Deadline-aware Broadcasting in Wireless Networks with Local Network Coding

Pouya Ostovari, Jie Wu, and Abdallah Khreishah
Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122

***Abstract*—Energy limitation is one of the most important challenges in wireless networks. Reducing the number of transmissions is one of the most effective ways to reduce the energy consumption. For this purpose, network coding can be used to mix packets together to reduce the number of transmissions. In addition to the importance of energy efficiency, in many applications, delay and deadline constraints are also important metrics. On the other hand, in order to increase the coding opportunity and efficiency of network coding, relay nodes need to wait to receive more packets, which increases the delay of the packets. In this paper, we study the problem of using network coding in wireless networks with deadline constraints. We provide three heuristics in an all-to-all broadcast application, to compute the local waiting time of the packets at relay nodes to improve the efficiency of the network coding without missing deadlines. Our simulation results show that our techniques reduce the number of transmissions while allowing all of the nodes to receive the packets on-time.**

***Index Terms*—Broadcasting, local network coding, partial dominant pruning, energy efficiency, deadline.**

I. Introduction

Broadcasting is a frequently used method in ad hoc networks for disseminating data and control messages in many applications. The simplest broadcasting method in wireless networks is flooding, where each node forwards the received packets to its neighbors. It is obvious that flooding is not an efficient way for broadcasting due to the unnecessary, redundant transmissions. In order to prevent redundant transmissions and to decrease energy consumption, probabilistic [1], [2] and deterministic approaches [3], [4] can be used.

Network coding is a method which can be used to reduce the number of transmissions in wireless networks. With network coding [5], [6], intermediate nodes mix different packets using mathematical operations to reduce the number of transmitted packets. Network coding can be classified into local and global coding. In local network coding, each forwarder node codes the packets such that the next-hops can decode it using the packets in their buffers. The next-hops first decode the received coded packets. Next, the forwarder nodes code the original packets again such that the next hops can decode them immediately. On the other hand, in global network coding, the intermediate nodes cannot decode the received coded packets immediately, and they code the received coded packets again without decoding them.

Network coding can be combined with probabilistic and deterministic forwarding methods to improve transmission efficiency. The work in [7] combines a deterministic approach with local coding. This approach uses network coding with the Partial Dominant Pruning (PDP) method [4]. In [8], the problem of broadcasting using network coding with directional antennas is addressed. It is shown in [2] that for fixed networks, network coding can offer a constant factor of benefit in terms of energy efficiency. The authors calculate these benefits for circular and square grid networks, and they propose a distributed broadcasting algorithm for random topologies.

Most of the research on network coding focuses on minimizing energy consumption [2], [8], maximizing the throughput [5], or achieving fairness [9], [10]. Delay and deadline are other important metrics which have been studied in [11], [12] for single-hop coding and broadcast channels. In this paper, we study the network coding problem by addressing the problem of *energy efficient broadcasting in wireless networks using network coding subject to deadline constraints*. To increase the energy efficiency of network coding we propose three methods, Velocity-based Waiting Time (VWT), Random Waiting Time (RWT), and Proportional Distribution of Waiting Time (PDWT) methods, to compute the waiting times of the packets at relay nodes such that no packet misses its deadline. These waiting times increase the chance of coding the packets, which decreases the number of transmissions and increases the energy efficiency of network coding.

II. Background and motivation

A. Network Coding

Network coding can be classified into local coding and global coding (linear coding). In local network coding, each node mixes non-coded packets such that its neighbors can decode the coded packet using the packets in their buffer. This means that the nodes do not need to wait to receive further packets, and they can decode the received coded packets immediately. On the other hand, in global network coding, the coding nodes code the native (non-coded) packets without considering the status of their neighbors. In this approach, when a receiver node receives a coded packet, it cannot decode the packet immediately, and it has to wait to receive a sufficient number of packets to be able to decode the coded packet.

In local coding, each node based on local two-hop information decides which packets to code together, such that all of the neighbors will be able to decode it using the packets in their buffers. Assume that in Fig. 1, nodes u_1, u_2, and u_3 want to broadcast their respective packets, p_1, p_2, and p_3, and node u_4 is the relay node. Using two-hop information, node

This research was supported in part by NSF grants ECCS 1128209, CNS 1065444, CCF 1028167, CNS 0948184, and CCF 0830289.

