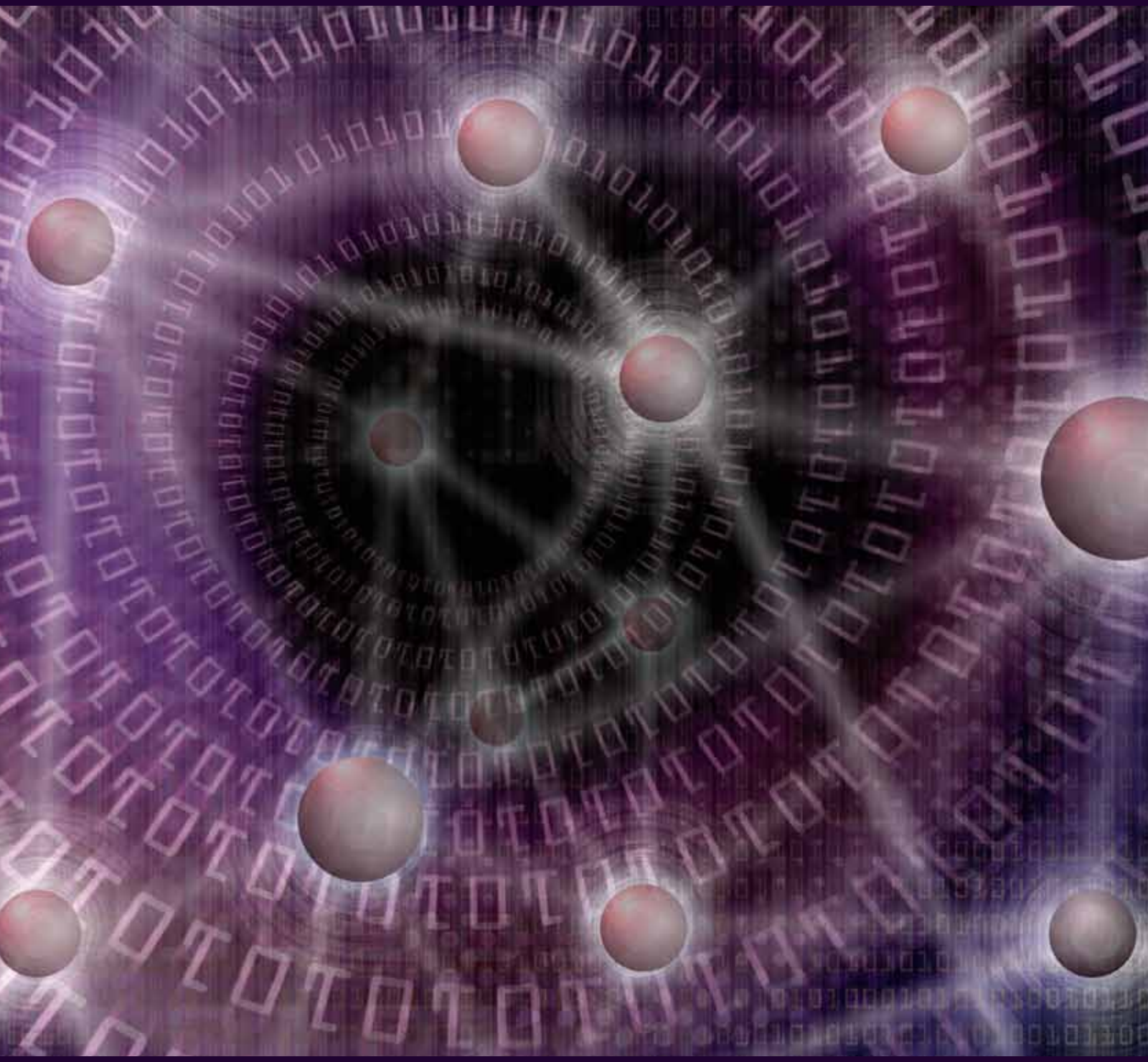
EURASIP Journal on Wireless Communications and Networking

Network Coding for Wireless Networks

Guest Editors: Heung-No Lee, Sae-Young Chung,
Christina Fragouli, and Zhi-Hong Mao



Network Coding for Wireless Networks

EURASIP Journal on
Wireless Communications and Networking

Network Coding for Wireless Networks

Guest Editors: Heung-No Lee, Sae-Young Chung,
Christina Fragouli, and Zhi-Hong Mao

Contents

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 359475, 2 pages
doi:10.1155/2010/359475

Editorial
Network Coding for Wireless Networks

Heung-No Lee,[1] Sae-Young Chung,[2] Christina Fragouli,[3] and Zhi-Hong Mao[4]

[1] *School of Information and Communications, GIST, Gwangju, Republic of Korea*
[2] *School of EECS, KAIST, Daejeon, Republic of Korea*
[3] *School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland*
[4] *ECE Department, The University of Pittsburgh, Pittsburgh, PA, USA*

Correspondence should be addressed to Heung-No Lee, heungno@gist.ac.kr

Received 31 December 2010; Accepted 31 December 2010

The main idea in network coding was introduced in 2000 by Ahlswede et al. With network coding, an intermediate node can not only forward its incoming packets but also encode them. It has been shown that the use of network coding can enhance the performance of wired networks significantly. Recent works have indicated that network coding can also offer significant benefits for wireless networks.

Communications over wireless channels are error-prone and unpredictable due to fading, mobility, and intermittent connectivity. Moreover, in wireless networks, transmissions are broadcasted and can be overheard by neighbors, which is treated in current systems as interference. Furthermore, security poses new challenges in wireless networks, where both passive and active attacks have quite different premises than in wired networks. Ideas in network coding promise to help toward all these issues, allowing to gracefully add redundancy to combat errors, take advantage of the broadcast nature of the wireless medium and achieve opportunistic diversity, exploit interference rather than be limited by it, and provide secure network communication against adversarial attacks.

In this special issue, we have been able to put together six original research articles which we believe can carry the momentum further and take the wireless network coding research to the next level. One article investigates the energy efficiency benefit of using network code. The other article exploits the superposition principle of radio waves in improving network coding performance. Two articles suggest various network coding strategies for relay networks. The last two aim to design minimal decoding delay network coding schemes for broadcast networks. While pointing detailed explanation to these individual articles, we will briefly introduce them here one by one.

J. Goseling et al., in the first paper of this special issue "*Lower bounds on the maximum energy benefit of network coding for wireless multiple unicast,*" investigate the benefit of using network coding for reducing energy consumption in wireless networks. The energy benefit of using network coding in d-dimensional networks, the paper indicates, is at least $2d/\lfloor\sqrt{d}\rfloor$-fold, compared to the case of using the plain routing solution.

S. Zhang and S. C. Liew in the second paper, "*Application of physical-layer network coding in wireless networks*," investigate the use of physical-layer network coding (PNC) for wireless networks. The idea of PNC is to exploit the inherent property of the radio channel that radio waves from different users superpose at the receiver antenna. This property can be used to carry out the addition operation needed in network coding and can be utilized to achieve substantial increase in throughput compared to conventional network coding schemes.

In the third paper, "*Joint channel-network coding for the Gaussian two-way two-relay network,*" P. Hu et al. investigate a two-way relay channel problem and consider five different network coding strategies made from a combination of basic ones such as Amplify-Forward (AF), Decode-Forward, and Decode-Amplify Forward. They have done extensive performance evaluations of these strategies for various relay channel environments.

B. Du and J. Zhang in the fourth paper, "*Parity-check network coding for multiple access relay channel (MARC) in wireless sensor cooperative communications,*" aim to design a parity-check network coding scheme for a two-source multiple access relay channel. The parity-check network code, they imply, is a multidimensional low-density parity-check (LDPC) code. Each user employs an LDPC code to

encode one's own source data, and the relay adds extra parity-check bits. The extra bits can be used as a "binning" process and make the overall coding scheme to be stronger in its error correction capability.

X. Wang et al., in the fifth paper "*Data dissemination in wireless sensor networks with network coding,*" aim at applying network coding in data dissemination problems which may arise in wireless sensor networks, such as software upgrades and an addition of new functionality. A package of data, often very large in its complete size, should be delivered in its entirety to each individual sensor. But the package of packets may not be delivered to all the sensors in a timely manner because some of them may have been put to operate in a sleep mode. Sleep scheduling is frequently used in wireless sensor networks as a means to save the battery, perhaps the most critical resource. The authors focus on the design of an effective XOR-based network coding strategy given a sleep scheduling information.

P. Sadeghi et al. investigate the design of feedback-based adaptive network coding schemes for packet erasure networks in the last paper "*An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels.*" The aim is to deliver high throughputs and low decoding delays. Two main throughput optimal schemes are fountain codes and random linear network codes since they do not require feedback about erasures. But their throughout optimality may come at the cost of large decoding delays. In the application layer, having to wait for the entire coded block to arrive can result in unacceptable delays. This paper focuses on designing network coding schemes with the help of feedback that can deliver innovative packets in any order to the destination and guarantee fast decoding.

Acknowledgments

We would like to thank all the participating authors who have responded to our call for papers with submission of high quality original papers and many experts who have accepted our review invitations, anonymously worked to review the submitted papers thoroughly, and provided constructive comments and advice which have improved the quality of this special issue.

Heung-No Lee
Sae-Young Chung
Christina Fragouli
Zhi-Hong Mao

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 605421, 13 pages
doi:10.1155/2010/605421

Research Article

Lower Bounds on the Maximum Energy Benefit of Network Coding for Wireless Multiple Unicast

Jasper Goseling,[1,2] Ryutaroh Matsumoto,[3] Tomohiko Uyematsu,[3] and Jos H. Weber[1]

[1] *Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands*
[2] *Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands*
[3] *Department of Communication and Integrated System, Tokyo Institute of Technology, Tokyo 152-8552, Japan*

Correspondence should be addressed to Jasper Goseling, j.goseling@ieee.org

Received 28 September 2009; Accepted 21 April 2010

Academic Editor: Heung-No Lee

We consider the energy savings that can be obtained by employing network coding instead of plain routing in wireless multiple unicast problems. We establish lower bounds on the benefit of network coding, defined as the maximum of the ratio of the minimum energy required by routing and network coding solutions, where the maximum is over all configurations. It is shown that if coding and routing solutions are using the same transmission range, the benefit in d-dimensional networks is at least $2d/\lfloor\sqrt{d}\rfloor$. Moreover, it is shown that if the transmission range can be optimized for routing and coding individually, the benefit in 2-dimensional networks is at least 3. Our results imply that codes following a *decode-and-recombine* strategy are not always optimal regarding energy efficiency.

1. Introduction

Emerging applications in wireless networks, like environment monitoring in rural areas by ad hoc networks, require more and more resources. One of the most important limitations is formed by battery life. Since battery technology is not keeping up with the increasing demand from resource-consuming applications, it is imperative that more efficient use is made of the available energy. There has been significant recent attention to the problem of minimizing energy consumption in networks. Some of the topics considered are minimum cost routing [1–3], power control algorithms [4–6], and cross-layer protocol design for energy minimization [7]. In this work, we are interested in the use of network coding [8–14] for reducing the energy consumption in wireless networks. We compare the reduction with traditional routing solutions. The contributions of this work are lower bounds on the energy reduction that can be achieved by using network coding for multiple unicast problems in wireless networks.

In recent years, there has been significant interest in network coding with the aim of reducing energy consumption in networks. More generally, network coding with a cost criterion has been considered. Much progress has been made in understanding the case of multicast traffic. In fact, it has been shown by Lun et al. that a minimum-cost network coding solution can be found in a distributed fashion in polynomial time [15]. The fact that the complexity of finding this solution is polynomial in time is surprising, since the corresponding routing problem is a Steiner tree problem that is known to be NP-complete [16].

Besides constructing minimum-cost coding solutions, it is also of interest to know what the benefits of network coding are compared to routing. In this work we, are interested in the energy benefit of network coding, which is the ratio of the minimum energy solution in a routing solution compared to the minimum energy network coding solution, maximized over all configurations. It has been shown by Goel and Khanna [17] that the energy benefit of network coding for multicast problems in wireless networks is upper bounded

by a constant. The problem of reducing energy consumption for many-to-many broadcast traffic in wireless networks has been studied by Fragouli et al. in [18] and Widmer and Le Boudec in [19], providing lower bounds on the energy benefit of network coding for specific topologies. More importantly, algorithms have been presented in [18, 19] that allow to exploit these benefits in practical scenarios, that is, in a distributed fashion.

The above demonstrates that for multicast traffic and for many-to-many broadcast traffic, there is some understanding of the energy benefits of network coding and how to exploit them. In order to reduce energy consumption in practical networks, however, it is important to consider also multiple unicast traffic. Indeed, in practice a large part of the data will be generated by unicast sessions. For the case of multiple unicast traffic, contrary to multicast and broadcast, not much is known. This paper deals with the energy benefits of network coding for wireless multiple unicast. Remember from the above that for multicast, the problem of minimum-cost routing is hard, whereas minimum-cost network coding is easy. In stark contrast, the problem of minimum-cost multiple unicast routing is easy. One constructs the minimum-cost solution, that is, the shortest path, for each session individually. The minimum-cost multiple unicast network coding problem, however, seems hard and in general very little is known.

Network coding for the multiple unicast problem was first studied by Wu et al. in [20], in which it was shown that in the information exchange problem on the line network, the energy saving achieved by network coding is a factor two. The network codes that we construct in this work are in a sense a generalization of the results on one-dimensional networks [20], to higher-dimensional networks. The networks considered in this work are lattices. More specifically, the hexagonal lattice and the rectangular lattice. Effros et al. [21] and Kim et al. [22] have considered energy-efficient network codes on the hexagonal lattice. We improve the lower bounds on the energy savings of network on the hexagonal lattice given in [21]. More precisely, we improve the previously known bound of 2.4 and obtain a new bound of 3.

Kramer and Savari have developed techniques that can be used to upper bound the achievable throughputs in a multiple unicast problem [23]. No methods are known, however, to lower bound the cost of network coding solutions for a configuration. A lower bound to the ratio of the minimum energy consumption of routing and coding solutions for a given multiple unicast configuration was provided by Keshavarz-Haddad and Riedi in [24]. For the type of configurations used in this paper, however, the results from [24] give the trivial lower bound of one. We will see, however, that network coding has large energy savings for these configurations.

An important class of network codes operates according to a principle that we will refer to in the remainder as *decode-and-recombine*. These codes satisfy the constraint that each symbol in each linear combination that is transmitted is explicitly known by the node transmitting that linear combination. Note, that this is a restriction from the general linear coding strategy, in which linear combinations of coded messages can be retransmitted. The motivation behind using decode-and-recombine codes is that it prevents information from spreading too much in the network, away from the path between source and destination, a heuristic introduced by Katti et al. [25]. The use of a decode-and-recombine strategy results in reduced complexity. However, an important question that has to be addressed is whether the use of decode-and-recombine codes leads to a higher energy consumption than is strictly necessary. We answer this question affirmatively. An upper bound of three on the energy benefit of decode-and-recombine codes has been given by Liu et al. [26]. One of the contributions of this work is to show that larger energy benefits can be obtained by considering also other types of codes.

This paper is organized as follows. In Section 2 we specify our model and problem statement more precisely. Our main results are presented in Section 3. Constructions of configurations that allow a large energy benefit for network coding and proofs of our results are given in Sections 4 and 5. In Section 6, finally, we discuss our work.

2. Model and Problem Statement

Let $V \subset \mathbb{R}^d$ be the nodes of a d-dimensional wireless network. We consider a wireless network model with broadcast, where all nodes within range r of a transmitting node can receive, and nodes outside this range cannot. More precisely, given a transmission range r, a node v is broadcasting to all nodes in the set

$$\{u \in V \mid \|u - v\| \leq r\}, \tag{1}$$

where $\|u - v\|$ denotes the Euclidean norm of $u - v$. The energy required to transmit one unit of information to all other nodes within range r equals cr^α, where α is the path loss exponent and c is some constant. In analyzing the energy consumption of nodes, we will consider only the energy consumed by transmitting. Receiver energy consumption as well as energy consumed by processing are assumed to be negligible compared to transmitter energy consumption. In particular, note that little additional processing is required for network coding, compared to the processing that is performed in a traditional wireless protocol stack.

The traffic pattern that we consider is multiple unicast. All symbols are from the field $\mathbb{F}_2$, that is, they are bits and addition corresponds to the xor operation. The source of each unicast session has a sequence of source symbols that need to be delivered to the corresponding destination. Let M be the set of unicast sessions. We call $\{V, M, r\}$ a wireless multiple unicast configuration.

We will compare energy consumption of routing and network coding. Our goal is to establish lower bounds on the maximum of the ratio of the minimum energy required by routing and network coding solutions, where the maximum is over all configurations. We will refer to this ratio as the *energy benefit of network coding*. Let $\mathcal{E}_{\text{coding}}(V, M, r)$ and $\mathcal{E}_{\text{routing}}(V, M, r)$ be the minimum energy required for network coding and routing solutions, respectively, for

a configuration $\{V, M, r\}$. The energy consumption of a coding or routing scheme is defined as the time-average of the total energy spent by all nodes in the network to deliver one symbol for each unicast session. In analyzing coding schemes, we will ignore the energy consumption in an initial startup phase and consider only steady-state behavior.

Note that since energy consumption per transmission equals cr^{α}, the transmission range r is an important factor in the energy consumption. Therefore, it is of particular interest to optimize the transmission range such that energy consumption is minimized. In this work, we consider two different quantities: (1) B_{fixed}, denoting the energy benefit that can be obtained if the transmission range is given and fixed and (2) B_{var}, denoting the energy benefit that can be obtained if one is allowed to optimize the transmission range. Note that the transmission range can be individually optimized for the routing and network coding scenarios. More precisely, the goal of this work is to establish lower bounds on

$$B_{\text{fixed}}(d) = \max_{V,M,r} \frac{\mathcal{E}_{\text{routing}}(V, M, r)}{\mathcal{E}_{\text{coding}}(V, M, r)}, \tag{2}$$

where the maximization is over all node locations $V \subset \mathbb{R}^d$, multiple unicast sessions M, and transmission ranges r, with the transmission range equal for the routing and network coding solutions, and

$$B_{\text{var}}(d) = \max_{V,M} \frac{\min_r \mathcal{E}_{\text{routing}}(V, M, r)}{\min_r \mathcal{E}_{\text{coding}}(V, M, r)}, \tag{3}$$

where the maximization is over all node locations $V \subset \mathbb{R}^d$ and multiple unicast sessions M, with the transmission range being optimized individually for the routing and network coding solutions. If no confusion can arise, we will omit dependency on d in the notation for B_{fixed} and B_{var}.

Since in B_{fixed}, r is equal for $\mathcal{E}_{\text{routing}}$ and $\mathcal{E}_{\text{coding}}$, the energy per transmission is equal in $\mathcal{E}_{\text{routing}}$ and $\mathcal{E}_{\text{coding}}$ and the benefit is equal to the ratio of the number of transmissions required in routing and network coding solutions.

Since we are interested in energy consumption only, we can assume that all transmissions are scheduled sequentially and/or that there is no interference. All coding and routing schemes that we consider proceed in time slots or rounds. In each time slot, all nodes are allowed to transmit one or more messages. We assume that the length of the time slot is large enough to accommodate sequential transmission of all messages in that round. Coding operations will be based on messages received in previous time slots only. Finally, we assume that all nodes have complete knowledge of the network topology and the network code that is being used.

To conclude this section, we introduce here some of the notation that will be used in the remainder of the paper. The symbol transmitted by a node $v \in V$ in time slot t is denoted by $x_t(v)$. If v transmits more than one symbol in time slot t, these will be distinguished by a superscript, giving, for instance, $x_t^1(v)$ and $x_t^2(v)$. Nodes are represented by vectors. Given vectors $u = (u_1, \ldots, u_d)$ and $v = (v_1, \ldots, v_d)$, let $u_k^l \triangleq (u_k, \ldots, u_l)$, $(u, v) \triangleq (u_1, \ldots, u_d, v_1, \ldots, v_d)$, and $u^{\setminus i} \triangleq (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_d) = (u_1^{i-1}, u_{i+1}^d)$.

Unicast sessions are denoted by $m^i(u)$, with i being an integer and u a vector. We will see in Sections 4 and 5 that u defines the location of the source and i the relative location of the destination, that is, the direction of the session. In some cases $m^i(u)$ will be denoted as $m^i(u_1, u_2^d)$ or similar forms. The tth source symbol of a session $m^i(u)$ is denoted by $m_t^i(u)$. The source and destination of session $m^i(u)$ are denoted by $s^i(u)$ and $r^i(u)$, respectively.

3. Results

We provide lower bounds on B_{var} and B_{fixed}.

Theorem 1. *The ratio of the minimum energy consumption of routing solutions and the minimum energy consumption of network coding solutions, maximized over all node locations, multiple unicast sessions, and transmission ranges, with the transmission range equal for the routing and network coding solutions, is at least* $2d/\lfloor \sqrt{d} \rfloor$*, that is,*

$$B_{\text{fixed}}(d) \geq \frac{2d}{\lfloor \sqrt{d} \rfloor}. \tag{4}$$

The result states that B_{fixed} is at least 2, 4, and 6 for 1-, 2- and 3-dimensional networks, respectively. The result that B_{fixed} is at least 2 in one-dimensional networks also follows from the results in [20]. The lower bound 4 for 2-dimensional networks exceeds the previously known bound of 2.4 [21]. This new lower bound is of particular interest, since it exceeds the upper bound of 3 for decode-and-recombine type network codes [26]. Indeed, the code that we construct does not follow a decode-and-recombine strategy. This shows that energy can be saved by considering strategies other than decode-and-recombine. No lower bounds for three-dimensional networks have been previously established.

Before proving Theorem 1 in Section 5, we provide some intuition. The configuration used to proof Theorem 1 has nodes placed at a d-dimensional rectangular lattice, connectivity $r = \sqrt{d}$ and is parameterized by an integer K controlling the size of the network. The network is given in Figure 1 for $d = 2$ and $K = 5$. For $d = 2$, the result of Theorem 1 is obtained as follows. First consider the case of routing. Note, that the minimum-energy solution is to route all packets along the shortest path between source and destination. Therefore, all nodes in the interior of the network will need to transmit four times. Now, for the case of network coding, we will show in Section 5 that it is possible to construct a network code in which each node in the interior of the network is transmitting only once in each time slot. Therefore, by considering large K and neglecting the energy consumption at the borders of the network, the obtained energy benefit is 4.

In Section 5 we will consider the general case of arbitrary d. Again, the network coding solution will be such that each of the $K^d + \mathrm{O}(K^{d-1})$ nodes in the interior of the network is transmitting only once in each time slot. In analyzing the routing solution, some care needs to the taken. Since $r = \sqrt{d}$, the number of hops that need to be taken on the shortest path

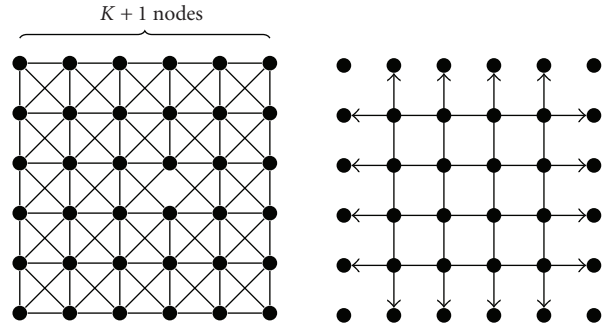


Figure 1: Configuration for which $\mathcal{E}_{\text{routing}}/\mathcal{E}_{\text{coding}} = 2d/\lfloor\sqrt{d}\rfloor$, with $d = 2$ depicted here, is achievable. Nodes are located at integer coordinates in a d-dimensional space, with connectivity given by $r = \sqrt{d}$, as depicted in (a). Unicast sessions are placed according to (b).

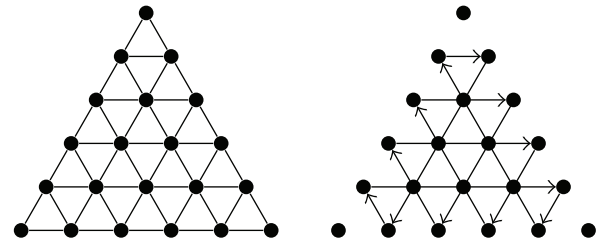


Figure 2: Configuration for which $\mathcal{E}_{\text{routing}}/\mathcal{E}_{\text{coding}} = 3$ is achievable. Nodes are a subset of the hexagonal lattice, with connectivity as depicted in (a). Unicast sessions are placed according to (b).

between source and destination equals $\lceil K/\lfloor\sqrt{d}\rfloor\rceil$. By noting that the number of sessions is roughly equal to the number of nodes at the border of the network, that is, $2dK^{d-1}+\mathrm{O}(K^{d-2})$, and ignoring all transmission from nodes at the border of the network, we establish

$$\begin{aligned} B_{\text{fixed}}(d) &\geq \lim_{K\to\infty} \frac{\left[2dK^{d-1}+O\left(K^{d-2}\right)\right]\left[K/\left\lfloor\sqrt{d}\right\rfloor\right]}{K^d+O(K^{d-1})} \\ &= \lim_{K\to\infty} \frac{2d/\left\lfloor\sqrt{d}\right\rfloor K^d+O\left(K^{d-1}\right)}{K^d+O(K^{d-1})} \\ &= \frac{2d}{\left\lfloor\sqrt{d}\right\rfloor}. \end{aligned} \tag{5}$$

Details of the configuration and a proof of Theorem 1 are given in Section 5.

The configuration and network code construction used for Theorem 1 are not useful for obtaining bounds on B_{var}. Since $r = \sqrt{d}$, the cost per transmission in the network coding scheme is $cd^{\alpha/2}$. One can verify, however, that the optimal transmission range under routing is $r = 1$. This requires K hops per session, with the cost per transmission being equal to c. Using the network code described above and the optimal routing solution at $r = 1$ gives

$$\begin{aligned} B_{\text{var}}(d) &\geq \lim_{K\to\infty} \frac{cK\left[2dK^{d-1}+O\left(K^{d-2}\right)\right]}{cd^{\alpha/2}[K^d+O(K^{d-1})]} \\ &= 2d^{1-\alpha/2}, \end{aligned} \tag{6}$$

which is at most 2, since $\alpha \geq 2$. Note that it was already shown in [20] that $B_{\text{var}}(1) \geq 2$ and in [21] that $B_{\text{var}}(2) \geq 2.4$.

By considering a different configuration, we show that $B_{\text{var}}(2) \geq 3$.

Theorem 2. *For 2-dimensional wireless networks, the ratio of the minimum energy consumption of routing solutions and the minimum energy consumption of network coding solutions, maximized over all node locations and multiple unicast sessions, with the transmission range optimized individually for the routing and network coding solutions, is at least 3, that is,* $B_{\text{var}}(2) \geq 3$.

Here we provide an intuitive explanation of this result; details of the configuration and a proof of Theorem 2 are provided in Section 4. The result is established using a multiple unicast configuration on a subset of the 2-dimensional hexagonal lattice as depicted in Figure 2. The minimum cost routing solution on this network follows shortest paths for all sessions and will require all nodes in the interior of the network to transmit three times in order to deliver one symbol for each session. In Section 4, we construct a network code in which each node in the interior is only transmitting once per delivered symbol. By making the size of the network large, the influence of the borders becomes negligible. Hence, the energy benefit is 3.

Besides providing new lower bounds on the energy benefit of network, the network codes that are constructed in this paper are of interest by themselves. They might lead to insight in how to operate in networks with another structure.

Finally, even though the case $d > 3$ is not of any practical relevance, the bounds as well as the code constructions might lead to a better insight for lower-dimensional networks.

4. An Efficient Code on the Hexagonal Lattice

In this section, we present a multiple unicast configuration in which the nodes form a subset of the hexagonal lattice. It will be shown that the energy benefit on this configuration is 3, proving Theorem 2. Since the code construction used here is less involved then the construction used to prove Theorem 1, we start with the proof of Theorem 2. This section is organized as follows. In Section 4.1 we present the configuration in more detail after which we give the construction of the network code in Section 4.2. Section 4.3 is used to prove that the code is valid. Finally, in Section 4.4 we analyze the energy consumption of the network code and prove Theorem 2.

4.1. Configuration. The size of the configuration is parameterized by a positive integer K. The nodes V form a subset of the hexagonal lattice. We index nodes with a tuple $(v_1, v_2) \in \mathbb{N}^2$. V is given by

$$V = \{(v_1, v_2) \mid v_1, v_2 \geq 0, v_1, v_2 \leq K, v_1 + v_2 \leq K\}. \quad (7)$$

The location of node $v \in V$ in $\mathbb{R}^2$ is given by vG, where

$$G = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}. \quad (8)$$

Let $\overset{\circ}{V}$ denote the interior of the network, that is,

$$\overset{\circ}{V} = \{v \in V \mid v_1, v_2 > 0, v_1, v_2 < K, v_1 + v_2 < K\}. \quad (9)$$

The transmission range that we are interested in is $r = 1$. This leads to connectivity between the six nearest neighbours. Hence, the neighbours of a node $(u_1, u_2) \in \overset{\circ}{V}$ are

$$\begin{aligned} &(u_1 - 1, u_2 + 1), \quad (u_1, u_2 + 1), \quad (u_1 - 1, u_2), \\ &(u_1 + 1, u_2), \quad (u_1, u_2 - 1), \quad (u_1 + 1, u_2 - 1). \end{aligned} \quad (10)$$

The nodes V and the connectivity are depicted in Figure 3.

There are $3(K-1)$ unicast sessions, denoted by $m^1(i)$, $m^2(i)$, and $m^3(i)$, $1 \leq i \leq K-1$. Sources and destinations of the sessions are positioned as follows:

$$\begin{aligned} m^1(i) : s^1(i) &= (0, i), & r^1(i) &= (K - i, i), \\ m^2(i) : s^2(i) &= (i, K - i), & r^2(i) &= (i, 0), \\ m^3(i) : s^3(i) &= (K - i, 0), & r^3(i) &= (0, K - i), \end{aligned} \quad (11)$$

as depicted in Figure 4. Remember from Section 2, that session $m^j(i)$ has the sequence of source symbols $m_0^j(i), m_1^j(i), m_2^j(i), \ldots$ to be transferred.

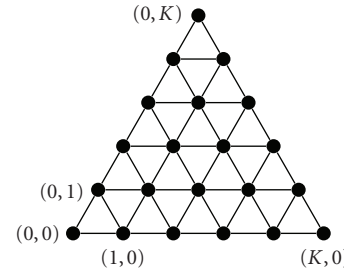


Figure 3: Nodes at a subset of the hexagonal lattice with the connectivity induced by a transmission range $r = 1$. The size of the network is controlled by K, with $K = 5$ in this figure.

4.2. Network Code. The network code is such that in each time slot a new source symbol from each session is transmitted. Also, one symbol of each session is decoded by its destination in each time slot. After successfully decoding a symbol, it is retransmitted by the destination in the next time slot. Nodes at the border will, therefore, transmit twice in each time slot. Nodes in the interior of the network transmit only once. The symbol that they transmit is a linear combination of one symbol from each of the sessions for which the shortest path between source and destination includes that node.

The operation of the network code is demonstrated in Figure 5 in which the transmissions of all nodes in the first four time slots are depicted. Different transmissions by the same node are separated by a comma. Note, moreover, that there is a startup phase, time slots 0 to 2, in which not all destinations are able to decode a symbol. From time slot 3 onwards, all destinations decode one symbol in every time slot. In analyzing the energy consumption of the coding scheme, we will ignore the startup phase.

The symbol transmitted at $t = 3$ by the node with the dotted border can be obtained by summing all transmissions from nodes with a dashed border in earlier time slots. Indeed

$$\begin{aligned} &m_1^1(3) + m_1^2(2) + m_0^1(1) + m_0^3(2) + m_2^2(1) \\ &\quad + m_1^1(3) + m_2^1(2) + m_0^3(2) + m_0^1(1) + m_1^2(2) \\ &\quad + m_1^3(1) = m_2^1(2) + m_2^2(1) + m_1^3(1). \end{aligned} \quad (12)$$

This coding operation (i.e., in time slot t, a node transmits the sum of what was transmitted by its top-left neighbour in time slot $t-2$, by its top right-neighbour in time slot $t-1$, and so forth, as visualized in Figure 5) is performed by all nodes that are in the interior of the network. The idea behind the coding operation is to cancel, by means of the XOR operation, all symbols that should not be retransmitted. In (12), for instance, we have $m_1^1(3) + m_1^1(3) = 0$. The exact operation of the network code is made more precise in the remainder of this subsection. The coding operation for interior nodes is given in exact form in (17).

Nodes at the border of the network operate as follows. Let $0 < u_2 < K$. In time slot t node $(0, u_2)$ transmitstwo symbols

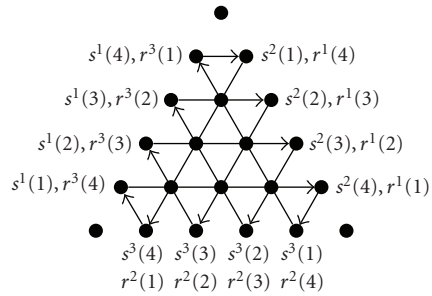


Figure 4: The unicast sessions on the network from Figure 3.

