Spring 2017

Decodable network coding in wireless network

Junwei Su

Department:

DECODABLE NETWORK CODING IN WIRELESS NETWORK

by

JUNWEI SU

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER SCIENCE

2017

Approved by

Dr. Maggie Cheng, Advisor
Dr. Wei Jiang
Dr. Fikret Ercal

ABSTRACT

Network coding is a network layer technique to improve transmission efficiency. Coding packets is especially beneficial in a wireless environment where the demand for radio spectrum is high. However, to fully realize the benefits of network coding two challenging issues that must be addressed are: (1) Guaranteeing separation of coded packets at the destination, and (2) Mitigating the extra coding/decoding delay. If the destination has all the needed packets to decode a coded packet, then separation failure can be averted. If the scheduling algorithm considers the arrival time of coding pairs, then the extra delay can be mitigated. In this paper, we develop a network coding method to address these (decoding and latency) issues for multi-source multi-destination unicast and multicast sessions. We use linear programming to find the most efficient coding design solution with guaranteed decodability. To reduce network delay, we develop a scheduling algorithm to minimize the extra coding/decoding delay. Our coding design method and scheduling algorithm are validated through experiments. Simulation results show improved transmission efficiency and reduced network delay.

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor Dr. Maggie Cheng from the computer science department at Missouri University of Science and Technology. Working for Dr. Cheng is a significant experience for me that I gained valuable research skills. There was a hard time when I studied this area at the beginning. Dr. Cheng always provided guidance and led me to the right direction. I'm grateful for all Dr. Cheng's help.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

Since its introduction in 2000 [1], network coding has gained a lot of research interest. Applications of network coding techniques in communication networks are still growing, from improving communication throughput and fairness [2], improving storage and content distribution efficiency [3], [4], to error detection/correction [5]–[11] and distortion optimization [12]. The main stream of research on network coding has been on finding the throughput capacity through random [13] or deterministic [14] coding schemes.

The fundamental difference between network coding and routing is the transmission reduction. In a wireless network shown in Fig. 1.1, nodes A and B need to send to each other via a relay node C. If routing is used, which simply does store-and-forward without changing the packets passing by, it takes four transmissions; but if network coding is used, it only takes three transmissions— there lay node combines two packets using a bitwise XOR operation and broadcasts the coded packet to nodes A and B simultaneously. Fewer transmissions reduce bandwidth demand which can directly improve communication throughput. Reduced medium contention can also indirectly improve the delay performance. However, when the network topology becomes complex, there is no easy solution for network coding. Some researchers studied the characteristics of the network topology that has a network coding solution [15]. Such characteristics include the well-known butterfly network, the grail network, etc. However, to our knowledge, there is no general analytical method to address the following question: given a network and its traffic load, does network coding offer more benefit than routing?

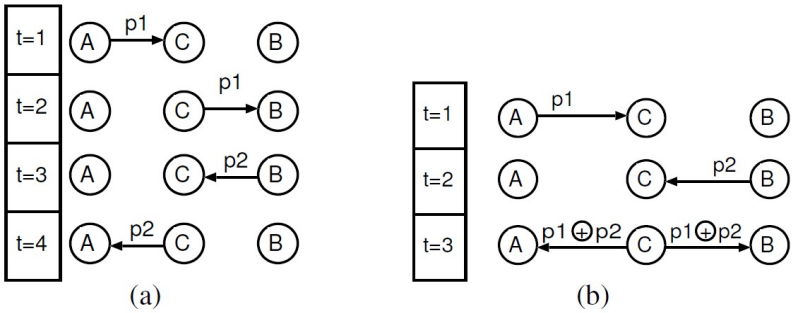

Fig. 1.1: Example of routing transmission. (a) Routing requires 4 transmissions in 4 time slots, (b) Network coding requires 3 transmissions in 3 time slots.

It is not difficult to find a common relay node between two flows, which harbors opportunity for using network coding. However, if the two flows are combined at some intermediate node, can the destinations successfully decode and recover the needed packets? The answer depends on the decodability of a network coding solution. If the needed data are mixed with sources it does not need it is called “pollution” [16].In fact, even if the needed data are mixed with other sources it needs, a pollution-free coded packet still does not guarantee it can be decoded. For instance, if a node receives a packet $C = A \oplus B$, without knowing either A or B, it cannot recover either one. How can we design a network coding solution that is decodable? What is the most bandwidth-efficient coding solution? The current lack of tools for decodability analysis and optimal coding design motivated this study.

The goal of this paper is to develop a general analytical method that is both decodable and efficient. To our knowledge, this is the first paper that concerns itself with these practical design issues. Our method finds a decodable coding solution which minimizes the required number of transmissions. Regardless of the destination decodability, we first determine the number of transmission reductions of two combined flows. Then we employ an integer linear program to find the optimal coding combination with the constraint that every destination must be able to decode the needed packets. Since our method does not require explicit graph-theoretical characterization, it is applicable for any complex network with arbitrary traffic loads. Such a deterministic coding design approach also offers better security features against the pollution attack—a relay node will not encode a packet with another random packet it receives if they are not a coding pair by design.

An important discovery of [15] is that the complexity of finding a good coding solution lies in finding the good paths rather than finding good encoding functions. It is also revealed in [17] that systematic network coding using XORs can provide the same or close to the same performance in terms of completion time as a random linear network coding scheme that uses a large field size, with the added advantage of requiring fewer and simpler operations during the decoding process. Therefore, in this paper we use simple pairwise XOR as the encoding function and use a deterministic network coding scheme to find the coding pairs.

The first step towards efficient communications is reducing the number of transmissions. Intuitively, fewer transmissions in wireless networking environments improve bandwidth efficiency. But the efficiency in spectrum does not always translate to shorter delay, especially when the waiting time is increased because of the use of network coding. A large number of studies focused on the tradeoff between the throughput gain of network coding and network delay. Apparently, if an opportunistic coding scheme is used, a relay node may need to hold a packet until its coding pair occurs or the threshold waiting time has passed [18], and a destination may need to wait until all the needed information is received to perform decoding, thus increasing delay. Sometimes the delay time at the destination can be unbounded. However, if a deterministic coding scheme is used, the transmission time can be scheduled in such a way that the total end-to-end delay is minimized. Although the transmission scheduling problem has been extensively studied, there is no scheduling scheme available that specifically addresses network coding traffic. In this paper, we focus on transmission scheduling when the traffic is a mixture of uncoded (forwarded without network coding) and coded packets.

Our main two contributions are: 1) finding the most bandwidth-efficient decodable network coding combination for wireless communications, and 2) developing a deterministic scheduling scheme to minimize network delay. First, we steer traffic to stay on the original routes and find the coding pairs resulting in the fewest number of transmissions. Integrated as a single linear program, this network coding scheme achieves an optimal solution with guaranteed decodability at each destination. Second, we develop a media access control (MAC) layer scheme that incorporates the new network coding conflict relation and generates a transmission schedule with minimum network delay. To preserve the original routes and keep the coding design as an add-on module is a design choice, which has the benefit of allowing different routing algorithms to couple with the coding scheme, and the routes in the routing table do not need to be updated; Moreover, the computation of coding combinations has less complexity than the one that uses joint design of routing and coding. The latter will be studied in our future work.

The rest of the paper is organized as follows. In Section II, we briefly survey the previous related work. In Section III, we describe the network setting in which network coding is explored. In Section IV, we formulate the coding design problem, provide a

decodability analysis framework and a linear programming model to find the optimal coding solution. In Section V, we develop a scheduling scheme for the mixed coding and routing traffic. Section VI validates the proposed models and algorithms by using the standard butterfly network. In Section VII, we present the ideas of random linear network coding and the comparison with our work. We present simulation results in Section VIII to study the performance of the algorithms in randomly chosen network settings. Section IX concludes the paper and points out future research directions.