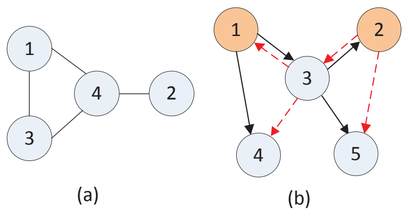

Fig. 1. (a):Local coding. (b): Waiting time and deadline.

u_4 knows that nodes u_1 and u_3 are neighbors, and they receive each others packets. Without network coding, node u_4 has to send tree packets. However, this node can code p_2 with p_1 or p_3. Assume that node u_4 sends $P = p_1 \oplus p_2$. Node u_1 and u_3 can recover packet p_2 by performing $P \oplus p_1$, and node u_2 can recover p_1 by performing $P \oplus p_2$.

Local network coding does not achieve optimality compared to global coding, but it has some advantages over global network coding. Firstly, the computational complexity of coding and decoding processes in local coding is much less than in linear network coding. Thus, for nodes with limited computational power, such as in sensor networks, local network coding is more attractive. Next, global network coding has more overhead than local coding because of coefficient vectors. Therefore, in this paper we use XOR-based local network coding in our deadline-aware methods.

B. Partial Dominant Pruning

To prevent broadcast flooding, we can use global or local approaches. Since we want to address local network coding in this paper, we use PDP broadcasting, which is a local method. However, in our methods, PDP can be replaced by other deterministic broadcasting methods. In PDP, each source node broadcasts its packet and selects a set of its one-hop neighbors as relay nodes such that this set covers two-hop neighbors of the source. Each relay node performs the same process, and all of the nodes receive the broadcasted packet. This approach forms a tree from a source node to all other nodes.

We represent the set of neighbors of node u (including u) as $N(u)$ and the set of neighbors of $N(u)$ as $N(N(u))$ (nodes that are within two-hop from u). Assume that node u sends a broadcast packet to node v and chooses this node as a relay node. Now, node v has to relay the packet and select a set of its one-hop neighbors as relay nodes to cover its two-hop neighbors. To minimize the number of transmissions, this set has to contain the minimum number of nodes. Nodes in $N(v)$ will receive the packet when node v broadcasts the packet and the nodes in $N(u)$ have already received it. Also, neighbors of common neighbors of nodes $N(u)$ and $N(v)$ will receive it. Therefore, node v has to select its relay nodes $R(u, v)$ from nodes in $B(u, v) = N(v) - N(u)$ to cover the nodes in $U(u, v) = N(N(v)) - N(u) - N(v) - N(N(u) \cap N(V))$. To find this set, a greedy set cover algorithm is used in [4]. At each step, a node in set B that covers the maximum number of nodes in U is added to the relay nodes. This process is repeated until all of the nodes of set U are covered.

Algorithm 1 Coding

```
$P \leftarrow p_i$
for each remaining packet $p_k$ in the queue do
    if all neighbors can decode $P \oplus p_k$ then
        $P \leftarrow P \oplus p_k$
```

III. Setting

In this paper, we study the problem of all-to-all broadcasting with deadline constraints. In our model, each packet has the same deadline to be received by all of the destinations, but the deadlines of different packets can be different. The deadlines are in terms of time slots and each transmission takes one time slot to be received by the next hop. We assume that each node has two-hop neighborhood information about the network. Our goal is to minimize the number of transmissions. The constraint is that each packet has to be received by all of the destination nodes before the deadline, if possible.

IV. Deadline-Aware Network Coding

In our problem, all of the nodes can be source nodes. All of the sources broadcast their packets based on the PDP algorithm. If a node is a relay node of more than one packet, it has the opportunity to mix the received packets. In local network coding, if node u sends a coded packet P, a neighbor v of u should be able to decode this packet without waiting for further packets. In other words, each node u with a set of native packets p in its sending buffer seeks to find a subset of the native packets q to XOR. To decrease the number of transmissions, for each transmitted packet, node u has to maximize the number of neighbors which can decode a missing packet. In [7], it is proven that this problem is NP-complete. Therefore, a greedy algorithm is used to address this problem. This algorithm takes the packet p at the head of the sending queue and sequentially looks for other packets in the queue such that if they are combined with p, all neighbors of node u will be able to decode the coded packet. The procedure is described in Alg. 1.