$x_t^1(0,u_2)$ and $x_t^3(0,u_2)$, where

Left border:

$$\boxed{\begin{aligned} x_t^1(0,u_2) &= m_t^1(u_2), \\ x_t^3(0,u_2) &= m_{t-u_2}^3(K-u_2). \end{aligned}} \tag{13}$$

Since $(0,u_2)$ is the source of session $m^1(u_2)$ it has source symbol $m_t^1(u_2)$, available. Also, $(0,u_2)$ is the destination for session $m^3(K-u_2)$. It remains to be shown that symbol $m_{t-u_2}^3(K-u_2)$ can be decoded by $(0,u_2)$ using the information obtained from its neighbours up to time slot t. For notational convenience, let

Left border:

$$\boxed{x_t(0,u_2) \triangleq x_t^1(0,u_2) + x_t^3(0,u_2).} \tag{14}$$

In a similar fashion, we have the following transmissions at the right and bottom borders of the network.

Right border:

$$\boxed{\begin{aligned} x_t^1(v_1,v_2) &= m_{t-v_1}^1(v_2), \\ x_t^2(v_1,v_2) &= m_t^2(v_1), \\ x_t(v_1,v_2) &\triangleq x_t^1(v_1,v_2) + x_t^2(v_1,v_2), \end{aligned}} \tag{15}$$

Bottom border:

$$\boxed{\begin{aligned} x_t^2(u_1,0) &= m_{t-K+u_1}^2(u_1), \\ x_t^3(u_1,0) &= m_t^3(K-u_1), \\ x_t(u_1,0) &\triangleq x_t^2(u_1,0) + x_t^3(u_1,0), \end{aligned}} \tag{16}$$

where $u_1, v_1, v_2 > 0$, $u_1, v_1, v_2 < K$, and $v_1 + v_2 = K$. Moreover, $x_t(v_1,v_2)$ and $x_t(u_1,0)$ are not symbols that are transmitted, but only notational shortcuts.

Nodes in the interior of the network transmit once in each time slot. Let $(u_1,u_2) \in \overset{\circ}{V}$. The coding operation it performs is given by

$$\boxed{\begin{aligned} x_t(u_1,u_2) &= x_{t-1}(u_1-1,u_2) + x_{t-2}(u_1-1,u_2+1) \\ &\quad + x_{t-1}(u_1,u_2+1) + x_{t-3}(u_1,u_2) \\ &\quad + x_{t-2}(u_1+1,u_2) + x_{t-2}(u_1,u_2-1) \\ &\quad + x_{t-1}(u_1+1,u_2-1). \end{aligned}} \tag{17}$$

4.3. Validity of the Network Code. We need to show that destinations can decode in time in order to retransmit the required symbols according to (13), (15), and (16). In order to do so we first analyze how data propagates through the network. If we look at the nodes in the network that transmit linear combinations that contain a certain source symbol, we see that symbols propagate exactly along the shortest paths between source and destination. This is made more precise in the following two lemmas.

Lemma 1. *Let $0 < u_2 < K$. Assume that the only nonzero source symbol transmitted in the network is $m_0^1(u_2)$ by node $(0,u_2)$ in time slot 0. Then, for all $t \geq 0$ and $(v_1,v_2) \in \overset{\circ}{V}$*

$$x_t(v_1,v_2) = \begin{cases} m_0^1(u_2) & \text{if } v_1 = t, v_2 = u_2, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Proof. We use induction over time. The base case is time slot $t = 0$, for which it is readily verified that the statement is true. Now, for the induction step, suppose that the lemma holds for all t' smaller than t. This implies that for all $\tau > 0$ and $(v_1,v_2) \in \overset{\circ}{V}$,

$$x_{t-\tau}(v_1,v_2) = x_{t-\tau-1}(v_1-1,v_2). \tag{19}$$

Hence,

$$\begin{aligned} x_t(v_1,v_2) &= x_{t-1}(v_1-1,v_2) + x_{t-2}(v_1-1,v_2+1) \\ &\quad + x_{t-1}(v_1,v_2+1) + x_{t-3}(v_1,v_2) + x_{t-2}(v_1+1,v_2) \\ &\quad + x_{t-2}(v_1,v_2-1) + x_{t-1}(v_1+1,v_2-1) \\ &= x_{t-1}(v_1-1,v_2) + x_{t-2}(v_1-1,v_2+1) \\ &\quad + x_{t-2}(v_1-1,v_2+1) + x_{t-3}(v_1,v_2) + x_{t-3}(v_1,v_2) \\ &\quad + x_{t-2}(v_1,v_2-1) + x_{t-2}(v_1,v_2-1) \\ &= x_{t-1}(v_1-1,v_2), \end{aligned} \tag{20}$$

which by the induction hypothesis is equal to $m_0^1(u_2)$ if $v_1 = t$ and $v_2 = u_2$ and zero otherwise. □

Lemma 2. *Let $(u_1,u_2) \in \overset{\circ}{V}$.*

$$\begin{aligned} x_t(u_1,u_2) &= m_{t-u_1}^1(u_2) + m_{t-K+u_1+u_2}^2(u_1) \\ &\quad + m_{t-u_2}^3(K-u_1-u_2). \end{aligned} \tag{21}$$

Proof. From Lemma 1, the time-invariance of the system, and the symmetry of the coding operation (17) of the internal nodes. □

We are now ready to prove that the destinations can correctly decode source symbols. We present the decoding procedure for nodes on the right border of the network. The decoding procedures at the other borders can be obtained by exploiting the symmetry of the system.

Lemma 3. *Consider* (u_1, u_2)*, with* $u_1 + u_2 = K$*,* $0 < u_2 < K$*, that is, the destination of session* $m^1(u_2)$*. It can decode symbol* $m^1_{t-u_1}(u_2)$ *at the end of time slot* $t-1$ *as*

$$x^2_{t-2}(u_1 - 1, u_2 + 1) + x_{t-1}(u_1 - 1, u_2) + x^2_{t-3}(u_1, u_2) + x_{t-2}(u_1, u_2 - 1) + x^1_{t-1}(u_1 + 1, u_2 - 1). \tag{22}$$

Proof. From Lemma 2, (15), it follows that (22) equals

$$m^1_{t-u_1}(u_2) + m^1_{t-u_1-2}(u_2 - 1) + m^1_{t-u_1-2}(u_2 - 1) + m^2_{t-2}(u_1 - 1) + m^2_{t-2}(u_1 - 1) + m^2_{t-3}(u_1) + m^2_{t-3}(u_1) m^3_{t-u_2-1}(1) + m^3_{t-u_2-1}(1) = m^1_{t-u_1}(u_2). \tag{23}$$

□

4.4. Energy Consumption. The energy consumption of the network coding scheme presented above is given in the following lemma.

Lemma 4. $\min_r \mathcal{E}_{\text{coding}}(V, M, r) \le \mathcal{E}_{\text{coding}}(V, M, 1) \le (c/2)K^2 + O(K)$.

Proof. From (13)–(17), we have that each of the $3(K-1)$ nodes at the border that are source or destination are transmitting twice in each time slot. Each of the $(K-1)(K-2)/2$ internal nodes is transmitting once in each time slot. Since $r = 1$, the energy consumption per transmission is c. This gives

$$\mathcal{E}_{\text{coding}}(V, M, 1) \le 6c(K-1) + \frac{c(K-1)(K-2)}{2} = \frac{c}{2}K^2 + \text{O}(K). \tag{24}$$

□

Next, we give the minimum energy required by a routing solution.

Lemma 5. $\min_r \mathcal{E}_{\text{routing}}(V, M, r) = \mathcal{E}_{\text{routing}}(V, M, 1) = (3c/2)K^2 + O(K)$.

Proof. Since we consider routing, we need to take the shortest path for each session. Since the energy consumption per hop equals cr^α, the energy consumption under routing is minimized for $r = 1$. Now, we see that the number of transmissions required to deliver a symbol for the sessions $m^1(1), \ldots, m^1(K-1)$ equals $K(K-1)/2$. Adding the transmissions for sessions of type 2 and 3 gives

$$\mathcal{E}_{\text{routing}}(V, M, 1) = \frac{3c}{2}K(K-1) = \frac{3c}{2}K^2 + \text{O}(K). \tag{25}$$

□

Using the above two lemmas, we are able to prove Theorem 2.

Proof of Theorem 2. Remember that B_{var} is defined as the maximum of $\min_r \mathcal{E}_{\text{routing}}(V, M, r)/\min_r \mathcal{E}_{\text{coding}}(V, M, r)$ over V and M. Hence, $\min_r \mathcal{E}_{\text{routing}}(V, M, r)/\min_r \mathcal{E}_{\text{coding}}(V, M, r)$ for any specific V and M will provide a lower bound to B_{var}. In addition, any upper bound to $\min_r \mathcal{E}_{\text{coding}}(V, M, r)$ will result in a lower bound to B_{var}. Hence, from Lemmas 4 and 5, we have

$$\begin{aligned} B_{\text{var}}(2) &\ge \lim_{K\to\infty} \frac{\min_r \mathcal{E}_{\text{routing}}(V, M, r)}{\min_r \mathcal{E}_{\text{coding}}(V, M, r)} \\ &\ge \lim_{K\to\infty} \frac{\mathcal{E}_{\text{routing}}(V, M, 1)}{\mathcal{E}_{\text{coding}}(V, M, 1)} \\ &\ge \lim_{K\to\infty} \frac{(3c/2)K^2 + \text{O}(K)}{(c/2)K^2 + \text{O}(K)} = 3. \end{aligned} \tag{26}$$

□

5. An Efficient Code on the d-Dimensional Rectangular Lattice

In this section, we present a multiple unicast configuration in which the nodes are placed at integer coordinates in a d-dimensional space, that is, at the rectangular lattice.

5.1. Configuration. The size of the configuration is parameterized by a positive integer K. We have

$$V = \{(v_1, \ldots, v_d) \mid 0 \le v_i \le K, i = 1, \ldots, d\}. \tag{27}$$

The interior of the network is given by

$$\overset{\circ}{V} = \{v \in V \mid 0 < v_i < K, i = 1, \ldots, d\}. \tag{28}$$

We will make use of

$$\overline{V} = \{v \in V \mid \exists \text{ unique } i : v_i \in \{0, K\}\}, \tag{29}$$

which corresponds to those nodes that are part of exactly one face of the network.

The transmission range that will be used is $r = \sqrt{d}$. This transmission range induces a neighbourhood consisting of all neighbours within distance $\sqrt{d}$. The coding operation of our network code is based on only part of the neighbourhood, that is, it uses

$$N_v = \{u \in V \mid |u_i - v_i| \le 1 \ \forall i, u \ne v\}. \tag{30}$$

Note, that for $d \le 3$, N_v corresponds to the complete neighbourhood of v. We will be using $\text{dist}(u, v) \triangleq \|u - v\|_1 = \sum_{i=1}^d |u_i - v_i|$, that is, $\text{dist}(u, v)$ denotes the Manhattan

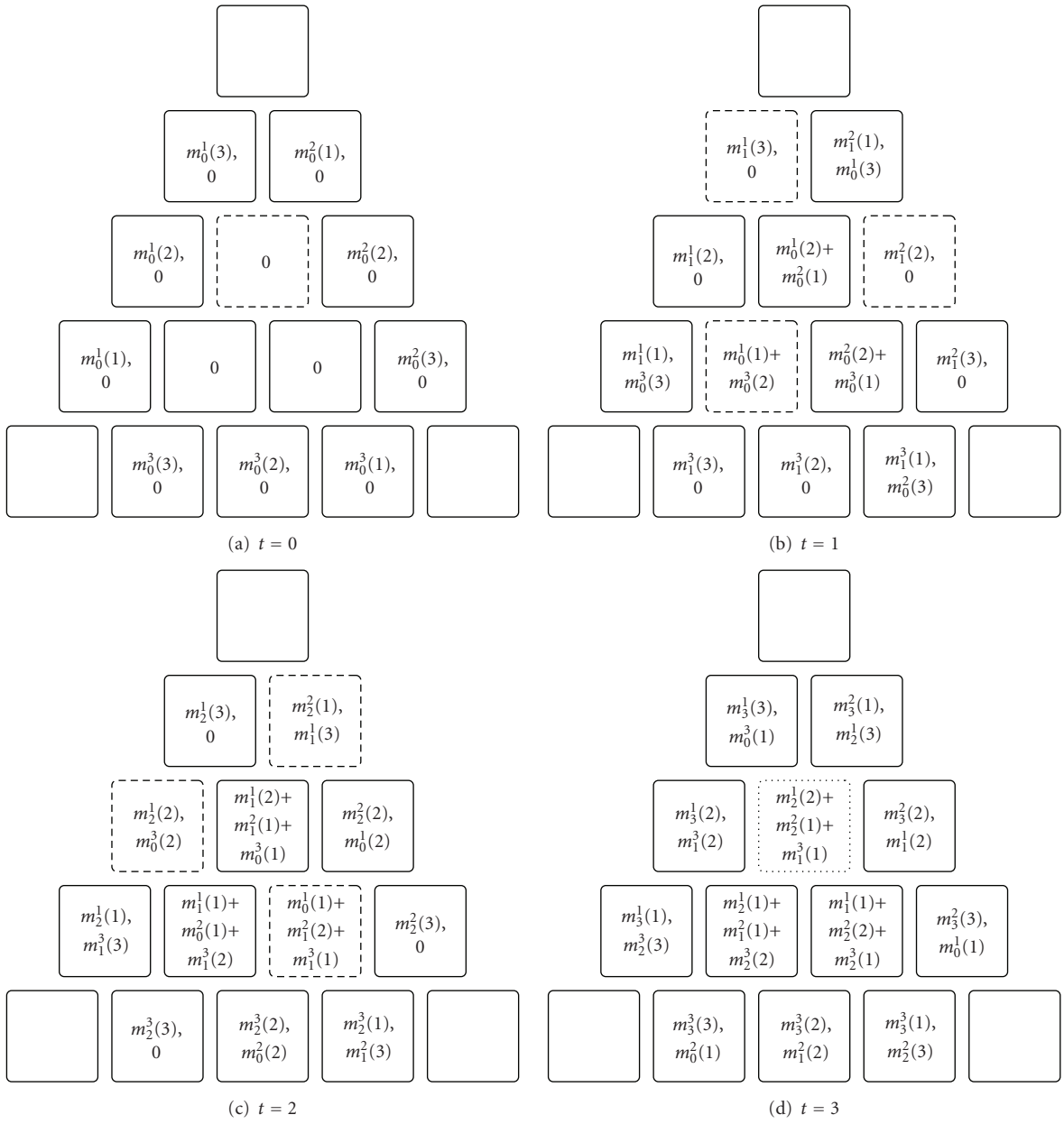


Figure 5: Example operation of the network code of Section 4, with $K = 4$. The transmissions of all nodes in the time slots $0, \ldots, 3$ are depicted. Different transmissions by the same node are separated by a comma. Note, that the symbol transmitted at $t = 3$ by the node with dotted border can be obtained by summing all transmissions from nodes with a dashed border in earlier time slots. All nodes in the interior of the network perform this simple coding operation.

distance from u to v. The network and its connectivity are depicted for $d = 2$ in Figure 6.

A source is located at each $v \in \overline{V}$. Therefore, there are $|\overline{V}| = 2d(K-1)^{d-1}$ sessions. If $v_i = 0$, we denote the session corresponding to this source by $m^i(v^{\setminus i})$. Recall from Section 2 that $v^{\setminus i}$ denotes the d-1 dimensional vector obtained by removing the ith element from v. If $v_i = K$, we denote the session by $m^{d+i}(v^{\setminus i})$. The destination of each session is located at the other side of the network, that is, we have $r^i(v^{\setminus i}) = s^{d+i}(v^{\setminus i})$ and $r^{d+i}(v^{\setminus i}) = s^i(v^{\setminus i})$. The positions of sources and destinations are depicted for $d = 2$ in Figure 7. It can be seen that $m^i(v^{\setminus i})$ and $m^{d+i}(v^{\setminus i})$ form oppositely directed sessions.

5.2. Network Code. We introduce sets $\Theta_\delta \subset \{1, \ldots, 2d\}$, $0 \leq \delta \leq d$, which are defined recursively as follows:

$$\begin{aligned} \Theta_d &= \{d\}, \\ \Theta_\delta &= (\Theta_{\delta+1} - 1)\Delta(\Theta_{\delta+1} + 1), \quad 0 < \delta < d, \\ \Theta_0 &= (\Theta_1 - 1)\Delta(\Theta_1 + 1) \setminus \{0\}, \end{aligned} \tag{31}$$

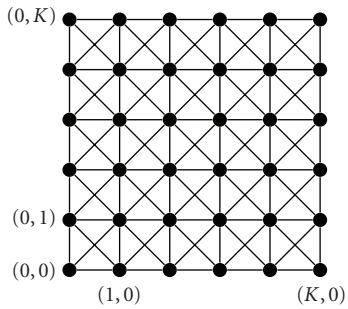


Figure 6: Nodes at a subset of the d-dimensional rectangular lattice, $d = 2$ depicted in the figure, with the connectivity induced by a transmission range $r = \sqrt{d}$. The size of the network is controlled by K, with $K = 5$ in this figure.

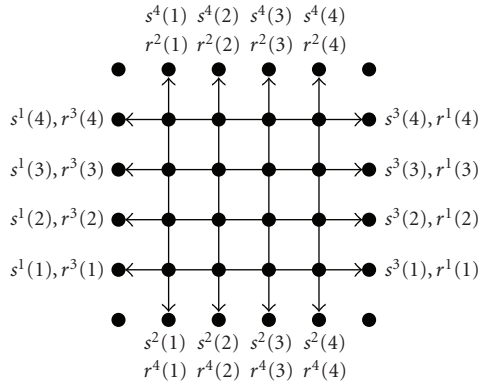


Figure 7: The unicast sessions on the network from Figure 6.

where Δ denotes symmetric difference and $\Theta_\delta \pm 1 = \{\tau \pm 1 \mid \tau \in \Theta_\delta\}$. Note that irrespective of d we have $1 \in \Theta_1$. As an example for $d = 2$ we have $\Theta_2 = \{2\}$, $\Theta_1 = \{1, 3\}$ and $\Theta_0 = \{4\}$.

The scheme is very similar in flavour to the scheme presented in Section 4; its operation is demonstrated in Figure 8 in which, for $d = 2$ and $K = 3$, the transmissions of all nodes in the first four time slots are depicted. The operation of the scheme is such that in time slot t sources transmit the tth source symbol and destinations decode the $(t - K)$th source symbol. Besides transmitting a new source symbol in each time slot, sources/destinations will also retransmit the symbol that has been decoded in that time slot, that is, they transmit two different symbols in each time slot. In the figure, different transmissions by the same node are separated by a comma. Nodes in the interior of the network transmit only once. The symbol that they transmit is a linear combination of one symbol from each of the sessions for which the shortest path between source and destination includes that node. The symbol transmitted at $t = 3$ by the node with the dotted border can be obtained by summing all transmissions from nodes with a dashed border in earlier time slots. This coding operation is performed by all nodes that are in the interior of the network. The exact operation of the network code is made more precise in the remainder of this subsection. The coding operation for interior nodes is given in exact form in (34).

Let node $v \in \overline{V}$. Remember that $v \in \overline{V}$ implies that there exists a unique i such that $v_i \in \{0, K\}$. Node v transmits

$$\boxed{\begin{aligned} x_t^i(v) &= m_{t-v_i}^i\left(v^{\setminus i}\right), \\ x_t^{d+i}(v) &= m_{t-K+v_i}^{d+i}\left(v^{\setminus i}\right). \end{aligned}} \tag{32}$$

For notational convenience, let

$$\boxed{x_t(v) \triangleq x_t^i(v) + x_t^{d+i}(v).} \tag{33}$$

The coding operation performed by an internal node is as follows:

$$\boxed{x_t(v) = \sum_{u \in N_v \cup \{v\}} \sum_{\tau \in \Theta_{\mathrm{dist}(u,v)}} x_{t-\tau}(u).} \tag{34}$$

5.3. Validity of the Network Code. The following result follows directly from the definition of the sets Θ_δ, but is stated here as a lemma because of its importance in the remainder of the paper.

Lemma 6. *Let $\{x_t\}$ be a sequence of symbols from $\mathbb{F}_2$ and let $0 < \delta < d$. We have*

$$\begin{aligned} \sum_{\tau \in \Theta_\delta} x_{t-\tau} &= \sum_{\tau \in \Theta_{\delta+1}} [x_{t-\tau+1} + x_{t-\tau-1}], \\ \sum_{\tau \in \Theta_0} x_{t-\tau} &= \sum_{\tau \in \Theta_1 \setminus \{1\}} x_{t-\tau+1} + \sum_{\tau \in \Theta_1} x_{t-\tau-1}. \end{aligned} \tag{35}$$

Lemma 7. *Consider node $(0, u_2^d) \in \overline{V}$. Assume that the only nonzero source symbol transmitted in the network is $m_0^1(u_2^d)$ by node $(0, u_2^d)$ in time slot 0. Then*

$$x_t(v) = \begin{cases} m_0^1\left(u_2^d\right), & \text{if } v_1 = t, v_2^d = u_2^d, \\ 0, & \text{otherwise}, \end{cases} \tag{36}$$

for all $v \in V$ and $t \geq 0$.

Proof. We use induction over t. At time $t = 0$, the lemma holds, giving us our base case. Now suppose that the lemma holds for all time slots smaller than t. If $v \in \overline{V}$, the lemma follows directly from (32)–(33). In the remainder we consider $u \in \mathring{V}$. From the induction hypothesis, it follows that for any $t' < t$

$$x_{t'}(u) = x_{t'-1}\left(u_1 - 1, u_2^d\right). \tag{37}$$

If $u_1 = K - 1$, it follows from (32) and the induction hypothesis that

$$x_{t'-1}(u) = x_{t'}\left(u_1 + 1, u_2^d\right). \tag{38}$$

Now, at t the coding operation performed by u can be decomposed as

$$x_t(u) = \sum_{w \in N_u \cup \{v\}} \sum_{\tau \in \Theta_{\mathrm{dist}(w,u)}} x_{t-\tau}(w) = \sum_{\substack{w \in N_u: \\ w_1 = u_1}} g(w), \tag{39}$$

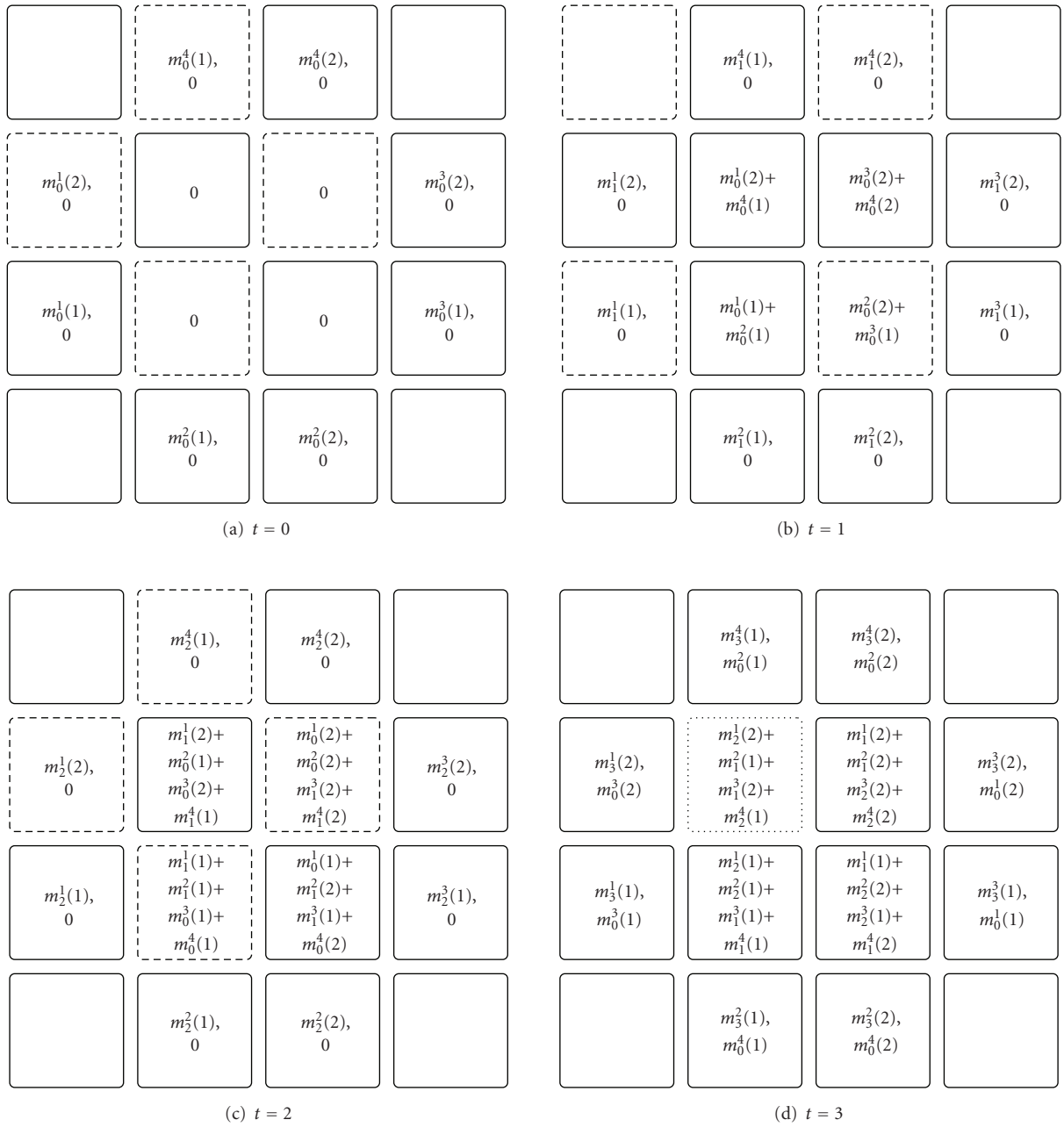


FIGURE 8: Example operation of the network code of Section 5, with $K = 3$. The transmissions of all nodes in the time slots $0, \ldots, 3$ are depicted. Different transmissions by the same node are separated by a comma. Note, that the symbol transmitted at $t = 3$ by the node with dotted border can be obtained by summing all transmissions from nodes with a dashed border in earlier time slots. All nodes in the interior of the network perform this simple coding operation.

where

$$g(w) = \sum_{\tau \in \Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1 - 1, w_2^d\right) + \sum_{\tau \in \Theta_{\mathrm{dist}(w,u)}} x_{t-\tau}(w) + \sum_{\tau \in \Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1 + 1, w_2^d\right). \tag{40}$$

In the remainder, we show that

$$g(w) = \begin{cases} x_{t-1}\left(w_1 - 1, w_2^d\right) & \text{if } w = u \\ 0, & \text{otherwise,} \end{cases} \tag{41}$$

which proves the lemma, since by the induction hypothesis $x_{t-1}(u_1 - 1, u_2^d) = m_0^1(u_2^d)$ if $u_1 = t$ and zero otherwise.

For $w \neq u$ we, have

$$\begin{aligned} g(w) &= \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1-1, w_2^d\right) + \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)}} x_{t-\tau}(w) \\ &\quad + \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1+1, w_2^d\right) \\ &= \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1-1, w_2^d\right) + \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau+1}(w) \\ &\quad + \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau-1}(w) + \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1+1, w_2^d\right) \\ &= \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1-1, w_2^d\right) + \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1-1, w_2^d\right) \\ &\quad + \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1+1, w_2^d\right) \sum_{\tau\in\Theta_{\mathrm{dist}(w,u)+1}} x_{t-\tau}\left(w_1+1, w_2^d\right) \\ &= 0, \end{aligned} \tag{42}$$

where the second equality follows from Lemma 6, the third equality follows from (37)-(38), and the last equality holds because we work over $\mathbb{F}_2$.

For $w = u$, we have

$$\begin{aligned} g(u) &= \sum_{\tau\in\Theta_1} x_{t-\tau}\left(u_1-1, u_2^d\right) + \sum_{\tau\in\Theta_0} x_{t-\tau}(u) \\ &\quad + \sum_{\tau\in\Theta_1} x_{t-\tau}\left(u_1+1, u_2^d\right) \\ &= \sum_{\tau\in\Theta_1} x_{t-\tau}\left(u_1-1, u_2^d\right) + \sum_{\tau\in\Theta_1\setminus\{1\}} x_{t-\tau+1}(u) \\ &\quad + \sum_{\tau\in\Theta_1} x_{t-\tau-1}(u) + \sum_{\tau\in\Theta_1} x_{t-\tau}\left(w_1+1, w_2^d\right) \\ &= \sum_{\tau\in\Theta_1} x_{t-\tau}\left(u_1-1, u_2^d\right) + \sum_{\tau\in\Theta_1\setminus\{1\}} x_{t-\tau}\left(u_1-1, u_2^d\right) \\ &\quad + \sum_{\tau\in\Theta_1} x_{t-\tau}\left(u_1+1, u_2^d\right) + \sum_{\tau\in\Theta_1} x_{t-\tau}\left(u_1+1, u_2^d\right) \\ &= x_{t-1}\left(u_1-1, u_2^d\right). \end{aligned} \tag{43}$$

□

Lemma 8. *Let* $u \in \overset{\circ}{V}$

$$x_t(u) = \sum_{i=1}^{d}\left[m_{t-u_i}^i\left(u^{\setminus i}\right) + m_{t-K+u_i}^{d+i}\left(u^{\setminus i}\right)\right]. \tag{44}$$

Proof. By linearity, time-invariance and symmetry of (34) together with Lemma 7. □

We are now ready to prove that the destinations can correctly decode source symbols. We present the decoding procedure for nodes on the right border of the network, that is, for nodes of type $(K, u_2^d) \in \overline{V}$. The decoding procedures at the other borders can be obtained by exploiting the symmetry of the system.

Lemma 9. *Consider node* $u = (K, u_2^d) \in \overline{V}$. *At the end of time slot* $t-1$, *it can decode symbol* $m_{t-K}^1(u_2^d)$ *as*

$$\begin{aligned} &\sum_{\substack{v\in N_u:\\ v_1<K}} \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)}} x_{t-\tau}(v) \\ &\quad + \sum_{\substack{v\in N_u:\\ v_1=K}} \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} \left[x_{t-\tau+1}^1(v) + x_{t-\tau-1}^{d+1}(v)\right] \\ &\quad + \sum_{\tau\in\Theta_1\setminus\{1\}} x_{t-\tau+1}^1(u) + \sum_{\tau\in\Theta_1} x_{t-\tau-1}^{d+1}(u) \end{aligned} \tag{45}$$

Proof. First note that all terms in (45) correspond to symbols that have been received by (K, u_2^d) before or in time slot $t-1$.