2. RELATE WORK

Recently, a new area of research has emerged called network coding that allows packet mixing at intermediate nodes [1]. Wang et al. [15] studied the problem of network coding with two simple unicast sessions for directed acyclic graphs (DAG). Unlike our work, the work in [15] characterizes the graphs that offer a coding opportunity. Such graphs include the well-known butterfly and grail subgraphs, but it does not address whether such a coding opportunity has advantages over routing. Our work addresses both feasibility and optimality — it answers whether the coding opportunity exists and whether coding is advantageous over routing. Moreover, the work in [15] is for two unicast sessions only. Our work includes a general model that can be applied to multiple unicast sessions and multiple multicast sessions. It goes beyond feasibility analysis and addresses whether there is performance gain in network coding and how to maximize this performance gain.

While the concept of random network coding seems promising, failure to separate the coded data can be the biggest barrier to its full potential. Unless the coded data can be successfully decoded, it is useless. The probability to decode has been addressed in [19]–[22]. In [19], Li et al. used the coupon collect or model to compute the expected number of coded packets needed for successful decoding, and provided abound on the probability of decoding failure. In [21] Ho et al. provided a lower bound on the probability of successful decoding for randomized network coding.

In a different direction, finding the capacity region enabled by network coding has been an active research area. Some studies focused on the outer bound, which is defined by the necessary conditions for the existence of network coding solutions [23]–[27], and some studies addressed the inner bound, which is the maximum achievable throughput under a certain coding scheme. The inner bound can be determined by linear programming, using a butterfly-seeking implementation [15], [28], [29], or a constructive coding design approach [16].

The issue of transmission scheduling with network coding on a time-division channel has also been investigated. Sagduyu et al. [30] investigated joint scheduling and wireless network coding. In this work, the whole network is partitioned into some

conflict-free disjoint subnetworks, each with minimum cost assignment. Then the network throughput is optimized through joint scheduling and network coding. In [31], opportunistic scheduling for wireless network coding is studied. The basic idea is to dynamically change the network coding group size by using opportunistic scheduling to maximize the average throughput. Our work uses separate modules for coding design and scheduling, with the objective of the former being to reduce transmissions and the latter being to reduce delay. Such an approach has the advantage of allowing the same scheduling algorithm to work with different coding schemes, or even with joint design of routing and coding as mentioned in our future work in Section IX.

In our previous work [32], we proposed a scheduling scheme for multicasting. Using this scheduling algorithm, unicast can be considered as a special case with a destination group of size one. However, to consider coded packets in unicast or multicast, additional care must be taken to consider the new conflict relation, since one coded packet contains the contents of two sources. The new scheduling scheme in this paper is designed to explore this feature in order to further improve delay performance.

3. NETWORK SETTING

In this paper, we consider multicasting in a multihop wireless network. Multicast generalizes unicast and broadcast by varying the destination group size. We assume the multicast routing information is known, and the packets will be forwarded along their original routes. After a packet is encoded with another packet, the coded packet still stays on the original routes towards the destinations. The network layer can use any routing algorithm. Our optimal coding algorithm uses the output of the routing algorithm as input. We also assume the packets are transmitted over a time-division multiplexing channel, and each time slot is equivalent to one packet transmission time.

The proposed work involves the network layer and the MAC layer. In the network layer, we discuss a process to decide the coding pairs—which flows will be coded together, and the location of coding—which relay node will perform the encoding function. The encoding function is pairwise XOR. We choose XOR for its simplicity, since our main objective is to find the coding combinations instead of the encoding function. In the MAC layer, we present an algorithm that schedules transmissions. Section IV addresses the network layer function and section V addresses the MAC layer function.

4. OPTIMAL CODING

The objective of the coding design is to reduce the number of transmissions. Among all the feasible coding solutions, the optimal solution is the one that uses the minimum number of transmissions to deliver data. Feasible solution means the receivers must either receive the needed data in its original form, or in a coded packet that can be decoded. So far, to the best of our knowledge, there is no other available tool to address the decodability issue other than a probability model. Our work is the first in its kind to provide a deterministic answer to the question.

4.1. TRANSMISSION REDUCTION

Let W_{ij} denote the number of transmissions that can be reduced by encoding packets of flow i and flow j. W_{ij} is an indicator of the benefit of coding flow i and flow j together. Let S denote the set of source nodes. To compute W_{ij} for all pairs $i, j \in S$, we can look at how many hops are on the shared paths of flow i and flow j. For example, in Fig. 4.1 (a), source i uses the path $\{1 \rightarrow 2 \rightarrow 3 \rightarrow 4\}$, and source j uses the path $\{4 \rightarrow 3 \rightarrow 2 \rightarrow 1\}$, then $W_{ij} = 1$. Flow i and flow j meet at node 2, then node 2 combines them and sends a coded packet i + j. Therefore, the number of transmissions reduced is 1. On the other hand, if source j uses $\{2 \rightarrow 3 \rightarrow 4\}$ (Fig. 4.1 (b)), then $W_{ij} = 3$. Since nodes 2, 3, and 4 each only need to transmit one coded packet i + j instead of separate packets i and j, therefore the number of transmissions reduced is 3.

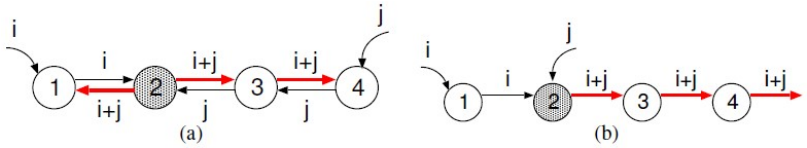

Fig. 4.1: Calculate the weight W_{ij} between flow i and flow j. (a) $W_{ij} = 1$, (b) $W_{ij} = 3$.

For unicast, the route is a simple path, and therefore can be described as a totally ordered list of nodes. The task of computing W_{ij} for unicast becomes trivial, since to compute the number of shared nodes in two totally ordered lists is equivalent to compute the longest common subsequence of two sequences, but this approach cannot be applied to multicast. For multicast, the route is a tree that can be described as a partially ordered list of nodes. If the packets generated by two sources can be coded, the two sets must

share at least one relay node. Sharing relay node(s) is a necessary but not sufficient condition to ensure that the coded packets can be decoded, as the more complicated analysis shows in the following.

Computing W_{ij} for multicast is more involved than for unicast. We first need to decide the partial order in a multicast tree, and represent the partially ordered set as a precedence matrix. Among all shared nodes of two partially ordered sets, we compute a new transmission order that preserves the original order of each multicast tree. The following procedure Weight calculates W_{ij}, in which the multicast trees T_1 of source i and T_2 of source j are given as input. If the returned value $W_{ij} = 0$, then there is no potential benefit from coding flows i and j; if $W_{ij} > 0$, then there is potential benefit from coding (regardless of the decodability at the destination), and the resulting graph G_T suggests where coding should occur.

```
WEIGHT(T1,T2)
1     Let V be the common relay nodes in T1and T2
2     Let n = |V|
3     for k = 1 to 2
4       do Mk = Transitive_Closure(Tk)
5         Reduce matrix Mk to be n ×n by eliminating
          the non-common vertices and relabeling
          the rows and columns
6         for i = 1 to n
7           do Set Mk(i, i) = 0
8           for j = 1 to n
9             do if Mk(i, j) = 1
10            then Mk(j, i) = -1
11        Build a tree of n nodes that preserves the
            precedence relation in Mk, call it tk
12    Graft t1 to t2 (or t2 to t1) to get a minimum graph GT
          that preserves the original transmission order
13    Let m = number of vertices in GT
14    Return 2n – m
```

Remark 1: At line 4, the precedence matrix M_k is obtained from procedure Transitive_Closure [33] to find the precedence relation of the nodes in the original multicast tree. M_k is binary. An entry '1' in cell (i, j) indicates node i is before node j by the partial order specified in the tree; and '0' otherwise, which has two possible implications: (a) j is before i by the partial order; (b) i and j are not ordered so it can be either way.