To increase coding opportunities, instead of sending the packets immediately, each forwarder node has to wait for a given time to receive more packets. Choosing the appropriate waiting times is critical in this approach. Long waiting times can result in deadline misses, and short waiting times can decrease coding opportunities. Assume that in Fig. 1(b), nodes u_1 and u_2 are sources. Assume that the sending time of p_1 and p_2 are 1 and 3, respectively. Also, assume that the deadline of p_1 and p_2 is time slot 6. Node u_3 receives p_1 and p_2 at times 2 and 4, respectively. If we set the waiting time of p_1 at node u_3 to 1, this node sends packet p_1 at time slot 3. Therefore, node u_3 has to send two non-coded packets. On the other hand, if we set the waiting time to 4, node u_5 receives p_1 at time slot 7, which is after the deadline. By choosing a waiting time of packet p_1 equal to 3, node u_3 sends one coded packet instead of two non-coded packets.

To address the problem of choosing the waiting times, we propose three approaches in the following sections.

Algorithm 2 Velocity (Initializing phase)

```
if Node $u_j$ is a leaf node then
    Send feedback $H_{ij} = 0$
else
    On receiving a feedback from node $u_k$ to node $u_j$
    Store $H_{ik}$
    if Received feedback from all children nodes then
        $H_{ij} \leftarrow \max(H_{ik}) + 1$, Forward $H_{ij}$ to the parent node
```

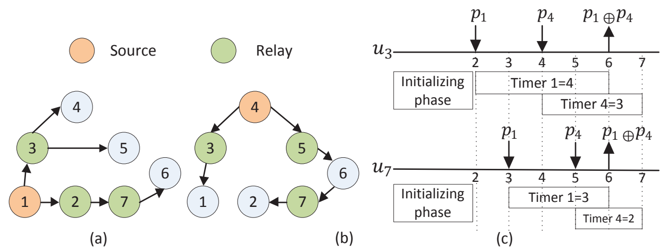

Fig. 2. Velocity-based approach.

A. Velocity-based Waiting Time

Our constraint is meeting all of the deadlines. Thus, a relay node u_j can postpone the transmission of a packet if it finds that by postponing the transmission, none of the nodes will receive that packet after the deadline. To make sure that all of the next-hop nodes receive packet p_i on-time, node u_j has to consider the receiving time of the packet by the deepest leaf node. The deepest leaf node is the farthest node of all of the branches in tree T_i that node u_j is the root of. Based on the maximum remaining hops, node u_j can calculate the extra time, which is the maximum allowable waiting time, and can use a portion of this time as its waiting time.

The Velocity-based Waiting Time approach (VWT) contains initialization and running phases. In the first phase, based on the PDP algorithm, each source node u_s sends a request to construct a tree T_s. Then, for each tree T_i, each leaf node u_k sends feedback which contains the length of the longest branch from node u_k. We call this value *the maximum remaining hops*, and we represent it by H_{ik}. For the leaf nodes $H_{ik} = 0$ since there is no remaining hop from node u_k. Node u_j collects the feedback from its children nodes. It adds 1 to the maximum received value, stores it as H_{ij} and relays it to its parent. Based on the feedback, each relay node u_j knows the remaining hops of the longest branch in tree T_i. The pseudo-code of the feedback part of the initializing phase is shown in Alg. 2.