Now, from Lemma 8, we have

$$\begin{aligned} &\sum_{\substack{v\in N_u:\\ v_1<K}} \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)}} x_{t-\tau}(v) \\ &= \sum_{\substack{v\in N_u:\\ v_1<K}} \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)}} \sum_{i=1}^{d}\left[m_{t-v_i-\tau}^i\left(v^{\setminus i}\right) + m_{t-K+v_i-\tau}^{d+i}\left(v^{\setminus i}\right)\right] \\ &= \sum_{\substack{v\in N_u:\\ v_1<K}} \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)}} \left[m_{t-v_1-\tau}^1\left(v^{\setminus 1}\right) + m_{t-K+v_1-\tau}^{d+1}\left(v^{\setminus 1}\right)\right] \\ &\quad + \sum_{i=2}^{d}\Bigg[\sum_{\substack{v\in N_u:\\ v_1<K, v_i=u_i}} \Bigg[\sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} m_{t-v_i+1-\tau}^i\left(v^{\setminus i}\right) \\ &\qquad + \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)}} m_{t-v_i-\tau}^i\left(v^{\setminus i}\right) \\ &\qquad + \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} m_{t-v_i-1-\tau}^i\left(v^{\setminus i}\right) \\ &\qquad + \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} m_{t-v_i+1-\tau}^{d+i}\left(v^{\setminus i}\right) \\ &\qquad + \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)}} m_{t-v_i-\tau}^{d+i}\left(v^{\setminus i}\right) \\ &\qquad + \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} m_{t-v_i-1-\tau}^{d+i}\left(v^{\setminus i}\right)\Bigg]\Bigg] \\ &\overset{(a)}{=} \sum_{\substack{v\in N_u:\\ v_1<K}} \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)}} \left[m_{t-v_1-\tau}^1\left(v^{\setminus 1}\right) + m_{t-K+v_1-\tau}^{d+1}\left(v^{\setminus 1}\right)\right] \\ &= \sum_{\tau\in\Theta_1}\left[m_{t-K+1-\tau}^1\left(u^{\setminus 1}\right) + m_{t-1-\tau}^{d+1}\left(u^{\setminus 1}\right)\right] \\ &\quad + \sum_{\substack{v\in N_u:\\ v_1=K}} \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} \left[m_{t-K+1-\tau}^1\left(v^{\setminus 1}\right) + m_{t-1-\tau}^{d+1}\left(v^{\setminus 1}\right)\right], \end{aligned} \tag{46}$$

where (a) holds, because for $\mathrm{dist}(u,v) > 0$, Lemma 6 gives

$$\sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} m^i_{t-\nu_i+1-\tau}\left(v^{\setminus i}\right) + \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)}} m^i_{t-\nu_i-\tau}\left(v^{\setminus i}\right) + \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} m^i_{t-\nu_i-1-\tau}\left(v^{\setminus i}\right) = 0, \tag{47}$$

and

$$\sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} m^{d+i}_{t-\nu_i+1-\tau}\left(v^{\setminus i}\right) + \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)}} m^{d+i}_{t-\nu_i-\tau}\left(v^{\setminus i}\right) + \sum_{\tau\in\Theta_{\mathrm{dist}(u,v)+1}} m^{d+i}_{t-\nu_i-1-\tau}\left(v^{\setminus i}\right) = 0. \tag{48}$$

From (32) it follows that

$$\sum_{\substack{v\in N_u:\,\tau\in\Theta_{\mathrm{dist}(u,v)+1}\\ \nu_1=K}} \left[x^1_{t-\tau+1}(v) + x^{d+1}_{t-\tau-1}(v)\right] = \sum_{\substack{v\in N_u:\,\tau\in\Theta_{\mathrm{dist}(u,v)+1}\\ \nu_1=K}} \left[m^1_{t-K+1-\tau}\left(v^{\setminus 1}\right) + m^{d+1}_{t-1-\tau}\left(v^{\setminus 1}\right)\right], \tag{49}$$

and

$$\sum_{\tau\in\Theta_1\setminus\{1\}} x^1_{t-\tau+1}(u) + \sum_{\tau\in\Theta_1} x^{d+1}_{t-\tau-1}(u) = \sum_{\tau\in\Theta_1\setminus\{1\}} m^1_{t-K+1-\tau}\left(u^{\setminus 1}\right) + \sum_{\tau\in\Theta_1} m^{d+1}_{t-1-\tau}\left(u^{\setminus 1}\right). \tag{50}$$

The proof of the lemma follows by adding the final expressions from (46), (49) and (50) observing that the outcome is $m^1_{t-K}(u_2^d)$. □

5.4. Energy Consumption. The energy consumption of the network coding scheme presented above provides an upper bound to $\min_r \mathcal{E}_{\mathrm{coding}}(V,M,r)$.

Lemma 10. $\mathcal{E}_{\mathrm{coding}}(V,M,\sqrt{d}) \le 4cd^{1+\alpha/2}(K-1)^{d-1} + cd^{\alpha/2}(K-1)^d$.

Proof. All transmissions are over distance $\sqrt{d}$ and cost $cd^{\alpha/2}$. The nodes in $\overline{V}$ are transmitting twice. On each of the $2d$ sides of the network, there are $(K-1)^{d-1}$ nodes from $\overline{V}$; hence $|\overline{V}| = 2d(K-1)^{d-1}$. This gives $2|\overline{V}| = 4d(K-1)^{d-1}$ transmissions. In addition, there are $(K-1)^d$ nodes in the interior, that are all transmitting once. □

Next, we give the minimum energy required by a routing solution.

Lemma 11. $\mathcal{E}_{\mathrm{routing}}(V,M,\sqrt{d}) = 2cd^{1+\alpha/2}\lceil K/\lfloor\sqrt{d}\rfloor\rceil(K-1)^{d-1}$.

Proof. Since the transmission range is equal to $\sqrt{d}$, a routing solution requires $\lceil K/\lfloor\sqrt{d}\rfloor\rceil$ transmissions per session. Moreover, there are $|\overline{V}| = 2d(K-1)^{d-1}$ sessions. □

Using the above two lemmas, we are able to prove Theorem 1.

Proof of Theorem 1. Lemmas 10 and 11 give

$$B_{\mathrm{fixed}}(d) \ge \lim_{K\to\infty} \frac{\mathcal{E}_{\mathrm{routing}}\left(V,M,\sqrt{d}\right)}{\mathcal{E}_{\mathrm{coding}}\left(V,M,\sqrt{d}\right)} \ge \lim_{K\to\infty} \frac{2cd^{1+\alpha/2}\left\lceil K/\left\lfloor\sqrt{d}\right\rfloor\right\rceil(K-1)^{d-1}}{cd^{\alpha/2}\left\lceil 4d(K-1)^{d-1} + (K-1)^d\right\rceil} \tag{51}$$

$$= \frac{2d}{\left\lfloor\sqrt{d}\right\rfloor}. \tag{52}$$

□

6. Discussion

We have given several constructions of energy-efficient network codes. These constructions serve to show that compared to plain routing, network coding has the potential of reducing energy consumption in wireless networks. Since we have provided only codes that are based on a centralized design, it remains to be shown in future work if and how this potential can be exploited using practical codes. Moreover, it would also be of interest to consider the energy-benefit in topologies in which the nodes are not positioned at a lattice, for instance, random networks.

In this work we have provided lower bounds on the energy benefit of network coding for wireless multiple unicast. Another open problem is to find upper bounds on the benefit.

References

[1] S. Chen and K. Nahrstedt, "An overview of quality of service routing for next-generation high-speed networks: problems and solutions," *IEEE Network*, vol. 12, no. 6, pp. 64–79, 1998.

[2] L. Tassiulas and J. Chang, "Energy conserving routing in wireless ad hoc networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Network (MobiCom '98)*, Dallas, Tex, USA, August 1998.

[3] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.

[4] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proceedings of the IEEE 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, pp. 404–413, March 2000.

[5] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol. 7, no. 4, pp. 343–358, 2001.

[6] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 291–303, 2002.

[7] A. J. Goldsmith and S. B. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 8–27, 2002.

[8] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[9] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[10] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[11] T. Ho, M. Médard, R. Koetter et al., "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[12] R. W. Yeung and N. Cai, "Network coding theory," *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 4, pp. 241–381, 2006.

[13] C. Fragouli and E. Soljanin, "Network coding fundamentals," *Foundations and Trends in Networking*, vol. 2, no. 1, pp. 1–133, 2007.

[14] C. Fragouli and E. Soljanin, "Network coding applications," *Foundations and Trends in Networking*, vol. 2, no. 2, pp. 135–269, 2007.

[15] D. S. Lun, N. Ratnakar, M. Médard et al., "Minimum-cost multicast over coded packet networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2608–2623, 2006.

[16] P. Winter, "Steiner problem in networks: a survey," *Networks*, vol. 17, no. 2, pp. 129–167, 1987.

[17] A. Goel and S. Khanna, "On the network coding advantage for wireless multicast in Euclidean space," in *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN '08)*, pp. 64–69, April 2008.

[18] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 450–463, 2008.

[19] J. Widmer and J.-Y. Le Boudec, "Network coding for efficient communication in extreme networks," in *Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp. 284–291, ACM Press, New York, NY, USA, 2005.

[20] Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Proceedings of the 39th Annual Conference on Information Sciences and Systems (CISS '05)*, 2005.

[21] M. Effros, T. Ho, and S. Kim, "A tiling approach to network code design for wireless networks," in *Proceedings of the IEEE Information Theory Workshop (ITW '06)*, pp. 62–66, IEEE, punta del Este, Uruguay, March 2006.

[22] S. Kim, M. Effros, and T. Ho, "On low-power multiple unicast network coding over a wireless triangular grid," in *Proceedings of the 45th Annual Allerton Conference on Communication, Control and Computing*, 2007.

[23] G. Kramer and S. A. Savari, "Edge-cut bounds on network coding rates," *Journal of Network and Systems Management*, vol. 14, no. 1, pp. 49–66, 2006.

[24] A. Keshavarz-Haddad and R. Riedi, "Bounds on the benefit of network coding: throughput and energy saving in wireless networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 376–384, April 2008.

[25] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (ACM SIGCOMM '06)*, pp. 243–254.

[26] J. Liu, D. Goeckelt, and D. Towsley, "Bounds on the gain of network coding and broadcasting in wireless networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 6–12, May 2007.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 870268, 12 pages
doi:10.1155/2010/870268

Research Article

Applying Physical-Layer Network Coding in Wireless Networks

Shengli Zhang[1,2] and Soung Chang Liew[2]

[1] *Department of Communication Engineering, Shenzhen University, Shenzhen, China*
[2] *Department Information Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong*

Correspondence should be addressed to Shengli Zhang, slzhang@ie.cuhk.edu.hk

Received 26 September 2009; Revised 31 December 2009; Accepted 7 February 2010

Academic Editor: Zhi-Hong Mao

A main distinguishing feature of a wireless network compared with a wired network is its broadcast nature, in which the signal transmitted by a node may reach several other nodes, and a node may receive signals from several other nodes, simultaneously. Rather than a blessing, this feature is treated more as an interference-inducing nuisance in most wireless networks today (e.g., IEEE 802.11). This paper shows that the concept of network coding can be applied at the physical layer to turn the broadcast property into a capacity-boosting advantage in wireless ad hoc networks. Specifically, we propose a physical-layer network coding (PNC) scheme to coordinate transmissions among nodes. In contrast to "straightforward" network coding which performs coding arithmetic on digital bit streams after they have been received, PNC makes use of the additive nature of simultaneously arriving electromagnetic (EM) waves for equivalent coding operation. And in doing so, PNC can potentially achieve 100% and 50% throughput increases compared with traditional transmission and straightforward network coding, respectively, in 1D regular linear networks with multiple random flows. The throughput improvements are even larger in 2D regular networks: 200% and 100%, respectively.

1. Introduction

One of the biggest challenges in wireless communication is how to deal with the interference at the receiver when signals from multiple sources arrive simultaneously. In the radio channel of the physical-layer of wireless networks, data are transmitted through electromagnetic (EM) waves in a broadcast manner. The interference between these EM waves causes the data to be scrambled.

To overcome its negative impact, most schemes attempt to find ways to either reduce or avoid interference through receiver design or transmission scheduling [1]. For example, in 802.11 networks, the carrier-sensing mechanism allows at most one source to transmit or receive at any time within a carrier-sensing range. This is obviously inefficient when multiple nodes have data to transmit.

While interference causes throughput degradation on wireless networks in general, its negative effect for multihop ad hoc networks is particularly significant. For example, in 802.11 networks, the theoretical throughput of a multihop flow in a linear network is less than 1/4 of the single-hop case due to the "self-interference" effect, in which packets of the same flow but at different hops collide with each other [2, 3].

Instead of treating interference as a nuisance to be avoided, we can actually embrace interference to improve throughput performance with the "right mechanism". To do so in a multihop network, the following goals must be met.

(1) A relay node must be able to convert simultaneously received signals into interpretable output signals to be relayed to their final destinations.

(2) A destination must be able to extract the information addressed to it from the relayed signals.

The capability of network coding to combine and extract information through simple Galois field GF(2^n) additions [4, 5] provides a potential approach to meet such goals. However, network coding arithmetic is generally only applied on bits that have already been correctly received. That is, when the EM waves from multiple sources overlap and mutually interfere, network coding cannot be used to resolve the data at the receiver. So, criterion 1 above cannot be met.

This paper proposes the application of network coding directly within the radio channel at the physical-layer. We call this scheme Physical-layer Network Coding (PNC). The main idea of PNC is to create an apparatus similar to that of network coding, but at the physical-layer that deals with EM signal reception and modulation. Through a proper modulation-and-demodulation technique at the relay nodes, additions of EM signals can be mapped to GF(2^n) additions of digital bit streams, so that the interference becomes part of the arithmetic operation in network coding. The basic idea of PNC was first put forth in our conference paper in [6]. Going beyond [6], this paper addresses a number of practical issues of applying PNC in wireless networks. In particular, we evaluate the performance of PNC based on specific scheduling algorithms for 1D and 2D regular networks that make use of PNC (The PNC scheduling schemes in this paper can be easily extended to more general networks as in [6]) . Compared to the traditional transmission and the straightforward network coding, our analytical results show that PNC can improve the network throughput by a factor of 2 and 1.5, respectively, for the 1D network, and by a factor of 3 and 2 respectively for the 2D network.

1.1. Related Work. In 2006, we proposed PNC in [6] as demodulation mappings based on different modulation schemes. A similar idea was also published independently in [7] at the same time by another group. After that, a large body of work from other researchers on PNC began to appear. The work can be roughly divided into three categories.

In the first category, PNC is regarded as a modulation-demodulation technique. Many new PNC mapping schemes have been proposed since [6]. For example, [8] proposed a scheme based on Tomlinson-Harashima precoding. Following [6], [9] proposed a simple relay strategy called analog network coding (ANC), in which the relay amplifies and forwards the received superimposed signal without any processing. Analog network coding turns out to be similar to a scheme earlier by researchers in the satellite communication society [10]. In [11], a number of memoryless relay functions, including PNC mapping and the BER optimal function, were identified and analyzed assuming phase synchronization between signals of the transmitters. In [12], we observed that there is a one-to-one correspondence between a relay function and a specific PNC scheme under the general definition of memoryless PNC. Besides the precise definition of memoryless PNC which distinguishes it from the traditional straightforward network coding (SNC), [12] also gave a number of new PNC schemes. Reference [13] proposed a new PNC scheme where the relay maps a group constellation points to one signal according to the phase difference of the two end nodes' signals. The mechanism also takes care of the phase difference between the two end nodes implicitly.

In the second category, PNC and channel coding are studied jointly. In [14–16], PNC was combined with Lattice code or LDPC code. It was proved that the capacity of the two-way relay channel can be approached in high SNR and low SNR. In [14–16], channel coding and PNC mapping are performed independently (i.e., successively). In [17], we proposed a novel scheme which treats channel coding and PNC in an integrated manner. We show that joint channel-PNC decoding can outperform the previous schemes significantly.

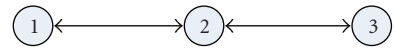


Figure 1: A three-node linear network.

In the third category, the focus is on the performance impact and significance of PNC in large-scale wireless networks. For one-dimensional wireless networks, [18] showed that PNC can improve the capacity by a fixed factor, although it does not change the scaling law. For two-dimensional wireless networks, [19] showed that PNC can increase capacity by a factor of 2.5 for the rectangular networks and a factor 2 for the hexagonal networks. However, the result in [18] is obtained based on a rough scheduling scheme which is established traditional network coding rather than physical-layer network coding (the special properties of PNC are ignored). Our paper here also discusses the application of PNC in large-scale wireless networks. It is different from [18] in that we provide the construction of an explicit PNC-scheduling algorithm (specially designed for PNC), upon which all our results are established. Compared with [19], we consider the many-to-many scenario with multiple sources and destinations, while [19] only considered the one-to-many scenario with one source.

The rest of this paper is organized as follows. Section 2 overviews the basic idea of PNC with a linear 3-node multihop network. Sections 3 and 4 investigate the application of PNC in the 1D regular linear network and 2D regular grid network, respectively. Section A concludes the paper.

2. Illustrating Example: A Three-Node Wireless Linear Network

Consider the three-node linear network in Figure 1. N_1 (Node 1) and N_3 (Node 3) are nodes that exchange information, but they are out of each other's transmission range. N_2 (Node 2) is the relay node between them.

This three-node wireless network is a basic unit for cooperative transmission and it has previously been investigated extensively [20–25]. In cooperative transmission, the relay node N_2 can choose different transmission strategies, such as Amplify-and-Forward or Decode-and-Forward [22], according to different Signal-to-Noise (SNR) situations. This paper focuses on the Decode-and-Forward strategy. We consider frame-based communication in which a time slot is defined as the time required for the transmission of one fixed-size frame. Each node is equipped with an omnidirectional antenna, and the channel is half duplex so that transmission and reception at a particular node must occur in different time slots. Slow fading is assumed throughout this paper for the ease of synchronization.

Before introducing the PNC transmission scheme, we first describe the traditional transmission scheduling

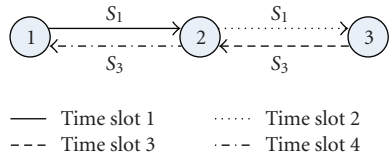


Figure 2: Traditional scheduling scheme.

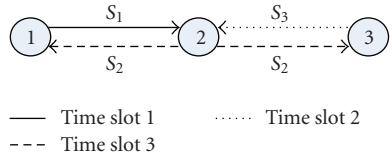


Figure 3: Straightforward network coding scheme.

scheme and the "straightforward" network-coding scheme for mutual exchange of a frame in the three-node network [20, 25].

2.1. Traditional Transmission Scheduling Scheme. In traditional networks, interference is usually avoided by prohibiting the overlapping of signals from N_1 and N_3 to N_2 in the same time slot. A possible transmission schedule is given in Figure 2. Let S_i denote the frame initiated by N_i. N_1 first sends S_1 to N_2, and then N_2 relays S_1 to N_3. After that, N_3 sends S_3 in the reverse direction. A total of four time slots are needed for the exchange of two frames in opposite directions.

2.2. Straightforward Network Coding Scheme. References [20, 25] outline the straightforward way of applying network coding in the three-node wireless network. Figure 3 illustrates the idea. First, N_1 sends S_1 to N_2 and then N_3 sends frame S_3 to N_2. After receiving S_1 and S_3, N_2 encodes frame S_2 as follows:

$$S_2 = S_1 \oplus S_3, \tag{1}$$

where $\oplus$ denotes bitwise exclusive OR operation being applied over the entire frames of S_1 and S_3. N_2 then broadcasts S_2 to both N_1 and N_3. When N_1 receives S_2, it extracts S_3 from S_2 using the local information S_1, as follows:

$$S_1 \oplus S_2 = S_1 \oplus (S_1 \oplus S_3) = S_3. \tag{2}$$

Similarly, N_2 can extract S_1. A total of three time slots are needed, for a throughput improvement of 33% over the traditional transmission scheduling scheme.

2.3. Physical-Layer Network Coding (PNC). We now introduce PNC as shown in Figure 4. Let us assume that the use of BPSK modulation at all the nodes. We further assume symbol-level time and carrier-phase synchronization, and the use of power control, so that the frames from N_1 and N_3 arrive at N_2 with the same phase and amplitude (Power control can be achieved in a slow fading channel with current techniques. Additional discussion about carrier-phase and symbol time synchronization can be found in [26]) . The

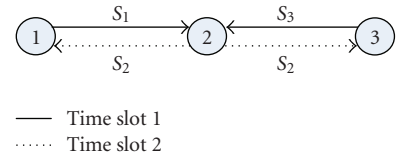


Figure 4: Physical-layer network coding.

combined bandpass signal received by N_2 during one symbol period is

$$\begin{aligned} r_2(t) &= s_1(t) + s_3(t) \\ &= a_1 \cos(\omega t) + a_3 \cos(\omega t) \\ &= (a_1 + a_3)\cos(\omega t), \end{aligned} \tag{3}$$

where $s_i(t)$, $i = 1$ or 3, is the bandpass signal transmitted by N_i, $r_2(t)$ is the bandpass signal received by N_2 during one symbol period, a_i is the BPSK modulated information bit of N_i, and ω is the carrier frequency. Then, N_2 will obtain a baseband signal $a_1 + a_3$.

Note that N_2 cannot extract the individual information transmitted by N_1 and N_3, that is, a_1 and a_3, from the combined signal in $a_1 + a_3$. However, N_2 is just a relay node. As long as N_2 can transmit the necessary information to N_1 and N_3 for extraction of a_1 and a_3 over there, the end-to-end delivery of information will be successful. For this, all we need is a special modulation/demodulation mapping scheme, referred to as PNC mapping in this paper, to obtain the equivalence of GF(2) summation of bits from N_1 and N_3 at the physical-layer.

Table 1 illustrates the idea of PNC mapping. In Table 1, $s_j \in \{0, 1\}$ is a variable representing the data bit of N_j and $a_j \in \{-1, 1\}$ is a variable representing the BPSK modulated bit of s_j such that $a_j = 2s_j - 1$.

With reference to Table 1, N_2 obtains the information bits:

$$s_2 = s_1 \oplus s_3. \tag{4}$$

It then transmits

$$s_2(t) = a_2 \cos(\omega t). \tag{5}$$

The BER analysis in [6] shows that the end-to-end BER for the three schemes is similar when the per-hop BER is low (the BER is less than 10^{-5} for 10 dB). Ignoring the slight BER difference, we have the following conclusion. For a frame exchange, PNC requires two time slots, 802.11 requires four, while straightforward network coding requires three. Therefore, PNC can improve the system throughput of the three-node wireless network by a factor of 100% and 50% relative to traditional transmission scheduling and straightforward network coding, respectively.

3. Applying PNC in Regular 1D Networks

Our discussions so far has only focused on the simple 3-node network with one bidirectional flow. In this section,

Table 1: PNC Mapping: modulation mapping at N_1, N_2; demodulation and modulation mappings at N_3.

Modulation mapping at N_1 and N_3				Demodulation mapping at N_2		
				Input	Output	
Input		Output			Modulation mapping at N_2	
					Input	Output
s_1	s_3	a_1	a_3	$a_1 + a_3$	s_2	a_2
1	1	1	1	2	0	−1
0	1	−1	1	0	1	1
1	0	1	−1	0	1	1
0	0	−1	−1	−2	0	−1

we discuss the application of PNC in 1D regular networks. There are two reasons for this discussion. First, the schemes proposed in regular network still work in random networks. And the analytical results in regular networks also provide some insights about applying PNC in random networks. Second, the regular network can also find applications in real world. For example, APs (access points) positioned along a highway form a regular linear chain in a vehicular network.

3.1. Regular Linear Network with One Bidirectional Flow. Consider a regular linear network with N nodes with equal spacing between adjacent nodes. Label the nodes as node 1, node 2, ..., node N, successively with nodes 1 and N being the two source and destination nodes, respectively. Figure 5 shows a network with $N = 5$. Suppose that node 1 is to transmit frames X_1, X_2, ... to node N, and node N is to transmit frames Y_1, Y_2, ... to node 1.

We could divide the time slots into two types: odd slots and even slots. In the odd time slots, the odd-numbered nodes transmit and the even-numbered nodes receive. In the even time slots, the even-numbered nodes transmit and the odd-numbered nodes receive.

Figure 5 shows the sequence of frames being transmitted by the nodes in a 5-node network. In slot 1, node 1 transmits X_1 to node 2 and node 5 transmits Y_1 to node 4 at the same time. In slot 2, node 2 and node 4 transmit X_1 and Y_1 to node 3 simultaneously; both node 2 and node 4 also store a copy of X_1 and Y_1 in their buffer, respectively. In slot 3, node 1 transmits X_2 to node 2, node 5 transmits Y_2 to node 4, and node 3 broadcasts $X_1 \oplus Y_1$ simultaneously; node 3 stores a copy of $X_1 \oplus Y_1$ in its buffer. Adding the stored X_1 to $X_2 \oplus X_1 \oplus Y_1$ received with PNC detection, node 2 can obtain $Y_1 \oplus X_2$. Node 4 can obtain $Y_2 \oplus X_1$ similarly. In slot 4, node 2 and node 4 broadcast $Y_1 \oplus X_2$ and $Y_2 \oplus X_1$, respectively. In this way, node 5 receives a copy of X_1 and node 1 receives Y_1 in slot 4. Also, in slot 4, node 3 obtains $Y_2 \oplus X_2$ by adding stored packet $X_1 \oplus Y_1$ to the received packet $X_1 \oplus Y_2 \oplus X_2 \oplus Y_1$.

With reference to Figure 5, we see that a relay node forwards two frames, one in each direction, every two time slots. So, the throughput is 0.5 frame/time slot in each direction. Due to the half duplex assumption, this is the maximum possible throughput we can achieve.

As detailed above, when applying PNC on the linear network, each node transmits and receives alternately in successive time slots; and when a node transmits, its adjacent

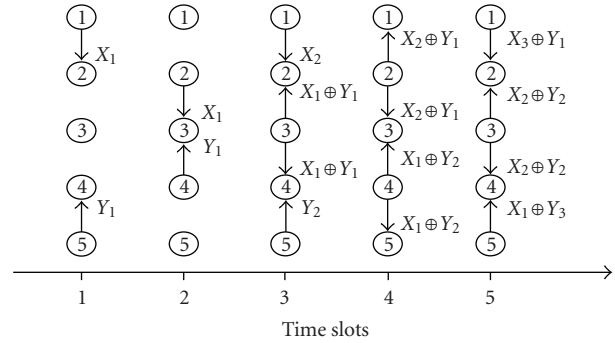


Figure 5: Bidirection PNC transmission in linear network.

nodes receive, and vice versa (see Figure 5). Let us investigate the signal-to-inference ratio (SIR) given this transmission pattern to make sure that it is not excessive. Consider the worst-case scenario of an infinite chain. We note the following characteristics of PNC from a receiving node's point of view.

(a) The interfering nodes are symmetric on both sides.

(b) The simultaneous signals received from the two adjacent nodes do not interfere due to the nature of PNC.

(c) The nodes that are two hops away are also receiving at the same time, and therefore will not interfere with the node.

Therefore, the two nearest interfering nodes are three hops away. We have the following SIR:

$$\text{SIR} = \frac{P_0/d^\alpha}{2 * \sum_{l=1}^{\infty} P_0/[(2l+1)d]^\alpha}, \tag{6}$$

where P_0 is the common (In a regular network, a trivial result of power control is that every node uses the same transmission power if the distances between adjacent nodes are constant) transmitting power of the nodes and α is the path-loss exponent. According to [27], $\alpha = 2$ for free space, $\alpha = 2.7 \sim 3.5$ for urban cellular networks, and $\alpha = 4 \sim 6$ for in-building transmission. We calculate the SIR for different α and the results are shown in Table 2. As can be seen, when $\alpha \geq 3$ (this is typical in wireless networks), the SIR is no less than 10 dB and the impact of the interference on BER is

Table 2: Signal to Noise Ratio with different path loss exponent

α	2	3	4	5	6
SIR (dB)	3.3	9.8	15.3	20.4	25.4

negligible for BPSK based on [28] (the capture threshold is often set to 10 db in wireless networks [3]). More generally, a thorough treatment should take into account the actual modulation scheme used, the difference between the effects of interference and noise, and whether or not channel coding is used. However, we can conclude that as far as the SIR is concerned, PNC is not worse than *traditional scheduling* (see Section 4) when generalized to the n-node linear network (In this paper, we assume that channel coding [17] is properly used at all the nodes and the packets can be correctly decoded to avoid error propagation once the targeted SIR is achieved. Reference [17] provides and investigates a hop-to-hop channel coding scheme for PNC) .

3.2. Regular Linear Network with Multiple Flows. Part A considers only one bidirectional flow. Here we consider a general setting in which there are K unidirectional flows in the N-node linear network. Note that this generalization includes the scenario in which there is a combination of unidirectional and bidirectional flows in the network, since each bidirectional flow can be considered as two unidirectional flows.

To allow PNC to be applied, we compose bidirectional flows out of the K unidirectional flows by matching pairs of unidirectional flows in opposite directions. The bidirectional flows can then make use of PNC for transmission, while the remaining unmatched unidirectional flows make use of the traditional strategy of multihop data transmission.

The optimal way to compose the bidirectional flows and schedule the transmission of the links in the flows is a tough problem. Here we consider a simple heuristic which is asymptotically optimal for the regular N-node linear network when N goes to infinity as shown in Part C. For simplicity, we assume that all flows have equal traffic.

We define the following terms with respect to the linear network. Let us label the nodes from left to right by 1 to N sequentially. Let (s_i, d_i) denote the source-destination pair of flow i. For a right-bound flow, $s_i < d_i$; for a left-bound flow, $s_i > d_i$. Let F denote the overall set of flows, and $F_R \subseteq F$ be the set of right-bound flows and $F_L \subseteq F$ be the set of left-found flows.

Two right-bound (left-bound) flows i and j are said to be *nonoverlapping* if $d_i < s_j$ or $d_j < s_i$ ($s_i < d_j$ or $s_j < d_i$). A *right packing* (*left packing*) is a set of nonoverlapping right-bound flows (left-bound flows). A dual packing consists of a right packing and a left packing. Figure 6 shows an example of a dual packing. Flows 2 and 3 form a right packing, and Flow 1 forms a left packing. Note that some of the nodes are traversed by both a right-bound flow and a left-bound flow. Let us call these nodes the common nodes, and the other nodes the noncommon nodes. A sequence of adjacent common nodes, flanked by but not including two noncommon nodes at two ends (an ellipse in Figure 6), forms a *PNC unit*, and we can use the PNC mechanism for transporting the bidirectional traffic over it. A sequence of adjacent noncommon nodes, together with the two common nodes flanking them (a rectangle in Figure 6), may or may not have traffic flowing over them. When there is traffic, the traffic is in one direction only, and the traditional multihop communication technique can be used to carry the unidirectional traffic. Essentially, by forming a dual packing, we also form many "virtual" bidirectional flows (each corresponding to a PNC unit) on which PNC can be applied.

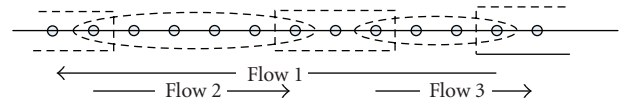


Figure 6: An example of a dual packing formed by a right packing and a left packing. An ellipse corresponds to a PNC unit. The nodes between two adjacent ellipses (including the terminal nodes of the ellipses) are grouped together by a rectangle.

Our heuristic as showing in Algorithm 1 consists of a method of forming dual packings from the K unidirectional flows.

The dual packings yield a set of "virtual" bidirectional flows, each corresponding to a PNC unit. Scheduling can then be performed as follows. Let us refer to the time needed for all the K unidirectional flows to transfer one packet from source to destination as one *frame*. Each link (hop) of a flow is allocated one time slot for transmission within a frame. A frame is further divided into two intervals, as follows.

(1) The first interval is dedicated to the PNC units (i.e., ellipses). Note that if there are M dual packings, $2M$ time slots are needed in the worst case; in the worst case, different dual packings use different time slots to transmit, and 2 time slots are needed for each dual packing (Two caveats are in order. The first is that according to our construction, there could be "trivial" PNC units with two nodes only. In this case, the PNC mechanism is not needed, and each node gets to transmit directly to the other node. Regardless of whether the PNC unit is trivial or not, two time slots are needed for the bidirectional flows. The second caveat is that there could be two PNC units in the same dual packing next to each other. For example, suppose nodes 1, 2, and 3 form a PNC unit, and nodes 4, 5, 6 form another. To avoid conflict, the scheduling of the transmissions on these two PNC units should be such that nodes 1, 3, 4, and 6 transmit in one time slot while nodes 2 and 5 transmit in another time slot. Again, two time slots are needed.).

(2) The second interval is dedicated to the nonPNC units (i.e., rectangles). The nodes of all rectangles of all dual packings are scheduled to transmit using the conventional scheme.

```
while (F ≠ ∅) { /* Each iteration in the while loop forms a dual packing. */
while (F_R ≠ ∅) { /* Each iteration in the while loop tries to find a "tight" right packing */
   largestDest=0;
   while (true) {
/* Each iteration in the while loop includes one more flow into the right packing being assembled. */
      i = arg min_{j∈F_R:s_j>largestDest} s_j
/* Select a flow with the smallest source larger than LargestDest; assume "null" is returned if there is no more flow
left in F_R with s_j > largestDest. */
      if (i ≠ null) {
         include flow i into the current right packing being assembled;
         largestDest = d_i;
         remove flow i from F;
      } else
         break;
/* Break out of the while(true) loop. */
      }
   }
while (F_L ≠ ∅) {
/* Each iteration in the while loop tries to find a "tight" left packing. */
/* Comment: details omitted here; the procedure is similar to the "while (F_R ≠ ∅)" loop above
except that largestDest is replaced by smallestDest; s_j > largestDest is replaced by s_j < smallestDest etc. */
   }
/* Combine the right packings and left packings one by one to obtain dual packings */
}
```

Algorithm 1

The number of time slots needed in the second interval depends on both the number and the lengths of the rectangles. As will be shown in Part C, it can be ignored compared to the time slots needed in the first interval as N goes to infinity.

3.3. Throughput of 1D Network with PNC. We now show that the packing and scheduling strategies presented in Part B can allow the upper-bound capacity of 1D network to be approached when the number of nodes N goes to infinity. Furthermore, compared with the conventional schemes discussed in [29], PNC can achieve a constant factor of throughput improvement.

We first detail the system model. To avoid edge effects, we consider a "large" circle instead of a line. The N nodes are uniformly distributed over the circle with a constant distance between adjacent nodes. Without loss of generality, let the distance between two adjacent nodes be a unit distance. Each transmission is over only one unit distance (i.e., a node only transmits to its two adjacent nodes). Consider the receiver of a link. We assume that simultaneous transmission by another link whose transmitter is two or more hops away from the receiver of the first link will not cause a collision to the first link. In our model, $N/2$ nodes are randomly chosen as the source nodes. The remaining $N/2$ nodes are the potential destination nodes. For each source node, a unique destination node is chosen among the $N/2$ potential destination nodes with equal probability. We assume matching without replacement in that the destination node chosen for a source node will not be put back to the pool before the destination node of another source is chosen. The route for a source-destination pair is also predetermined in a random way (note: there are two routes from a source to its destination, one in the clockwise direction and the other in the counterclockwise direction).