After line 10, the precedence matrices M_k becomes ternary and has entries 1, -1 or 0:

$$M_k \begin{cases} 1, & \text{if i must transmit before j transmits} \\ -1, & \text{if j must transmit before i transmits} \\ 0, & \text{if i and j are not ordered} \end{cases}$$

Entries with 1 and -1 in M_k indicate strong precedence relations that must be preserved in the subsequent procedure.

Remark 2: The grafting procedure in line 12 can be done while preserving the strong precedence relation in M_1 and M_2. In Fig. 4.2, to graft t_1 into t_2, we add edges (3, 4) and (1, 3) from t_1 (see Fig.4.2 (c)), but we do not need to add edge (1, 2), since node 1 is already a predecessor of node 2. The graph in (c) shows the transmission order if coding occurs at node 4; to graft t_2 into t_1 (see Fig. 4.2 (d)), we add edges (2, 1) and (4, 2) from t_2, but not edge (4, 3), since node 4 is already a predecessor of node 3. The graph in (d) shows the transmission order if coding occurs at node 1. In either (c) or (d), graph G_T has 6 vertices, so a total of 6 transmissions will be sufficient instead of 4 transmissions for each. The number of transmissions that can be reduced by coding is $2 \times 4 - 6 = 2$.

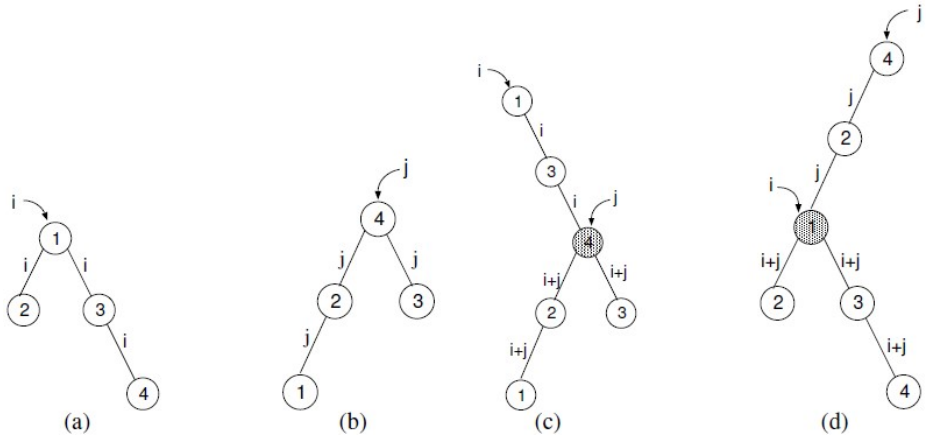

Fig. 4.2: Example of tree grafting. (a) t_1, (b) t_2, (c) graft t_1 into t_2, (d) graft t_2 into t_1. Labels on edges show the source of the packets.

The grafting procedure takes one entire tree and adds edges from the other tree to it to preserve the transmission order specified in both trees. This procedure takes O(E) time, where E = n — 1 is the number of edges in t_1 or t_2. To graft t_1 into t_2 and the other way around yield the same m, since the number of edges that need to be added to the tree is the same.

4.2. LINEAR PROGRAM MODEL FOR OPTIMAL CODING DESIGN

After we obtain the weight W_{ij}, we can use it as a constant in the mathematical optimization model. Recall that W_{ij} denotes the number of transmissions that can be reduced by encoding flow i and flow j. Let the decision variable C_{ij} indicate if flow i should be coded with flow j: C_{ij} = 1 indicates flow i is coded with flow j; =0 otherwise. So the objective should be maximizing the sum of $W_{ij}C_{ij}$ for all pairs of i and j.

We try to find the coding design with the fewest number of transmissions. The rules are: (1) encoding of two packets can only occur at a relay node, so the source node must transmit the original information; (2) decoding is the task of destination nodes, so the relay nodes do not attempt to decode a coded packet; (3) for simplicity, we also assume a flow will be coded with at most one other flow, and keep the same coding pair through its lifetime; (4) packets stay on their predetermined routes. Therefore, if C_{kj} = 1 (flow k is coded with flow j), then the destinations of k and j both receive the coded packet from their predetermined routes.

The constraint of the optimization model is the guaranteed decodability— all destinations must receive data directly or after decoding. Due to the simple rule of XOR coding, if node i receives h = k $\oplus$ j, having knowledge of k can help node i decode j: h $\oplus$ k = j. Therefore node i can decode a received packet h = k $\oplus$ j to recover j if there exists another flow k such that X_{ik} = 1 (i.e., i knows k).

Let D_j denote the destinations of source j. The following rules specify the requirement that every destination i of j must receive from j. X_{ij}, T_{ij} , and C_{kj} are all binary variables. The first indicates i can receive from j either directly (X_{ij} = 1) or by decoding (T_{ij} = 1); The second indicates as long as there is one k such that i knows k (X_{ik} = 1) and k is coded with j (C_{kj} = 1), then i can decode j (T_{ij} = 1);The third one indicates if there is no such k then T_{ij} = 0; The last one indicates a source j has to be either coded with another source k or sent in its original form.

$$X_{ij} + T_{ij} = 1, \qquad \forall i \in D_j, \forall j \in S \tag{1a}$$

$$T_{ij} \geq X_{ik} \wedge C_{kj}, \qquad \forall k \in S, \forall i \in D_j, \forall j \in S \tag{1b}$$

$$T_{ij} \leq \bigvee_{k \in S} X_{ik} \wedge C_{kj}, \qquad \forall i \in D_j, \forall j \in S \tag{1c}$$

$$X_{ij} = 1 - \sum_{k \in S} C_{kj}, \qquad \forall i \in D_j, \forall j \in S \tag{1d}$$

Let L_{kij} denote the logic AND of X_{ik} and C_{kj}, so $T_{ij} \geq L_{kij}$. Recall that for binary variables a, b, and c, $a = b \wedge c$ is equivalent to $a \leq b$, $a \leq c$, and $a \geq b + c - 1$; $a = b \wedge c$ is equivalent to $a \geq b$, $a \geq c$, and $a \leq b + c$. If we know that (b, c) = (0, 1) or (1, 0), then $b \wedge c = b + c$. With these simple manipulations, the above relations can be written in linear inequalities in the following linear program:

Maximize

$$\sum_{j \in S} \sum_{i \neq j \in S} C_{ij} W_{ij} \tag{2}$$

Subject to

$$X_{ij} + T_{ij} = 1, \qquad \forall i \in D_j, \forall j \in S \tag{3a}$$

$$X_{ij} = 1 - \sum_{k \in S} C_{kj}, \qquad \forall i \in D_j, \forall j \in S \tag{3b}$$

$$L_{kij} \leq X_{ik}, \qquad \forall k \in S, \forall i \in D_j, \forall j \in S \tag{3c}$$

$$L_{kij} \leq C_{kj}, \qquad \forall k \in S, \forall i \in D_j, \forall j \in S \tag{3d}$$

$$L_{kij} \geq X_{ik} + C_{kj} - 1, \qquad \forall k \in S, \forall i \in D_j, \forall j \in S \tag{3e}$$

$$T_{ij} \geq L_{kij}, \qquad \forall k \in S, \forall i \in D_j, \forall j \in S \tag{3f}$$

$$T_{ij} \leq \sum_{k \in S} L_{kij}, \qquad \forall i \in D_j, \forall j \in S \tag{3g}$$

$$C_{ij} = C_{ji}, \qquad \forall i, j \in S \tag{3h}$$

$$X_{jj} = 1, C_{jj} = 0, \qquad \forall j \in S \tag{3i}$$

$$X_{ij} = 1, \qquad \forall i \in N_j, \forall j \in S \tag{3j}$$

$$\sum_{i \in S} C_{ij} \leq 1, \qquad \forall j \in S \tag{3k}$$

$$X_{ij} = \{0,1\}, T_{i,j} = \{0,1\}, \qquad \forall i \in D_j, \forall j \in S \tag{3l}$$

$$C_{ij} = \{0,1\}, \qquad \forall i, j \in S \tag{3m}$$

$$L_{kij} = \{0,1\}, \qquad \forall k \in S, i \in D_j, j \in S \tag{3n}$$

The solution to C_{ij} indicates which pairs of flow should be coded for maximum benefit. The relay node that performs the encoding function is already known when W_{ij} is calculated. If all C_{ij}'s are zero, then there is no decodable solution for the given network setting. Therefore the above model can also be used for decodability analysis.