In the running phase (Alg. 3), when a relay node receives a packet, it computes the remaining time of that packet by subtracting the current time from the deadline. Then, it subtracts the number of maximum remaining hops from the remaining time to compute the extra time. At the end, using a velocity-based approach [13], it computes the waiting time of the packet. A velocity-based approach means that the waiting time of the packet is calculated based on both the deadline and the maximum remaining hops. When the waiting time of a packet expires, the node uses Alg. 1 to code the packet with other packets in its buffer and transmits it. The relay

Algorithm 3 Velocity (Running phase)

```
On receiving a packet $P$ by node $u_j$
for each Native packet $p_i \in P$ do
    if $u_j \in Forwarders(p_i)$ then
        $R_{ij} = D_i - t$, $E_{ij} = R_{ij} - H_{ij}$
        $W_{ij} = \lfloor \frac{E_{ij}}{H_{ij}} \rfloor$, $Timer_i \leftarrow W_{ij}$
```

Algorithm 4 PDWT (Initialization phase)

```
if Node $u_j$ is a leaf node then
    Send feedback $H_{ij} = 0$ and $f_{ij} = 0$
else
    On receiving a feedback from node $u_k$ to node $u_j$
    store $H_{ik}$ and $f_{ik}$
    if has received feedback from all children nodes then
        $H_{ij} \leftarrow \max(H_{ik}) + 1$
        if $f_i > 1$ then
            $f_{i,j} \leftarrow \ mean(f_{ik}) + f_i$
        Forward $H_{ij}$ and $f_{ij}$ to the parent node
```

node computes the waiting time using equation $W_{ij} = \lfloor \frac{E_{ij}}{H_{ij}} \rfloor$, where $E_{ij} = R_{ij} - H_{ij}$ and $R_{ij} = D_i - t$. The remaining time and waiting time of packet p_i at node u_j are represented by R_{ij} and W_{ij}, respectively. Here, D_i is the deadline of the packet p_i, and t is the current time. Based on our assumption, the nodes have the MIMO capability, and each transmission takes one time slot. Therefore, H_{ij} is equal to the remaining transmission time, and E_{ij} represents the extra time of packet p_i at node u_j.

The selected paths based on the PDP approach, from sources u_1 and u_4 to other nodes are shown in Figures 2(a) and 2(b). Assume that u_1 and u_4 send packets p_1 and p_4 at time slots 1 and 3, respectively. Also, assume that $D_1 = 7$, and $D_4 = 8$. Nodes u_2 and u_3 receive packet p_1 at time slot 2. Node u_2 is a relay node in only one tree. Thus, it does not have a coding opportunity and forwards the received packet immediately. In contrast, node u_3 is a forwarder node in both trees, so it computes the waiting time of the packet. At time slot 2, the remaining hops from node u_3 in tree T_1 is 1. Therefore, $R_{1,3} = 7 - 2 = 5$, $E_{1,3} = 5 - 1 = 4$, and $W_{1,3} = 4$. Node u_7 receives packet p_1 at time slot 3, and its waiting time is $W_{1,7} = \frac{(7-3)-1}{1} = 3$. Node u_3 and u_7 receive packet p_4 at time slots 4 and 5, respectively. The waiting times of packet p_4 at nodes u_3 and u_7 are $W_{4,3} = \frac{(8-4)-1}{1} = 3$, and $W_{4,7} = \frac{(8-5)-1}{1} = 2$. Fig. 2(c) shows that at nodes p_3 and p_7, the timers of packet p_1 expire after receiving packet p_7 and before the expiration of the timers of p_7. As a result, when the timer of p_3 is expired, nodes p_3 and p_7 check to see if they can mix packet p_1 with p_4. Since all of their neighbors can decode $p_1 \oplus p_4$, they send this coded packet.

B. Proportional Distribution of Waiting Time

There is a higher coding chance at the nodes that are relay nodes in more trees compared to other nodes, as they receive packets more frequently than other nodes. Also, the chance of

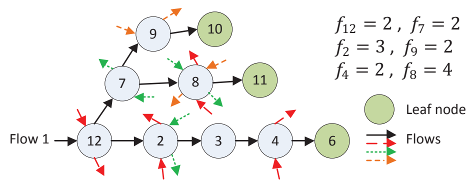

Fig. 3. Proportional distribution of waiting time.

mixing many packets together at these nodes is more than at other nodes. Thus, in our second method, which is based on the Proportional Distribution of Waiting Time (PDWT), we distribute the extra time among the nodes in a way that is proportional to the number of outgoing flows that pass from these nodes. Thus, if node u is a relay node of more flows than node v, we assign more waiting time to node u than to node v. We represent the number of outgoing flows from node u_i as f_i.