The analytical results for the traditional transmission scheme and straightforward network coding scheme in our circular model are similar to those in the 1D linear network in [29] when N goes to infinity. Using similar approach, it is not difficult to obtain the respective per-flow throughputs in our circular network as

$$\lambda_T(N) = \frac{2}{N}, \qquad \lambda_S(N) = \frac{8}{3N}, \tag{7}$$

where unit link bandwidth is assumed.

Let us now focus on the PNC throughput. We will show that PNC can achieve the per-flow throughput $4/N - \varepsilon$ for any small positive value ε as N goes to infinity. Let us first provide further details to the scheduling strategy presented in Part B.

The packing and scheduling are as follows. For packing, we first unwrap the circle to a noncircular linear network by randomly selecting the source node of a clockwise flow, labelled s, on the circle as the start point of the linear network. The adjacent node of the selected source node in the counterclockwise direction in the circle, labeled e, will serve as the end point of the linear network. Next, we obtain one packing of the clockwise flows according to the packing algorithm in Part B. It is possible that the last selected flow crosses the start point. In that case, we cut the flow into two subflows by performing the cut between the start point and the end point, and only consider the first subflow in the aforementioned packing. After forming the

above clockwise unidirectional packing, we form a matching counterclockwise unidirectional packing at choosing e as the start point and s as the end point. If there is an existing counterclockwise flow with e as its source node, we will start with this flow in the unidirectional packing. If not, we will choose the next flow with source node closest to e in the counterclockwise direction in our packing.

For "traffic balance", after getting the first dual packing as above, for the next dual packing, we will start with forming the counterclockwise unidirectional packing first (i.e., s and e will be defined with respect to the counterclockwise packing) before constructing the matching clockwise packing. Repeating the above procedure allows us to form a series of dual packings.

The scheduling of transmissions is the same as that in Part B except that here we also have to consider the transmission across the two subflows cut as above, if any. We assume the traffic from the destination of a preceding subflow to the source of its corresponding subflow is transmitted using the conventional scheme in the second interval.

With the above packing and scheduling strategies, we have the following theorem on the per-flow throughput of the 1D circular network when N goes to infinity.

Theorem 1. *With PNC, we can approach the upper bound of the per-flow throughput of the 1D network:*

$$\lambda_P(N) = \frac{4}{N}. \tag{8}$$

Sketch of Proof. A sketch of the proof for Theorem 1 is provided here and a detailed proof is given in the Appendix. With the help of the max-flow min-cut theorem, the upper bound of the per-flow throughput for our 1D circular network can be shown to be $4/N$. That this upper bound can be approached with the application of the aforementioned PNC packing and scheduling strategies is argued as follows. Consider the original $N/4$ unidirectional flows. With PNC packing and scheduling, these flows have been decomposed into PNC units and nonPNC units for transmission in the first and second intervals. For each round of first and second intervals (i.e., for each frame), one packet is transported from the source to the destination of each flow. We can show that the number of time slots needed in the first interval for all the flows is at most $(1 + \varepsilon_1)N/4$, where the small positive quantity ε_1 goes to zero as N goes to infinity. The number of time slots needed in the second interval, on the other hand, is $\varepsilon_2 N$, where the small positive quantity ε_2 goes to zero as N goes to infinity. Then we can obtain the per-flow throughput with PNC: $1/(N/4 + \varepsilon_1 N/4 + \varepsilon_2 N/4) = (1 - \varepsilon)N/4$.

A corollary of Theorem 1 is that PNC can improve the throughput of the 1D network by a factor of 2 and 1.5 relative to the traditional transmission scheme and the SNC scheme (7), respectively.

A notable fact is that PNC can approach the capacity with minimum energy. Recall that PNC exchanges one packet between the two end nodes within two time slots, during which each of the n nodes on the chain transmits once with energy E_t and receives once with energy E_r. And a total energy $n(E_t+E_r)$ is used. In fact, $n(E_t+E_r)$ is the lower bound of energy to exchange one packet. For one exchange, the two end nodes must transmit once to send their message and must receive once to obtain their needed message; the $n-2$ relay nodes must receive once and transmit once to finish one relay. Therefore, the energy of $n(E_t + E_r)$ is necessary.

4. Applying PNC in 2D Grid Network

Section 3 focused on the 1D regular network. This section investigates the application of PNC in a 2D regular gird network. We assume the same transmission protocol as in Section 3.

4.1. 2D Grid Network with One Bidirectional Flow in Each Line. Figure 7 shows the grid network under consideration, in which N nodes are uniformly located at the cross points as shown. In this part, we first consider the case in which each line (horizontal or vertical) on the grid has one and only one bidirectional flow. Specifically, the two end nodes in each line, node 1 and node $\sqrt{N}$, exchange information through the relay nodes in between.

The flows transmit with the following PNC schedule. Consider the horizontal lines (similar schedule applies for the vertical lines). The first two time slots are dedicated to transmissions on lines $1, J+1, 2J+1, \ldots$; the next two time slots are dedicated to transmissions on lines nodes on the lines $2, J+2, 2J+2, \ldots$; and so on. The separation J must be large enough for acceptable SIR. In the example of Figure 7, $J = 4$.

For a group of simultaneous active lines, to reduce SIR, when the odd nodes transmit on one active line, then the even nodes will transmit on its two adjacent active lines, as shown in Figure 7.

Let us investigate the SIR of this transmission pattern given a J. Consider the worst-case scenario in which N goes to infinity. For a given receiver, the interference from the nodes within the same line is $I_1 = 2 * \sum_{l=1}^{\infty} P_0/[(2l+1)d]^{\alpha}$, where P_0, l, $d = 1$, and α are defined similarly as in Section 3.1. Without loss of generality, suppose that the receiver is an even node. The interference from the other active lines whose odd nodes are transmitting is $I_2 = 4\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} P_0/[(2l)^2 d^2 + J^2(2k+2)^2 d^2]^{\alpha/2}$, and the interference from the other active lines whose even nodes are transmitting is $I_3 = 4\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} P_0/[(2l+1)^2 d^2 + J^2(2k+1)^2 d^2]^{\alpha/2}$. Thus, the overall SIR is given by

$$\text{SIR} = \frac{P_0/d^{\alpha}}{I_1 + I_2 + I_3}. \tag{9}$$

For a typical value of $\alpha = 4$, the SIR in (9) is about 13.5 dB, 12.3 dB, and 10.0 dB for J equals 5, 4, and 3, respectively. With an assumed 10 dB target, $J = 3$ is enough to guarantee successful transmission.

4.2. 2D Grid Network with Multiple Random Flows. Let us now investigate the application of PNC in the 2D grid network with a more general traffic pattern. With respect to

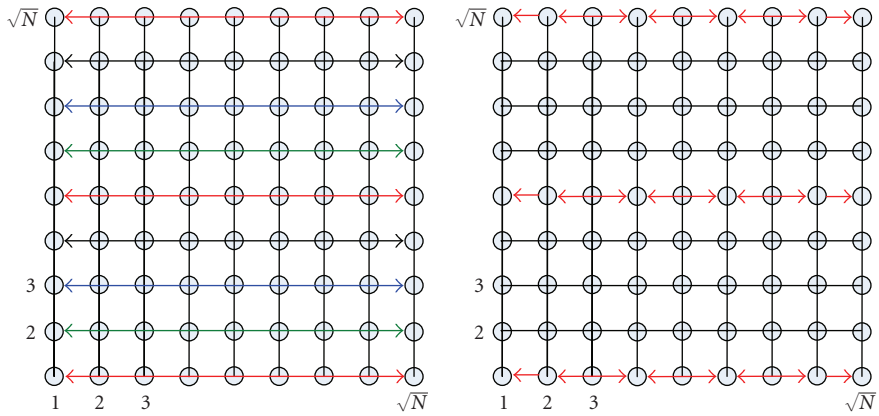


FIGURE 7: Subfigure (a) shows 2D grid network with one bidirectional flow in each line. The lines separated by $J - 1 = 3$ lines, that is, the lines with the same color, are allowed to transmit simultaneously. Subfigure (b) shows a scheduling for one group active lines (red lines) in a specific time slot

Figure 7, we now randomly choose $N/2$ of the nodes as the source nodes. The remaining $N/2$ nodes are the destination nodes.

Here we apply a simple routing scheme, as in [29]. For a source-destination pair at positions (x_s, y_s) and (x_d, y_d), the data will first be forwarded vertically to the node at (x_s, y_d) before being forwarded horizontally to the destination. The horizontal and vertical transmissions are separated into two different time intervals. For horizontal (or vertical) transmissions, the scheduling within each line (column) is the same as that in the Section 3.2 and the scheduling among different lines (columns) is the same as in part A.

When N goes to infinity, the number of nodes in each line or column, $\sqrt{N}$, also goes to infinity, and the per-flow PNC throughput in each line or column will approach $4/\sqrt{N}$, as argued in Section 3. Since the horizontal transmission and vertical transmission are scheduled in different time interval and in each interval every J lines (columns) transmit simultaneously, the per-flow transmission of PNC in the 2D grid network can approach

$$\lambda_P(N) = \frac{4}{\sqrt{N}} \cdot \frac{1}{J} \cdot \frac{1}{2} = \frac{2}{J\sqrt{N}}. \tag{10}$$

For comparison purposes, let us look at the per-flow throughput under the traditional transmission strategy and under the straightforward network coding strategy. With the routing/scheduling strategy and the corresponding throughput analysis in [29], we can show that the traditional transmission scheme and SNC scheme can achieve the following throughputs, respectively:

$$\begin{aligned} \lambda_T(N) &= \frac{4}{(1+\Delta)\sqrt{N}} \cdot \frac{1}{3} \cdot \frac{1}{2} = \frac{2}{9\sqrt{N}}, \\ \lambda_C(N) &= \frac{4}{(1+\Delta/2)\sqrt{N}} \cdot \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{3\sqrt{N}}. \end{aligned} \tag{11}$$

In the 2D grid network, the nodes are tightly packed than in the 1D network, and the interfering nodes must be kept at least 3 hops away, that is, $\Delta = 2$, to obtain an SIR of no less than 10 dB (note: in the 1D network, Δ could be 1 for SIR of about 10 dB). When $\Delta = 2$, we can verify that throughputs better than (11) cannot be achieved. In other words, the throughput in (11) is also the upper bound for traditional transmission scheme and SNC scheme under all possible schedulings.

Therefore, setting $J = 3$ in (10), we conclude that PNC can achieve a throughput improvement factor of 3 and 2 relative to the traditional transmission scheme and the SNC scheme, respectively. Note that the improvement factors under the 2D network are larger than those under the 1D network, which are 2 and 1.5, respectively (see Section 3).

5. Conclusion

This paper has introduced a novel scheme called *Physical-layer Network Coding* (PNC) that significantly enhances the throughput performance of multihop wireless networks. Instead of avoiding interference caused by simultaneous electromagnetic waves transmitted from multiple sources, PNC embraces interference to effect network-coding operation directly from physical-layer signal modulation and demodulation. With PNC, signal scrambling due to interference, which causes packet collisions in the MAC layer protocol of traditional wireless networks (e.g., IEEE 802.11), can be eliminated.

We have proposed explicit scheduling algorithms for PNC in 1D and 2D regular networks with multiple random flows. It is shown that PNC can potentially achieve 100% and 50% throughput increases compared with traditional transmission and straightforward network coding, respectively, in the 1D regular linear network. The throughput improvements are even larger in the 2D regular network: 200% and 100%, respectively. In particular, PNC can allow the upper-bound throughput of the 1D regular network to be approached as the number of nodes goes to infinity.

Appendix

A. Proof of Theorem 1

This appendix proves Theorem 1 in three steps. First, the fact that $4/N$ is the upper bound for the throughput of the 1D circular linear network can be argued as follows. Let us consider the number of time slots needed so that each flow can transport one packet from its source to its destination. Due to half-duplexity, there can be at most $N/2$ transmitting nodes in a time slot. In general, each transmitting node can transmit to at most two of its adjacent nodes simultaneously. Hence, in total, there can be at most N one-hop transmissions being successfully completed in each time slot. The number of hops between the source and destination of a flow is on average $N/2$. There are altogether $N/2$ flows. Using Chernoff bound, we can show that the total number of one-hop transmissions required (aggregated over all flows) is $N^2/4$ w.h.p. as N goes to infinity. Thus, the time slots needed are lower bounded by $(N^2/4)/N = N/4$. Within this number of time slots, each flow transports a packet from source to destination. Thus, the per-flow throughput is upper bounded by $\lambda \leq 1/(N/4) = 4/N$.

Next, we prove that the number of time slots needed in the second interval is negligible compared to N, denoted by $\varepsilon_2 N$ where ε_2 is a small positive quantity that goes to zero as N goes to infinity. The total one-hop transmissions in the second interval can be divided into two parts, the one-hop transmissions in the rectangles and the one-hop transmissions between subflows (created when we unwrap the circular network into a linear network).

Let us first consider the rectangles. As shown in Figure 8, within a dual packing, the rectangles do not overlap. Furthermore, the two end nodes in a rectangle must be either a source or destination node of some flow. As a proof technique, let us artificially divide the rectangles into two groups according to the dual packings containing them. Recall that the dual packings are formed successively in our packing algorithm. Consider the first $(1 - \varepsilon_3)$ fraction of all flows (including the original flows and the generated subflows) that are included successively into the dual packings. The first group of rectangles arises from these flows. The second group of rectangles belongs to the remaining ε_3 fraction of the flows. We set ε_3 such that $\varepsilon_3 = 1/\sqrt{\log N}$.

As discussed in Section 3.2, when we perform packing on the circular network by unwrapping it to a linear network, it is possible for a flow to be cut into two subflows. Each clockwise unidirectional packing contains at least one flow that does not generate subflows (a flow cannot have more than N hops). As a corollary, if the clockwise packing contains a flow that has been cut into two subflows, then the packing must contain at least two flows to start with. One of these subflows will be relegated to a future packing exercise. So, each clockwise packing reduces the number of remaining flows to be packed by at least one. For the matching counterclockwise packing, at most one flow will be cut into two subflows. Thus, the matching counterclockwise packing does not increase the number of remaining counterclockwise flow. Recall from the discussion in Section 3.2 that for "traffic balance" successive dual packings will start with clockwise and counterclockwise packings in an alternate manner. Thus, successive dual packings will reduce the numbers of remaining clockwise and counterclockwise flows by at least one alternately.

In the beginning, there are $N/2$ original flows ($N/4$ of which are clockwise and $N/4$ of which are counterclockwise flows). From the argument in the previous paragraph, there are altogether at most $N/2$ dual packings. Each dual packing will at most generate at most two extra flows to the flow pool (because of cut between s and e). Thus, altogether there could be at most N extra flows being generated. Hence, the total number of flows (including the original flows and the subflows) is $3N/2$.

In general, since the two end nodes of a rectangle must be either a source or a destination of some flow, the number of rectangles in a dual packing is no more than the number of flows in that dual packing (note: some nonend nodes within a rectangle could also be sources or destinations; thus the "no more than" rather than "equal to"). Therefore, the number of rectangles in the first group is no more than $(1 - \varepsilon_3)N$. For these rectangles, as shown in Lemma 2 at the end of this appendix, the number of nodes in each group-1 rectangle is no more than $(1 - \varepsilon_4)\log(N) + \varepsilon_4 N$ w.h.p., where ε_4 is a small positive quantity that goes to zero when N goes to infinity. Similarly, the number of rectangles in the second group is upper bounded by $\varepsilon_3 N$. As a trivial bound, we will upper-bound the number of nodes in each group-2 rectangle by N. Note that each node will at most transmit once within a rectangle (group-1 or group-2) for traffic forwarding. Thus, the total number of one-hop transmissions needed for the rectangles is upper bounded by

$$T_1 = (1 - \varepsilon_3)N \cdot \left[(1 - \varepsilon_4)\log(N) + \varepsilon_4 N\right] + \varepsilon_3 N \cdot N. \tag{A.1}$$

Now, consider the transmissions across subflows. A one-hop transmission is needed for two adjacent subflows generated by the cut when we unwrap the circular network to a corresponding linear network. In other words, there is a one-hop transmission whenever there is an extra subflow, which is upper bounded by $N/2$ according to the above argument. Thus, the total number of one-hop transmissions between all adjacent subflows is upper bounded by $T_2 = N/2$.

Putting things together, the total one-hop transmissions in the second interval is upper bounded by $T_1 + T_2$. Since we determine the start and end nodes of each dual packing in a uniformly random way and pack each unidirectional packing in a uniformly random way, the one-hop transmissions in the rectangles are also uniformly distributed among all the N nodes along the circle. With the traditional transmission scheme, there are $N/2$ one-hop transmissions in each time slot. Therefore, the time slots needed in the second interval

are upper bounded by

$$\begin{aligned} k_2 &= \frac{T_1 + T_2}{N/2} \\ &= \frac{(1-\varepsilon_3)N \cdot \left[(1-\varepsilon_4)\log(N) + \varepsilon_4 N\right] + \varepsilon_3 N \cdot N + N/2}{N/2} \\ &= 2(1-\varepsilon_3)(1-\varepsilon_4)\log(N) + 2(1-\varepsilon_3)\varepsilon_4 N + \varepsilon_3 N + 1 \\ &= N\varepsilon_2, \end{aligned} \tag{A.2}$$

where ε_2 is determined by ε_3, ε_4, and N. It is easy to show that ε_2 will go to zero as N goes to infinity.

Finally, we prove that the number of time slots needed in the first interval is less than $(1+\varepsilon_1)N/4$. In a unidirectional packing, a residual node is an idle node that through which no packet passes (i.e., none of the flows of the unidirectional packing passes through the node). Thus, the number of nodes through which one packet passes in one unidirectional packing is N, minus the number of residual nodes. Consider a dual packing to which group-1 rectangles belong. According to Lemma 1 immediately after the proof of Theorem 1 here, the number of residual nodes in each of the unidirectional packings of the dual packings is less than $\log(N)$ w.h.p.. That is, the number of nonresidual nodes in a unidirectional packing is more than N-$\log(N)$ w.h.p., and the number of nonresidual nodes in both the unidirectional packings of the dual packing is more than $2(N - \log N)$. That is, the traffic handled by each dual packing (in terms of packet flows across all nodes in the dual packing) is more than $2(N - \log N)$.

Now, consider an arbitrary node in the network. According to our model, it is either the source or destination of some flow. The packet of that flow passes through it with probability 1. For the other $N/2 - 1$ original flows, a packet passes through the node with probability 1/2. By the Chernoff-Hoeffding theorem, the number of packets that go through each node is $1/2 \cdot (N/2 - 1) + 1$ w.h.p.. Considering all N nodes, the number of packets passing through them is $(1/2(N/2-1)+1)N$. Note that this is the total traffic which is more than the traffic in the dual packings to which group-1 rectangles belong.

Therefore, the number of dual packings to which the group-1 rectangles belong is upper bounded by

$$\frac{(1/2(N/2-1)+1)N}{(2(N-\log(N)))}\ \text{w.h.p.} \tag{A.3}$$

Similar to the argument for group-1 rectangles, for the flows containing the group-2 rectangles, there are at most ε_3N flows which will generate at most ε_3N unidirectional packings, that is, ε_3N/2 dual packings. Then we can obtain that the total number of dual packings is no more than

$$\frac{(1/2(N/2-1)+1)N}{(2(N-\log(N)))} + \frac{\varepsilon_3 N}{2} = \frac{(1+\varepsilon_1)N}{8}, \tag{A.4}$$

with high probability, where ε_1 is determined by ε_3 and N. It is easy to verify that ε_1 goes to zero as N goes to infinity. Since each packing needs at most two times slots, the time slots needed for the first interval are at most $k_1 = (1+\varepsilon_1)N/4$.

With the help of k_1 and k_2, we can obtain the lower bound of the per-flow throughput as

$$\begin{aligned} \lambda_P(N) &= \frac{1}{k_1 + k_2} \\ &= \frac{1}{(1+\varepsilon_1)N/4 + 2\log(N) + 2N\varepsilon_2 + 1} \\ &= \frac{4}{N}\frac{1}{1+\varepsilon_1 + 2\log(N)/N + 2\varepsilon_2 + 1/N} = \frac{4}{N}(1-\varepsilon), \end{aligned} \tag{A.5}$$

where ε can be obtained from ε_1, ε_2, and N, and it goes to zero as N goes to infinity. Then Theorem 1 is proved.

Lemma 1. *For any clockwise (counterclockwise) unidirectional packing contained in the dual packings to which group-1 rectangles belong, the number of residual nodes is less than* $\log(N)$ *w.h.p.*

Proof. Let P denote the set of dual packings to which group-1 rectangles belong. Let us focus on one clockwise unidirectional packing p in P. The proof for the counterclockwise case is similar. Let P_c be the clockwise packings in P. Let m denote the number of clockwise flows in P_c. According to our way of partitioning the rectangles into the two groups, we have $m \le (1-\varepsilon_3)N_1$, where N_1 is the total number of clockwise flows.

Recall that in our traffic model, we randomly select $N/2$ nodes to be sources and $N/2$ nodes to be destinations. In other words, any node among the N nodes is either a source or a destination. This applies to any residual node in p as well. In particular, a residual node in p is either (1) a destination node (of a clockwise or counter-clockwise flow), (2) a source node of a counter-clockwise flow, or (3) a source node of a clockwise flow. In case 3, since the residual node is a residual node in p, it must be a source node of a clockwise flow already packed (i.e., already belong to P_c) prior to packing p.

For a unidirectional packing, consider the first flow from the start point s. Suppose this flow ends at node i. Let us consider the probability of node $(i+1)$ being a residual node with respect to this unidirectional packing. Due to the randomness of our packing procedure and our random selection of sources and destinations for flows, node $(i+1)$ is a destination node with probability $p_1 = 1/2$; it is a source node of a counter-clockwise flow with probability $p_2 = 1/4$ w.h.p, and it is a source node of a prepacked clockwise flow with probability $p_3 \le (1-\varepsilon_3)/4$ w.h.p. Then the probability that node $(i+1)$ is a residual node given that node i is not a residual node is

$$P(1 \mid 0) = p_1 + p_2 + p_3 \le 1 - \frac{\varepsilon_3}{4}. \tag{A.6}$$

In out notation above, the 1 in $P(1 \mid 0)$ refers to the fact that we have found one residual thus far, and the 0 refers to the fact that we have not found any residual node so far.

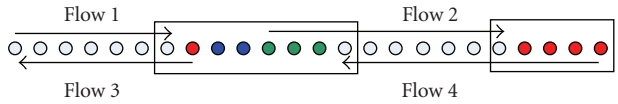


Figure 8: An example of a dual packing, where flow 1 and flow 2 belong to the clockwise unidirectional packing, flow 3 and flow 4 belong to the counterclockwise unidirectional packing. The white nodes are nonresidual nodes, the red nodes are the residual nodes of the clockwise unidirectional packing, the green nodes are the residual nodes of the counterclockwise packing, and the blue nodes are the residual nodes of both the two unidirectional packings. The nodes in the rectangles are the uncommon nodes.

Given node $(i+1)$ is a residual node, the probability that the node $(i+2)$ is also a residual node is $P(2 \mid 1) \leq P(1 \mid 0)$ (due to sampling without replacement). The probability of a sequence of l or more residual nodes is given by

$$P(1 \mid 0)P(2 \mid 1)P(3 \mid 2) \cdots P(l \mid l-1) \leq [P(1 \mid 0)]^l \leq \left[1 - \frac{\varepsilon_3}{4}\right]^l. \tag{A.7}$$

When $l = \log(N)$, as N-goes to infinity, the above probability is $\exp(-\sqrt{\log(N)}/4)$, which will approach zero. Thus, Lemma 1 is proved. □

Lemma 2. *For group-1 rectangles, the number of nodes in each rectangle is no more than* $2\log(N)$ *with probability* $1-\varepsilon_4$, *where* ε_4 *is a small positive quantity that goes to zero when N goes to infinity.*

Proof. With respect to Figure 8 and the explanation in its caption, let N_r, N_g, N_b denote the number of red, green, and blue nodes in a dual packing, respectively. By Lemma 1, $N_r + N_b \leq \log(N)$, and $N_g + N_b \leq \log(N)$ w.h.p. Thus, $N_r + N_g + N_b \leq N_r + N_g + 2N_b \leq 2\log(N)$. □

Acknowledgments

This work was partially supported by the Competitive Earmarked Research Grant (project number 414507) established under the University Grant Committee of the Hong Kong and the Natural Science Foundation of China (project number 60902016).

References

[1] T. Ojanperä and R. Prasad, "An overview of air interface multiple access for IMT-2000/UMTS," *IEEE Communications Magazine*, vol. 36, no. 9, pp. 82–95, 1998.

[2] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM '01)*, pp. 61–69, Rome, Italy, July 2001.

[3] P. C. Ng and S. C. Liew, "Throughput analysis of IEEE802.11 multi-hop ad hoc networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 309–322, 2007.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[5] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[6] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: physical-layer network coding," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 358–365, Los Angeles, Calif, USA, September 2006.

[7] P. Popovski and H. Yomo, "The anti-packets can increase the achievable throughput of a wireless multi-hop network," in *Proceedings of IEEE International Conference on Communications (ICC '06)*, vol. 9, pp. 3885–3890, Istanbul, Turkey, July 2006.

[8] Y. Hao, D. Goeckel, Z. Ding, D. Towsley, and K. K. Leung, "Achievable rates for network coding on the exchange channel," in *Proceedings of IEEE Military Communications Conference (MILCOM '07)*, Orlando, Fla, USA, October 2007.

[9] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: analog network coding," Tech. Rep. MIT-CSAIL-TR-2007-012, MIT, Cambridge, Mass, USA, 2007.

[10] M. Denkberg, "Paired carrier multiple access(PCMA) for satellite communications," in *Proceedings of the Pacafic Telecommunications Conference*, Honolulu, Hawaii, USA, 1998.

[11] T. Cui, T. Ho, and J. Kliewer, "Memoryless relay strategies for two-way relay channels: performance analysis and optimization," in *Proceedings of IEEE International Conference on Communications (ICC '08)*, pp. 1139–1143, Beijing, China, May 2008.

[12] S. Zhang, S. C. Liew, and L. Lu, "Physical layer network coding schemes over finite and infinite fields," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 3784–3789, New Orleans, La, USA, November-December 2008.

[13] T. Koike-Akino, P. Popovski, and V. Tarokh, "Denoising maps and constellations for wireless network coding in two-way relaying systems," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 3790–3794, New Orleans, La, USA, November-December 2008.

[14] S. Zhang and S. Liew, "Capacity of two-way relay channel," 3rd HK-BJ Doctoral forum, 2008, http://arxiv.org/ftp/arxiv/papers/0804/0804.3120.pdf.

[15] W. Nam, S.-Y. Chung, and Y. H. Lee, "Capacity bounds for two-way relay channels," in *Proceedings of the International Zurich Seminar on Digital Communications (IZS '08)*, pp. 144–147, Zurich, Germany, March 2008.

[16] K. Narayanan, M. P. Wilson, and A. Sprintson, "Joint physical layer coding and network coding for bi-directional relaying," in *Proceedings of the 45th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Ill, USA, September 2007.

[17] S. Zhang and S.-C. Liew, "Channel coding and decoding in a relay system operated with physical-layer network coding," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 788–796, 2009.

[18] K. Lu, S. Fu, Y. Qian, and H.-H. Chen, "On capacity of random wireless networks with physical-layer network coding," *IEEE*

Journal on Selected Areas in Communications, vol. 27, no. 5, pp. 763–772, 2009.

[19] C. Chen, K. Cai, and H. Xiang, "Scalable ad hoc networks for arbitrary-cast: practical broadcast-relay transmission strategy leveraging physical-layer network coding," *EURASIP Journal on Wireless Communications and Networking*, vol. 2008, Article ID 621703, 15 pages, 2008.

[20] Y. Wu, P. A. Chou, and S. Y. Kung, "Information exchange in wireless networks with network coding and physical layer broadcast," Tech. Rep. MSR-TR-2004-78, Microsoft Research, Redmond, Wash, USA, 2004.

[21] C. Hausl and J. Hagenauer, "Iterative network and channel decoding for the two-way relay channel," in *Proceedings of IEEE International Conference on Communications (ICC '06)*, vol. 4, pp. 1568–1573, Istanbul, Turkey, July 2006.

[22] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[23] T. M. Cover and A. A. El-Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, 1979.

[24] L. Lai, K. Liu, and H. El-Gamal, "On the achievable rate of three-node wireless networks," in *Proceedings of the IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 1, pp. 739–744, Maui, Hawaii, USA, June 2005.

[25] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.

[26] S. Zhang and S. Liew, "Synchronization analysis in physical layer network coding," Submitted, http://arxiv.org/abs/1001.0069.

[27] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.

[28] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, USA.

[29] J. Liu, D. Goeckelt, and D. Towsley, "Bounds on the gain of network coding and broadcasting in wireless networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 724–732, Anchorage, Alaska, USA, May 2007.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 708416, 13 pages
doi:10.1155/2010/708416

Research Article

Joint Channel-Network Coding for the Gaussian Two-Way Two-Relay Network

Ping Hu,[1] Chi Wan Sung,[1] and Kenneth W. Shum[2]

[1] *Department of Electronic Engineering, City University of Hong Kong, Hong Kong*
[2] *Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong*

Correspondence should be addressed to Kenneth W. Shum, wkshum@inc.cuhk.edu.hk

Received 1 October 2009; Revised 27 January 2010; Accepted 13 March 2010

Academic Editor: Sae-Young Chung

New aspects arise when generalizing two-way relay network with one relay to two-way relay network with multiple relays. To study the essential features of the two-way multiple-relay network, we focus on the case of two relays in our work. The problem of how two terminals, equipped with multiple antennas, exchange messages with the help of two relays is studied. Five transmission strategies, namely, amplify-forward (AF), hybrid decode amplify forward (HLC), hybrid decode amplify forward (HMC), decode forward (DF), and partial decode forward (PDF), are proposed. Their designs are based on a variety of techniques including network coding, multiplexed coding, multi-input multi-output transmission, and multiple access with common information. Their performance is compared with the cut-set outer bound. It is shown that there is no dominating strategy and the best strategy depends on the channel conditions. However, by studying their multiplexing gains at high signal-tonoise (SNR) ratio, it is shown that the AF scheme dominates the others in high SNR regime.

1. Introduction

Relay channel, which considers the communication between a source node and a destination with the help of a relay node, was introduced by van der Meulen in [1]. Based on this channel model, Cover and El Gamal developed coding strategies known as decode-forward (DF) and compress-forward (CF) in [2]. These techniques now become standard building blocks for cooperative and relaying networks, which have been extensively studied in the literature (e.g., [3, 4]).

For many applications, communication is inherently two-way. A typical example is the telephone service. In fact, the study of two-way channel is not new and can be traced back to Shannon's work in 1961 [5]. However, the model of two-way relay channel, though natural, did not attract much attention. Recently, probably due to the advent of network coding [6] in the last decade, there is a growing interest in this model. The application of DF and CF to two-way relay channel was considered in [7]. The half-duplex case was studied in [8, 9]. The results in [10] showed that feedback is beneficial only in a two-way transmission. Network coding for the two-way relay channel was studied in [11, 12]. Physical layer network coding based on lattices is considered recently [13], and shown to be within 0.5 bit from the capacity in some special cases [14].

All the aforementioned works are for one relaying node. It is easy to envisage that in real systems, more than one relay can be used. Schein in [15] started the investigation of the network with one source-destination pair and two parallel relays in between. This model was further studied in [16] under the assumption of half-duplex relay operations. For one-way multiple-relay networks in general, cooperative strategies were proposed and studied in [17]. We remark that a notable feature that does not exist in the single-relay case is that the multiple relays can act as a virtual antenna array so that beamforming gain can be reaped at the receiver. In this paper, we follow this line of research and consider two-way communications. Two-relays are assumed, for this simple model already captures the essential features of the more general multiple-relay case. We are interested in knowing how different techniques can be used to construct transmission strategies for the two-way two-relay network and how they perform under different channel conditions. In particular, we apply the idea of network coding to both the

physical layer and the network layer. Besides, channel coding techniques for multiple access channel (MAC) and multi-input multi-output (MIMO) channel are also employed. Several transmission strategies are thus constructed and their achievable rate regions are derived.

We remark that the channel model that we consider in this paper is also called the *restricted* two-way two-relay channel [7]. This means that the signal from a source node depends only on the message to be transmitted, but not on the received signal at the source. Besides, our results are obtained under the half-duplex assumption, which is realistic for practical systems. Each node is assumed to transmit one half of the time and receive during the other half of the time. The performance of our proposed strategies can be further improved if the ratio of transmission time and receiving time is optimized. We do not consider this more general case, since it complicates the analysis but provides no new insights.

This paper is organized as follows. Our network model is described in Section 2. Some basic coding techniques are reviewed in Section 3. Based on these coding techniques, several transmission strategies are devised in Section 4. Their performance at high signal-to-noise ratio regime is analyzed in Section 5. The rate regions of these strategies are compared under some typical channel realizations in Section 6. The conclusion is drawn in Section 7.