Integer linear programs are NP-hard to solve, but we can use relaxation and rounding to get an approximate solution. In fact, the linear program solver [34] has a built-in relaxation and rounding function. For the above integer linear program as well as the one in Section V, we used the solver directly to obtain integer solutions.

5. TRANSMISSION SCHEDULING

After we obtain the coding solution, the next step is to schedule the transmission of the coded packets and the original packets so that the coding/decoding delay won't degrade the performance of the network. Coding/decoding delay refers to the extra delay caused by the use of network coding, i.e., a packet has to stay in the buffer to wait for its coding pair (or decoding key) to arrive. Since a coded packet has contents of two original packets but only takes one slot to transmit, a new conflict graph is needed.

The result from the coding design in Section IV provides the flow information on each link, including the coded packets and uncoded packets. Given the flow information, we can build a conflict graph $G_C = (V_C, E_C)$, where each vertex $v \in V_C$ is a transmission denoted by a pair (transmitter, flow), and two vertices are connected by an edge if and only if the two transmissions conflict with each other. The flow is the identifier of the source node.

The definition of conflict relation depends on the MAC layer protocol. For instance, if the MAC layer ACK is used, any two links within 2 hops are considered conflicting with each other; But if the MAC layer ACK is not used, two transmissions are considered conflicting with each other if a receiver of one transmitter is in the interference range of the other transmitter. The latter is more appropriate for multicast since the multiple ACKs from receivers can overwhelm the sender. For example, in Fig. 5.1 (a), A and B conflict because a receiver of A is in B's transmission range, but in (b) A and B do not conflict. Different conflict relation definitions may result in different conflict graphs, but the scheduling algorithm proposed in this paper uses the established conflict graph as input and therefore can work with any definition of conflict relation.

If network coding is not used, the number of vertices $|V_C|$ in G_C is the actual number of transmissions (Fig. 5.2 (a)). However, when network coding is used, the actual number of transmissions may be smaller than the number of vertices in G_C (Fig. 5.2 (b)), because the transmission of one coded packet is represented as two vertices in G_C. Since it is essentially one transmission, there will be no edge between the two vertices in G_C (between (a, 1) and (a, 2), between (b, 1) and (b, 2)), which implies the transmission of flow 1 and flow 2 by the same wireless node can happen at the same time.

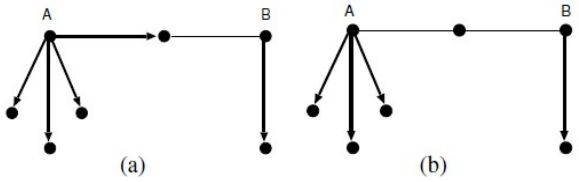

Fig. 5.1: Example of conflict relation. A and B conflict in (a) but not in (b).

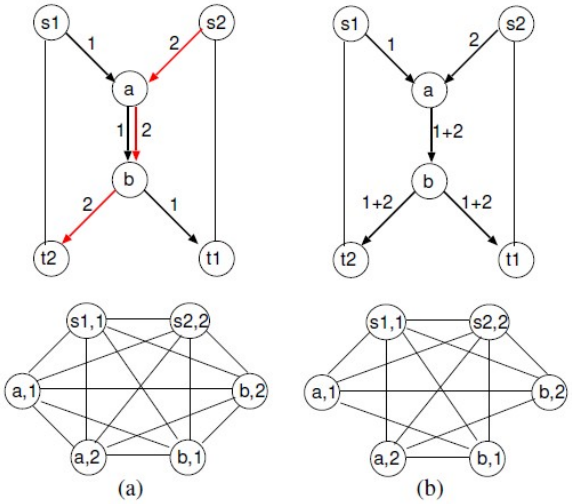

Fig. 5.2: Routing example. (a) Routing without network coding, (b) With network coding. Labels on edges are flow IDs. The bottom row shows the conflict graphs.

After obtaining the conflict graph, we can use an optimization model to compute the slot assignment. R_s is the data rate of source s, given in the number of packets transmitted in a TDMA frame. Let $d_{v,s,i}$ represent the delay at node v for a packet generated by source s, which includes the store-and-forward delay and waiting time before transmission; index i is for the i^{th} packet, and $i = 1..R_s$.

Since a coded packet stays on the original route, we can calculate its delay at a relay node for each source separately. For example, if flow 1 and flow 2 are combined at node v, then the delay for flow 1 is $d_{v,1,i}$ and the delay for flow 2 is $d_{v,2,i}$. Depending on the packet arrival time of flow 1 and flow 2 at node v, $d_{v,1,i}$ and $d_{v,2,i}$ could be different. The difference is the coding delay. At the destination, the time difference between receiving a coded packet and its decoding key is the decoding delay. If we minimize the total end-to-end delay for all flows, we have considered the effect of both coding and decoding delay.

In the following objective function, $P_{s,d}$ is the routing path from s to a destination node d. $v \in P_{s,d}$ is a transmitting node on the path. v could be the source node or a relay

node. D_s is the group of destination nodes of source s. We can minimize the total end-to-end delay using the following objective function:

Minimize

$$\sum_{s\in S}\sum_{i=1}^{R_s}\sum_{d\in D_s}\sum_{v\in P_{s,d}} d_{v,s,i} \tag{4}$$

The constraints for the optimization model are: (1) all transmissions must be conflict-free; (2) the slot assignment can accommodate the traffic load given by the network layer. In the following, $v \in P_s$, or $Path_{v,s} = 1$ means v is a transmitting node on the routing paths of source s. In (5d), $(u, v) \in P_s$ means the directed link (u, v) is on the routing paths, and node u and v both are transmitters. Let F be the total number of distinct slots in a TDMA frame. Let A_s be the packet generation time at source s, which is given as input. The time difference between the transmission time and A_s is the initial access delay at the source. If a packet is one of the coding pairs, it is important that the initial access delay is minimized to reduce the waiting time of the other packet. We introduce binary variables $sl_{v,s,f}$ and $sl_{v,s,f,i}$: $sl_{v,s,f}=1$ indicates slot f is assigned to node v for transmitting packets generated by source s; $sl_{v,s,f,i}$ is for the i^{th} packet among the R_s packets. We can express the constraints of the optimization in the following linear inequalities:

Subject to

$$sl_{v,s,f} + sl_{v',s',f} \le 1, \qquad \forall\big((v,s),(v',s')\big) \in E_C, \forall f = 1..F \tag{5a}$$

$$\sum_{f=1}^{F} sl_{v,s,f,i} = Path_{v,s}, \qquad \forall i = 1..R_s, \forall v \in P_s, \forall s \in S \tag{5b}$$

$$sl_{v,s,f} = \sum_{i=1}^{R_s} sl_{v,s,f,i}, \qquad \forall v \in P_s, \forall s \in S, \forall f = 1..F \tag{5c}$$

$$d_{v,s,i} = \sum_{f=1}^{F} sl_{v,s,f,i} \times f - \sum_{f=1}^{F} sl_{u,s,f,i} \times f + X_{v,s,i}F,$$

$$\forall (u,v) \in P_s, \forall s \in S, \forall i = 1..R_s \tag{5d}$$

$$d_{s,s,i} = \sum_{f=1}^{F} sl_{s,s,f,i} \times f - A_s + x_{s,s,i}F, \qquad \forall s \in S, \forall i = 1..R_s \tag{5e}$$

$$0 < d_{v,s,i} < F, \qquad \forall v \in P_s - \{s\}, \forall s \in S, \forall i = 1..R_s \tag{5f}$$

$$0 < d_{s,s,i} < F, \qquad \forall s \in S, \forall i = 1..R_s \tag{5g}$$

$$sl_{v,s,f} = \{0,1\}, sl_{v,s,f,i} = \{0,1\}, \qquad \forall v \in P_s, \forall s = S, \forall f = 1..F, \forall i = 1..R_s \tag{5h}$$

$$x_{v,s,i} = \{0,1\}, \qquad \forall v \in P_s, \forall s \in S, \forall i = 1..R_s \tag{5i}$$

(5a) requires that any two vertices connected by an edge in the conflict graph not use the same slot to transmit. (5b) – (5c) assign slots to nodes according to the traffic load from the network layer. (5d) – (5e) model the delay of each packet at each node, including the initial access delay at the source node.