Similarly to the previous approach, the new approach has initialization and running phases. The feedback part of the initialization phase is described in Alg. 4. Also like the previous approach, after constructing all of the trees, the leaf nodes of each tree T_i send back the number of remaining hops to their parent. They also send the number of outgoing flows that pass from them. When a parent node u_j receives all of the feedback from its children nodes in tree T_i, it records the average received value of f_{ik}'s. We represent the average received value from the children nodes of node u_j in tree T_i as f_{ij}. If f_j is more than one, it means that u_j is a coding node. In this case, node u_j adds f_j to f_{ij}, and if f_j is less than two it does not change f_{ij}. Then, the node forwards f_{ij} and H_{ij} to its parent. This process is repeated until the source node of tree T_i receives this feedback. After the initialization phase, each node u_j knows H_{ij} and f_{ij}.

When a relay node receives a packet in the running phase, it uses the following equation to compute the waiting time of the packet (Alg. 5):

$$W_{ij} = \lfloor \frac{E_{ij} \times f_j}{f_{ij}} \rfloor \tag{1}$$

Fig. 3 shows a part of tree T_1 and the outgoing flows from each node. We do not show the complete topology and all of the trees for brevity. In this example, after constructing all of the trees, node u_6 sends $f_6 = 0$ as feedback. Node u_4 receives this feedback. This node has two outgoing flows. Therefore, it stores and sends $f_{1,4} = 0 + f_4 = 2$. Node u_3 has one outgoing flow, so it is not a coding node. Thus, it does not change the received feedback and sends $f_{1,3} = 2$. Then, node u_2 forwards $f_{1,2} = 2 + 3 = 5$. This process is repeated for nodes u_{10} and u_{11}. Node u_7 receives two feedbacks, $f_{1,8} = 4$ and $f_{1,9} = 2$. This node sends $f_{1,7} = \frac{4+2}{2} + 2 = 5$ to node u_{12}. At the end, node u_{12} computes and sends $f_{1,12} = \frac{5+5}{2} + 2 = 7$. Assume that $E_{1,12} = 7$. In this case, $W_{1,12} = \lfloor \frac{7 \times 2}{7} \rfloor = 2$, so $E_{1,7} = 5$. Thus, $W_{1,7} = \lfloor \frac{5 \times 2}{5} \rfloor = 2$, and $W_{1,8} = \lfloor \frac{3 \times 4}{4} \rfloor = 3$.

C. Random Waiting Time

The third proposed method is Random Waiting Time (RWT). The initialization phase of this approach is similar to the PDWT method. In this approach, each node computes a range of waiting time values and selects a random value from this range. To prevent deadline misses, node u_j cannot postpone the transmission of packet p_i more than E_{ij}. Also, it is not logical to use E_{ij} as the upper bound of the range since it is possible to use the entire waiting time at the first nodes, which results in much smaller waiting times at the remaining nodes. Therefore, we use $\frac{Z_{ij}+E_{ij}}{2}$ as the upper bound. Here, Z_{ij} represents the waiting time of packet p_i at node u_j that is computed by the PDWT approach. On the other hand, we use $\frac{Z_{ij}}{2}$ as the lower bound of the waiting time range. In conclusion, in the RWT approach, the initialization phase is similar to the PDWT approach. Then, in the running phase, each node uses Equation 1 to compute Z_{ij}. Next, the node selects a random waiting time within the range of $(\frac{Z_{ij}}{2}, \frac{Z_{ij}+E_{ij}}{2})$. Alg. 6 shows the ruining phase of this method.