2. Channel Model and Notations

The two-way two-relay (TWTR) network consists of four nodes: two terminals A and B, and two parallel relays 1 and 2 (see Figure 1). Terminals A and B want to exchange messages with the help of the two relays. We assume there is no direct link between the two terminals and between the two-relays. Furthermore, all of the nodes are half-duplex. The total communication time, $2N$, are divided into two stages, each of which consists of N time slots. In the first stage, the terminals send signals and the relays receive. In the second stage, the relays send signals and the terminals receive. The solid arrows in Figure 1 correspond to stage 1 and the dashed arrows correspond to stage 2.

Suppose that terminals A and B are equipped with n antennas, whereas each of relays 1 and 2 has only one antenna. For $i \in \{A, B\}$ and $j \in \{1, 2\}$, we use $\mathbf{X}_i(t) \in \mathbb{R}^n$ to denote the transmit signal from node i, and $Z_j(t) \in \mathbb{R}$ to denote independently and identically distributed (i.i.d.) Gaussian noise with distribution $\mathcal{N}(0, \sigma^2)$. The channel is assumed static and the channel gain from node i to j is denoted by an n-dimensional column vector $\mathbf{h}_{ij}$. We assume channel reciprocity holds so that $\mathbf{h}_{ij} = \mathbf{h}_{ji}$. In the first stage, the outputs of the network at time $t = 1, 2, \ldots, N$, are given by

$$Y_1(t) = \mathbf{h}_{A1}^T \mathbf{X}_A(t) + \mathbf{h}_{B1}^T \mathbf{X}_B(t) + Z_1(t), \tag{1}$$

$$Y_2(t) = \mathbf{h}_{A2}^T \mathbf{X}_A(t) + \mathbf{h}_{B2}^T \mathbf{X}_B(t) + Z_2(t). \tag{2}$$

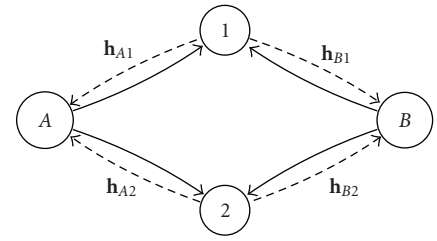


Figure 1: Model of two-way two-relay network. The labels of the arrows indicate the corresponding link gains.

In the second stage, for $t = N + 1, N + 2, \ldots, 2N$, the outputs at the terminal nodes are

$$\mathbf{Y}_A(t) = \mathbf{h}_{A1} X_1(t) + \mathbf{h}_{A2} X_2(t) + \mathbf{Z}_A(t), \tag{3}$$

$$\mathbf{Y}_B(t) = \mathbf{h}_{B2} X_2(t) + \mathbf{h}_{B1} X_1(t) + \mathbf{Z}_B(t), \tag{4}$$

where $X_j(t) \in \mathbb{R}$, $j \in \{1, 2\}$ is the transmit symbol of relay j, $\mathbf{Z}_i(t) \in \mathbb{R}^n$ for $i \in \{A, B\}$ is a Gaussian random vector with each component i.i.d according to $\mathcal{N}(0, \sigma^2)$. We assume that the link gains $\mathbf{h}_{A1}$, $\mathbf{h}_{A2}$, $\mathbf{h}_{B1}$, and $\mathbf{h}_{B2}$ are time-invariant and known to all nodes. We have the following power constraints in each stage:

$$\frac{1}{N} \sum_{t=1}^{N} \mathbf{X}_i(t)^T \mathbf{X}_i(t) \le P_i \tag{5}$$

for $i \in \{A, B\}$, and

$$\frac{1}{N} \sum_{t=N+1}^{2N} X_j^2(t) \le P_j \tag{6}$$

for $j \in \{1, 2\}$, where P_A, P_B, P_1, and P_2 denote the power constraints on terminals A and B and relays 1 and 2, respectively.

Let R_A and R_B be the data rates of terminal A and B, respectively. In a period consisting of $2N$ channel symbols (N symbols for each phase), terminal A wants to send one of the 2^{2NR_A} symbols to terminal B, and terminal B wants to send one of the 2^{2NR_B} symbols to terminal A. A $(2^{2NR_A}, 2^{2NR_B}, 2N)$ code for the TWTR network consists of two message sets $M_A = \{1, 2, \ldots, 2^{2NR_A}\}$ and $M_B = \{1, 2, \ldots, 2^{2NR_B}\}$, two encoding functions

$$f_i : M_{\mathrm{i}} \longrightarrow (\mathbb{R}^n)^N, \quad i \in \{A, B\}, \tag{7}$$

two relay functions

$$\phi_j : \mathbb{R}^N \longrightarrow \mathbb{R}^N, \quad j \in \{1, 2\}, \tag{8}$$

and two decoding functions

$$\begin{aligned} g_A &: (\mathbb{R}^n)^N \times M_A \longrightarrow M_B, \\ g_B &: (\mathbb{R}^n)^N \times M_B \longrightarrow M_A. \end{aligned} \tag{9}$$

For $i = A, B$, terminal i transmits the codeword $f_i(m_i)$ in stage one, where m_i is the message to be transmitted. For

$j = 1, 2$, relay j applies the function ϕ_j to its received signal and transmits the resulting signal in the second stage. Let the received signals at terminals A and B be $\mathbf{Y}_A^N$ and $\mathbf{Y}_B^N$, respectively. In this paper, we will use a superscript "$\mathbf{Y}^N$" to indicate a sequence of length N. So $\mathbf{Y}_A^N$ and $\mathbf{Y}_B^N$ are sequences of length N, with each component equal to a vector in $\mathbb{R}^n$. After the second stage, terminal i decodes the message from the other source node by g_i. We note that the decoding function g_i uses the message from source terminal i as input as well. We say that a decoding error occurs if $g_A(\mathbf{Y}_A^N, m_A) \neq m_B$ or $g_B(\mathbf{Y}_B^N, m_B) \neq m_A$. The average probability of error is

$$P_e^{2N} \triangleq \frac{1}{|M_A||M_B|} \times \sum_{\substack{(m_A,m_B)\\ \in M_A \times M_B}} \Pr\Big\{ g_A\left(\mathbf{Y}_A^N, m_A\right) \neq m_B, \text{or } g_B\left(\mathbf{Y}_B^N, m_B\right) \neq m_A \mid (m_A, m_B) \text{ is sent} \Big\}. \tag{10}$$

A rate pair (R_A, R_B) is said to be achievable if there exists a sequence of $(2^{2NR_A}, 2^{2NR_B}, 2N)$ codes, satisfying the power constraints in (5) and (6), with $P_e^{2N} \to 0$ as $N \to \infty$.

Although the terminals are equipped with n antennas, the transmitted signals from the terminals are essentially 2 dimensional. To see this, we observe that the first term in the right hand side of (1), namely, $\mathbf{h}_{A1}^T \mathbf{X}_A(t)$, is a projection of $\mathbf{X}_A(t)$ in the direction of $\mathbf{h}_{A1}$. Any signal component of $\mathbf{X}_A(t)$ orthogonal to $\mathbf{h}_{A1}$ will not be picked up by relay 1. Likewise, from (2), we see that any signal component of $\mathbf{X}_A(t)$ orthogonal to $\mathbf{h}_{A2}$ will not be sensed by relay 2. There is no loss of generality, if we assume that the signals transmitted from the terminals take the following form:

$$\mathbf{X}_i(t) = \mathbf{H}_i \boldsymbol{\lambda}_i(t) \tag{11}$$

for $i \in \{A, B\}$, where $\mathbf{H}_i \triangleq [\mathbf{h}_{i1} \ \ \mathbf{h}_{i2}]$ is an $n \times 2$ matrix, and the two components in $\boldsymbol{\lambda}_i(t) \triangleq [\lambda_{i1}(t) \ \ \lambda_{i2}(t)]^T$ represent the projections of $\mathbf{X}_i(t)$ on $\mathbf{h}_{i1}$ and $\mathbf{h}_{i2}$. We consider the 2-dimensional vector $\boldsymbol{\lambda}_i(t)$ as the input to the channel at node i. The power constraint in (5) can be written as

$$\frac{1}{N} \sum_{t=1}^{N} \boldsymbol{\lambda}_i(t)^T \mathbf{H}_i^T \mathbf{H}_i \boldsymbol{\lambda}_i(t) \leq P_i, \tag{12}$$

for $i \in \{A, B\}$.

Notations. We will treat 2×1 random vectors $\boldsymbol{\lambda}_A$ and $\boldsymbol{\lambda}_B$ as input signals at terminal A and B, respectively, and let $\mathbf{K}_A$ and $\mathbf{K}_B$ denote their corresponding 2×2 covariance matrices. For $i \in \{A, B\}$ and $j \in \{1, 2\}$, let

$$\Gamma_j^i \triangleq \frac{\mathbf{h}_{ij}^T \mathbf{H}_i \mathbf{K}_i \mathbf{H}_i^T \mathbf{h}_{ij}}{\sigma^2} \tag{13}$$

be the signal to noise ratio of the signal received at relay j from terminal i. Shannon's capacity formula is denoted by $C(x) \triangleq 0.25 \log_2(1 + x)$. Also, for $n \times n$ matrices, we let $C_n(\mathbf{X}) \triangleq 0.25 \log_2 \det(\mathbf{I}_n + \mathbf{X})$, where $\mathbf{I}_n$ denote the $n \times n$ identity matrix. The reason for the factor of 0.25 before the log function, instead of a factor of 0.5 in the original capacity formula, is due to the fact that the total transmission time is divided into two stages of equal length. All logarithms in this paper are in base 2. The set of non-negative real numbers is denoted by $\mathbb{R}_+$. Gaussian distribution with mean zero and covariance matrix $\mathbf{K}$ is denoted by $\mathcal{N}(0, \mathbf{K})$.

3. Review of Coding Techniques and Capacity Regions from Information Theory

The proposed transmission strategies are based on a host of existing coding techniques and capacity results. A review of them is given in this section.

3.1. Physical-Layer Network Coding. In wireless channel, the channel is inherently additive; the received signal is a linear combination of the transmitted signals. This fact is exploited for the two-way relay channel in [18–21]. Consider the following single-antenna two-way network with two sources and one relay in between. There is no direct link between the two sources, and the exchange of data is done via the relay node in the middle. Let $x_i(t)$ be the transmitted signal from source i, for $i = 1, 2$. The transmission is divided into two phases. In the first phase, the relay receives

$$y(t) = x_1(t) + x_2(t) + z(t), \tag{14}$$

where $z(t)$ is an additive noise. For simplicity, it is assumed that both link gains from the sources to the relay are equal to one. In the second phase, the relay amplifies the received signal $y(t)$, and transmits a scaled version $\zeta y(t)$ of $y(t)$, where ζ is a scalar chosen so that the power requirement is met. Since source 1 knows $x_1(t)$, the component $\zeta x_1(t)$ within the received signal at source 1 can be treated as known interference, and hence be subtracted. Similarly, source 2 can subtract $\zeta x_2(t)$ from the received signal. Decoding is then based on the signal after interference subtraction.

3.2. Multiplexed Coding. Multiplexed coding [22] is a useful coding technique for multi-user scenarios in which some user knows the message of another user *a priori*. Consider the two-way relay channel as in the previous paragraph. Node 1 wants to send message m_1 to node 2 via the relay node, and node 2 wants to send message m_2 to node 1 via the relay node. For $i = 1, 2$, let n_i be the number of bits used to represent message m_i. The transmission of the nodes is divided into two phases. In the first phase, the two source nodes transmit. Suppose that the relay node is able to decode m_1 and m_2. For the encoder at the relay, we generate a $2^{n_1} \times 2^{n_2}$ array of codewords. Each codeword is independently drawn according to the Gaussian distribution such that the total power of each codeword is less than or equal to P. In the second phase, the relay node sends the codeword in the (m_1, m_2)-entry in this array. Suppose that the received signal at source node i is corrupted by additive white Gaussian

noise with variance σ_i^2, for $i = 1, 2$. At source 1, since m_1 is known, the decoder knows that one of the 2^{n_2} codewords in the row corresponding to m_1 had been transmitted. Out of these 2^{n_2} codewords, it then declares the one based on the maximal likelihood criterion. By the channel coding theorem for the point-to-point Gaussian channel, source 1 can decode reliably at a rate of $0.5\log(1+P/\sigma_1^2)$. Likewise, by considering the columns in the array of codewords, source 2 can decode at a rate of $0.5\log(1+P/\sigma_2^2)$.

Multiplexed coding can be implemented using concepts from network coding. We assume, without loss of generality, that $n_2 \geq n_1$. We identify the 2^{n_2} possible messages from source node 2 with the vectors in the n_2-dimensional vector space over the finite field of size 2, $\mathbb{F}_2^{n_2}$, and identify the 2^{n_1} messages from source node 1 with a subspace of $\mathbb{F}_2^{n_2}$ of dimension n_1, say $\mathcal{V}_1$. We generate 2^{n_2} Gaussian codewords independently, one for each vector in $\mathbb{F}_2^{n_2}$. To send messages m_1 and m_2 in the second phase, the relay node transmits the codeword corresponding to $m_1 + m_2$, where the addition is performed using arithmetics in $\mathbb{F}_2^{n_2}$. The output of the decoder at node 1 is a vector in $\mathbb{F}_2^{n_2}$. We subtract from it the vector in $\mathcal{V}_1$ corresponding to m_1. If there is no decoding error, this gives the codeword corresponding to m_2, and the value of m_2 is recovered.

Now let us consider node 2. Since m_2 is known a priori, node 2 is certain that the signal transmitted from the relay is associated with one of the vectors in the affine space $m_2 + \mathcal{V}_1$. The message m_1 can be estimated by comparing the likelihood function of the 2^{n_1} codewords associated with $m_2 + \mathcal{V}_1$. It can be seen that the maximal data rate is the same as in the array approach mentioned in the previous paragraph, but the size of the codebook at the relay reduces from $2^{n_2+n_1}$ to 2^{n_2}.

3.3. Capacity Region for MIMO Channel. Consider a MIMO channel with n_T transmit antennas and n_R receive antennas, with the link gain matrix denoted by a real $n_R \times n_T$ matrix $\mathbf{H}$. The channel output equals

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}, \tag{15}$$

where $\mathbf{X}$ is the n_T-dimensional channel input and $\mathbf{Z}$ is an n_R-dimensional zero-mean colored Gaussian noise vector with covariance matrix $\mathbf{K}_Z$. Without loss of information, we whiten the noise by pre-multiplying both sides of (15) by $\mathbf{K}_Z^{-1/2}$. The transformed channel output is thus

$$\mathbf{Y}' = \mathbf{K}_Z^{-1/2}\mathbf{H}\mathbf{X} + \mathbf{K}_Z^{-1/2}\mathbf{Z}. \tag{16}$$

The covariance matrix of the noise vector $\mathbf{K}_Z^{-1/2}\mathbf{Z}$ is now the $n_R \times n_R$ identity matrix. By the capacity formula for MIMO channel with white Gaussian noise [23], the capacity for the MIMO channel in (15) is given by

$$\frac{1}{2}\log\det\left(\mathbf{I}_{n_R} + \mathbf{K}_Z^{-1/2}\mathbf{H}\mathbf{K}_X\mathbf{H}^T\mathbf{K}_Z^{-1/2}\right), \tag{17}$$

where $\mathbf{K}_X$ denotes the $n_R \times n_R$ covariance matrix of $\mathbf{X}$. Using the identity

$$\det(\mathbf{I}_n + \mathbf{A}\mathbf{B}) \equiv \det(\mathbf{I}_m + \mathbf{B}\mathbf{A}), \tag{18}$$

which holds for any $n \times m$ matrix $\mathbf{A}$ and $m \times n$ matrix $\mathbf{B}$, we rewrite (17) as

$$\frac{1}{2}\log\det\left(\mathbf{I}_{n_T} + \mathbf{H}^T\mathbf{K}_Z^{-1}\mathbf{H}\mathbf{K}_X\right). \tag{19}$$

3.4. Capacity Region for Multiple-Access Channel (MAC). The channel output of the two-user single-antenna Gaussian multiple-access channel is given by

$$y = x_1 + x_2 + z, \tag{20}$$

where x_i is the signal from user i, for $i = 1, 2$, and z is an additive white Gaussian noise with variance σ^2. Each of the two users wants to send some bits to the common receiver. Suppose that the power of user i is limited to P_i, for $i = 1, 2$. The rate pair (R_1, R_2), where R_i is the data rate of user i, is achievable in the above 2-user MAC if and only if it belongs to

$$\mathcal{C}_{\text{mac}}\left(\frac{P_1}{\sigma^2}, \frac{P_2}{\sigma^2}\right) \triangleq \Big\{(R_1, R_2) \in \mathbb{R}_+^2 : \tag{21}$$

$$R_1 \leq 0.5\log_2\left(1 + \frac{P_1}{\sigma^2}\right) \tag{22}$$

$$R_2 \leq 0.5\log_2\left(1 + \frac{P_2}{\sigma^2}\right) \tag{23}$$

$$R_1 + R_2 \leq 0.5\log_2\left(1 + \frac{(P_1 + P_2)}{\sigma^2}\right)\Big\}. \tag{24}$$

We refer the reader to [24] for more details on the optimal coding scheme for MAC.

4. Channel-Network Coding Strategies

We develop five transmission schemes for TWTR network. In the first scheme (AF), the received signals at both relay nodes are amplified and forwarded back to terminals A and B. In the second and third scheme (HLC, HMC), one of the relays employs the amplify forward strategy, while the other decodes the messages from terminals A and B. In the fourth scheme (DF), both relays decode the messages from terminals A and B. In the last strategy (PDF), another mixture of decode-forward and amplify-forward strategy is described.

4.1. Amplify Forward (AF). In this strategy, relay node j ($j \in \{1, 2\}$) buffers the signal received in the first stage, and amplifies it by a factor of ζ_j. The amplified signal

$$X_j(t) = \zeta_j\left(\mathbf{h}_{Aj}^T\mathbf{X}_A(t) + \mathbf{h}_{Bj}^T\mathbf{X}_B(t) + Z_j(t)\right) \tag{25}$$

is then transmitted in the second stage. At the end of the second stage, each terminal, who has the information of itself, subtracts the corresponding term and obtains the desired message from the residual signal.

By putting (25) into (3), we can write the received signal at terminal A as

$$\begin{aligned}\mathbf{Y}_A(t) = &\left(\zeta_1\mathbf{h}_{A1}\mathbf{h}_{A1}^T + \zeta_2\mathbf{h}_{A2}\mathbf{h}_{A2}^T\right)\mathbf{H}_A\boldsymbol{\lambda}_A(t)\\ &+ \left(\zeta_1\mathbf{h}_{A1}\mathbf{h}_{B1}^T + \zeta_2\mathbf{h}_{A2}\mathbf{h}_{B2}^T\right)\mathbf{H}_B\boldsymbol{\lambda}_B(t)\\ &+ \zeta_1\mathbf{h}_{A1}Z_1(t) + \zeta_2\mathbf{h}_{A2}Z_2(t) + \mathbf{Z}_A(t).\end{aligned} \tag{26}$$

Here, we have replaced $\mathbf{X}_A(t)$ and $\mathbf{X}_B(t)$ by their 2-dimensional representations $\mathbf{H}_A\boldsymbol{\lambda}_A(t)$ and $\mathbf{H}_B\boldsymbol{\lambda}_B(t)$. Since terminal A knows its own input $\boldsymbol{\lambda}_A(t)$ as well as the link gains and amplifying factors, the signal component containing $\boldsymbol{\lambda}_A(t)$ as a factor can be subtracted from $\mathbf{Y}_A(t)$. The residual signal is

$$\begin{aligned}&\left(\zeta_1\mathbf{h}_{A1}\mathbf{h}_{B1}^T + \zeta_2\mathbf{h}_{A2}\mathbf{h}_{B2}^T\right)\mathbf{H}_B\boldsymbol{\lambda}_B(t)\\ &\quad+ \zeta_1\mathbf{h}_{A1}Z_1(t) + \zeta_2\mathbf{h}_{A2}Z_2(t) + \mathbf{Z}_A(t).\end{aligned} \tag{27}$$

The message from terminal B can then be decoded using a decoding algorithm for point-to-point MIMO channel. The received signal at terminal B is treated similarly.

Theorem 1. *A rate pair (R_A, R_B) is achievable by the AF strategy if*

$$\begin{aligned}R_A &\le C_2\left(\mathbf{H}_A^T\mathbf{H}_{\text{af}}^T\left(\mathbf{N}_{\text{af}}^B\right)^{-1}\mathbf{H}_{\text{af}}\mathbf{H}_A\mathbf{K}_A\right),\\ R_B &\le C_2\left(\mathbf{H}_B^T\mathbf{H}_{\text{af}}\left(\mathbf{N}_{\text{af}}^A\right)^{-1}\mathbf{H}_{\text{af}}^T\mathbf{H}_B\mathbf{K}_B\right),\end{aligned} \tag{28}$$

where

$$\begin{aligned}\mathbf{N}_{\text{af}}^i &\triangleq \left(\zeta_1^2\mathbf{h}_{i1}\mathbf{h}_{i1}^T + \zeta_2^2\mathbf{h}_{i2}\mathbf{h}_{i2}^T + \mathbf{I}_n\right)\sigma^2, \quad i \in \{A, B\},\\ \mathbf{H}_{\text{af}} &\triangleq \zeta_1\mathbf{h}_{B1}\mathbf{h}_{A1}^T + \zeta_2\mathbf{h}_{B2}\mathbf{h}_{A2}^T,\end{aligned} \tag{29}$$

$\zeta_1, \zeta_2 \in \mathbb{R}$ *and* $\mathbf{K}_A$ *and* $\mathbf{K}_B$ *are* 2×2 *covariance matrices, such that the following power constraints:*

$$\text{Tr}\left(\mathbf{H}_i\mathbf{K}_i\mathbf{H}_i^T\right) \le P_i, \quad \text{for } i = A, B, \tag{30}$$

$$\left(\Gamma_j^A + \Gamma_j^B + 1\right)\zeta_j^2\sigma^2 \le P_j, \quad \text{for } j = 1, 2, \tag{31}$$

are satisfied.

Proof. The residual signal (27) at terminal A can be written as $\mathbf{H}_{\text{af}}^T\mathbf{H}_B\boldsymbol{\lambda}_B(t)$ plus a noise vector with covariance matrix $\mathbf{N}_{\text{af}}^A$. The residual signal at terminal B equals $\mathbf{H}_{\text{af}}\mathbf{H}_A\boldsymbol{\lambda}_A(t)$ plus a noise vector with covariance matrix $\mathbf{N}_{\text{af}}^B$. Therefore, after self-signal subtraction, the resultant channels can be considered MIMO channels with two transmit antennas and n receive antennas. From (19), we obtain the rate constraints in (28). The inequalities in (30) are the power constraints for terminals A and B, and those in (31) are the power constraints for relays 1 and 2. □

4.2. Hybrid Decode-Amplify Forward with Linear Combination (HLC). In this strategy, relay 1 decodes the messages from terminals A and B, and meanwhile, relay 2 employs the amplify-forward strategy. In order to obtain beamforming gain, after decoding the two messages, relay 1 reconstructs the codewords corresponding to the decoded messages and sends a linear combination of them in the second stage.

In the first stage, relay 1 and terminals A and B form a multiple-access channel with relay 1 as the destination node. We use the optimal encoding scheme for MAC at terminals A and B, and the optimal decoding scheme at relay 1. In the second stage, relay 1 decodes and reconstructs $\mathbf{X}_A(t)$ and $\mathbf{X}_B(t)$, and then transmits a linear combination

$$X_1(t) = \mathbf{z}_A^T\mathbf{X}_A(t) + \mathbf{z}_B^T\mathbf{X}_B(t) \tag{32}$$

for some $\mathbf{z}_A$ and $\mathbf{z}_B \in \mathbb{R}^n$. Relay 2 amplifies $Y_2(t)$ by a scalar factor ζ and transmits $X_2(t) = \zeta Y_2(t)$.

At terminal A, after subtracting the signal component that involves $\mathbf{X}_A(\text{t})$, we get

$$\left(\mathbf{h}_{A1}\mathbf{z}_B^T + \zeta\mathbf{h}_{A2}\mathbf{h}_{B2}^T\right)\mathbf{H}_B\boldsymbol{\lambda}_B(t) + \zeta\mathbf{h}_{A2}Z_2(t) + \mathbf{Z}_A(t). \tag{33}$$

At terminal B, the residual signal after subtraction is

$$\left(\mathbf{h}_{B1}\mathbf{z}_A^T + \zeta\mathbf{h}_{B2}\mathbf{h}_{A2}^T\right)\mathbf{H}_A\boldsymbol{\lambda}_A(t) + \zeta\mathbf{h}_{B2}Z_2(t) + \mathbf{Z}_B(t). \tag{34}$$

The decoding is done by using decoding method for MIMO channel.

Theorem 2. *A rate pair (R_A, R_B) is achievable by the HLC strategy if*

$$(R_A, R_B) \in \frac{1}{2}\mathcal{C}_{\text{mac}}\left(\Gamma_1^A, \Gamma_1^B\right), \tag{35}$$

$$R_A \le C_2\left(\left(\mathbf{H}_{\text{hlc}}^A\right)^T\left(\mathbf{N}_{\text{hlc}}^B\right)^{-1}\mathbf{H}_{\text{hlc}}^A\mathbf{K}_A\right), \tag{36}$$

$$R_B \le C_2\left(\left(\mathbf{H}_{\text{hlc}}^B\right)^T\left(\mathbf{N}_{\text{hlc}}^A\right)^{-1}\mathbf{H}_{\text{hlc}}^B\mathbf{K}_B\right), \tag{37}$$

where

$$\begin{aligned}\mathbf{H}_{\text{hlc}}^A &\triangleq \left(\mathbf{h}_{B1}\mathbf{z}_A^T + \zeta\mathbf{h}_{B2}\mathbf{h}_{A2}^T\right)\mathbf{H}_A,\\ \mathbf{H}_{\text{hlc}}^B &\triangleq \left(\mathbf{h}_{A1}\mathbf{z}_B^T + \zeta\mathbf{h}_{A2}\mathbf{h}_{B2}^T\right)\mathbf{H}_B,\\ \mathbf{N}_{\text{hlc}}^i &\triangleq \left(\zeta^2\mathbf{h}_{i2}\mathbf{h}_{i2}^T + \mathbf{I}_n\right)\sigma^2, \quad \text{for } i = A, B,\end{aligned} \tag{38}$$

$\mathbf{z}_A$, $\mathbf{z}_B \in \mathbb{R}^n$, $\zeta \in \mathbb{R}$, *and* $\mathbf{K}_A$ *and* $\mathbf{K}_B$ *are* 2×2 *covariance matrices such that the following power constraints:*

$$\text{Tr}\left(\mathbf{H}_i\mathbf{K}_i\mathbf{H}_i^T\right) \le P_i, \quad \text{for } i = A, B, \tag{39}$$

$$\mathbf{z}_A^T\mathbf{H}_A\mathbf{K}_A\mathbf{H}_A^T\mathbf{z}_A + \mathbf{z}_B^T\mathbf{H}_B\mathbf{K}_B\mathbf{H}_B^T\mathbf{z}_B \le P_1, \tag{40}$$

$$\left(\Gamma_2^A + \Gamma_2^B + 1\right)\zeta^2\sigma^2 \le P_2 \tag{41}$$

are satisfied.

In (35), the product of a real number x and a set $\mathcal{A}$ is defined as $x\mathcal{A} \triangleq \{xa : a \in \mathcal{A}\}$.

Proof. From the rate constraints for MAC channel in (22)–(24), we have the rate constraints for relay 1 in (35). We multiply by a factor of one half because the first phase only occupies half of the total transmission time.

The conditions in (36) and (37) are derived from the capacity formula for MIMO channel with colored noise in (19). The inequalities in (39) are the power constraints for sources A and B. The inequalities in (40) and (41) are the power constraints for relays 1 and 2, respectively. □

The parameters $\mathbf{z}_A$, $\mathbf{z}_B$, $\mathbf{K}_A$, and $\mathbf{K}_B$ can be obtained by running an optimization algorithm. For example, we can aim at maximizing a weighted sum $w_A R_A + w_B R_B$. The values of $\mathbf{z}_A, \mathbf{z}_B, \mathbf{K}_A$ and $\mathbf{K}_B$ which maximize the weighted sum $w_A R_A + w_B R_B$ are chosen.

4.3. Hybrid Decode-Amplify Forward with Multiplexed Coding (HMC). As in the previous strategy, relay 1 decodes and forwards the messages from A and B, and relay 2 amplifies and transmits the received signal. However, in this strategy, relay 1 re-encodes the messages into a new codeword to be sent out in the second stage. Terminals A and B decode the desired messages based on multiplexed coding.

Theorem 3. *A rate pair (R_A, R_B) is achievable by the HMC strategy if R_A and R_B satisfy*

$$(R_A, R_B) \in \frac{1}{2}\, \mathcal{C}_{\text{mac}}\left(\Gamma_1^A, \Gamma_1^B\right), \tag{42}$$

$$R_A \le C_n\left(\mathbf{G}_{\text{hmc}}^A \left(\mathbf{N}_{\text{hmc}}^B\right)^{-1}\right), \tag{43}$$

$$R_B \le C_n\left(\mathbf{G}_{\text{hmc}}^B \left(\mathbf{N}_{\text{hmc}}^A\right)^{-1}\right), \tag{44}$$

where

$$\mathbf{G}_{\text{hmc}}^A \triangleq \mathbf{h}_{B1}\mathbf{h}_{B1}^T P_1 + \zeta^2 \mathbf{h}_{B2}\mathbf{h}_{A2}^T \mathbf{H}_A \mathbf{K}_A \mathbf{H}_A^T \mathbf{h}_{A2}\mathbf{h}_{B2}^T, \tag{45}$$

$$\mathbf{G}_{\text{hmc}}^B \triangleq \mathbf{h}_{A1}\mathbf{h}_{A1}^T P_1 + \zeta^2 \mathbf{h}_{A2}\mathbf{h}_{B2}^T \mathbf{H}_B \mathbf{K}_B \mathbf{H}_B^T \mathbf{h}_{B2}\mathbf{h}_{A2}^T, \tag{46}$$

$$\mathbf{N}_{\text{hmc}}^i \triangleq \left(\zeta^2 \mathbf{h}_{i2}\mathbf{h}_{i2}^T + \mathbf{I}_n\right)\sigma^2, \quad \text{for } i \in \{A, B\}, \tag{47}$$

$\mathbf{K}_A, \mathbf{K}_B$ *are* 2×2 *covariance matrices satisfying* (39), *and* $\zeta \in \mathbb{R}$ *satisfies* (41).

Proof. The proof is by random coding argument and we will sketch the proof below. More details can be found in [25].

Our objective is to show that any rate pair (R_A, R_B) that satisfies the condition in the theorem is achievable. For $i = A, B$, terminal i randomly generates a Gaussian codebook with 2^{2NR_i} codewords with length N, satisfying the power constraint in (5). Label the codewords by $\mathbf{X}_i^N(m_i)$, for $m_i \in M_i$. For relay 1, we generate a $2^{2NR_A} \times 2^{2NR_B}$ array of Gaussian codewords of length N and power P_1. The codeword in row m_A and column m_B is denoted by $X_1^N(m_A, m_B)$, and satisfies the power constraint in (6).

After the first stage, relay 1 is required to decode both messages from terminals A and B. This can be accomplished with arbitrarily small probability of error if the rate constraints for MAC in (22) to (24) are satisfied. This corresponds to the rate constraint in (42). Let the estimated messages from A and B be $\hat{m}_A$ and $\hat{m}_B$.

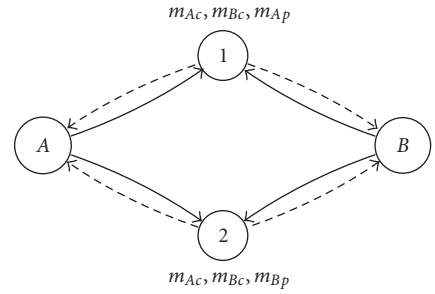


Figure 2: Decoded messages at the two-relays in the DF strategy.

In the second stage, relay 1 transmits $X_1^N(\hat{m}_A, \hat{m}_B)$. Relay 2 amplifies its received signal and transmits $\zeta Y_2(t)$. From (41), the amplified signal has average power no more than P_2.