In case that a relay node v is transmitting a coded packet from s_1 and s_2, the two vertices in the conflict graph representing the transmission must be assigned to use the same slot, and therefore the following additional constraint is added:

$$\sum_{f=1}^{F} sl_{v,s_1,f,i} \times f = \sum_{f=1}^{F} sl_{v,s_2,f,i} \times f, \forall i = 1..R_s \tag{5j}$$

where $R_s = \min\{R_{s_1}, R_{s_2}\}$. The one with a higher data rate will send the remaining packets uncoded and therefore are not subject to the constraint.

6. MODEL CONSISTENCY

We use the well-known butterfly network to validate the proposed scheme. For the network shown in Fig. 5.2, there are two unicast sessions: s1～>t1 and s2～>t2. If we do not use network coding, flow s1 → a → b → t1 requires three transmissions, and flow s2 → a → b → t2 requires three transmissions. The conflict graph has a clique of size 6, so a total of 6 mutually conflicting transmissions need 6 time slots. If we use network coding, only 4 slots are needed. The conflict graph has a maximum clique of size 4. We first run the Weight procedure to get $W_{12} = 2$ and get graph G_T, which consists of node a and node b and a directed edge from node a to node b. Solving the integer linear program for the optimal coding design, we get $C_{12} = 1$, which indicates flow 1 and flow 2 should be coded at node a.

At the MAC layer, we run the scheduling procedure based on the integer linear program model (4) – (5j). Fig. 6.1 shows the slot assignment on the nodes. The results generated from the proposed scheme are consistent with the prediction.

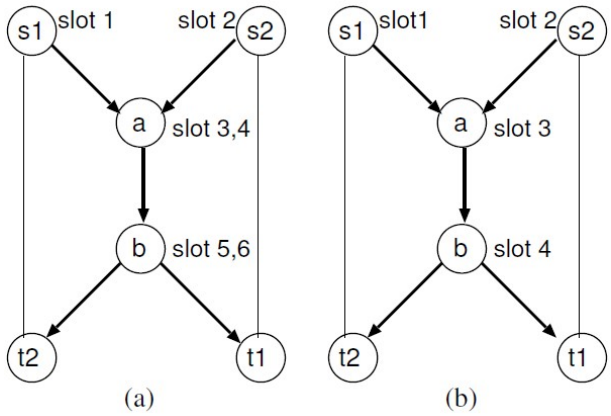

Fig. 6.1: Slot assignment for the Butterfly Network. (a) without using network coding; (b) using network coding.

7. RANDOM LINEAR NETWORK CODING

7.1. LITERATURE SURVEY

Ahlswede et al. [1] showed that with network coding, a source can multicast information at a rate approaching the smallest minimum cut between the source and any receiver when the symbol size approaches infinity. Li et al. [19] proved that a finite symbol size is sufficient for linear coding in multicast. Ho et al. [35] showed a novel randomized coding approach which achieves robustness in a way different from the traditional approaches, and presented the lower bound on the success probability of a random network code. Their following research [13] presented the specific random linear network coding approach in general multisource multicast networks. Feder et al. [36] gave the lower bounds of the coding field size, and the upper bounds between flows from source to destinations based on the number of clashes. Furthermore, they computed the exact probability of the random linear network coding over a Galois Field of size q.

7.2. ALGORITHM IDEAS

When data is sent from one or more sources to one or more destinations using RLNC, each original packet can be divided into s symbols [7]. These symbols can be interpreted from the Galois Field GF(2^s), which has finite number of elements. All the operations are performed over the GF and result in the same field elements. For the original packets $X_1, X_2, \ldots, X_n$, the sources node chooses a set of coding coefficients $g_i = [g_{i1}, g_{i2}, \ldots, g_{in}]$ from the GF(2^s). Hence each original packet has one coefficient. The new coded packet C becomes:

$$C_j = \sum_{j}^{N} g_{ji} \times X_i \tag{6}$$

Since the coefficients are randomly selected and independently from the GF, this approach is referred to as random linear coding.

7.3. DECODABILITY ANALYSIS

When the destination nodes received the set $(g_j, C_j), \ldots, (g_N, C_N)$ of encoded packets, they need to solve the equation (6) to retrieve the original packets. X_i are the

unknowns. This is a linear system with K equations and N unknowns, which can be considered as a matrix form:

$$X = g^{-1} \times C \tag{7}$$

To recovering the original packet, we need $K \geq N$, i.e., the number of received packets has to be at least larger than the number of original packets. However, this condition is not sufficient, because it does not guarantee all the combinations are independent. Specific decoding process will be discussed later.

7.4. COMPARISON

We apply the XOR based network coding and the random linear network coding on the butterfly network as shown in Fig. 7.1 (a) and (b). Without using network coding, it needs 6 hops to send messages b_1 and b_2 to the destinations. With XOR based network coding, it needs 5 hops, and the random linear network coding only needs 4 hops. However, that does not mean the random linear network has the best performance. First of all, its coding process is more complicated than the XOR based network coding. It requires computation over a GF(2^s). Second, if it decides to encode k packets, each relay node will have to wait a period of time to gather k packets. Hence its delay can be much longer. As for the decodability, we can confirm that b_1 and b_2 will be decoded, whereas in random linear network coding, there is a chance that the received combinations are not linear independent. Therefore the XOR based network coding with our scheme performs better in decodability.

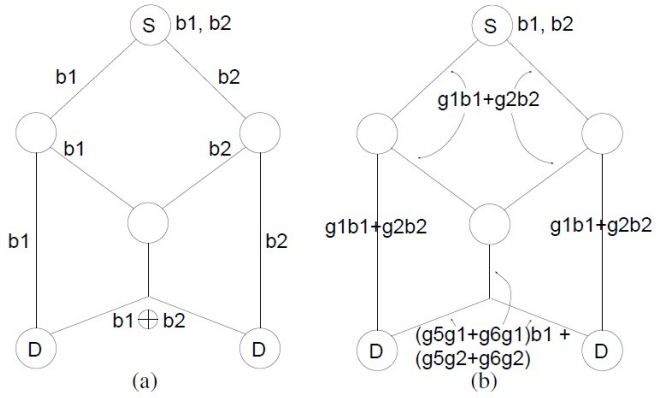

Fig. 7.1: An example in the Butterfly Network. (a) with XOR based network coding; (b) with random linear network coding.

8. SIMULATION

8.1. COMPARE WITH THE NETWORK WITHOUT USING NETWORK CODING

To test if the proposed scheme provides any benefit for networks beyond the well-known butterfly network, we randomly deploy wireless nodes on a 150m × 150m square region. Node transmission range is set to 30m. If two nodes are within 30m of each other, they are connected by a wireless link.

In the first simulation, we test the scheme on networks of 10 to 80 nodes, among which, 20% of the nodes are used as sources of multicasting. Each source has 5 destinations. We randomly choose destinations of each source across the network. The routing information is given, so all packets are transmitted without changing their predetermined routes. We use the network coding design to explore the coding opportunity, and then use the proposed LP-based scheduling scheme to compute the slot assignment. The objective function (4) is used to compute the total end-to-end delay. The results are compared with the simple First-Come, First-Served (FCFS) scheme, in which a node is assigned to use the next available slot as soon as it arrives at a relay node. For a fair comparison, we use the centralized FCFS that is aware of the network topology to make sure the new assignment has no conflict with existing assignments.