Algorithm 5 PDWT (Running phase)

```
On receiving a packet $P$ by node $u_j$
for each Native packet $p_i \in P$ do
    if $u_j \in Forwarders(p_i)$ then
        $R_{ij} = D_i - t$, $E_{ij} = R_{ij} - H_{ij}$
        $W_{ij} = \lfloor \frac{E_{ij} \times f_j}{f_{ij}} \rfloor$, $Timer_i \leftarrow W_{ij}$
```

Algorithm 6 Random (Running phase)

```
On receiving a packet $P$ by node $u_j$
for each Native packet $p_i \in P$ do
    if $u_j \in Forwarders(p_i)$ then
        $R_{ij} = D_i - t$, $E_{ij} = R_{ij} - H_{ij}$, $Z_{ij} = \lfloor \frac{E_{ij} \times f_j}{f_{ij}} \rfloor$
        $W_{ij} \leftarrow rand(\frac{Z_{ij}}{2}, \frac{Z_{ij}+E_{ij}}{2})$, $Timer_i \leftarrow W_{ij}$
```

V. Simulation Results

We compare our proposed methods with the network coding method in [7]. The work in [7] sets a random waiting time for the packets in case of delay tolerant networks. For the case of non-delay tolerant networks, there is no waiting time for the packets. We refer to this method as the *no waiting* method.

We evaluate the methods on 100 random topologies. We vary the deadlines from 0.5 times to 2.5 times the diameter[1] (d) of the network to ensure that there are cases with and without deadline misses. We represent the packet generation period by G. This means that each node in every G time slots generates one packet. Therefore, G is inversely proportional to the packet generation rate. We vary the packet generating period from $\frac{M}{2}$ to $2 \times M$. Here, M is the number of nodes.

In the first experiment, we study the effect of the packet generation period on the number of transmissions. In Fig. 4(a), the deadlines are in the range of d to $1.5 \times d$. The number of transmissions of the methods increases as we increase G. The reason is that by increasing the generation period of the packets, the expected number of received packets at each time

[1]The diameter of the network is the distance in terms of hop count between the farthest nodes of the network.

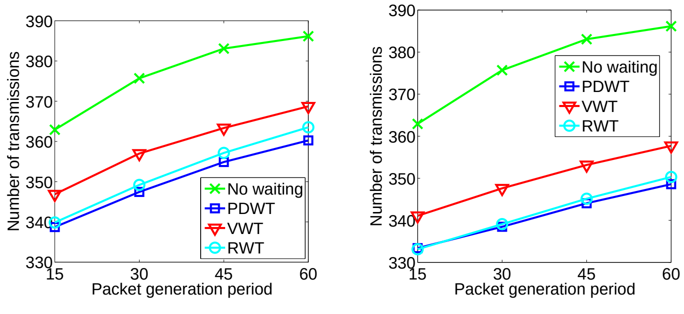

Fig. 4. Effect of packet generation period on the number of transmissions- M=30. (a):$D \in (d, 1.5 \times d)$. (b):$D \in (1.5 \times d, 2 \times d)$

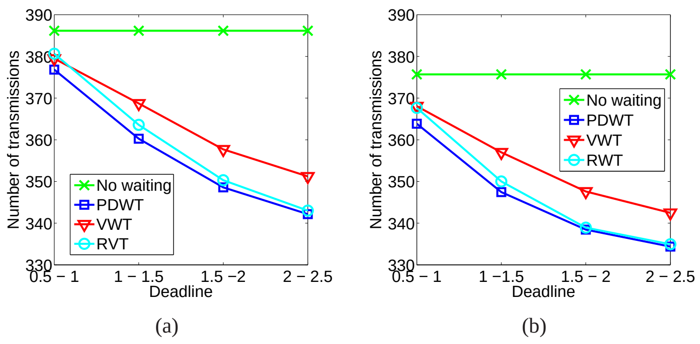

Fig. 5. Effect of deadlines on the number of transmissions- M=30. (a):$G = 60$. (b):$G = 30$.

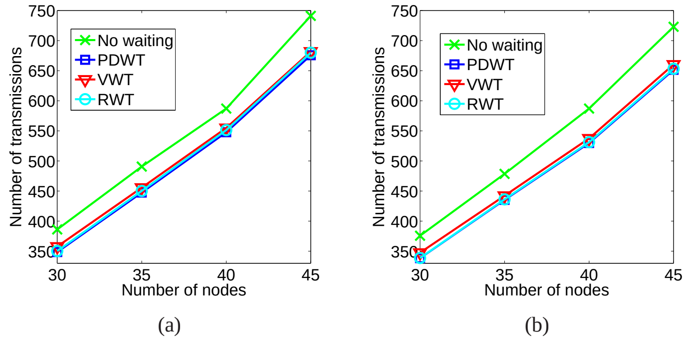

Fig. 6. Effect of number of nodes on the number of transmissions- $D \in (1.5 \times d, 2 \times d)$. (a): $G = M$. (b): $G = 2 \times M$

slot decreases. Thus, the coding opportunity decreases. In Fig. 4(b), we change the deadlines to be in the the range of $1.5 \times d$ to $2 \times d$. It can be inferred from this figure that the difference between our proposed methods and the no waiting method increases as we increase the generation period of the packets.