After subtracting the term $\zeta \mathbf{h}_{A2}\mathbf{h}_{A2}^T \mathbf{X}_A(t)$, which is known to terminal A, the residual signal at terminal A is

$$\left[\mathbf{h}_{A1} X_1(\hat{m}_A, \hat{m}_B)(t) + \zeta \mathbf{h}_{A2}\mathbf{h}_{B2}^T \mathbf{X}_B(t)\right] + \zeta \mathbf{h}_{A2} Z_2(t) + \mathbf{Z}_A(t). \tag{48}$$

Note that terminal A knows its message m_A, and $\hat{m}_A = m_A$ with probability arbitrarily close to one if (42) is satisfied. The idea of multiplexed coding can then be used. In (48), the covariance matrix of the signal in square bracket is given by $\mathbf{G}_{\text{hmc}}^B$ in (46), and the covariance of the noise term is given by $\mathbf{N}_{\text{hmc}}^A$. Applying the capacity expression, we obtain the rate constraint in (44). In a similar manner, we obtain (43). □

4.4. Decode Forward (DF). In the DF strategy, terminal node i, $(i \in \{A, B\})$ splits the message m_i into two parts: the common part m_{ic} and the private part m_{ip}. The two common messages are transmitted via both relay nodes. The private message m_{Ap} is decoded by relay 1 only, and can be interpreted as going through the path from terminal A to relay 1 to terminal B. Symmetrically, the private part of message m_{Bp} is decoded by relay 2 only, and can be interpreted as going through the path from terminal B to relay 2 to terminal A. After the first stage, relay 1 decodes the common messages of both terminals and the private message of terminal A. Relay 2 decodes the common messages of both terminals and the private message of terminal B. The encoding and decoding schemes in the first stage is similar to those developed by Han and Kobayashi for the interference channel (IC) in [26]. Since both relays have access to the common messages, the channel in the second stage can be considered a multiple access channel with common information. Furthermore, since terminals A and B have information of themselves, we can further improve the rate region by the idea of multiplexed coding.

We have the following characterization of the rate region for the DF strategy:

Theorem 4. *For $i \in \{A,B\}$, let R_{ip} and R_{ic} be the rates of the private and common messages, respectively, from terminal i. Let Γ_j denote P_j/σ^2 for $j = 1,2$, and let $\mathbf{K}_{Ac}$, $\mathbf{K}_{Ap}$, $\mathbf{K}_{Bc}$, and $\mathbf{K}_{Bp}$ denote 2×2 covariance matrices, and*

$$\Gamma_j^{ik} \triangleq \frac{\mathbf{h}_{ij}^T \mathbf{H}_i \mathbf{K}_{ik} \mathbf{H}_i^T \mathbf{h}_{ij}}{\sigma^2} \tag{49}$$

for $i \in \{A,B\}$, $j \in \{1,2\}$ and $k \in \{p,c\}$. For $j = 1,2$. A rate pair (R_A, R_B) is achievable if we can decompose $R_A = R_{Ap} + R_{Ac}$ and $R_B = R_{Bp} + R_{Bc}$ such that

$$(R_A, R_{Bc}) \in \frac{1}{2}\mathcal{C}_{\text{mac}}\left(\frac{\Gamma_1^{Ap}+\Gamma_1^{Ac}}{\Gamma_1^{Bp}+1}, \frac{\Gamma_1^{Bc}}{\Gamma_1^{Bp}+1}\right), \tag{50}$$

$$(R_{Ac}, R_B) \in \frac{1}{2}\mathcal{C}_{\text{mac}}\left(\frac{\Gamma_2^{Ac}}{\Gamma_2^{Ac}+1}, \frac{\Gamma_2^{Bp}+\Gamma_2^{Bc}}{\Gamma_2^{Ac}+1}\right), \tag{51}$$

$$R_{Ap} \le C\left(\overline{\alpha}_1 \|\mathbf{h}_{B1}\|^2 \Gamma_1\right), \tag{52}$$

$$\begin{aligned} R_A \le C_n\Big(&\Gamma_1 \mathbf{h}_{B1}\mathbf{h}_{B1}^T + \Gamma_2 \mathbf{h}_{B2}\mathbf{h}_{B2}^T \\ &+\sqrt{\alpha_1\alpha_2\Gamma_1\Gamma_2}\left(\mathbf{h}_{B1}\mathbf{h}_{B2}^T + \mathbf{h}_{B2}\mathbf{h}_{B1}^T\right)\Big), \end{aligned} \tag{53}$$

$$R_{Bp} \le C\left(\overline{\alpha}_2 \|\mathbf{h}_{A2}\|^2 \Gamma_2\right), \tag{54}$$

$$\begin{aligned} R_B \le C_n\Big(&\Gamma_1 \mathbf{h}_{A1}\mathbf{h}_{A1}^T + \Gamma_2 \mathbf{h}_{A2}\mathbf{h}_{A2}^T \\ &+\sqrt{\alpha_1\alpha_2\Gamma_1\Gamma_2}\left(\mathbf{h}_{A1}\mathbf{h}_{A2}^T + \mathbf{h}_{A2}\mathbf{h}_{A1}^T\right)\Big), \end{aligned} \tag{55}$$

$$\text{Tr}\left(\mathbf{H}_A\left(\mathbf{K}_{Ac}+\mathbf{K}_{Ap}\right)\mathbf{H}_A^T\right) < P_A, \tag{56}$$

$$\begin{aligned} &\text{Tr}\left(\mathbf{H}_B\left(\mathbf{K}_{Bc}+\mathbf{K}_{Bp}\right)\mathbf{H}_B^T\right) < P_B, \\ &\alpha_1 + \overline{\alpha}_1 < 1, \quad \alpha_2 + \overline{\alpha}_2 < 1 \end{aligned} \tag{57}$$

for some nonnegative α_j and $\overline{\alpha}_j$.

Details of the DF coding scheme and the proof of Theorem 4 are given in the Appendix.

4.5. Partial Decode Forward (PDF). In the PDF strategy, both relays decode the message of terminal A. Each relay then subtracts the reconstructed signal of terminal A from the received signal. Call the resulting signal the residual signal. The message of terminal A is re-encoded into a new codeword, and linearly combined with the residual signal. This linear combination is then transmitted in the second stage. Since both relays know the message of terminal A, the two-relays can jointly re-encode the message of terminal A using some encoding scheme for a MIMO channel with two transmit antennas and n receive antennas.

Theorem 5. *A rate pair (R_A, R_B) is achievable by the PDF strategy if it satisfies*

$$R_A \le \min\left\{C\left(\frac{\Gamma_1^A}{\left(\Gamma_1^B+1\right)}\right), C\left(\frac{\Gamma_2^A}{\left(\Gamma_2^B+1\right)}\right)\right\}, \tag{58}$$

$$R_A \le C_2\left((\mathbf{H}_B)^T\left(\mathbf{N}_{\text{pdf}}^B\right)^{-1}\mathbf{H}_B\mathbf{K}_R\right), \tag{59}$$

$$R_B \le C_2\left(\left(\mathbf{H}_{\text{pdf}}^B\right)^T\left(\mathbf{N}_{\text{pdf}}^A\right)^{-1}\mathbf{H}_{\text{pdf}}^B\mathbf{K}_B\right), \tag{60}$$

where

$$\begin{aligned} \mathbf{N}_{\text{pdf}}^i &\triangleq \left(\zeta_1^2 \mathbf{h}_{i1}\mathbf{h}_{i1}^T + \zeta_2^2 \mathbf{h}_{i2}\mathbf{h}_{i2}^T + \mathbf{I}_n\right)\sigma^2, \\ \mathbf{H}_{\text{pdf}}^B &\triangleq \left(\zeta_1 \mathbf{h}_{A1}\mathbf{h}_{B1}^T + \zeta_2 \mathbf{h}_{A2}\mathbf{h}_{B2}^T\right)\mathbf{H}_B, \end{aligned} \tag{61}$$

and $\zeta_j \in \mathbb{R}$ and $\mathbf{K}_A, \mathbf{K}_B, \mathbf{K}_R$ are 2×2 covariance matrices such that the following power constraints hold

$$\text{Tr}\left(\mathbf{H}_i\mathbf{K}_i\mathbf{H}_i^T\right) \le P_i, \quad \text{for } i = A,B, \tag{62}$$

$$\mathbf{K}_R(j,j) + \left(\Gamma_j^B+1\right)\sigma^2\zeta_j^2 \le P_j \tag{63}$$

for $j = 1,2$. (Here, $\mathbf{K}_R(j,j)$ denotes the jth diagonal entry in $\mathbf{K}_R$.)

Proof. The two-relays treat the signal originated from terminal B as noise, and decode the message of terminal A. The rate requirement in (58) guarantees that the message of terminal A can be decoded with arbitrarily small probability of error at both relays. Let the decoded message of terminal A be denoted by $\hat{m}_A$.

For $j = 1,2$, the reconstructed signal $\mathbf{h}_{Aj}^T\mathbf{X}_A(t)$ is then subtracted from $Y_j(t)$. The residual signal at relay j is $\mathbf{h}_{Bj}^T\mathbf{X}_B(t) + Z_j(t)$.

At the relays, we employ two Gaussian codebooks for the re-encoding of the message from terminal A. For each message m_A, we generate two correlated codewords $U_{1,m_A}(t)$ and $U_{2,m_A}(t)$, with mean zero and each pair of symbols at any t distributed according to a 2×2 covariance matrix $\mathbf{K}_R$. At relay j, the decoded message $\hat{m}_A$ is re-encoded into $U_{j,\hat{m}_A}(t)$, which is a codeword with power $\mathbf{K}_R(j,j)$. In the second stage, relay j transmits

$$U_{j,\hat{m}_A}(t) + \zeta_j\left(\mathbf{h}_{Bj}^T\mathbf{X}_B(t) + Z_j(t)\right), \tag{64}$$

for some amplifying factor ζ_j. The inequality in (63) ensures that the power constraint is satisfied at the relays.

At the end of stage 2, terminal A subtracts the signal component that involves U_{1,m_A} and U_{2,m_A} from its received signal and obtains

$$\mathbf{H}_{\text{pdf}}^B\boldsymbol{\lambda}_B(t) + \zeta_1\mathbf{h}_{A1}Z_1(t) + \zeta_2\mathbf{h}_{A2}Z_2(t) + \mathbf{Z}_A(t). \tag{65}$$

From the capacity formula for MIMO channel (19), terminal A can recover the message from terminal B reliably if (60) is satisfied.

For the decoding in terminal B, we subtract all terms involving $\mathbf{X}_B(t)$, and get

$$\mathbf{H}_B \begin{bmatrix} U_{1,\hat{m}_A}(t) \\ U_{2,\hat{m}_A}(t) \end{bmatrix} + \zeta_1 \mathbf{h}_{B1} Z_1(t) + \zeta_2 \mathbf{h}_{B2} Z_2(t) + \mathbf{Z}_B(t). \quad (66)$$

This is equivalent to a MIMO channel with link gain matrix $\mathbf{H}_B$ and colored noise. Recall that $\mathbf{K}_R$ is the covariance matrix of the encoded signal. By the capacity formula of MIMO channel (19), we obtain the rate constraint in (59). □

Remark 1. We note that the matrices $\mathbf{N}^i_{\text{af}}$, $\mathbf{N}^i_{\text{hlc}}$, $\mathbf{N}^i_{\text{hmc}}$ and $\mathbf{N}^i_{\text{pdf}}$, for $i = A, B$, are invertible. Indeed, by checking that $\mathbf{v}^T\mathbf{N}\mathbf{v}$ is strictly positive for all non-zero $\mathbf{v} \in \mathbb{R}^n$, we see that the matrix is positive definite, and hence invertible.

5. Performance in High SNR Regime

In this section, we compare the performance of the five strategies described in the previous section in the high Signal-to-Noise Ratio (SNR) regime.

For fixed powers and link gains, let $C_{\text{sum}}(\sigma^2)$ denote the sum rate $R_A + R_B$ as a function of the noise variance σ^2. We use the *multiplexing gain* (also called *degree of freedom*) [27], defined by

$$M \triangleq \lim_{\sigma^2 \to 0} \frac{C_{\text{sum}}(\sigma^2)}{(1/2)\log(\sigma^{-2})}, \quad (67)$$

as the performance measure at high SNR. At high SNR, that is, when σ^2 is very small, we can approximate the sum rate by $(M/2)\log(\sigma^{-2})$ if the multiplexing gain is equal to M.

Consider the multiplexing gain of the AF scheme. When the sum rate $R_A + R_B$ is maximized subject to the rate constraints (28) in Theorem 1, the equalities in (28) hold. We can assume without loss of generality that

$$R_A = C_2\left(\mathbf{H}_A^T\mathbf{H}_{\text{af}}^T\left(\mathbf{N}_{\text{af}}^B\right)^{-1}\mathbf{H}_{\text{af}}\mathbf{H}_A\mathbf{K}_A\right), \quad (68)$$

$$R_B = C_2\left(\mathbf{H}_B^T\mathbf{H}_{\text{af}}\left(\mathbf{N}_{\text{af}}^A\right)^{-1}\mathbf{H}_{\text{af}}^T\mathbf{H}_B\mathbf{K}_B\right). \quad (69)$$

We first suppose that the covariance matrices $\mathbf{K}_A$ and $\mathbf{K}_B$, and the amplifying constants ζ_1 and ζ_2, are fixed. Note that if the power constraint in (31) holds, then it continues to hold if σ^2 becomes smaller. Therefore, when $\sigma^2 \to 0$, the power constraints in (30) and (31) are satisfied.

Each of the expressions in (68) and (69) can be written in the form

$$\frac{1}{4}\log\det\left(\mathbf{I}_2 + \frac{\mathbf{M}}{\sigma^2}\right), \quad (70)$$

where $\mathbf{M}$ is a 2×2 matrix that equals

$$\mathbf{H}_A^T\mathbf{H}_{\text{af}}^T\left(\mathbf{N}_{\text{af}}^B\right)^{-1}\mathbf{H}_{\text{af}}\mathbf{H}_A\mathbf{K}_A, \quad \text{or} \quad (71)$$

$$\mathbf{H}_B^T\mathbf{H}_{\text{af}}\left(\mathbf{N}_{\text{af}}^A\right)^{-1}\mathbf{H}_{\text{af}}^T\mathbf{H}_B\mathbf{K}_B. \quad (72)$$

By singular value decomposition [28, Chapter 7], we can factor $\mathbf{M}$ as $\mathbf{U}\mathbf{\Lambda}\mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ are 2×2 unitary matrices, and $\mathbf{\Lambda} = [\lambda_{ij}]$ is a diagonal matrix with non-negative diagonal entries $\lambda_{11} \geq \lambda_{22} \geq 0$. The number of positive diagonal entries in $\mathbf{\Lambda}$ is precisely the rank of $\mathbf{M}$. We can rewrite (70) as

$$\frac{1}{4}\log\det\left(\mathbf{U}^{-1}\mathbf{V}^{-1} + \frac{\mathbf{\Lambda}}{\sigma^2}\right). \quad (73)$$

Suppose that $\mathbf{U}^{-1}\mathbf{V}^{-1}$ is equal to $[a_{ij}]_{i,j=1}^2$. The determinant

$$\begin{vmatrix} a_{11} + \dfrac{\lambda_{11}}{\sigma^2} & a_{12} \\ a_{21} & a_{22} + \dfrac{\lambda_{22}}{\sigma^2} \end{vmatrix} \quad (74)$$

in (73) can be expanded as a polynomial in σ^{-2}, with the degree equal to the rank of $\mathbf{M}$. Therefore, the limit

$$\lim_{\sigma^2 \to 0} \frac{(1/4)\log\det(\mathbf{I}_2 + \mathbf{M}/\sigma^2)}{(1/2)\log(\sigma^{-2})} \quad (75)$$

depends only on the rank of the matrix $\mathbf{M}$, and equals 0, 0.5, or 1, if the rank of $\mathbf{M}$ is 0, 1, or 2, respectively. The problem of determining the multiplexing gain now reduces to determining the rank of the matrices in (71) and (72).

Recall that the rank function satisfies the following properties [28, page 13]: (i) if $\mathbf{A}$ and $\mathbf{C}$ are square invertible matrices, then $\text{rank}(\mathbf{ABC}) = \text{rank}(\mathbf{B})$ for all matrix $\mathbf{B}$, whenever the matrix multiplications are well-defined; (ii) for all $m \times n$ matrices $\mathbf{A}$, we have $\text{rank}(\mathbf{A}^T\mathbf{A}) = \text{rank}(\mathbf{A})$. Consider the matrix in (72). After replacing $\mathbf{H}_{\text{af}}$ by its definition, we can express the matrix in (72) as

$$\mathbf{H}_B^T\mathbf{H}_B\mathbf{Z}\mathbf{H}_A^T\left(\mathbf{N}_{\text{af}}^A\right)^{-1}\mathbf{H}_A\mathbf{Z}\mathbf{H}_B^T\mathbf{H}_B\mathbf{K}_B, \quad (76)$$

where $\mathbf{Z}$ denotes the diagonal matrix $\text{diag}(\zeta_1^2, \zeta_2^2)$. We assume that $\mathbf{H}_A$ and $\mathbf{H}_B$ have full rank. This assumption holds with probability one if the link gains are generated from a continuous probability distribution function such as Rayleigh. Also, we assume that $\mathbf{Z}$, $\mathbf{K}_A$, and $\mathbf{K}_B$ are of full rank. This assumption does not incur any loss of generality, because they are design parameters that we can choose. We can perturb them infinitesimally, and the resulting matrices will be of rank two, but the value on the right hand side of (69) deviates negligibly. By property (i), and the fact that $\mathbf{H}_B^T\mathbf{H}_B$, $\mathbf{Z}$, and $\mathbf{K}_B$ are invertible 2×2 matrices, the rank of the matrix in (76) is equal to the rank of $\mathbf{H}_A^T(\mathbf{N}_{\text{af}}^A)^{-1}\mathbf{H}_A$. Then we get

$$\begin{aligned} &\text{rank}\left(\mathbf{H}_A^T\left(\mathbf{N}_{\text{af}}^A\right)^{-1}\mathbf{H}_A\right) \\ &\quad = \text{rank}\left(\mathbf{H}_A^T\left(\mathbf{N}_{\text{af}}^A\right)^{-1/2}\left(\mathbf{N}_{\text{af}}^A\right)^{-1/2}\mathbf{H}_A\right) \\ &\quad = \text{rank}\left(\left(\mathbf{N}_{\text{af}}^A\right)^{-1/2}\mathbf{H}_A\right) \quad \text{(by Property (ii))} \\ &\quad = \text{rank}(\mathbf{H}_A) \quad \text{(by Property (i))} \\ &\quad = 2. \end{aligned} \quad (77)$$

Similarly, we can show that the rank of the matrix in (71) is equal to two.

For fixed invertible covariance matrices $\mathbf{K}_A$ and $\mathbf{K}_B$, and positive real numbers ζ_1 and ζ_2,

$$\lim_{\sigma^2 \to 0} \frac{\text{R.H.S. of (69)} + \text{R.H.S. of (69)}}{0.5\log(\sigma^{-2})} = 2. \tag{78}$$

Since the above argument holds for all invertible $\mathbf{K}_A$ and $\mathbf{K}_B$, and positive ζ_1 and ζ_2, we conclude that the multiplexing gain of the AF strategy is equal to 2.

For HLC and HMC, relay 1 is required to decode the messages of the terminals, and in both schemes the sum rate is subject to the sum rate constraint in the MAC channel in the first phase. The multiplexing gains of both the HLC and HMC strategies are limited by

$$\lim_{\sigma^2 \to 0} \frac{C\left(\Gamma_1^A + \Gamma_1^B\right)}{0.5\log(\sigma^{-2})} = 0.5. \tag{79}$$

Similarly, the multiplexing gain of DF is also limited by the decoding of messages at the relays. The rate constraints (50) and (51) imply that it is no more than 0.5.

The multiplexing gain of the PDF scheme is somewhere in between the multiplexing gains of AF and DF. The transmission from terminal B to terminal A can be considered AF, while the transmission from terminal A to terminal B in the other direction is limited by the message decoding after stage 1. From (58), we get

$$\lim_{\sigma^2 \to 0} \frac{R_A(\sigma^2)}{0.5\log(\sigma^{-2})} \le 0.5, \tag{80}$$

and from (60), we have

$$\lim_{\sigma^2 \to 0} \frac{R_B(\sigma^2)}{0.5\log(\sigma^{-2})} = \frac{1}{2}\operatorname{rank}(\mathbf{H}_A) = 1, \tag{81}$$

provided that the $\mathbf{H}_A$ has full rank. Therefore, its maximal multiplexing gain is 1.5.

We summarize the performance of the five schemes at high SNR in Table 1. We can see that the AF strategy has the highest multiplexing gain. It is well known that the maximal multiplexing gain of the Gaussian MIMO channel with two transmit antennas and two received antennas is equal to two [23]. We see that at high SNR, the AF strategy behaves like a transmission scheme achieving full multiplexing gain in the MIMO channel with two transmit antennas and two received antennas.

6. Numerical Examples

We compare the information rates achievable by the proposed strategies in Section 4 with the cut-set outer bound in [29]. Since the derivation is straightforward, we state the outer bound without proof. For $i, j \in \{1, 2\}$, and $k \in \{A, B\}$, let

$$\Gamma_{ij}^k \triangleq \frac{\mathbf{h}_{ki}^T \mathbf{H}_k \mathbf{K}_k \mathbf{H}_k^T \mathbf{h}_{kj}}{\sigma^2}. \tag{82}$$

Theorem 6 (Outer bound). *A rate pair (R_A, R_B) is achievable in the TWTR network only if it satisfies*

$$\begin{aligned}
R_A \le \min\Big\{ & C\left(\Gamma_1^A + \Gamma_2^A + \Gamma_1^A\Gamma_2^A - \Gamma_{12}^A\Gamma_{21}^A\right), \\
& C\left(\Gamma_2^A\right) + C_n\left(\mathbf{h}_{B1}\mathbf{h}_{B1}^T(1-\rho^2)\Gamma_1\right), \\
& C\left(\Gamma_1^A\right) + C_n\left(\mathbf{h}_{B2}\mathbf{h}_{B2}^T(1-\rho^2)\Gamma_2\right), \\
& C_n\Big(\mathbf{h}_{B1}\mathbf{h}_{B1}^T\Gamma_1 + \mathbf{h}_{B2}\mathbf{h}_{B2}^T\Gamma_2 \\
& \qquad + \rho\left(\mathbf{h}_{B1}\mathbf{h}_{B2}^T + \mathbf{h}_{B2}\mathbf{h}_{B1}^T\right)\sqrt{\Gamma_1\Gamma_2}\Big)\Big\}, \\
R_B \le \min\Big\{ & C\left(\Gamma_1^B + \Gamma_2^B + \Gamma_1^B\Gamma_2^B - \Gamma_{12}^A\Gamma_{21}^B\right), \\
& C\left(\Gamma_2^B\right) + C_n\left(\mathbf{h}_{A1}\mathbf{h}_{A1}^T(1-\rho^2)\Gamma_1\right), \\
& C\left(\Gamma_1^B\right) + C_n\left(\mathbf{h}_{A2}\mathbf{h}_{A2}^T(1-\rho^2)\Gamma_2\right), \\
& C_n\Big(\mathbf{h}_{A1}\mathbf{h}_{A1}^T\Gamma_1 + \mathbf{h}_{A2}\mathbf{h}_{A2}^T\Gamma_2 \\
& \qquad + \rho\left(\mathbf{h}_{A1}\mathbf{h}_{A2}^T + \mathbf{h}_{A2}\mathbf{h}_{A1}^T\right)\sqrt{\Gamma_1\Gamma_2}\Big)\Big\},
\end{aligned} \tag{83}$$

for some real number ρ between 0 and 1, and 2×2 covariance matrices $\mathbf{K}_A$ and $\mathbf{K}_B$ such that $\operatorname{Tr}(\mathbf{H}_i\mathbf{K}_i\mathbf{H}_i^T) \le P_i$ holds for $i = A, B$.

We select several typical channel realizations and show the corresponding achievable rate regions in Figure 3 to Figure 8. To simplify the calculation, we consider the single antenna case where $n = 1$. The power constraint is set to $P = 1$ and the noise variance is set to $\sigma^2 = 1$.

In Figure 3, we plot the rate regions when all link gains are large (the link gain is 10 for all links). As mentioned in the previous section, the AF strategy has the largest multiplexing gain in the high SNR regime. We can see in Figure 3 that the AF strategy achieves the largest sum rate.

In Figures 4 and 5, we consider the case where relay 1 has larger link gains than relay 2. In Figure 4, the link gains h_{A1} and h_{B1} are the same. In this case, HMC dominates all other strategies. In Figure 5, the two link gains, h_{A1} and h_{B1}, are not equal. In this case, HLC dominates HMC. HLC performs better in this asymmetric case because of its ability to adjust power between signals and utilize the beamforming gain.

When both relays are close to one of the terminals, PDF has the best performance, as can be seen in Figure 6. The reason is that both relays are able to decode reliably the message from the closer terminal, and then they cooperatively forward the message to the other terminal using MIMO techniques.

Figures 7 and 8 presents two scenarios in which DF dominates all other transmission strategies. We remark that DF is quite flexible in that it has many tunable parameters. The case where both h_{A1} and h_{B2} are relatively large is shown in Figure 7. Another case where h_{A1} and h_{A2} are larger than h_{B1} and h_{B2} is shown in Figure 8. In both cases, DF is much better than other strategies.

We can further summarize the numerical results in Table 2. It is not supposed to be a precise description on the

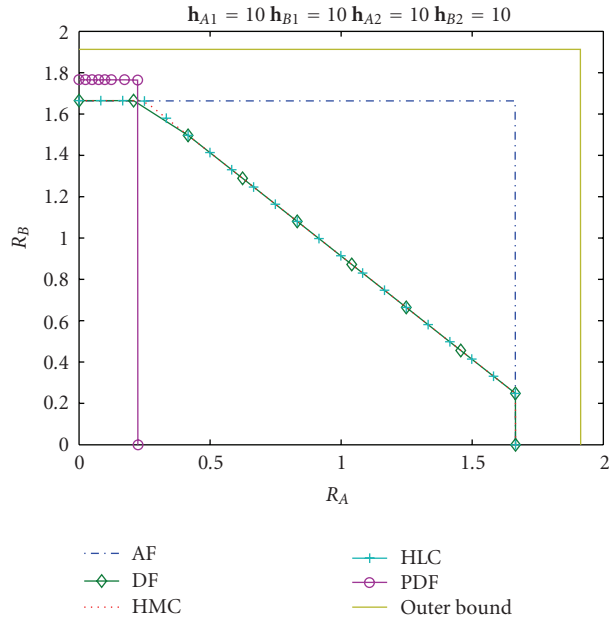


Figure 3: The achievable rate regions when all link gains are large.

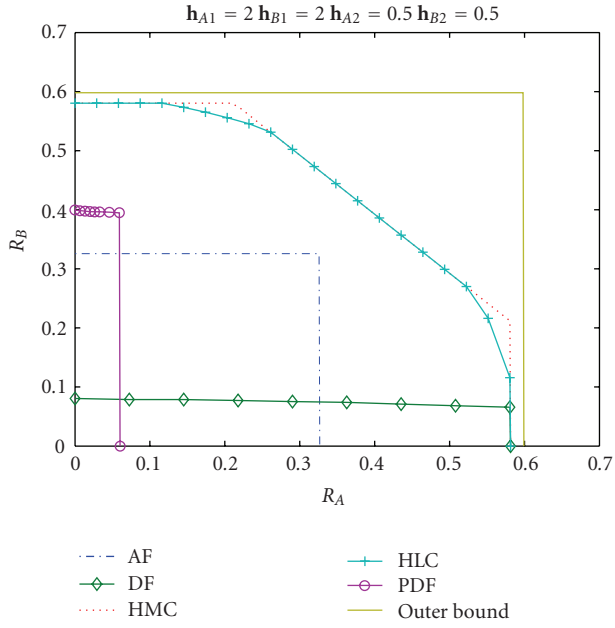


Figure 4: The achievable rate regions when one relay has large link gains (symmetric case).

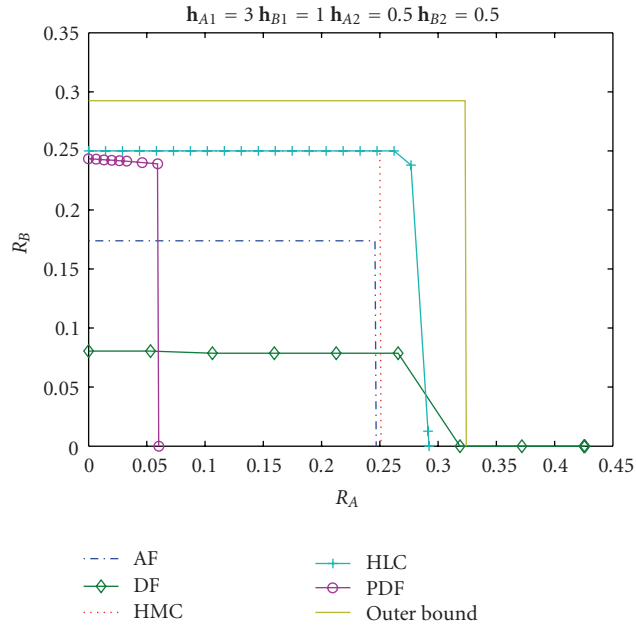


Figure 5: The achievable rate regions when one relay has large link gains (symmetric case).

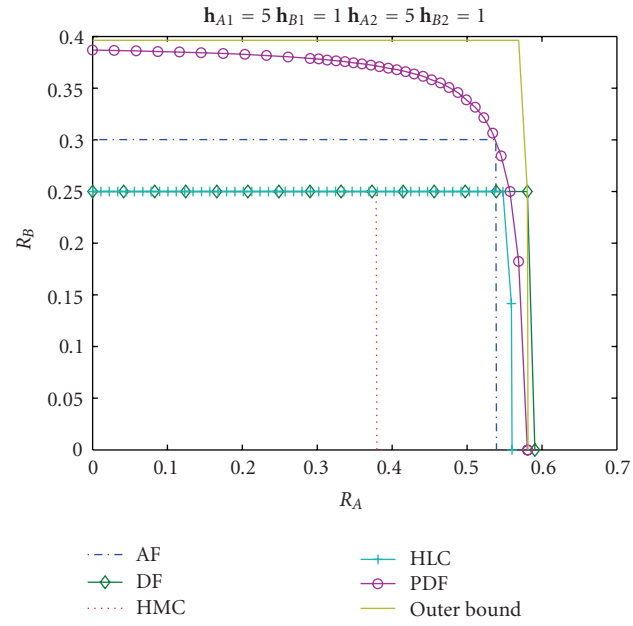


Figure 6: The achievable rate regions when both relays are close to terminal A.

relative merits of the schemes. Instead, it provides a rough guideline for easy selection of a suitable scheme. In the table, "G" refers to "the channel condition is good" and "B" refers to "the channel condition is bad." We say that a channel is good if its link gain is two to three times, or more, than the link gain of a bad channel. When all the link gains are large, we should use AF. In the case when one pair of the opposite links of the network is good, whereas the other pair is weak, DF provides larger throughput. If one of the relays is good but the other relay is bad, HMC or HLC should be used.

Table 1: Multiplexing gains of the transmission schemes in the high SNR regime.

Scheme	AF	HMC, HLC, DF	PDF
Multiplexing gain	2	0.5	1.5

PDF scheme is the best one in the scenario where one of the sources has large link gains but the other does not.

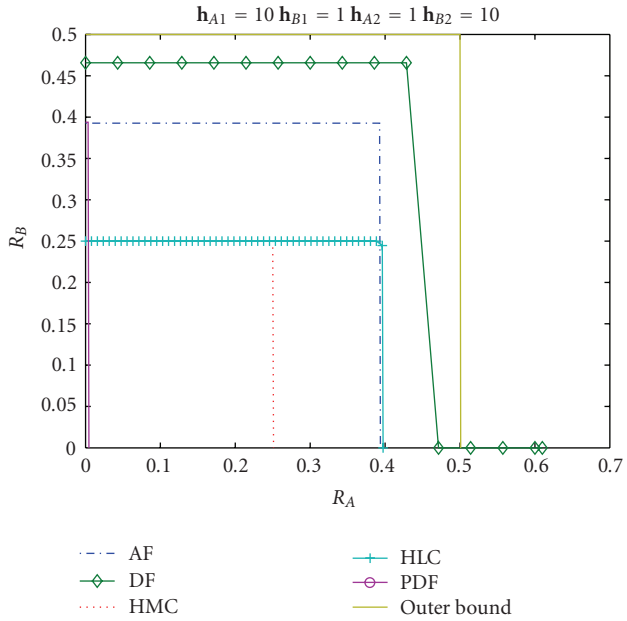


Figure 7: The achievable rate regions and the outer bound.

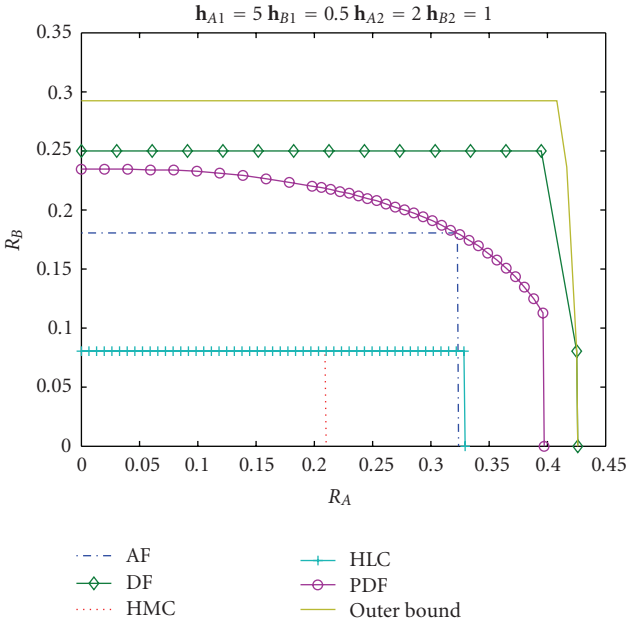


Figure 8: The achievable rate regions and the outer bound.

