of the number of transmissions.**

***Keywords*—broadcast, energy efficiency, network coding, wireless networks**

I. Introduction

Broadcasting is a fundamental operation in wireless multihop networks, which allows a node to disseminate data to all other nodes. Naive broadcast schemes often are inefficient. For example, flooding causes severe waste of channel use and severe packet collision, which leads to the broadcast storm problem [1]. To avoid broadcast storm, a crucial issue is to develop a one-to-all broadcast scheme with fewest transmissions, which is referred to as the minimum transmission broadcast (MTB) problem [2]. When there are multiple source nodes in a wireless network, the broadcast storm becomes severer. This motivates this paper to study the MTB problem for the many-to-all scenario.

Network coding [3] is a promising technique to improve transmission efficiency. Instead of relaying received packets separately, network coding enables nodes to combine several packets and send out fewer combined/coded packet. It has been shown that network coding can effectively reduce the number of transmissions needed in broadcast [4],[5],[6],[7],[8].

A challenge of network-coding-based broadcast is, upon each packet reception, to determine whether to forward the received packet immediately or to wait subsequent packets to gain coding opportunity. Take in Fig. 1a as an example. Node 3 had better wait until both p_i and p_j arrive and sends out one coded packet $p_i + p_j$; otherwise, coding opportunity is lost and node 3 has to forward p_i and p_j separately, which take two transmissions. On the contrary, nodes 1 and 2 should forward what they receive immediately, because their waiting results in a longer delay but gains no benefit.

The goal of this paper is to design a network-coding-based many-to-all broadcast protocol, which decides, at run time, to forward received packets immediately or to wait an additional period of time to gain coding opportunity so as to minimize the total number of transmissions. Our contributions are two-fold. First, we relate the many-to-all-with-network-coding MTB problem to a maximum out-degree problem. The solution of the latter can serve as a lower bound for the number of transmissions in practice. Second, we propose a distributed protocol, called *p*riority-based *n*etwork-*c*oding *b*roadcast protocol (PNCB). PNCB constructs efficient broadcast tree in a distributed manner and dictates nodes to transmit packets in a network coding fashion. To prevent deadlock, PNCB has a priority-based deadlock prevention mechanism built in.

The remainder of this paper is organized as follows. Section II describes the related work. The network model is described in Section III. Section IV presents a lower bound of the MTB problem for the many-to-all with network coding scenario. Section V presents our proposed protocol, which is evaluated in Section VI. Section VII gives some concluding remarks.

II. Related Work

The minimum transmission broadcast (MTB) problem for the one-to-all scenario is to find a set of forwarding nodes (*i.e.,* forwarders) such that all nodes in the network receive a broadcast packet originated from the source, whereby the number of transmissions is minimized. This problem is NP-hard since it is equivalent to the NP-hard problem, the minimum connected dominating set (MCDS) problem [9],[10]. A number of heuristic/approximation algorithms have been proposed. These schemes can be classified into either probabilistic approaches where packets are forwarded with a given probability, or deterministic approaches where a connected dominating set of forwarders is formed.

A challenge in probabilistic approaches [11] is to determine a proper value of the forwarding probability (aka the forwarding factor). There is a tradeoff between efficiency and reliability of data dissemination. A challenge in deterministic approaches is to find as few forwarders as possible in a deterministic and distributed manner while guaranteeing full delivery. A number of approximation algorithms [12],[13] have been proposed.

Network coding [3],[9],[10],[11],[14] which allows combining packets before forwarding them, has been shown its ability to improve the multicast/broadcast efficiency. With

This research was supported in part by National Science Council, Taiwan, under grant no. NSC 102-2221-E-007-011.

978-1-4799-1297-1/14/$31.00 ©2014 IEEE

network coding, the minimum-energy multicast problem (which subsumes the MTB problem) is proven solvable in polynomial time [15],[16]; a centralized, polynomial-time solution is introduced in [16]. Many real applications prefer decentralized methods; thus distributed implementations of network coding are proposed in [17] for single-source multicast and in [4],[5] for 1-to-all broadcast.