The TDMA frame size is 30 slots, and each slot time is one packet transmission time. If the source generates one packet each frame, then the source rate is 1/30 B, where B is the wireless link bandwidth. We define the baseline rate = 1/30 B. We compare the delay performance obtained from the proposed scheme Network Coding with Optimal Scheduling (NC-OptSchedule) with that obtained from the shortest path multicast routing with FCFS scheduling (MOSPF-FCFS). We observed that with multicast destinations randomly distributed across the network, there is little chance for two flows to benefit from network coding. Unicast traffic is worse in terms of coding opportunity. This observation further testifies that if an opportunistic coding scheme is used, in which packets stay on their original routes and relay nodes opportunistically encode packets passing by, some destinations may never be able to receive enough information to decode a coded packet, or have to wait for a long time to collect the needed information. Fig. 8.1

shows that the number of transmissions is the same, but the proposed scheme still outperforms the FCFS scheme. The performance gain comes from using the optimal scheduling scheme. The proposed scheduling scheme outperforms FCFS by 25% to 40%.

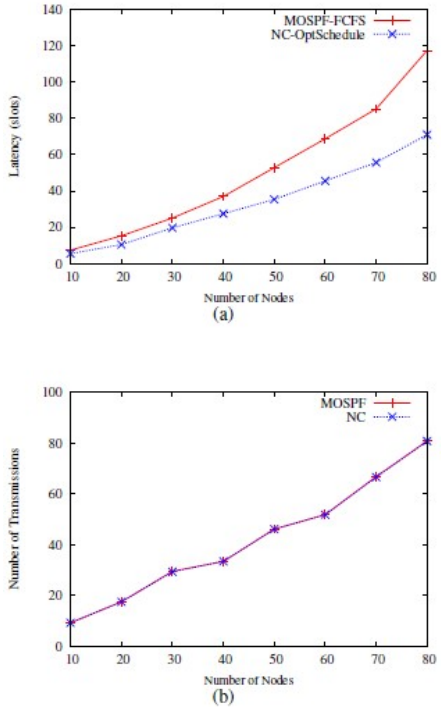

Fig. 8.1: Result with no group communication. (a) End-to-end delay; (b) number of transmissions. The two algorithms have the same number of transmissions.

In the second simulation, we choose N nodes to have group communication (i.e., all-to-all communication). This group of N nodes is randomly chosen from networks of m nodes. Fig. 8.2 shows the results for N = 10, m = 10 to 80. When nodes are having group communication, there are more chances that two flows share a path (or a segment of a path), which creates an opportunity to use network coding. The benefit of using network coding is shown in the number of transmissions and the demand for spectrum bandwidth. The demand for bandwidth is the minimum number of distinct slots needed in order to have conflict-free transmissions. The proposed network coding scheme (NC) shows significant reduction in both as shown in Fig. 8.2 (b) and (c). The overall reduction in delay (see Fig. 8.2 (a)) is achieved from both the network layer by using network coding and the MAC layer by using the proposed optimal scheduling scheme. The results for N = 20, m = 20 to 80 (Fig. 8.3) are consistent with Fig. 8.2.

8.2. COMPARE WITH OTHER NETWORK CODING: RLNC

We first apply the random linear coding strategy on the example network in Fig. 8.4. The network is deployed on a 150m×150m square field. The transmission range is set to 30m. Nodes within 30m to each other are connected by a wireless link. Node positions are randomly generated. None of the node is isolated, which means that there is at least one routing path from every node to reach every other node. The routing information is given, so the routing paths of all packets are predetermined.

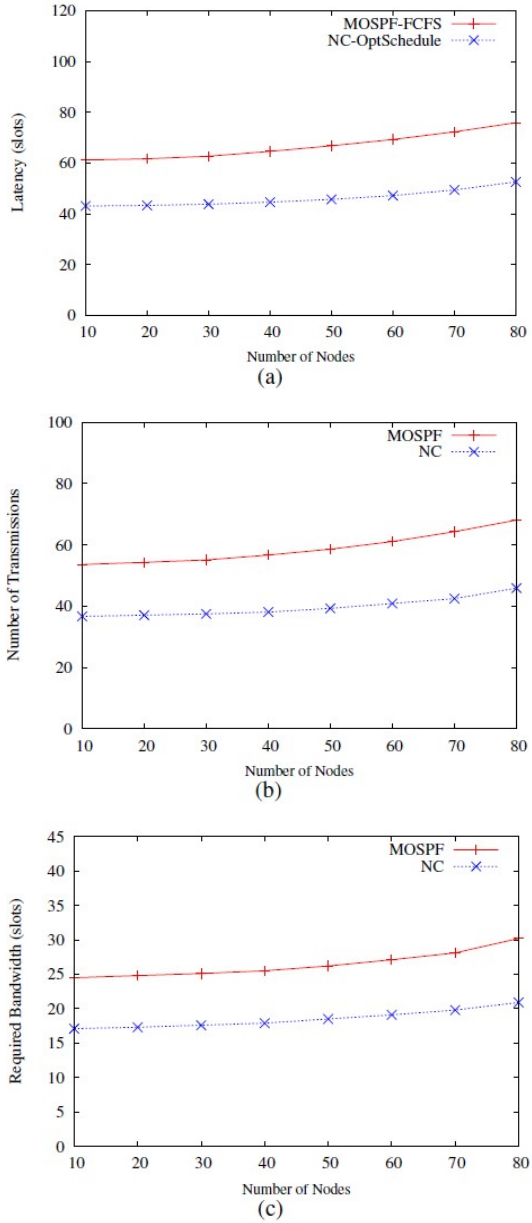

Fig. 8.2: Result with 10 nodes having group communication. (a) end-to-end delay; (b) number of transmissions; (c) required bandwidth.

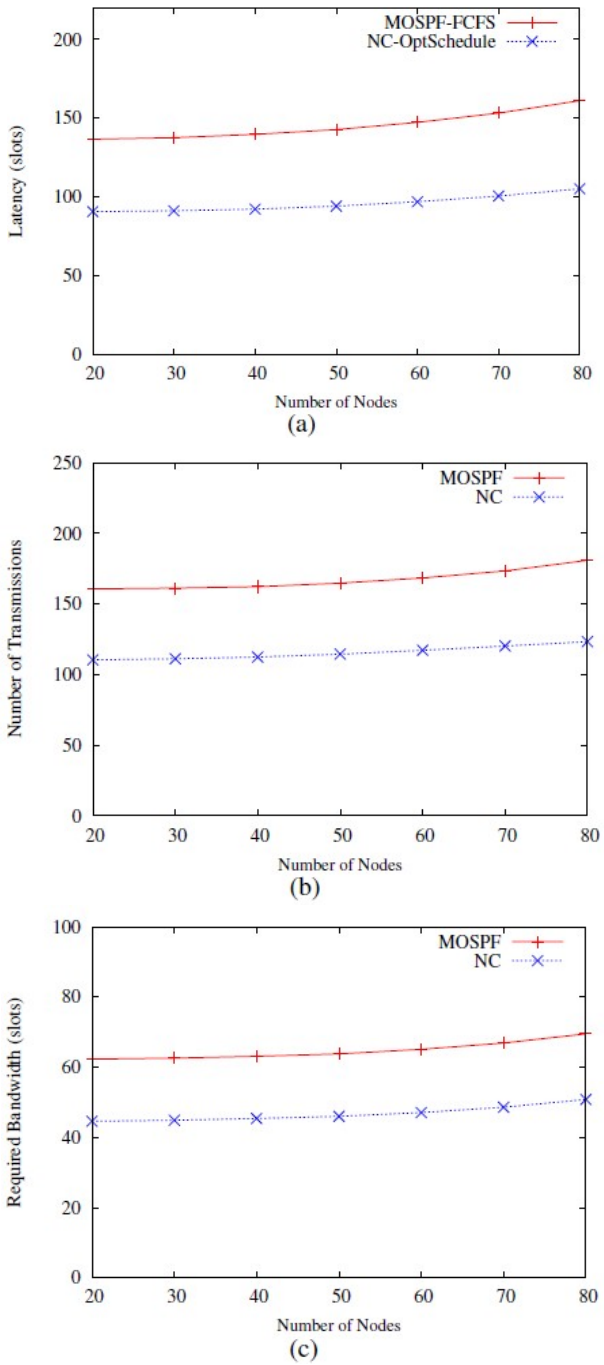

Fig. 8.3: Result with 20 nodes having group communication. (a) end-to-end delay; (b) number of transmissions; (c) required bandwidth.