Table 2: Performance guideline for the two-way two-relay network in the medium SNR regime.

$\|\mathbf{h}_{A1}\|$	$\|\mathbf{h}_{B1}\|$	$\|\mathbf{h}_{A2}\|$	$\|\mathbf{h}_{B2}\|$	Scheme
G	G	G	G	AF
G	B	B	G	DF
G	G	B	B	HMC, HLC
G	B	G	B	PDF

7. Conclusion

We have devised several transmission strategies for the TWTR network, each of which is derived from a mix-and-match of several basic building blocks, namely, amplify-forward strategy, decode-forward strategy, and physical-layer network coding, and so forth. We can see from the numerical examples that there is no single transmission strategy that can dominate all other strategies under all channel realizations. In other words, transmission strategy should be tailor-made for a given environment. In this paper, we have investigated the pros and cons of different building blocks and demonstrated how they can be used to construct transmission strategies for the TWTR network. We believe that the idea can be applied to other relay networks as well.

While in this paper we only consider the case where there are only two-relays, the ideas of our proposed schemes can be applied to the case with more than two-relays. In particular, AF and PDF can be directly implemented without any change. As for DF, HMC, and HLC, the design may be more complicated, since we have to determine which relay to decode which source's message. On the other hand, the idea behind remains the same.

In our work, we have assumed that the channels are static. When link gains are time varying, our result reveals that a static strategy can only be suboptimal. To fully exploit the available capacity of the network, adaptive strategies that can switch between several modes are needed. How to determine a good strategy based on channel state information is an open problem. It is especially difficult if the switching is based on local information only, and we leave it for future work.

Appendix

Proof of Theorem 4

The following information-theoretic argument shows that any rate pair (R_A, R_B) satisfying the conditions in Theorem 4 is achievable.

Codebook Generation. For $i = A, B$, the common message of terminal i is drawn uniformly in $M_{ic} \triangleq \{1, 2, \ldots, 2^{2NR_{ic}}\}$ and the private message from $M_{ip} \triangleq \{1, 2, \ldots, 2^{2NR_{ip}}\}$. For $i = A, B$, we generate $2^{2NR_{ic}}$ independent sequences of length N. In each sequence, the components are 2×1 vectors drawn independently with distribution $\mathcal{N}(0, \mathbf{K}_{ic})$. Label the generated sequences by $\mathbf{U}_i^N(m_{ic})$ for $m_{ic} \in M_{ic}$. Generate $2^{2NR_{ip}}$ independent sequences of length N, with each component drawn independently with distribution $\mathcal{N}(0, \mathbf{K}_{ip})$. Label the generated sequences by $\mathbf{W}_i^N(m_{ip})$ for $m_{ip} \in M_{ip}$. Set

$$\mathbf{X}_i^N\left(m_{ic}, m_{ip}\right) = \mathbf{H}_i\left(\mathbf{U}_i^N(m_{ic}) + \mathbf{W}_i^N\left(m_{ip}\right)\right). \tag{A.1}$$

By (56) and (57), with very high probability the power constraints on node A and node B are satisfied.

There is a common codebook for relay 1 and relay 2. We generate an array of codewords with $2^{2NR_{A_c}}$ rows and $2^{2NR_{B_c}}$

columns. The codewords have length N and each component is drawn independently from $\mathcal{N}(0,1)$. Label the codewords by $V_0^N(m_{Ac}, m_{Bc})$, for $m_{Ac} \in M_{Ac}$ and $m_{Bc} \in M_{Bc}$.

For relay 1, we generate $2^{2N(R_{Ap}+R_{Ac}R_{Bc})}$ codewords, indexed by $m_{Ap} \in M_{Ap}$, $m_{Ac} \in M_{Ac}$, $m_{Bc} \in M_{Bc}$, and denoted by

$$\widetilde{X}_1^N\left(m_{Ap}, m_{Ac}, m_{Bc}\right). \tag{A.2}$$

Each of them is drawn independently with each component generated from $\mathcal{N}(0, \overline{\alpha}_1 P_1)$. Let $X_1^N(m_{Ac}, m_{Bc}, m_{Ap})$ be the linear combination

$$\sqrt{\alpha_1 P_1}\, V_0^N(m_{Ac}, m_{Bc}) + \widetilde{X}_1^N\left(m_{Ap}, m_{Ac}, m_{Bc}\right). \tag{A.3}$$

Since $\alpha_1 + \overline{\alpha}_1$ is strictly less than 1, $X_1^N(m_{Ac}, m_{Bc}, m_{Ap})$ satisfies the power constraint of node 1 with very high probability.

For relay 2, we generate $2^{2N(R_{Bc}+R_{Bp}+R_{Ac})}$ codewords, labeled by

$$\widetilde{X}_2^N\left(m_{Bp}, m_{Bc}, m_{Ac}\right), \tag{A.4}$$

for $m_{Bp} \in M_{Bp}$, $m_{Bc} \in M_{Bc}$, $m_{Ac} \in M_{Ac}$. The components of each codeword are generated independently from $\mathcal{N}(0, \overline{\alpha}_2 P_2)$. Let $X_2^N(m_{Ac}, m_{Bc}, m_{Bp})$ be

$$\sqrt{\alpha_2 P_2}\, V_0^N(m_{Ac}, m_{Bc}) + \widetilde{X}_2^N\left(m_{Bp}, m_{Bc}, m_{Ac}\right). \tag{A.5}$$

The codeword $X_2^N(m_{Ac}, m_{Bc}, m_{Bp})$ satisfies the power constraint of node 2 by the hypothesis that $\alpha_2 + \overline{\alpha}_2 < 1$.

Encoding: For source node $i \in \{A, B\}$, to send the message (m_{ic}, m_{ip}), it sends $\mathbf{X}_i^N(m_{ic}, m_{ip})$ to the relays.

In the second stage, relay 1 and relay 2 transmit $X_1^N(\hat{m}_{Ac}, \hat{m}_{Bc}, \hat{m}_{Ap})$ and $X_2^N(\hat{m}_{Ac}, \hat{m}_{Bc}, \hat{m}_{Bp})$. The messages indicated by $\hat{}$ is the estimated version of the original message.

Decoding: For $i = 1, 2$, the channel output at relay i is

$$\begin{aligned}&\mathbf{h}_{Ai}^T \mathbf{H}_A\left(\mathbf{U}_A(m_{Ac})(t) + \mathbf{W}_A\left(m_{Ap}\right)(t)\right)\\ &\quad + \mathbf{h}_{Bi}^T \mathbf{H}_B\left(\mathbf{U}_B(m_{Bc}(t)) + \mathbf{W}_B\left(m_{Bp}\right)(t)\right) + Z_i(t).\end{aligned} \tag{A.6}$$

The receiver at relay 1 treats the signal component $\mathbf{h}_{B1}^T \mathbf{H}_B \mathbf{W}_B(m_{Bp})(t)$ as noise, and tries to decode m_{Ac}, m_{Bc} and m_{Ap}. It reduces to a MAC with two users, but three independent messages; two messages from node A and one message from node B. In order to decode these three messages reliably, we need the requirement in (50). Likewise, we have the requirement in (51) for correct decoding at node 2.

Relay 2 treats the signal component $\mathbf{h}_{A2}^T \mathbf{H}_A \mathbf{W}_A(m_{Ap})(t)$ as noise, and tries to decode m_{Ac}, m_{Bc} and m_{Bp}. This can be done with arbitrarily small error if the condition in (51) holds.

In the second stage, terminal A receives

$$\begin{aligned}\mathbf{Y}_A(t) = &\left[\sqrt{\alpha_1 P_1}\mathbf{h}_{A1} + \sqrt{\alpha_2 P_2}\mathbf{h}_{A2}\right] V_0(\hat{m}_{Ac}, \hat{m}_{Bc})(t)\\ &+ \mathbf{h}_{A1}\widetilde{X}_1\left(\hat{m}_{Ap}, \hat{m}_{Ac}, \hat{m}_{Bc}\right)(t)\\ &+ \mathbf{h}_{A2}\widetilde{X}_2\left(\hat{m}_{Bp}, \hat{m}_{Bc}, \hat{m}_{Ac}\right)(t) + \mathbf{Z}_A(t).\end{aligned} \tag{A.7}$$

Assuming that $\hat{m}_{Ac} = m_{Ac}$ and $\hat{m}_{Ap} = m_{Ap}$, the channel is equivalent to a two-user MAC with common information, in which both users send m_{Bc}, and one of the users sends the private message m_{Bp}. The decoding is done by typicality as in [30, chapter 8], with the additional functionality of multiplexed coding. The decoder at terminal A searches for $\hat{\hat{m}}_{Bc}$ and $\hat{\hat{m}}_{Bp}$ such that Y_A^N, $V_0^N(m_{Ac}, \hat{\hat{m}}_{Bc})$, $\widetilde{X}_1^N(m_{Ap}, m_{Ac}, \hat{\hat{m}}_{Bc})$ and $\widetilde{X}_2^N(\hat{\hat{m}}_{Bp}, \hat{\hat{m}}_{Bc}, m_{Ac})$ are jointly typical. From the capacity region of MAC with common information [30, page 102], we obtain the following rate requirements

$$\begin{aligned}R_{Bp} &\le I\left(\widetilde{X}_2; Y_A \mid \widetilde{X}_1, V_0\right),\\ R_{Bp} + R_{Bc} &\le I\left(\widetilde{X}_1, \widetilde{X}_2, V_0; Y_A\right),\end{aligned} \tag{A.8}$$

where I is the mutual information function. This gives the conditions in (54) and (55).

Similarly, we have the conditions in (52) and (53) for successful decoding in terminal B. This completes the proof of Theorem 4.

Acknowledgment

This work is supported by a grant from the City University of Hong Kong (Project no. SRG 7002386).

References

[1] E. C. van der Meulen, *Transmission of information in a T-terminals discrete memoryless channel*, Ph.D. dissertation, University of California, Berkeley, Calif, USA, June 1968.

[2] T. M. Cover and A. A. El Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, 1979.

[3] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—part I: system description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.

[4] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[5] C. E. Shannon, "Two-way communications channels," in *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 611–644, June 1961.

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[7] B. Rankov and A. Wittneben, "Achievable rate regions for the two-way relay channel," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '06)*, pp. 1668–1672, Seattle, Wash, USA, July 2006.

[8] P. Larsson, N. Johansson, and K.-E. Sunell, "Coded bi-directional relaying," in *Proceedings of the 63rd IEEE Vehicular Technology Conference (VTC '06)*, vol. 2, pp. 851–855, Melbourne, Australia, May-July 2006.

[9] S. J. Kim, P. Mitran, and V. Tarokh, "Performance bounds for bidirectional coded cooperation protocols," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 5235–5241, 2008.

[10] D. Dash, A. Khoshnevis, and A. Sabharwal, "An achievable rate region for a multiuser half-duplex two-way channel," in *Proceedings of the 40th Asilomar Conference on Signals, Systems,*

and Computers (ACSSC '06), pp. 707–711, Pacific Grove, Calif, USA, October-November 2006.

[11] C.-H. Liu and F. Xue, "Network coding for two-way relaying: rate region, sum rate and opportunistic scheduling," in *Proceedings of IEEE International Conference on Communications (ICC '08)*, pp. 1044–1049, Beijing, China, May 2008.

[12] I.-J. Baik and S.-Y. Chung, "Network coding for two-way relay channels using lattices," in *Proceedings of IEEE International Conference on Communications (ICC '08)*, pp. 3898–3902, Beijing, China, May 2008.

[13] K. Narayanan, M. P. Wilson, and A. Sprintson, "Joint physical layer coding and network coding for bi-directional relaying," in *Proceedings of the 45th Annual Allerton Conference on Communication, Control, and Computing*, University of Illinois, June 2007.

[14] W. Nam, S.-Y. Chung, and Y. H. Lee, "Capacity bounds for two-way relay channels," in *Proceedings of International Zurich Seminar on Communications (IZS '08)*, pp. 144–147, Zurich, Germany, March 2008.

[15] B. Schein, *Distributed coordination in network information theory*, Ph.D dissertation, MIT, Cambridge, Mass, USA, 2001.

[16] F. Xue and S. Sandhu, "Cooperation in a half-duplex Gaussian diamond relay channel," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3806–3814, 2007.

[17] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

[18] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: physical-layer network coding," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 358–365, Los Angeles, Calif, USA, September 2006.

[19] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: analog network coding," in *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (ACM SIGCOMM '07)*, pp. 397–408, Kyoto, Japan, August 2007.

[20] S. Zhang, S. C. Liew, and L. Lu, "Physical layer network coding schemes over finite and infinite fields," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 3784–3789, New Orleans, La, USA, November-December 2008.

[21] B. K. Dey, S. Katti, S. Jaggi, D. Katabi, M. Médard, and S. Shintre, ""Real" and "complex" network codes: promises and challenges," in *Proceedings of the 4th Workshop on Network Coding, Theory, and Applications (NetCod '08)*, pp. 1–6, Hong Kong, January 2008.

[22] A. Høst-Madsen, "Capacity bounds for cooperative diversity," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1522–1544, 2006.

[23] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

[24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.

[25] P. Hu, *Cooperative strategies for Gaussian parallel relay networks*, M.S. thesis, City University of Hong Kong, Hong Kong, September 2009.

[26] T. S. Han and K. Kobayashi, "A new achievable rate region for the interference channel," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 49–60, 1981.

[27] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1073–1096, 2003.

[28] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[29] M. A. Khojastepour, A. Sabharwal, and B. Aazhang, "Bounds on achievable rates for general multiterminal networks with practical constraints," in *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks (IPSN '03)*, vol. 2634 of *Lecture Notes in Computer Science*, pp. 146–161, Palo Alto, Calif, USA, 2003.

[30] G. Kramer, *Topics in Multi-User Information Theory*, vol. 4 of *Foundations and Trends in Communications and Information Theory*, NOW Publishers, 2007.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 945765, 15 pages
doi:10.1155/2010/945765

Research Article

Parity-Check Network Coding for Multiple Access Relay Channel in Wireless Sensor Cooperative Communications

Bing Du and Jun Zhang

School of Electronics and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China

Correspondence should be addressed to Bing Du, ice.dudu@yahoo.com.cn

Received 27 September 2009; Revised 30 December 2009; Accepted 12 April 2010

Academic Editor: Zhi-Hong Mao

A recently developed theory suggests that network coding is a generalization of source coding and channel coding and thus yields a significant performance improvement in terms of throughput and spatial diversity. This paper proposes a cooperative design of a parity-check network coding scheme in the context of a two-source multiple access relay channel (MARC) model, a common compact model in hierarchical wireless sensor networks (WSNs). The scheme uses Low-Density Parity-Check (LDPC) as the surrogate to build up a layered structure which encapsulates the multiple constituent LDPC codes in the source and relay nodes. Specifically, the relay node decodes the messages from two sources, which are used to generate extra parity-check bits by a random network coding procedure to fill up the rate gap between Source-Relay and Source-Destination transmissions. Then, we derived the key algebraic relationships among multidimensional LDPC constituent codes as one of the constraints for code profile optimization. These extra check bits are sent to the destination to realize a cooperative diversity as well as to approach MARC *decode-and-forward* (DF) capacity.

1. Introduction

The demand for ubiquitous communications has motivated the deployment of a variety of wireless devices and technologies that accommodate ad hoc communications. In large numbers, such devices, despite their different sizes, processing constraints, and levels of affordability, form a Wireless Sensor Network (WSN). The WSN cooperatively monitors the physical world and enables sharing of computing capabilities, bandwidth, and energy resources, offering more integrated and essential information than with any single-sensor node. The WSN is generally built as a hierarchical structure by placing a sparse network of access points connected by a high-bandwidth network within a random homogeneous ad hoc network, in which wireless relay nodes serve exclusively as forwarders [1], as in Figure 1. In addition, the hierarchical sensor network with an access point and a single forwarding node can be modeled as a Multiple Access Relay Channel (MARC), which is a multisource extension of the well-known single-user relay channel [2]. With dedicated relay nodes, cooperative communications [3–7] among WSN exploit the broadcast characteristics and inherent spatial diversity to form a large transmit and/or receive antenna array (also known as Multiple Input Multiple Output, MIMO). Collaborative clusters are able to achieve spatial diversity as well as rate multiplexing by making "negotiations" among neighboring nodes to fully utilize the rich wireless propagation environments across multiple protocol layers and offers numerous opportunities to improve network performance in terms of throughput [2], reliability [8–10], longevity, and flexibility.

The most important element in cooperative communications is coding protocols responsible for interaction between cooperative nodes. Over the past few years, several coding strategies have been deployed for cooperative communications. Distributed space-time coding was originally proposed for MIMO systems [6]; nevertheless, synchronization among cooperative nodes is the unavoidable problem when the space-time coding strategy is brought into cooperative communications. Lately, as the network grows, traditional relay schemes have become increasingly bandwidth-inefficient. To break through the bandwidth bottleneck, network coding [11]—a technique originally developed for routing in lossless wireline networks—has been recently applied to wireless

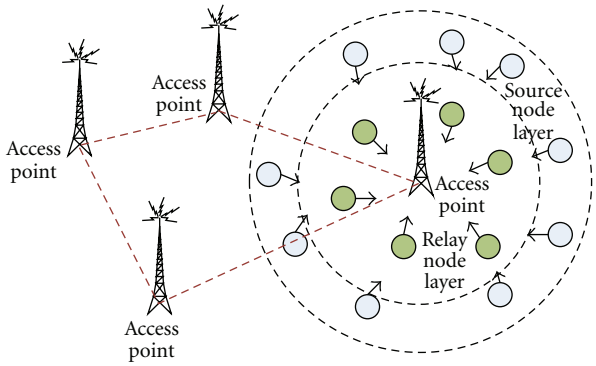


Figure 1: Hierarchical communication structure with multiple sources and dedicated relays.

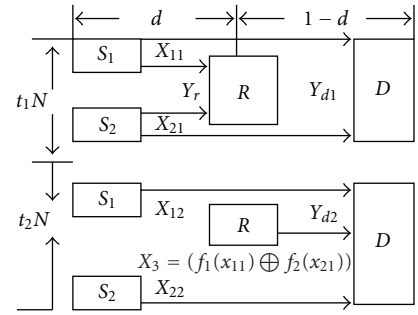


Figure 2: Cooperative protocol in MARC with one-block transmission. In the t_1 slot, S_1 and S_2 broadcast x_{11} and x_{21}; in the t_2 slot, R forwards the network-coded message x_3, and S_1 and S_2 transmit x_{12} and x_{22}.

relay networks. Traditional relaying [12–14] entails a loss in spectral efficiency that can be mitigated through network coding in cooperative communications, for its information theoretical scheme and cooperative nature. However, certain fundamental aspects of wireless communication, interference, fading, and mobility make the problem of applying network coding to cooperative communications particularly challenging.

The application of a cooperative network coding strategy is based on the fact that network coding has automatically been associated with cooperative communications as it employs intermediate nodes to combine packets [15–22]. Some approaches with practical advantages have been established to introduce network coding strategies into relay cases. In a two-way relay channel, the relay node combines received messages via network coding and broadcasts them to the opposite sited sources [15, 16]. Such a strategy has been demonstrated to reduce the number of time slots required to exchange a packet from 4 to 2, and thus a significant gain in throughput. A recently developed idea based on joint network coding with channel coding or source coding [17, 18, 21, 22] suggests that network coding is a generalization of source coding and channel coding [23]. Effros et al. [19] used network information theory to show that joint design of source, channel, and network coding in end-to-end transmission could yield much better performance, especially for the situation in which source, channel, and network separation between these codes does not hold in underlying networks.

In essence, the contribution of this paper is to employ network coding with additional parity-check bits generated from the two sources' information bits in relay nodes with linear acceptable complexity. The extra parity-check bits are designed as side information to fill up the mutual information gap between Source-Destination and Relay-Destination transmissions and hence approach the MARC "Cut-Set" bound, which is not addressed in most of the previous research works. Specifically, this paper constructs a multidimensional LDPC code to realize the network coding in a cooperative pair of nodes, as the graphical description of LDPC can flexibly bridge distributed processing and can be customized to emulate a random coding scheme of any rate. Although density evolution (DE) has high precision, the resulting increase in the complexity of DE poses a significant challenge to design a multidimensional LDPC decoder. Our work concentrates on practical implementation to present the behaviors of constituent decoders by Extrinsic Information Transfer Charts (EXITs) with a modified Gaussian approximation, which reduces the infinite dimensional problem of tracking densities to a one-dimensional problem of tracking means that is readily addressed with linear programming tools.

The remainder of the paper is organized as follows. Section 2 describes a MARC model as well as system settings. In Section 3, we analyze the achievable sum-rate with information theory as the motivation for coding design and propose network-coding cooperative transmit strategy with multidimensional LDPC codes. The work in Section 4 focuses on the optimization of multidimensional LDPC code profile using modified Gaussian approximation and EXIT as a linear-constraint optimization. Finally, simulations are conducted and discussed to demonstrate the effectiveness of the network-coded cooperative strategy.

2. System Model and Coding Strategy

This section briefly introduces the two-source MARC model used throughout the paper and LDPC code preliminaries as the basis of the paper.

2.1. System Model. To exhaustively describe the network coding strategy, we formulate our system to MARC, a model for network topologies in which multiple sources communicate with a single Destination in the presence of a Relay node. Basically, the system consists of two Sources (S_1, S_2), one Relay (R) and one Destination (D), as in Figure 2. This MARC model has a symmetric positioning of S_1, S_2 with respect to R and D. The relay moves along the line connecting D with the origin, which is normalized to 1. The distance between S and R is set to d. Path loss is proportional to $1/d^2$. The channels between each node are independent of each other. Perfect global channel knowledge is assumed at all nodes.

Since radio terminals cannot transmit and receive simultaneously in the same frequency band, most cooperative strategies are based on the half-duplex mode [24]. The nodes are allocated orthogonal channels by TDMA. S_1 and S_2 are assumed to send messages with no priority. One block transmission is separated into two consecutive time slots, normalized to $t_1 + t_2 = 1$. Furthermore, one block length of the source is N (for brevity and clarity, the symbols S_1 and S_2 are equal to and independent of each other) and is further divided into two subblocks with t_1N and t_2N-long codewords for two slots' transmissions.

We use X, Y to represent the signals sent and received. In particular, x_{ij}, $i, j \in \{1, 2\}$ denotes the signals sent by S_1, S_2. The subscript i identifies S_1 and S_2, and the subscript j represents the two consecutive channels. x_3 is the signal sent by R, and y_r is the signal received by R. The variables y_{d1} and y_{d2} are signals received by D in consecutive channels. Specifically, in time slot t_1, S_1 and S_2 broadcast their messages x_{11} and x_{21} to R and D. In time slot t_2, R forwards the network-coded message x_3, and S_1 and S_2 send the messages x_{12} and x_{22} (new or old) to D, as in Figure 2. The equivalent baseband transmission model is shown in (1):

$$\begin{aligned} y_r &= h_{s_1r}x_{11} + h_{s_2r}x_{21} + w_{r1}, \\ y_{d1} &= h_{s_1d}x_{11} + h_{s_2d}x_{21} + w_{d1}, \\ y_{d2} &= h_{s_1d}x_{12} + h_{s_2d}x_{22} + h_{rd}x_3 + w_{d2}. \end{aligned} \tag{1}$$

Rayleigh flat fading is adopted to model these links. Specifically, h_{ij} are channel coefficients capturing the effects of path-loss, shadowing, and fading, modeled by independent circularly symmetric complex Gaussian random variables with a mean of zero and a variance of σ_{ij}^2. Furthermore, w_i, $i = r$, d_1, and d_2 account for noise and other additive interferences at the receiver, modeled with an independent, zero-mean additive Gaussian white noise with variance σ^2.

2.2. Power Control. The transmit power of each source $P_i^t = E[(x_i(n)^2)]$, where $i = 1, 2, 3$ denote S_1, S_2, and R, respectively, is constrained by

$$\begin{aligned} P_1^{t_1} + P_2^{t_1} &\le P_{\text{tot}}^{t_1}, \\ P_1^{t_2} + P_2^{t_2} + P_3^{t_2} &\le P_{\text{tot}}^{t_2}. \end{aligned} \tag{2}$$

2.3. LDPC Codes. The cooperative coding scheme adopts LDPC code. A binary LDPC code is represented by a binary sparse parity-check matrix $\mathbf{H}_{k\times n}$ which connects to a bipartite graph with n variable nodes (corresponding to n columns) and k check nodes (corresponding to k rows). An attractive property of LDPC is that it can be designed graphically by a bipartite graph, which naturally matches the network topology for cooperation. The LDPC code is presented by its variable and check nodes degree distributions $(\lambda(x), \rho(x))$, where $\lambda_i(\rho_i)$ represents the fraction of edges connected to a variable (check) node with degree i. The rate of the code is given in terms of $(\lambda(x), \rho(x))$:

$$R = 1 - \frac{\int \rho(x)dx}{\int \lambda(x)dx}. \tag{3}$$

3. Parity-Check Network Code Design

There are two particular highlights of our cooperative strategy: one is the cooperative design of side information at the relay node to exactly fill up the gap of mutual information between SR and SD channels (based on the MARC model, relay is in the middle of S_1, S_2, and D, and SR thus subject to less path loss than SD channel); the other is the network coding procedure to combine extra check bits for one-slot transmission. Particularly, the insight of the first highlight is to approach the MARC DF "Cut-Set" bound, and the second is to ensure BER QoS as network coding is extended to wireless fading environment. This section will address these two challenges.

3.1. Achievable Rates. This subsection analyzes the parity-check network coding cooperative strategy, extended from *decode-and-forward* in MARC using information theory, as a fundamental instruction to develop the coding design as described below.

The key element in the proposed strategy is that the relay node forwards redundant bits as side information for both S_1 and S_2 to D, which is based on the essential idea of "channel coding with side information." This process is a "dual thought" of "source coding with side information" [25, 26]. Channel coding with side information is to append some extra check bits to codewords, which is a "binning" process assigning a set of codewords to different bins and enlarging the minimum distance between them. At the receiver, the side information provides an index of the message, and then the decoding process chooses the closest codeword in a box with a specific index. Application of such an idea in a traditional three-node relay network can be found in [8, 9]. This paper applies the binning approach to a MARC with network coding. Besides, the extra check bits are generated with the goal of approaching MARC DF capacity. The resulting network-coded strategy is capable of balancing the problem of spatial diversity and multiplexing.

Usually, the informational theoretical view deals with achievable rates. In the MARC scenario, we consider the sum-rate, which conveys more intuition. For *decode-and-forward* strategy in a general multiple source half-duplex relay channel, the bounds on all combinations of the rate tuples for reliable detection at R and D are as follows [1]:

$$\begin{aligned} &\left(R_{S_1} + R_{S_2}\right)_{\text{DF}} \\ &\quad \le \min_{t_1+t_2=1} \begin{pmatrix} t_1 I(X_{11}, X_{21}; Y_r) + t_2 I(X_{12}, X_{22}; Y_{d_2} \mid X_3), \\ t_1 I(X_{11}, X_{21}; Y_{d_1}) + t_2 I(X_{12}, X_{22}, X_3; Y_{d_2}) \end{pmatrix}. \end{aligned} \tag{4}$$

The first terms in $\min(\cdot)$ of (4) represent the maximum rate at which R can decode the messages x_{11} and x_{21} and

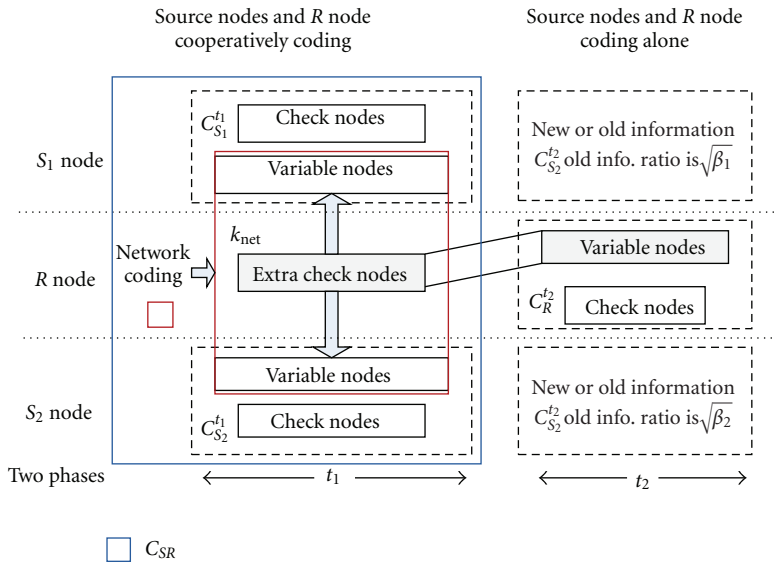


Figure 3: The cooperative strategy based on parity-check network coding.

the maximum rate at which D can decode x_{12} and x_{22} in the presence of x_3. The second terms in $\min(\cdot)$ of (4) represent the maximum rate at which D can decode the messages x_{11} and x_{21}, and the maximum rate at which D can decode all three messages x_{12}, x_{22}, andx_3.

The cooperative strategy in this study employs network coding in the sense of cooperation between S_1, S_2, and R to achieve MARC capacity in (4). The detailed protocol is as follows.

3.1.1. In Time Slot t_1: Source Nodes Operations. Each S_1 (S_2) encodes the message x_{11} (x_{21}) to codewords $C_{S_1}^{t_1}$ ($C_{S_2}^{t_1}$) at the rate of

$$R_{S_1R}^{t_1} + R_{S_2R}^{t_1} = I(X_{11}, X_{21}; Y_r). \tag{5}$$

Then, S_1 (S_2) broadcasts $C_{S_1}^{t_1}$ ($C_{S_2}^{t_1}$) to R and D. D receives the data and waits for decoding at the end of the block transmission.

To achieve maximum throughput, S_1 and S_2 broadcast messages at the sum-rate (5). R is able to decode x_{11}, x_{21} with an arbitrarily low error probability, since $R_{S_1R}^{t_1} + R_{S_2R}^{t_1}$ equals the capacity of the SR channels. According to the geometric configuration in Figure 2, intuitively, $I(X_{11}, X_{21}; Y_r) > I(X_{11}, X_{21}; Y_d)$; the physical channel of SD is more attenuated by the path loss than that of the SR channel. Consequently, although D also receives x_{11} and x_{21}, it is unable to uniquely decode them and requires extra bits $t_1N(I(X_{11}, X_{21}; Y_r) - I(X_{11}, X_{21}; Y_d))$ to make x_{11} and x_{21} decodable.

3.1.2. In Time Slot t_2: Relay Node Operation. R sends these extra bits, $t_1N(I(X_{11}, X_{21}; Y_r) - I(X_{11}, X_{21}; Y_d))$, to D at the rate

$$\begin{aligned} R_{RD} &= \frac{t_1N\left(I(X_{11}, X_{21}; Y_r) - I(X_{11}, X_{21}; Y_{d_1})\right)}{t_2N} \\ &= \frac{t_1}{t_2}\left(I(X_{11}, X_{21}; Y_r) - I(X_{11}, X_{21}; Y_{d_1})\right). \end{aligned} \tag{6}$$

Specifically, after decoding the codewords from S_i, R estimates $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$, and cooperatively uses the codewords $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$ to generate extra check bits for both S_1 and S_2, and then combines them with network coding to produce $k_{\text{net}} = t_1N(I(X_{11}, X_{21}; Y_r) - I(X_{11}, X_{21}; Y_{d_1}))$ extra check bits. The process is "network coding." For transmission, k_{net} is encapsulated by R's LDPC codeword $C_R^{t_2}$ and sent to D. Hence, the extra check bits with S_1 and S_2 codes $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$ construct the cooperative multidimensional LDPC code C_{SR}, as illustrated in Figure 3. The elements in the blue rectangle construct the cooperative code C_{SR} with respect to the information in time slot t_1. The procedure in the red rectangle is the network coding which produces and combines extra bits for both S_1 and S_2. In particular, k_{net} check bits encapsulated by codeword $C_R^{t_2}$ sent to D capture the RD channel's fading characteristics and provide an effective extinct message at D to realize a spatial diversity.

From the perspective of information theory, C_{SR} is cooperatively encoded by S_1, S_2, and R on the grounds of coding with side information. "Binning" is performed by extra check network-coded bits (or syndromes) in R's message generated from $C_{S_i}^{t_1}$, $i \in \{1,2\}$ to perform decoding of $x_{11}, x_{21} \in \{1, 2, \ldots, 2^{t_1NI(X_{11}, X_{21}; Y_r)}\}$ by restricting them into $2^{t_1NI(X_{11}, X_{21}; Y_{d_1})}$ bins of $2^{t_1N(I(X_{11}, X_{21}; Y_r) - I(X_{11}, X_{21}; Y_{d_1}))}$ in size each. From Figure 4, the "binning" process of R partitions the space of codewords of S_1 and S_2, enlarging their minimum distances to make the source's message decodable.