For the many-to-all scenario, there are probabilistic approaches based on network coding. Widmer *et al.* in [6] theoretically quantified the energy savings that network coding offers in two topologies and proposed low-complexity distributed algorithms. Practical protocols for all-to-all broadcast are proposed in [7].

There are also deterministic approaches. Li et al. proposed CODEB [8], which selects forwarders by partial dominant pruning (PDP) over two-hop topology. CODEB has two network coding algorithms, XOR-based coding and Reed-Solomon coding. However, the coding gain of CODEB is restricted because the forwarder selection process does not consider coding opportunities at all. This motivates our study on tree-based many-to-all network coding in which forwarders (and broadcast trees) are selected so as to maximize coding opportunities.

III. Network Model

We consider the many-to-all scenario in which a number of source nodes disseminate their data to all nodes in a wireless network. Define n as the total number of nodes in the network and m as the number of source nodes. The set of all nodes in the network is denoted by $V = \{i \mid 1 \le i \le n\}$, and the set of source nodes is denoted by $S = \{i \mid 1 \le i \le m\}$.

Node u is a (1-hop) neighbor of node v if node u is within the transmission range of node v. The set of all neighbors of a node, say node v, is denoted by $N(v)$. It is assumed that each node knows the information about its neighbors. This can be achieved, for example, by sending hello messages periodically.

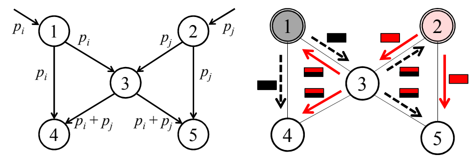
Any pair of neighbor nodes is assumed to have a reliable link connecting each other. Reliable data delivery over unreliable links can be done by a number of acknowledgement-based retransmission schemes [18] and collision resolution strategies [19].

Source nodes may generate packets at equal or unequal rates. This paper focuses on the the *equal-rate model*, in which all sources generate packets at the same rate. Without loss of generality, we assume each source generates one packet in a session and define d_k as the packet generated at source node $k, k \in S$.

IV. Problem Formulation and Lower Bound

The MTB problem for the many-to-all scenario under the equal-rate model, abbreviated as the *many-to-all MTB problem*, aims to find a many-to-all broadcast scheme which minimizes the total number of transmissions. Without network coding, this problem is equivalent to find the minimum connected dominating set (MCDS) for each source node. The MCDS rooted from a source is actually a spanning tree with fewest internal nodes; only internal nodes forward the broadcast packet. So the minimum number of transmissions in the many-to-all MTB problem is the sum of the numbers of the internal nodes in the MCDS trees constructed for all source nodes.

Leverage network coding can further decrease the number of transmissions in the many-to-all MTB problem. Take Fig. 1b as an example, where MCDS trees for source nodes 1 and 2 are drawn in black dotted arrows and red solid arrows, respectively. Without network coding, the minimum number of transmissions is equal to the sum, $2+2=4$. With network coding, node 3 can transmit a coded packet, rather than forwarding the two broadcast packets; therefore, only 3 transmissions are needed in this example.



(a) (b)

Fig. 1. (a) An example of network coding in a butterfly network. (b) Broadcast trees for sources 1 and 2. Drawn by red solid arrows is the tree rooted from node 1. Drawn by black dotted arrows is the tree rooted from node 2.

Observed from the example, the number of transmissions highly depends on the maximum out-degree of each node. The out-degree of node i to node j, denoted by B_{ij}, is defined as the number of directed edges from node i to node j. The maximum out-degree of node i, denoted by B_i^M, is defined as $\max_j B_{ij}$ over all $j \in N(i)$. Take node 3 in Fig. 1b as an example: $B_{31} = B_{32} = B_{34} = B_{35} = 1$ and $B_3^M = 1$. Clearly, the number of transmissions node i takes for many-to-all data dissemination is lower bounded by B_i^M. For example, node 3 which receives two packets needs to send out at least $B_3^M = 1$ packet. An efficient way for node 3 is to send one coded packet, instead of respectively forwarding the two packets.