lower level, but lower level nodes should not receive packets came from higher level. Hence, we used the Breadth-first search algorithm to mark the level on each node. The next step is to generate the original packets X_i. We want to see the performance changes between different packets encoded strategy and different numbers of sent packets. X_i is

set from 2 to 30. Every node contains a receive packets array and a sent packets array. At first, the source node will be inserted 2 packets in the receive packets array. If there are enough packets in this array, they will be encoded into 1 packet and stored in the send packets array. In the next level, nodes will receive all the send packets from the lower level nodes which are within 30m. When the destination nodes received packets, the coefficient of every original packet will be extracted and formed a matrix. We used this matrix to determine whether the encoded packets can be decoded. Specific decoded process will be explained in the following part.

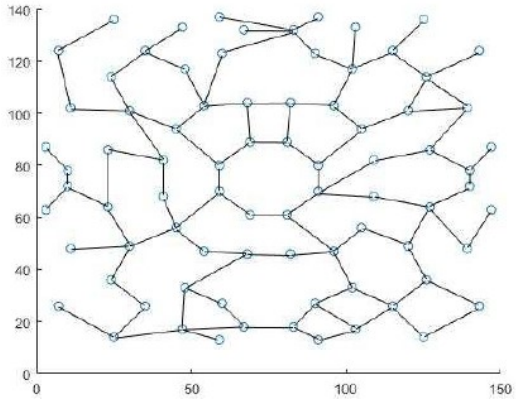

Fig. 8.4: Randomly generated network.

Fig. 8.5 shows the result after we run the test 400 times with different sources and destinations. In Fig.8.5 (b), we observed that the total decodability, which means the probability of all the destinations can decode all the encoded packets, can decrease to 0 when we encode every 10 packets and every 20 packets. Encoded every 2 packets can make sure some of the destinations decode them all, but in a very low decoded rate. Fig. 8.5 (a) shows the sum of the number of decode packets in each destination. When we encode every 2 packets, as expected, the number of decode packets gradually increased along with the number of sent packets. But when we encode every 10 packets, the result number suddenly drops when there are 10 sent packets. That is because these 10 packets are encoded into 1 packet in the source node so that the number of transmitted packets in the network reduces. Same as when we encode every 20 packets.

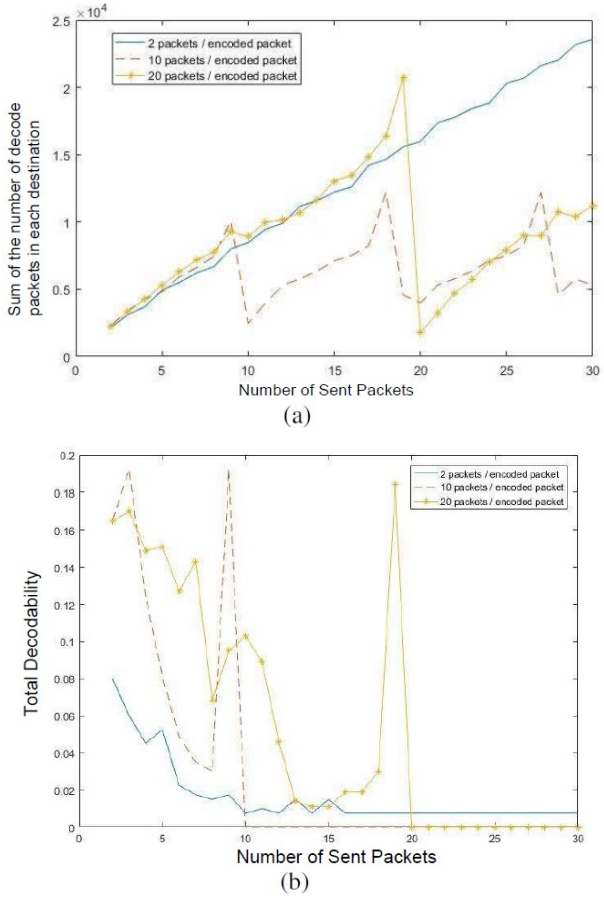

Fig. 8.5: Result of RLNC. (a)The changes of the sum of decode packet in each destination. (b)The total decidability decreases along with the increase number of sent packets.

To clarify the decoding process, we apply the random linear coding algorithm on a simple network in Fig.8.6. Every 2 packets will be encoded into 1, extra packet which can not be encoded will be sent out directly. X_1 and X_2 are the source messages being sent to the destination n_6 and n_7. At first, X_1 and X_2 are encoded into $g_1X_1+g_2X_2$ and sent to n_2 and n_3. But n_2 and n_3 do not receive enough packets, so they pass the packet to the next level. n_4 and n_5 re-encode the packets again and generate different coefficient for X_1 and X_2. The coefficient g_i is randomly selected elements from a finite field. For node n_6, its received coefficient can form a matrix:

$$\begin{pmatrix} g_3g_1 + g_4g_1 & g_3g_2 + g_4g_2 \\ g_5g_1 + g_6g_1 & g_5g_2 + g_6g_2 \end{pmatrix}$$

If the rank of the matrix K > N, which N is the number of original packet, then we can decode the coded packets and get the original data. In this case, we have 2 sent packets, so K needs to larger or equal to 2.

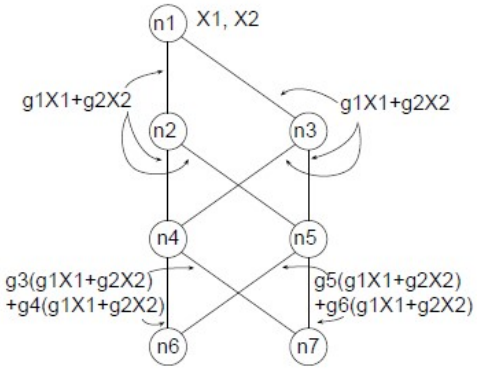

Fig. 8.6: Example of random linear network coding process.

Furthermore, Fig. 8.7 demonstrates the result of this simple network. As the number of packets increase, the total decodability will decrease to 0, no matter what the encode strategy is. As for the number of decoded packet in Fig. 8.7 (a), only 2-packets-per-encoded-packet strategy has a relatively stable decode rate. The results of the other two strategies fluctuate in certain patterns.