In the next experiment, we study the effect of deadlines on the number of transmissions. When the deadlines of the packets are in the range of $0.5 \times d$ to d, the extra time of the most of packets at relay nodes is near zero. Thus, the number of transmissions of all of the methods are close, which can be seen in Figures 5(a) and (b). The number of transmissions is inversely proportional to the the deadlines. The reason is that by increasing the deadlines, the packets have more waiting time, which increases the chance of coding. The number of transmissions of the no waiting method is fixed since relay nodes forward the received packets immediately without considering the deadlines. It can be inferred from Fig. 5(a) that the PDWT method has the lowest number of transmissions compared to the other approaches.

The period of generating packets in Fig. 5(b) is half of the period in Fig. 5(a). The number of transmissions in Fig. 5(b) is less than that in Fig. 5(a). The reason is that the relay nodes have more chances to code the received packets.

Figures 6(a) and (b), show the effect of the number of nodes on the number of transmissions. The number of transmissions of all of the methods increases as we increase the number of nodes since the number of generated packets increases. The packet generation period in Fig. 6(b) is half of that in Fig. 6(a); the number of transmissions is less than in Fig. 6(a).

VI. Conclusion

In this paper, we study the problem of energy-efficient, all-to-all broadcasting with deadline constraints. We combine local network coding with the PDP broadcasting method to reduce the number of transmissions. We propose three methods to compute the waiting times of the packets at the relay nodes such that these waiting times do not result in deadline misses. Using the simulation results, we show that our approaches increase the efficiency of local network coding. The simulation results show that the Proportional Distribution of Waiting Time (PDWT) method has the lowest number of transmissions compared to the other approaches.

References

[1] Y. Sasson, D. Cavin, and A. Schiper, “Probabilistic broadcast for flooding in wireless mobile ad hoc networks,” in *IEEE WCNC 2003*, vol. 2, Mar 2003, pp. 1124–1130.
[2] C. Fragouli, J. Widmer, and J. L. Boudec, “Efficient broadcasting using network coding,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 450–463, Apr 2008.
[3] J. Wu and H. Li, “On calculating connected dominating sets for efficient routing in ad hoc wireless networks,” in *Proc. of International Workshop Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999, pp. 7–14.
[4] W. Lou and J. Wu, “On reducing broadcast redundancy in ad hoc wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 111– 122, Apr-Jun 2002.
[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “Xors in the air: practical wireless network coding,” in *ACM SIGCOMM*, 2006.
[6] S. Li, R. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
[7] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, “Network coding-based broadcast in mobile ad-hoc networks,” in *IEEE INFOCOM*, May 2007.
[8] S. Yang and J. Wu, “Efficient broadcasting using network coding and directional antennas in MANETs,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 148–161, Feb 2010.
[9] A. Khreishah, C. Wang, and N. Shroff, “Cross-layer optimization for wireless multihop networks with pairwise intersession network coding,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 606–621, 2009.
[10] C. Wang, A. Khreishah, and N. Shroff, “Cross-layer optimizations for intersession network coding on practical 2-hop relay networks,” in *Asilomar*, vol. 41, 2009, pp. 771–775.
[11] X. Li, C.-C. Wang, and X. Lin, “Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints,” in *IEEE INFOCOM*, 2010.
[12] C. Zhan and Y. Xu, “Broadcast scheduling based on network coding in time critical wireless networks,” in *IEEE NetCod*, June 2010.
[13] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, and T. He, “Rap: A real-time communication architecture for large-scale wireless sensor networks,” in *Proc. of IEEE RTAS*, 2002, pp. 55–66.