B_i^M is a tight lower bound for the number of transmissions node i takes. Usually, node i takes exactly B_i^M transmissions, but there are some topologies in which node i may take more than B_i^M transmissions, as discussed later in Section V.B.

Obviously, the sum of maximum out-degrees over all nodes is a lower bound for the total number of transmissions. Our goal is to construct a many-to-all broadcast scheme, consisting of a set of broadcast trees for individual source nodes, such that the sum of maximum out-degrees over all nodes is minimized.

V. PNBC Protocol Design

This section presents a fully-distributed and efficient approach, called the *p*riority-based *n*etwork *c*oding *b*roadcast (PNCB) protocol. PNCB consists of two phases—broadcast trees construction phase and priority-based coding-aware forwarding phase. In the first phase, broadcast trees for all source nodes are constructed in such a way that attempts to minimize the number of internal nodes and maximize coding opportunities. In the second phase, each node disseminates broadcast packets in a network coding manner, based on several coding-aware forwarding rules. In addition, the priority-based deadlock prevention mechanism is adopted to avoid deadlocks that may happen in few topologies.

A. Broadcast Trees Construction Phase

In this phase, a broadcast tree is constructed for each source node in a distributed way. Each source initiates a parent selection process in the breadth-first traversal order, in which all nodes select their parents associated with that source. To minimize the total number of transmissions, this phase attempts to minimize the sum of maximum out-degrees over all nodes and leverage coding opportunity. To this end, the broadcast trees construction algorithm we devise adopts three empirical parent selection rules.

1) Parent Selection Rules

For each source node, each network node selects its own parent. Denote the broadcast tree associated with source k by $B\text{-}Tree_k$. The constructed broadcast trees should have the properties:

- For full delivery, the constructed graphs must be loop-free and a connected component.
- Because the number of transmissions sent by a node is lower bounded by that node's maximum out-degree, the maximum out-degree of each node should be minimized.
- Because only internal nodes in a broadcast tree forward the broadcast packet, the total number of internal nodes should be minimized.

Parent selection rules are made to achieve the above properties:

- R_1: Parent candidates for $B\text{-}Tree_k$ of a node, say node i, must be node i's neighbors that are already in $B\text{-}Tree_k$. (R_1 prevents a loop and an isolated tree from being formed.)
- R_2: It is preferred that selecting node j as node i's parent for $B\text{-}Tree_k$ does not increase the maximum out-degree of node j. (R_2 aims to minimize the maximum out-degree of a node.)
- R_3: A node prefers to select the neighbor node with most children in $B\text{-}Tree_k$ as its parent for $B\text{-}Tree_k$. (R_3 helps to reduce the number of internal nodes.)

Among the rules, the order of priority is $R_1 > R_2 > R_3$. R_1 is used to identify parent candidates. If there are multiple parent candidates, node i will select one of them by considering R_2. R_3 will be considered if there are multiple parent candidates satisfying R_2. If there are still multiple parent candidates, node i will randomly select one of them selected as its parent.

2) Broadcast Trees Construction Algorithm

A distributed broadcast trees construction algorithm is proposed to enforce the parent selection rules at run time. Two kinds of control packets—$p_{\text{select}}(t,u,v)$ and $p_{\text{update}}(t,u,v)$—are used to trigger parent selection and to update parent-children information. After node v selects node u as its parent for broadcast tree t, node v sends the control packet $p_{\text{select}}(t,u,v)$, which triggers the neighbor nodes that have not decided their parents in this tree to select their parents according to the parent selection rules. At the meantime, u sends the control packet $p_{\text{update}}(t,u,v)$ to update its neighbor nodes. All node receiving $p_{\text{select}}(t,u,v)$ or $p_{\text{update}}(t,u,v)$ will update the neighbor information about the tree topology.