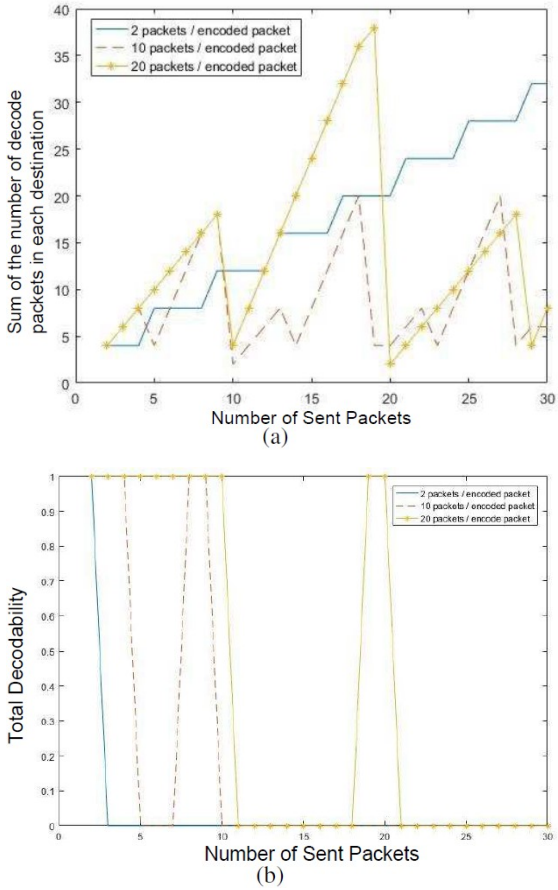

Fig. 8.7: Result of RLNC (Special Case). (a)The changes of the sum of decode packet in each destination. (b)The total decidability decreases along with the increase number of sent packets (cont.).

9. CONCLUSION AND FUTURE WORK

In this paper, we develop a deterministic network coding method and scheduling scheme using linear programming in multi-source multicasting wireless networks. Our network coding method at the network layer is designed to find the most bandwidth-efficient coding solution with guaranteed packet decodeability at all destinations. Our conflict-free, node transmission scheduling algorithm at the MAC layer is designed to minimize network delay. Indeed, our coding and scheduling schemes outperformed the shortest path routing using first-come first-serve scheduling by 25-40%. The coding and scheduling scheme produce consistent result for the well-known butterfly network but are also extensible to any complex network with arbitrary traffic. Our simulation results confirm that network coding is beneficial when a group of nodes are engaged in group communication. Overall, our approach reduces end-to-end delay, improves transmission efficiency, and minimizes bandwidth requirements when a network coding opportunity exists.

Although we assumed a pairwise XOR for encoding and original packet routes are preserved before and after coding, even more efficient solutions may be possible by relaxing one or more of these assumptions. Therefore, future research will explore the joint computation of routing and coding.

BIBLIOGRAPHY

[1] R. Ahlswede, N. Cai, S.-Y.R. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204-1216, July, 2000.

[2] A. Khreishah, C.-C. Wang, and N. B. Shroff, "Optimization based rate control for communication networks with inter-session network coding," in *IEEE INFOCOM*, 2008.

[3] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems,"in *INFOCOM*, 2007.

[4] S. Acedanski, S. Deb, M. Medard, and R. Koetter, "How good is random linear coding based distributed networked storage?" in *In1st Workshop on Network Coding, Theory and Applications (NetCod)*, April 2005.

[5] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks usingrandomized network coding," in *IEEE ISIT 04*, June 2004.

[6] S. Jaggi, M. Langberg, T. Ho, and M. Effros, "Correction of adversarial errors in networks," in *IEEE ISIT* 05, July 2005.

[7] N. Cai and R. W. Yeung, "Network coding and error correction," in *IEEE ITW 02*, 2002.

[8] R. W. Yeung and N. Cai, "Network error correction, part i: Basic concepts and upper bounds," *Communications in Information andSystems*, vol. 6, 2006.

[9] N. Cai and R. W. Yeung, "Network error correction, part ii: lower bounds," *Communications in Information and Systems*, 2006.

[10] Z. Zhang, "Network error correction coding in packetized networks," in *IEEE Information Theory Workshop*, 2006, pp. 433–437.

[11] S. Yang and R. W. Yeung, "Characterizations of network errorcorrection/ detection and erasure correction," in *in Proc. NetCod*, 2007.

[12] T. Cui, T. Ho, and L. Chen, "Distributed distortion optimization for correlated sources," in *ISIT 2007*, Nice, France, June 24 – June29, 2007.

[13] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi, and D. Karger, "A random linear network coding approach to multicast," *IEEE/ACMTrans. on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[14] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network codeconstruction," *IEEE Trans. Inform. Theory*, vol. 51, no. 6, pp. 1973–1982, June 2005.

[15] C.-C. Wang and N. B. Shroff, “Beyond the butterfly - a graph-theoretic characterization of the feasibility of network coding with twosimple unicast sessions,” in *IEEE International Symposium on Information Theory*, Nice, France, June 2007.

[16] Y. Wu, “On constructive multi-source network coding,” in *Intl Symp. Inform. Theory*, Seattle, USA, July 2006.

[17] D. E. Lucani, M. Medard, and M. Stojanovic, “Systematic network coding for time-division duplexing,” in *ISIT 2010*, Austin, Texas,U.S.A., June 13 - 18, 2010.

[18] Y.-P. Hsu, N. Abedini, S. Ramasamy, N. Gautam, A. Sprintson, and S. Shakkottai, “Opportunities for network coding: To wait or notto wait,” in *IEEE International Symposium on Information Theory Proceedings*, 2011.

[19] Y. Li, E. Soljanin, and P. Spasojevi’c, “Collecting coded coupons over generations,” in *ISIT 2010*, Austin, Texas, U.S.A., June 13 -18, 2010.

[20] A. T. Campo and A. Grant, “On random network coding for multicast,” in *ISIT 2007*, Nice, France, June 24 – June 29, 2007.

[21] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, “On randomized network coding,” in *41st Annual Allerton Conference on Communication, Control and Computing, Monticello*, USA, 2003.

[22] R. Dougherty, C. Freiling, and K. Zeger, “Insufficiency of linear coding in network information flow,” *IEEE Trans. Inform. Theory*,vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[23] L. Song, R. Yeung, and N. Cai, “Zero-error network coding for acyclic networks,” *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp.3129–3139, Dec. 2003.

[24] K. Jain, V. Vazirani, R. Yeung, and G. Yuval, “On the capacity of multiple unicast sessions in undirected graphs,” in *Intl Symp. Inform. Theory*, Seattle, USA, July 2006.

[25] X. Yan, J. Yang, and Z. Zhang, “An outer bound for multisource multisink network codign with minimum cost consideration,” *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2373–2385, June 2006.

[26] N. Harvey, R. Kleinberg, and A. Lehman, “On the capacity of information network,” *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp.2345–2364, June 2006.

[27] G. Gramer and S. Savari, “Edge-cut bounds on network coding rates,” *Journal of Network and Systems Management*, vol. 14, no. 1,pp. 49–67, March 2006.

[28] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, “Network coding for multiple unicasts: An approach based on linearoptimization,” in *Intl Symp. Inform. Theory*, Seattle, USA, July 2006.

[29] A. Eryilmaz and D. Lun, “Control for inter-session network coding,” in *Workshop on Network Coding, Theory and Applications*, Jan.2007.

[30] Y. E. Sagduyu and A. Ephremides, “Joint scheduling and wireless network coding,” in *First Workshop on Network Coding, Theory, and Applications*, Riva Del Garda, Italy, Apr. 2005.

[31] H. Yomo and P. Popovski, “Oppportunistic scheduling for wireless network coding,” in *IEEE International Conference on Communications’ 07*, June 2007, pp. 5610–5615.

[32] M. Cheng and Q. Ye, “Transmission scheduling based on a new conflict graph model for multicast in multihop wireless networks,” in *IEEE Globecom*, 2012.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, 3rd edition*. MIT Press, 2009.

[34] “http://lpsolve.sourceforge.net/,” October 2016.

[35] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, “The benefits of coding over routing in a randomized setting,” *Information Theory, 2003. Proceedings. IEEE International Symposium on*, pp. 442–, June-4 July 2003.

[36] D. Ron M. Feder and A. Tavory, “Exact Decoding Probability under Ran-dom Linear Network Coding". in *Electronic Colloquium on Computational Complexity*, vol. 10, no. 33, 2003.

VITA

Junwei Su, earned Bachelor Degree in December, 2014 in Computer Science from Drury University. He received his Master Degree in Computer Science from Missouri University of Science and Technology in May 2017. His research interest was in computer networks, wireless networks and network coding.