Each broadcast tree is constructed in a breadth-first traversal order from the source node. Specially, a source node, say source k, sends out a control packet $p_{\text{select}}(k,0,k)$ to its neighbor nodes to begin the construction of $B\text{-}Tree_k$. Upon receiving this control packet, a neighbor node, say node i, sets a defer timer. During this defer period, it may hear other nodes sending out control packets and can update neighbor information. When the defer timer expires, node i selects a parent for $B\text{-}Tree_k$ according to the parent selection rules. The length of a defer period is randomly chosen to avoid collisions. The parent selection process keeps going in such a way that the broadcast trees are constructed in a breadth-first traversal order. The pseudo code of the broadcast tree construction algorithm is given in Fig. 2. The complexity of this algorithm is $O(N^2)$ where N is the total number of nodes in the network.

```
Algorithm: The broadcast trees construction algorithm at node i
if node i is a source node then
    node i broadcasts p_select(i, 0, i)
end if
if node i receives p_select(k, u, v) from node v then
    if i = u then
        broadcast p_update(k, u, v)
    else if node i has no parent in B-Tree_k then
        set a defer timer Timer_Decision_k
    end if
    update neighbor information
else if node i receives p_update(k, u, v) from node u then
    update neighbor information
end if
if Timer_Decision_k expires then
    select node p_k as its parent for B-Tree_k according to the parent
        selection rules
    broadcast p_select(k, p_k, i)
end if
```

Fig. 2. Pseudo code of the broadcast trees construction algorithm.

B. Priority-Based Coding-Aware Forwarding Phase

In this phase, native packets $d_k, 1 \le k \le m$, originated from m source nodes are broadcast over the entire network in a *network coding fashion*. Coded packets are generated at nodes by random linear network coding [20] with their coefficient vectors. Each coefficient vectors consists of m coefficients.

Each node has a coding pool to store innovative packets it overhears. A coded packet is innovative to a node if its coefficient vector is not within the span of the coefficient vectors of all packets stored in that node's coding pool. The progressive Gauss-Jordan elimination method is used to check whether a newly received packet is innovative. If it is innovative, the node adds that packet to its coding pool.

Upon reception of a coded packet, a node decides whether to generate and transmit a coded packet immediately, or to postpone the transmission for more coding opportunities. To facilitate the coding-aware forwarding as well as the priority-based deadlock prevention mechanism we propose in this section, each node maintains a *T-graph*. The T-graph at node u is initialized at the beginning of each session to be the subgraph of the broadcast trees constructed in the first phase that consists of all nodes adjacent to u and all directed edges entering or leaving u. Fig. 3 gives a high-level overview of the second phase.

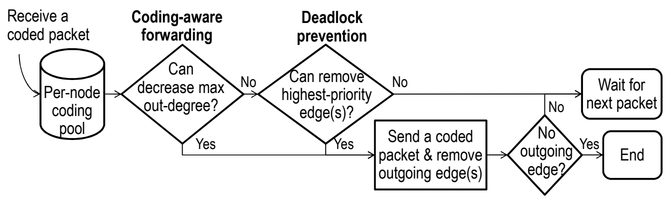


Fig. 3. Flowchart of the priority-based coding-aware forwarding phase.

1) Coding-Aware Forwarding

A parent node in *B-Tree*$_k$, $1 \le k \le m$, is responsible to deliver the information of d_k to its children in *B-Tree*$_k$. To this end, the parent node has to transmit a coded packet that contains the information of d_k. A coded packet is said to contain the information of d_k if the corresponding coefficient in the coefficient vector is nonzero.

To leverage coding opportunity, we consider multiple broadcast trees together. A pair of nodes, say parent node u and child node v, may have multiple parent-child relations, represented by B_{uv} directed edges from node u to node v. To deliver all information required by node v, node u needs to transmit B_{uv} coded packets which have linearly independent coefficient vectors, so that node v can decode all the packets after all node v's parents finish transmitting coded packets.

A parent-child relation, or a directed edge in the T-graph, implies that the parent should deliver certain information to the child. A source node, say source k, immediately transmits d_k. On the other hand, a non-source node, upon reception of a coded packet, decides whether to generate and transmit a coded packet immediately or to postpone the transmission for more coding opportunities. To transmit fewest packets, a non-source node, say node u, will transmit a coded packet if the transmission can decrease the maximum out-degree of the T-graph by one, after removing the corresponding outgoing edges. Otherwise, node u should wait for more packets to arrive so that the transmission of a coded packet can deliver useful information to more children.

After sending a coded packet, the parent node u removes the corresponding outgoing edge(s) in the T-graph. In a wireless network, the coded packet is received by all of node u's neighbor nodes, due to wireless multicast advantage. So node u could remove more than one outgoing edge in the T-graph. More precisely, if the information contained in this coded packet is useful to a certain child, node u should remove a corresponding outgoing edge to that child. Node u stop to transmit any coded packets once it has no outgoing edge left in the T-graph.

We use an example to illustrate the coding-aware forwarding scheme. In Fig. 4, node 6 is an internal node in four broadcast trees (*B-Tree*$_1$, *B-Tree*$_2$, *B-Tree*$_3$, and *B-Tree*$_4$) and needs to delivery corresponding information to node 7, node 8, and node 9. Suppose that the first coded packet p generated by node 6 contains the information of d_1 and d_2. Node 6 transmits p immediately because this transmission can decrease the maximum out-degree by one. This transmission delivers the information of d_1 to node 8 and node 9, and delivers the information of d_2 to node 7. Note that a transmission can deliver at most one unit of information. Therefore, after transmitting a coded packet, node 6 removes one outgoing edge to node 7, one edge to node 8, and one edge to node 9.

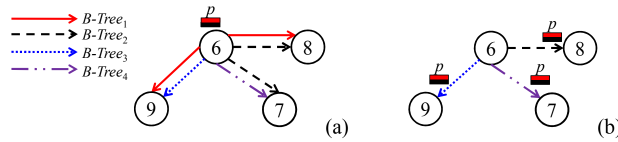


Fig. 4. Topologies before and after node 6 transmits its 1st coded packet.

2) Priority-Based Deadlock Prevention

In our proposed coding-aware forwarding scheme, if each node is allowed to transmit a coded packet only when that transmission can decrease the maximum out-degree of the T-graph, deadlock may occur in some topologies because each node may hold some packets and wait for other packets. Take Fig. 5 as an example, where nodes 1 and 2 are source nodes. After receiving the packet d_1 sent from node 1, node 3 will not send/forward any packet because no transmission at this moment can decrease maximum out-degree. Node 3 will wait forever to receive a packet from node 4. However, for the same reason, node 4 will hold the packet d_2 sent from node 2 and wait forever for a packet from node 3. Hence, nodes 3 and 4 both waits for a packet from each other, which causes a deadlock. To avoid deadlock while still having high coding opportunity, we propose a priority-based deadlock prevention mechanism.

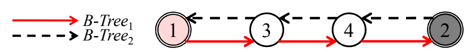


Fig. 5. An example of deadlock problem. Node 3 and node 4 will wait for packet from each other.

Deadlock situations arise when all nodes wait for packets which are held by others. An idea to prevent deadlock is to guarantee that at any time there always exist some packets which cannot be held and must be transmitted. To this end, we assign priority to each native packet d_k, $1 \le k \le m$. Correspondingly, edges in *B-Tree*$_k$ have the same priority as d_k. The priorities can be set according to a variety of needs such as packets' importance. Without loss of generality, we set the priorities according to the identifier of source node. That implies, d_i has higher priority than d_j for $i < j$.

Recall that after a node sends out a coded packet containing the information of d_k to its children, the node removes corresponding outgoing edges in the T-graph. To prevent deadlock, we add one more rule: A node must transmit a coded packet if it is able to generate a coded packet that can remove the highest-priority edges in its T-graph. Obviously, deadlock can be prevented by this priority-based forwarding rule.

Fig. 5 illustrates how the priority-based deadlock prevention scheme works. By this rule, node 3 will forward packet d_1 instead of holding d_1 because this transmission removes the outgoing edge to node 4, which is the highest-priority edge in node 3's T-graph. As can be seen from this example, nodes 3 and 4 no longer wait forever for a coded packet from each other and thus the deadlock is prevented. With this deadlock-prevention rule added, the total number of transmissions is five, which is the optimal.

In summary, Fig. 6 gives the pseudo code of the priority-based coding-aware forwarding algorithm.

Algorithm: The priority-based coding-aware forwarding algorithm at node i

```
if node i receives an innovative packet then
  node i stores the packet in its coding pool and generates a coded packet p
  if transmitting p can decrease maximum out-degree of i's T-grpah or remove the highest-priority edge in i's T-grpah then
    node i broadcasts p and removes the corresponding outgoing edges in its T-graph
    if node i has no outgoing edge left then
      node i broadcasts p_finished
    end if
  end if
else if node i receives p_finished from a node then
  node i updates the neighbor information
end if
```

Fig. 6. Pseudo code of priority-based coding-aware forwarding algorithm.

VI. Performance Evaluation

Our proposed protocol, PNCB, is evaluated using an in-house simulator developed in C. The main performance metric of our interest for a broadcast protocol is the number of transmissions. This metric is normalized by the number of nodes to show the efficiency. In addition, this metric is normalized by the number of sources since we consider the multiple-source scenario. The *ratio of broadcasts* is defined as the average number of transmissions required by each node for delivering a native packet to all nodes. More precisely, the ratio of broadcasts, R_B, is defined as:

$$R_B = \frac{\text{total number of transmissions}}{(\text{number of sources}) \times (\text{number of nodes})}.$$

In this simulation setup, we consider an environment where links within transmission range are reliable. Packets transmitted within transmission range of 200 meters are always delivered. In each simulation run, a number of nodes are distributed randomly over a square region of area $1000 \times 1000\text{m}^2$. Each simulation result is averaged over 20 instances.

PNCB and several broadcast schemes such as CODEB [8], RBS [21], and PDP [8] are run respectively to compare their performances. In addition, we also compare the performance of PNCB with those of several bounds to study how well the overall and individual components in PNCB perform.

A. Performance Comparison Given Broadcast Trees

Given the broadcast trees constructed by our proposed algorithm described in Section V.A, the performance in terms of R_B depends on the benefit from coding opportunity. Consider two extreme cases when network coding is not allowed and when network coding is fully and ideally exploited. In the first extreme case, no coding benefit is obtained and thus the corresponding R_B can be regarded as an upper bound. In the other extreme case, the total number of transmissions is equal to or smaller than the sum of maximum out-degrees (SMOD), which gives a lower bound for R_B. Compared to SMOD, PNCB requires slightly more transmissions due to the priority-based deadlock prevention mechanism.

We compare PNCB with the upper and lower bounds under a variety of number of sources. As shown in Fig. 7 where 200 nodes are distributed in the network, R_B of PNCB decreases as the number of sources increases. This makes sense because coding opportunity increases as the number of sources increases. On the contrary, the upper bound corresponding to the case of not allowing network coding increases with the number of sources, because the given broadcast trees are not optimized for the number of internal nodes only. Besides, we observe that R_B of PNCB is close to the SMOD lower bound when there are many sources. This small gap implies that the proposed mechanism avoids deadlocks at a small cost.

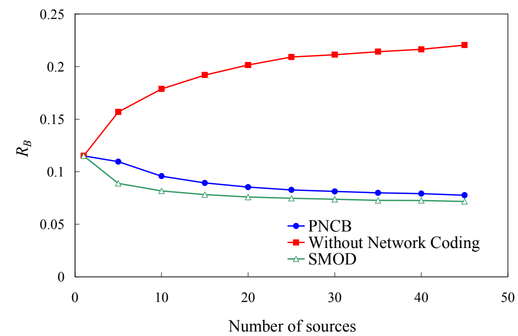


Fig. 7. The ratio of broadcasts vs. the number of sources.

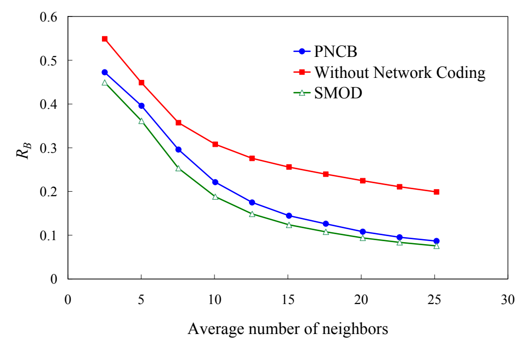


Fig. 8. The ratio of broadcasts vs. various average numbers of neighbors.

Fig. 8 shows the impact of network density on the ratio of broadcasts. The number of sources in this figure is set to 20. As we can see, R_B decreases as the network becomes denser. This is because in a denser network, the number of internal nodes in the constructed broadcast trees is smaller and coding opportunity is higher. Besides, observed from Fig. 8, R_B of PNCB is quite close to the SMOD lower bound, especially in a denser network.

B. Comparison with Existing Protocols

In this subsection, we compare the performance of PNCB with those of several protocols existing in the literature, including CODEB, RBS and PDP. The simulation topology is set as follows. There are 100 nodes randomly distributed in the network. On average each node has 30 neighbor nodes. 1 to 100 nodes are randomly chosen as source(s).

As can be seen in Fig. 9, PNCB outperforms RBS and CODEB. When there is only one source node, PNCB performs only slightly better than RBS since PNCB gains no coding benefit in this case and RBS is designed for one-to-all broadcast. As the number of sources increases, PNCB performs growingly better than RBS, which shows the coding benefits of our coding-aware forwarding.

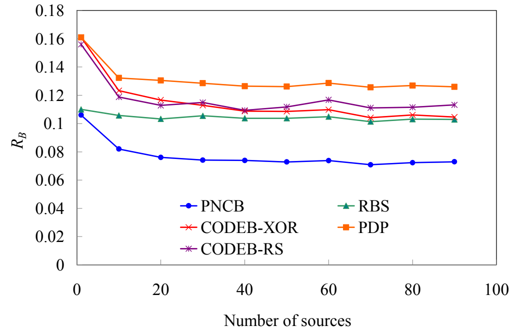


Fig. 9. The ratio of broadcasts vs. various numbers of sources (comparison with CODEB)

Compared to CODEB and PDP, PNCB performs much better. The main reason is on the forwarder selection process. In CODEB and PDP, each node chooses forwarders based on partial dominant pruning (PDP), which does not leverage coding opportunity. On the contrary, PNCB constructs broadcast trees in a way that attempts to maximize coding opportunity. Although CODEB obtains a high coding gain, the performance of CODEB is limited by the underlying PDP mechanism. This explains why RBS outperforms CODEB in our simulation.

VII. Conclusion

In this paper, we have addressed the many-to-all MTB problem in multi-hop wireless networks. The many-to-all MTB problem with network coding is related to a maximum out-degree problem. The solution to the maximum out-degree problem can be treated as a performance bound for any practical protocol.

To solve the many-to-all MTB problem with network coding in a fully distributed manner, we have developed a priority-based network-coding broadcast (PNCB) protocol. PNCB has two core functions—constructing efficient broadcast trees and dictating nodes to transmit packets in a network coding manner. Besides, PNCB includes the deadlock prevention mechanism, in order to ensure full delivery. Simulation results show that our proposed protocol outperforms existing protocols in terms of a measure of transmission efficiency.

References

[1] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, “The Broadcast Storm Problem in a Mobile Ad Hoc Network,” in *Proc. ACM/IEEE MobiCom*, Seattle, Washington, Aug. 1999.

[2] F.-W. Chen and J.-C. Kao, “Game-Based Broadcast over Reliable and Unreliable Wireless Links in Wireless Multihop Networks,” IEEE Transactions on Mobile Computing, vol. 12, no. 8, pp. 1613-1624, Aug. 2013.

[3] R. Ahlswede, C. Ning, S. Y. R. Li, and R. W. Yeung, “Network Information Flow,” *IEEE Trans. on Information Theory*, vol. 46, no. 4, pp. 1204-1216, July 2000.

[4] H. Shah Mansouri and M. R. Pakravan, “Reliable and Energy Efficient Single Source Broadcasting Using Network Coding in Wireless Ad-Hoc Networks,” in *Proc. ICT-MICC*, Penang, Malaysia, May 2007.

[5] H. Shah Mansouri and M. R. Pakravan, “Network Coding Based Reliable Broadcasting in Wireless Ad-hoc Networks,” in *Proc. IEEE Intl. Conference on Networks*, Adelaide, Australia, Nov. 2007.

[6] J. Widmer, C. Fragouli, and J. Le Boudec, “Low-Complexity Energy-Efficient Broadcasting in Wireless Ad-Hoc Networks Using Network Coding,” in *Proc. Workshop on Network Coding, Theory, and Applications*, Riva del Garda, Italy, April 2005.

[7] C. Fragouli, J. Widmer, and J. Y. Le Boudec, “A Network Coding Approach to Energy Efficient Broadcasting: From Theory to Practice,” in *Proc. IEEE INFOCOM*, Barcelona, Spain, April 2006.

[8] Li (Erran) Li, R. Ramjee, M. Buddhikot, and S. Miller, “Network Coding-Based Broadcast in Mobile Ad Hoc Networks” *in Proc. IEEE INFOCOM*, Phoenix, USA, Apr. 2008.

[9] D. Kim, Y. Wu, Y. Li, F. Zou, and D. Du, “Constructing Minimum Connected Dominating Sets with Bounded Diameters in Wireless Networks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 20, no. 2, pp. 147-157, Feb. 2009.

[10] Y. Li, My T. Thai, F. Wang, C. Yi, P. Wan, and D. Du, “On Greedy Construction of Connected Dominating Sets in Wireless Networks,” *Wireless Communications and Mobile Computing*, vol. 5, no. 8, pp. 927-932, Dec. 2005.

[11] Y. Sasson, D. Cavin, and A. Schiper, “Probabilistic Broadcast for Flooding in Wireless Mobile Ad Hoc Networks,” in *Proc. IEEE WCNC*, New Orleans, LA, March 2003.

[12] H.-I. Lu and R. Ravi, “Approximating Maximum Leaf Spanning Trees in Almost Linear Time,” *Journal of Algorithms,* vol. 29, no. 1, pp. 132-141, Oct. 1998.

[13] P.-J. Wan, K. M. Alzoubi, and O. Frieder, “Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks,” *Mobile Networks and Applications,* vol. 9, no. 2, pp. 141-149, Apr. 2004.

[14] S. Y. R. Li, R. W. Yeung, and C. Ning, “Linear Network Coding,” *IEEE Trans. on Information Theory,* vol. 49, no. 2, pp. 371-381, Feb. 2003.

[15] D. Lun, M. Medard, and R. Koetter, “Efficient Operation of Wireless Packet Networks Using Network Coding,” in *Proc Intl. Workshop on Convergent Technologies*, Oulu, Finland, June 2005.

[16] Y. Wu, P. A. Chou, and K. Sun-Yuan, “Minimum-Energy Multicast in Mobile Ad Hoc Networks Using Network Coding,” *IEEE Trans. on Communications,* vol. 53, no. 11, pp. 1906-1918, Nov. 2005.

[17] P. Joon-Sang, M. Gerla, D. S. Lun, Y. Yunjung, and M. Medard, “Codecast: a Network-Coding-Based Ad Hoc Multicast Protocol,” *IEEE Wireless Communications,* vol. 13, no. 5, pp. 76-81, Oct. 2006.

[18] F. J. Ros, P. M. Ruiz, and I. Stojmenovic, “Acknowledgment-Based Broadcast Protocol for Reliable and Efficient Data Dissemination in Vehicular Ad Hoc Networks,” *IEEE Transactions on Mobile Computing,* vol. 11, pp. 33-46, Jan. 2012.

[19] X. Zhang and K. G. Shin, “Chorus: Collision Resolution for Efficient Wireless Broadcast,” *Proc. IEEE INFOCOM*, San Diego, March 2010.

[20] P. Chou, Y. Wu, and K. Jain, “Practical Network Coding,” *in Proc. Allerton Annual Conference on Communication, Control, and Computing*, Monticello, Illinois, Oct. 2003.

[21] M. Khabbazian and V. K. Bhargava, “Efficient Broadcasting in Mobile Ad Hoc Networks,” *IEEE Trans. on Mobile Computing*, vol. 8, no.2, pp. 231-245, Feb. 2009.