3.1.3. Source Nodes Operations. In time slot t_2, each S_1 (S_2) sends a message to D independently because R is in the half-duplex mode. According to the channel status, S_1 and S_2 can choose to send new or old information using the independent codebook $C_{S_1}^{t_2}$ ($C_{S_2}^{t_2}$). The new information sent

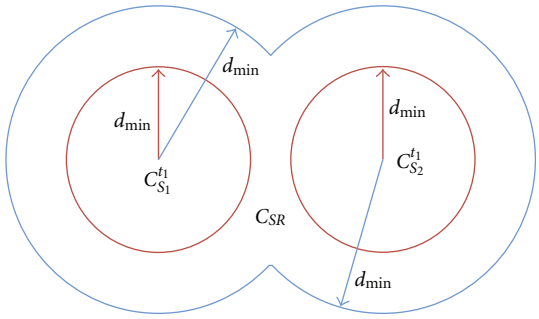


FIGURE 4: The minimum distances of C_{SR} and $C_{S_i}^{t_i}$, $i \in \{1,2\}$.

in the t_2 time slot at the sum-rate inherited from the DF rate region in (4) is

$$\left(R_{S_1D}^{t_2} + R_{S_2D}^{t_2}\right)_{\mathrm{DF}} \leq \min\left(\begin{array}{c} I(X_{12}, X_{22}; Y_{d_2} \mid X_3), \\ \left(\begin{array}{c} I(X_{12}, X_{22}, X_3; Y_{d_2}) \\ -\dfrac{t_1}{t_2}\left(I(X_{11}, X_{21}; Y_r) - I(X_{11}, X_{21}; Y_{d_1})\right) \end{array}\right) \end{array}\right). \tag{7}$$

Source transmissions in the t_2 slot are isolated from the operation of R, as in Figure 3, which illustrates the operation in time slot t_2 with independent information transmissions by S_1, S_2, and R. Thus, we deal with codebook $C_{S_1}^{t_2}$ ($C_{S_2}^{t_2}$) as a single LDPC code and choose a suitable LDPC codebook to satisfy the rate constraint in (7).

At the end of one block transmission, D successively decodes $C_R^{t_2}$, $C_{S_1}^{t_2}$, and $C_{S_2}^{t_2}$. Then, the extra check bits k_{net} are obtained for joint decoding of C_{SR} with $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$.The network coding cooperative strategy is summarized as follows in Table 1.

In the cooperative protocol mentioned above, MARC DF capacity in (4) is approximated via the rate allocation scheme in (5) through (7). Especially, if $I(X_{11}, X_{21}; Y_r) > I(X_{11}, X_{21}; Y_d)$, the rate at $I(X_{11}, X_{21}; Y_r)$ to transmit information of S_1 and S_2 in time slot t_1 to D will be achieved, resulting in a rate gain by cooperation between S_1, S_2, and R.

However, strictly speaking, the network coding performed here is not exactly the same as the network layer coding, which mainly focuses on routing problems and packet-level combination. Here, we borrow the kernel idea of the network layer coding to combine the extra check bits in R, which improves the bandwidth efficiency by R's extra check bits transmitted in one slot for both S_1 and S_2.

3.1.4. Parameters in the Cooperative Protocol. The achievable sum-rate is as a function of three parameters: d, t_1, and β_i. The definition of d and t_1 is in Section 2.1. β_i is the fraction of S_i allocate to the old messages in t_2. And $\beta_1 = \beta_2 = \beta$ is set for the symmetric geometry. The achievable sum-rate in (4) can be evaluated by these three parameters with the AWGN channel capacity. The outer bound of the sum-rate is the maximum of (8) subject to the value of $t_i\beta_i$, $i \in \{1,2\}$.

$$\begin{aligned}\left(R_{S_1} + R_{S_2}\right)_{\mathrm{DF}} &\leq \max_{t_i\beta_i}\left\{\min\left(\begin{array}{c} \left(\begin{array}{c} t_1 C\left(\displaystyle\sum_{i\in\{1,2\}} \gamma_{S_i}^{t_1} \left|h_{S_ir}\right|^2\right) \\ +t_2 C\left(\displaystyle\sum_{i\in\{1,2\}} (1-\beta_i)\gamma_{S_i}^{t_2} \left|h_{S_id}\right|^2\right) \end{array}\right), \\ \left(\begin{array}{c} t_1 C\left(\displaystyle\sum_{i\in\{1,2\}} \gamma_{S_i}^{t_1} \left|h_{S_id}\right|^2\right) \\ +t_2 C\left(\displaystyle\sum_{i\in\{1,2\}} \dfrac{\gamma_{S_i}^{t_2}\left|h_{S_id}\right|^2 + \gamma_R^{t_2}\left|h_{rd}\right|^2}{+\displaystyle\sum_{i\in\{1,2\}} 2\sqrt{\beta_i \gamma_{S_i}^{t_2}\left|h_{S_id}\right|^2 \gamma_R^{t_2}\left|h_{rd}\right|^2}}\right) \end{array}\right) \end{array}\right)\right\}. \\ &\tag{8}\end{aligned}$$

The received signal-to-noise ratio for R and D is listed with the channel gains as

$$\begin{aligned} P_{S_ir}^{t_1} &= \gamma_{S_i}^{t_1}\left|h_{S_ir}\right|^2, \quad P_{S_id}^{t_1} = \gamma_{S_i}^{t_1}\left|h_{S_id}\right|^2, \\ P_{S_id}^{t_2} &= \gamma_{S_i}^{t_2}\left|h_{S_id}\right|^2, \quad P_{rd}^{t_2} = \gamma_R^{t_2}\left|h_{rd}\right|^2, \end{aligned} \quad i \in \{1,2\}. \tag{9}$$

$\gamma = E_S/\sigma^2 = P/(W\sigma^2)$ is the input signal-to-noise ratio (SNR), where power P is constrained within 10 dB in both time slots using (2), and σ^2 is the variance of noise at the receivers of R and D, which are assumed to be equal.

The rates of (8) are plotted in Figure 5. Note that, when d is around 0.5, the sum-rate is at its maximum. The function of best β against d is more like a step function. When R is physically closer to the source $d < 0.3$, $\beta = 1$ is optimal, which means that the old information takes up all source transmissions in time slot t_2. This could be attributed to a path loss of the RD channel, and so S_1 and S_2 send the same information again to fill up the gap. The other extreme is when R is physically closer to D, $d > 0.7$, $\beta = 0$ which means that sources send new information in time slot t_2. However, R must successfully decode the source's information.

To obtain the time partition factor t_1, the sum-rate in (7) can be used to calculate t by manipulating the sum-rate at the corner point of the capacity region, which means that the two terms of $\min(\cdot)$ in (7) are equal. In Figure 5, we evaluate t by setting the two terms mentioned above to be equal, with different d. When $d = 0.5$ and $t_1 = 0.7$, the MARC DF sum-rate achieves its maximum value, which means that R's transmission takes up $t_2 = 0.3$ slot, resulting in a free degree of each source of $0.7/2 + 0.3 = 0.65$.

3.2. Cooperative Design Framework. This subsection depicts the network-coded cooperative framework to realize above achievable rates. Specifically, the layered structure is constructed with multidimensional LDPC constituent codes corresponding to S_1 and S_2, as in Figure 6. This coding strategy is based on a half-duplex TDD mode, so that the

Table 1: The network-coded cooperative strategy.

	S_1 and S_2	R	D
t_1	Broadcast $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$	Receives $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$	Receives $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$ and stores them for deocoding.
t_2	Send $C_{S_1}^{t_2}$ and $C_{S_2}^{t_2}$ to D.	Performs network coding to generate k_{net} extra check bits which is encoded by $C_R^{t_2}$ and sends $C_R^{t_2}$ to D	(1) Receives $C_{S_1}^{t_2}$ and $C_{S_2}^{t_2}$ (2) Receives $C_R^{t_2}$ (3) Successively decodes $C_{S_1}^{t_2}, C_{S_2}^{t_2}, C_R^{t_2}$ and obtains k_{net} (4) Joint decodes $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$ with k_{net}

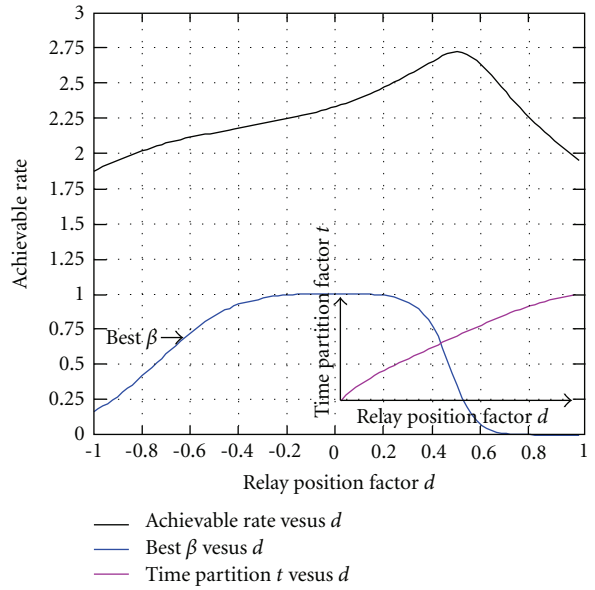


Figure 5: The achievable sum-rate with $P_1^{t_1} + P_2^{t_1} \le P_{\text{tol}}^{t_1} = 10\,\text{db}$, $P_1^{t_2} + P_2^{t_2} + P_3^{t_2} \le P_{\text{tol}}^{t_2} = 10\,\text{db}$.

operation of R only cooperates with the source transmissions in time slot t_1. In time slot t_2, S_1, S_2, and R send their information independently.

The cooperative codeword C_{SR}'s parity-check matrix $\mathbf{H}_{SR}$ is constructed with three LDPC constituent codes as in Figure 6, including sub-LDPC parity-check matrices $\mathbf{H}_{S_1}$ and $\mathbf{H}_{S_2}$, and the network code parity-check matrix $\mathbf{H}_{\text{net}}$. $\mathbf{H}_{S_1}$ ($\mathbf{H}_{S_2}$) is employed by S_1 (S_2) to encode the message x_{11}, (x_{21}) locally; thus, $\mathbf{H}_{S_1}$ ($\mathbf{H}_{S_2}$) is a complete parity-check matrix. $\mathbf{H}_{S_1}$ ($\mathbf{H}_{S_2}$) has n_1 (n_2) variable nodes and k_1 (k_2) check nodes. The sources' codeword $C_{S_1}^{t_1}(C_{S_2}^{t_1})$ is enforced to satisfy k_1 (k_2) check bits.

In addition, parity-checks k_1 and k_2 do not interact with each other or do not check each others' variable nodes since the independent sources S_1 and S_2 cannot produce checks for unknown information bits.

The extra check nodes k_{net} have the same variable nodes as the check nodes of S_1 and S_2; otherwise, they cannot provide any checks for the codewords of S_1 and S_2. Therefore, in $\mathbf{H}_{\text{net}}$, the variable nodes n_1 and n_2 are sequentially arranged as information bits and are enforced to satisfy k_{net} check bits. Hence, $\mathbf{H}_{\text{net}}$ has k_{net} rows and $(n_1 + n_2 + k_{\text{net}})$ columns, as $\mathbf{H}_{\text{net}}$: $k_{\text{net}} \times (n_1 + n_2 + k_{\text{net}})$. Above all, the network coding procedure uses $\mathbf{H}_{\text{net}}$ to merge the extra check bits.

Random linear codes are capacity-approaching for the Gaussian channel under maximum likelihood decoding. Therefore, the extra checks are randomly connected to the set of variable nodes $n_1 + n_2$ in $\mathbf{H}_{\text{net}}$. However, if $\mathbf{H}_{\text{net}}$ is constructed in a completely random way, encoder and decoder implementations become very difficult as the code size grows due to the pseudorandom interconnection and the large memory required. Structured LDPC codes would be a good option to facilitate implementation without compromising performance. Therefore, $\mathbf{H}_{\text{net}}$ is constructed in the partial dual-diagonal form so that most parity check bits can be obtained via back-substitution. Partial dual-diagonal form is merely in the k_{net} portion, as illustrated in Figure 6, and the remainders are still randomly constructed.

Linear-time encoding can be achieved by using the near-triangular parity portion. The extra check bits $b_1, b_2, \ldots, b_{k_{\text{net}}}$ are generated by a direct encoding procedure, as follows:

$$
\begin{aligned}
b_0 &= \sum_{j=1}^{t_1 N + t_1 N} \sum_{i=1}^{k_{\text{net}}} H_{\text{net}}(i,j) \left[C_{S_1}^{t_1}, C_{S_2}^{t_1} \right]^T, \\
b_1 &= \sum_{j=1}^{t_1 N + t_1 N} H_{\text{net}}(i,j) \left[C_{S_1}^{t_1}, C_{S_2}^{t_1} \right]^T + b_0, \quad i = 1, \\
b_{i+1} &= b_i + \sum_{j=1}^{t_1 N + t_1 N} H_{\text{net}}(i,j) \left[C_{S_1}^{t_1}, C_{S_2}^{t_1} \right]^T + b_0, \\
&\qquad\qquad i = 2, \ldots, k_{\text{net}}.
\end{aligned}
\tag{10}
$$

The addition of the above equations is in a binary field; b_0 is an additional variable used to calculate extra check bits $b_1, b_2, \ldots, b_{k_{\text{net}}}$.

As mentioned above, the cooperative LDPC code C_{SR} is satisfied by the parity-check constraints as

$$
\begin{aligned}
H_{S_1}\left[C_{S_1}^{t_1} \right] = 0; \qquad H_{S_2}\left[C_{S_2}^{t_1} \right] = 0; \\
H_{SR}\left[C_{S_1}^{t_1}, C_{S_2}^{t_1}, \{b_1, b_2, \ldots, b_{k_{\text{net}}}\} \right] = 0.
\end{aligned}
\tag{11}
$$

Moreover, once k_{net} extra check bits are obtained via optimization cooperatively conducted with $\mathbf{H}_{S_1}$ and $\mathbf{H}_{S_2}$, the quasidiagonal part of $\mathbf{H}_{\text{net}}$ is determined. Hence the parity-check matrix $\mathbf{H}_{SR}$ can be simplified to $\mathbf{H}'_{SR}$ by removing the columns of the quasidiagonal part, and the optimization is then performed on $\mathbf{H}'_{SR}$ instead.

In $\mathbf{H}'_{SR}$, variable nodes have two types of checks: their own checks and extra checks offered by network coding.

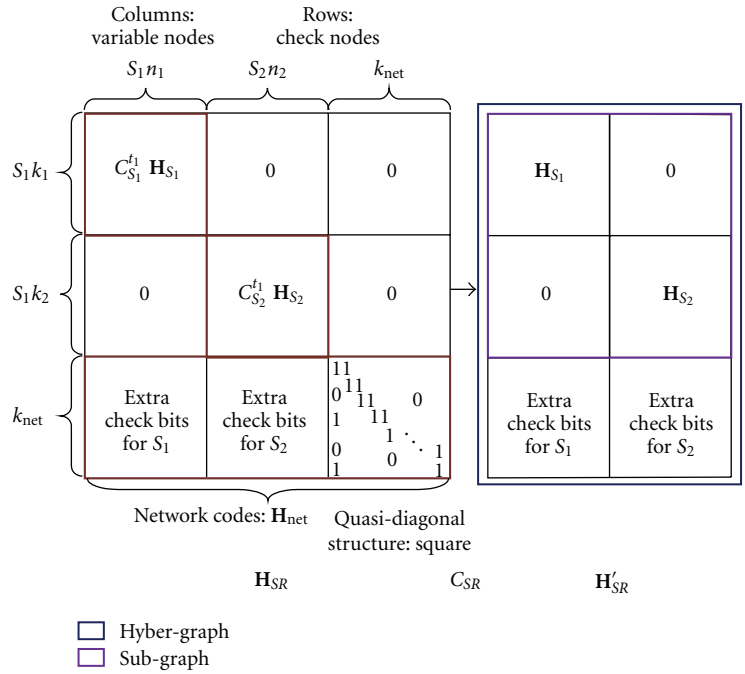


Figure 6: The cooperative design framework of parity-check network coding.

Accordingly, each variable node in $\mathbf{H}'_{SR}$ has two types of variable node degrees, expressed by $\lambda_{i,j}^{SR}$: sub-LDPC degree (in $\mathbf{H}_{S_1}$ or $\mathbf{H}_{S_2}$) i, $i \geq 2$, and extra degree j, $j \geq 0$ (in $\mathbf{H}_{\text{net}}$). Assuming that $0 < \eta_1(\eta_2) < 1$ is the ratio of the edges in $\mathbf{H}_{S_1}$ ($\mathbf{H}_{S_2}$) to the edges in $\mathbf{H}'_{SR}$, the variable node degree distributions $\gamma^{S_1}(x)$ ($\gamma^{S_2}(x)$) of $\mathbf{H}_{S_1}$ ($\mathbf{H}_{S_2}$) in terms of $\lambda_{i,j}^{SR}$ are

$$\gamma_i^{S_1} = \frac{1}{\eta_1} \sum_{j \geq 0} \frac{i}{i+j} \lambda_{i,j}^{SR}, \qquad \gamma_i^{S_2} = \frac{1}{\eta_2} \sum_{j \geq 0} \frac{i}{i+j} \lambda_{i,j}^{SR}. \tag{12}$$

The relationship of (12) is used for cooperative code profile optimization in next section.

Then, we will give the kernel constraint of the cooperative design, which determines how the extra check bits are connected to the variable node set in the cooperative code C_{SR}. Since the extra checks are appended to the sub-LDPC codes $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$, which have the same set of variable nodes as C_{SR}, the degree of C_{SR} variable nodes turns out to be greater than that of the same set of variable nodes in sub-LDPC codes $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$. However, due to the random construction, the extra checks connected to one specific variable node cannot be determined; in other words, it is impossible to list exactly which variable node receives the extra checks. Under this circumstance, we derive the relationship between $C_{S_1}^{t_1}$, $C_{S_2}^{t_1}$, and C_{SR} in terms of variable nodes' number with respect to a specific degree i, denoted by $N_i = (\lambda_i / i) \cdot E$, where E is the total number of edges of the parity-check matrix concerned.

Theorem 1. *If the cooperative code C_{SR} has a maximum degree $d_{v,SR}$ and a total number of edges E_{SR} and, similarly, two sub-LDPC codes $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$ have maximum degrees d_{v,S_1} and d_{v,S_2} and total edges E_{S_1} and E_{S_2}, respectively, then one has the following relationships:*

$$\sum_{i=j}^{d_{v,SR}} \frac{\lambda_{i,j}^{SR}}{i} E_{SR} \geq \sum_{i=j}^{\max(d_{v,S_1}, d_{v,S_2})} \left(\frac{\gamma_i^{S_1}}{i} E_{S_1} + \frac{\gamma_i^{S_2}}{i} E_{S_2} \right) \quad \forall j = 2, 3, \ldots, d_{v,SR}. \tag{13}$$

The proof of Theorem 1 is in the appendix. Theorem 1 ensures that the network-coded messages from the relay node as the extra check bits independently sent through the fading channel offer spatial diversity gain to the cooperative strategy. And the number of extra check bits is determined by

$$(n_1 + n_2)(I(X_{11}, X_{21}; Y_r) - I(X_{11}, X_{21}; Y_{d_1})). \tag{14}$$

4. Cooperative Code Profile Optimization

Next, the challenge to the construction of $\mathbf{H}'_{SR}$ lies in finding the optimal code profile of C_{SR}, including optimal profiles of sub-LDPC constituent codes $C_{S_1}^{t_1}$, $C_{S_2}^{t_1}$ together with extra check bits.

In engineering, optimization has always been a difficult problem due to its computational complexity, particularly for cost-constraint hardware. Therefore, to restrict our optimization algorithms to a linear programming is the mainly interest in this section. We will use Gaussian approximation and Extrinsic Information Transfer (EXIT) charts as the linear programming tool to obtain a C_{SR} code profile in a cooperative framework illustrated in Figure 6.

Generally, optimization of LDPC code profile can be done in two different ways. One is to fix noise variance and maximize information transmit rate to search for the optimal degree distributions $(\lambda(x), \rho(x))$. The other is to fix the rate to find the $(\lambda(x), \rho(x))$ that yields the largest noise threshold. The cooperative strategy discussed in this paper prefers bandwidth efficiency to noise threshold. Information transmit rate seems straightforward, which is defined as the ratio of information bits sent by sources to all bits transmitted for the concerned message (source messages in time slot t_1), and thus,

$$R_{SR} = \frac{n_1 - k_1 + n_2 - k_2}{n_1 + n_2 + k_{\text{net}}}. \tag{15}$$

Equation (15) can be expressed by the degree distribution as

$$R_{SR} = 1 - \frac{\sum_{i\geq 2} \rho_i^{SR}/i}{\sum_{i\geq 2, j\geq 0} \lambda_{i,j}^{SR}/(i+j)}. \tag{16}$$

The optimization algorithm maximizes rate R_{SR} to obtain the degree distribution of $\mathbf{H}'_{SR}$.

It is difficult to obtain $(\lambda(x), \rho(x))$ in one operational procedure, and so we fix $\rho(x)$ to get $\lambda(x)$ and then get $\rho(x)$ with fixed $\lambda(x)$, given the maximum number of iterations. With a constant $\rho(x)$, the maximizing rate is equivalent to maximizing $\sum_{i\geq 2, j\geq 0} \lambda_{i,j}^{SR}/(i+j)$.

EXIT [27] provides a computationally simple tool for predicting the asymptotic convergence behavior of iterative coding schemes by tracking trajectories of extrinsic information exchange between variable nodes and check nodes in the bipartite graph. Furthermore, operations of variable and check nodes are referred to the variable-node decoder (VND) and check-node decoder (CND), respectively. We also use mutual information as the surrogate to analyze and optimize LDPC codes by matching the EXIT functions with the constituent decoders (VND, CND) based on the area property of the functions. Figure 7 illustrates iterative joint decoding of VND and CND in $\mathbf{H}'_{SR}$. Specifically, $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$ are received by D at t_1 time slot, while $C_R^{t_2}$ is received by D at t_2 time slot. Channel S_1D captures its own fading factor via $C_{S_1}^{t_1}$; channel S_2D captures its own fading factor via $C_{S_2}^{t_1}$; channel RD captures its own fading factor via $C_R^{t_2}$. These three codewords are used to cooperative decode x_{11} and x_{21}. Each sub-LDPC code is related to a coupling of a VND-CND decoder. The network code plays a role as the interleaving function of the two CNDs with extra extrinsic information.

EXIT charts compute two curves, the VND curve and the CND curve, corresponding to the steps of each decoder's density evolution. With the VND curve, I_A is interpreted as the mutual information between the VND "input" LLR message and the transmitted symbol of the check node at iteration l. I_E is interpreted as the mutual information between the VND "output" LLR message and the transmitted symbol of the variable node at iteration l. With the CND function, the interpretations of I_E and I_A are opposite.

Gaussian approximation is an effective way to track the means of the log likelihood ratio (LLR) message, which is assumed to be symmetrically Gaussian distributed [28]. Even with an irregular LDPC code [27], the Gaussian approximation can still be precise after a few modifications; that is, the distribution of the variable node LLR message is a mixture of Gaussian approximations, and the corresponding VND EXIT function is

$$\begin{aligned} I_{E_V} &= f(I_{ch}, I_{A_V}) = \sum_{j=2}^{d_v} \lambda_j I_{E_{Vj}} \\ &= \sum_{j=2}^{d_v} \lambda_j J(J^{-1}(I_{ch}) + (j-1)J^{-1}(I_{A_V})), \end{aligned} \tag{17}$$

where $J(x)$ is defined by

$$J(x) = I(X, L) = \int \frac{1}{\sqrt{4\pi x}} e^{-(l-x)^2/4x} \left(1 - \log_2\left(1 + e^l\right)\right) dl. \tag{18}$$

The corresponding CND EXIT function is

$$\begin{aligned} I_{E_c} &= f(I_{A_c}) = \sum_{j=2}^{d_c} \rho_j I_{E_{cj}} \\ &= \sum_{j=2}^{d_c} \rho_j \frac{1}{\ln 2} \sum_{i=1}^{\infty} \frac{1}{(2i-1)(2i)} \left[\varphi_i(J^{-1}(I_{A_c}))\right]^{j-1}, \end{aligned} \tag{19}$$

where $\varphi_i(x)$ is defined by

$$\varphi_i(x) = \int_{-1}^{1} \frac{2t^{2i}}{(1-t^2)\sqrt{4\pi x}} e^{-(\ln(1+t/1-t)-x)^2/4x} dt. \tag{20}$$

The decoding process is expected to converge progressively after each decoding iteration. Therefore, we require $I_{E_v}(I_{E_c}(I_A)) > I_A$ for all $I_A \in [0, 1]$ to ensure successful decoding. This is equivalent to $I_{E_v}(I_A) > I_{E_c}^{-1}(I_A)$. The decoding process is thus predicted to converge if and only if the VND curve is strictly greater than the reversed-axis CND curve.

Next we will formulate the constraints to fulfill the optimization and obtain code profiles of C_{SR}, $C_{S_1}^{t_1}$, and $C_{S_2}^{t_1}$: $(\lambda_{i,j}^{SR}, \rho^{SR})$, $(\gamma^{S_1}, \rho^{S_1})$, and $(\gamma^{S_2}, \rho^{S_2})$. In this paper, for simplicity but still revealing the insights of the cooperative design, we let node S_1 and S_2 completely symmetric, that is, $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$ are equal, and thus in simulations we can treat them as one LDPC code $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$).

First, (13) in Section 3 is the kernel constraint. Specifically, if S_1 and S_2 are completely symmetric, which means that $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$ have an equal number of extra checks and an equal number of bipartite graph edges, that is, $E_{SR} = 2E_{S_1} = 2E_{S_2}$, we have

$$\begin{aligned} &\sum_{i=j}^{d_{v,SR}} \frac{\left(\lambda_{i,j}^{SR}\right)}{i} \\ &\qquad \geq \sum_{i=j}^{\max(d_{v,S_1}, d_{v,S_2})} \left(\frac{\gamma_i^{S_1}}{2i} + \frac{\gamma_i^{S_2}}{2i}\right) \quad \forall j = 2, 3, \ldots, d_{v,SR}. \end{aligned} \tag{21}$$

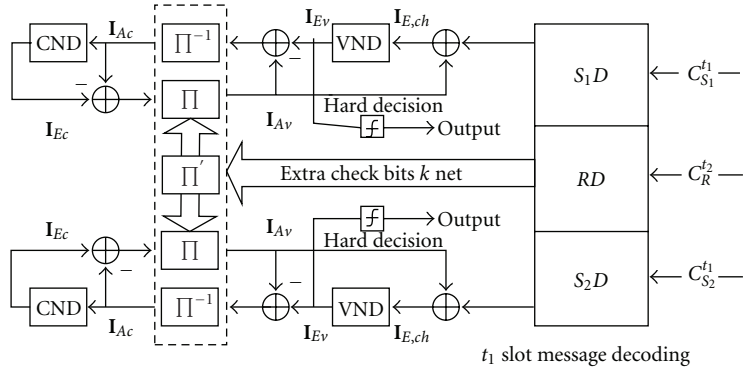


Figure 7: Decoder structure for joint decoding of C_{SR}.

Equation (21) poses a constraint to ensure the cooperative design of C_{SR} together with $C_{S_1}^{t_1}$ and $C_{S_2}^{t_1}$.

Moreover, the rate constraints are imposed to further restrict cooperative design. Cooperative C_{SR} has more check bits than both sources; so the cooperative code rate should be lower than any rate of S_1 and S_2:

$$R_{SR} < R_{(C_{S_1}^{t_1})}, \quad R_{SR} < R_{(C_{S_2}^{t_1})},$$
$$\left(R_{(C_{S_j}^{t_1})} = 1 - \left(\frac{\sum_i \rho_i / i}{\sum_i \gamma_i / i}\right)^{S_j}, \ j = 1, 2\right). \tag{22}$$

As mentioned above, using EXIT, the VND curve must be strictly greater than the reversed-axis CND curve to ensure the convergence of a propagation decoding algorithm, which requires that all the constituent codes satisfy the condition $I_{E_v}(I_A) > I_{E_c}^{-1}(I_A)$, with additional irregular LDPC modification for all I belonging to a discrete, fine grid over (0, 1):

$$\begin{aligned}
&\sum \gamma_i^{S_1} I_{E_V,i}^{S_1}(I_{A_V}, I_{ch}) > {I^{-1}}_{E_C}^{S_1}(I_{A_C}) + \delta,\\
&\sum \gamma_i^{S_2} I_{E_V,i}^{S_2}(I_{A_V}, I_{ch}) > {I^{-1}}_{E_C}^{S_2}(I_{A_C}) + \delta, \quad \delta > 0,\\
&\sum_{i,j} \lambda_{i,j}^{SR} I_{E_V,i,j}(I_{A_V}, I_{ch}) > {I^{-1}}_{E_C}^{SR}(I_{A_C}) + \delta.
\end{aligned} \tag{23}$$

Clearly, the degree distribution $\lambda(x)$ of a complete parity-check matrix sums to "1" wherever it occurs in $\mathbf{H}_{S_1}$ and $\mathbf{H}_{S_2}$ or in $\mathbf{H}'_{SR}$:

$$\gamma^{S_1}(1) = \gamma^{S_1}(1) = \lambda^{SR}(1) = 1. \tag{24}$$

The above four constraints (21)–(24) formulate the cooperative design and optimization and maintain the linear features of the variable node degree distribution $\lambda(x)$. Linear optimization with respect to $\lambda(x)$ yields a good C_{SR} profile $\lambda(x)$ with a fixed and concentrated check node degree $\rho(x)$. Meanwhile, fixing the variable node degree distribution $\lambda(x)$, similar optimization principles hold for the check node degree distribution $\rho(x)$.

5. Simulations and Results

This section validates the performance of parity-check network coding in MARC via numerical simulations. These simulations focus on two goals: (1) demonstrating that the cooperative framework produces a good cooperative code C_{SR} profile as well as a single code $C_{S_1}^{t_1}$ or $C_{S_2}^{t_1}$ profile and (2) investigating the BER performance of the cooperative code C_{SR} under different channel settings compared with $C_{S_1}^{t_1}$ or $C_{S_2}^{t_1}$.

5.1. EXIT Chats of Code Profile. EXIT charts of C_{SR}, $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$) are shown in Figure 8. With the rate $R_{S_1R}^{t_1} = 0.5$ ($R_{S_2R}^{t_1} = 0.5$) and the SNR = 1.2 dB, the EXIT curve of VND and CND is obtained. The curve of CND is strictly lower than the reversed-axis VND curve. Figure 8(b) draws the EXIT chart of the cooperative code C_{SR} subject to the EXIT chart of $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$) in Figure 8(a) with the linear optimization algorithm mentioned in Section 4. The curves of VND and CND approach asymptotically as the code rate increases. However, a comparison of the two subfigures shows that the gap between the VND and CND curves of C_{SR} is greater than that of $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$) because more check bits work on the same set of variable nodes for code C_{SR}. Table 2 lists the optimal degree distributions at the $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$) code rates of 0.3, 0.4, 0.5, and 0.6. In each column of the rate, $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$) code profile is in the left subcolumn, while the cooperative C_{SR} code profile is in the right subcolumn. It is obvious that the distributions satisfy the constraints of (21)–(24); this is especially true for the cooperative design constraints.

Moreover, Figure 9 shows the maximum of C_{SR} transmit rates and related $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$) transmit rates obtained by the linear optimization algorithm in Section 4. Here, we assume that the S_1 and S_2 have the same transmit rate (this is not necessary). We also plot the MARC *decode-and-forward* "Cut-Set" capacity in the same figure to compare the proposed parity-check network coding strategy. These results show that the cooperative strategy achievable rate is approximately 0.5 dB below the MARC capacity. And for better illustration the cooperative strategy, the direct link transmission capacity, that is, $I(X_{11}, X_{21}; Y_{d_1})$ without relay is also plotted.

Table 2: Code profile from the optimization algorithm.

Rate	0.3				0.4				0.5				0.6			
	$C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$)		C_{SR}		$C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$)		C_{SR}		$C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$)		C_{SR}		$C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$)		C_{SR}	
$\lambda(x)$	2	0.2264	2	0.1671	2	0.2479	2	0.1716	2	0.243	2	0.1956	2	0.3944	2	0.3463
	3	0.0173	3	0.0288	4	0.1395	4	0.1996	3	0.2994	3	0.1127	3	0.0462	3	0.0749
	4	5.03e-6	5	0.2689	5	0.2893	5	0.2887	7	0.2862	4	0.171	4	0.2411	4	0.2246
	5	0.2683	6	0.0338	18	0.0604	18	0.0539	8	5.98e-6	7	0.2304	5	0.3183	5	0.2086
	6	0.1074	8	0.0451	19	0.2629	19	0.1492	9	0.1714	8	4.63e-6			8	0.101
	27	0.1818	29	0.2256			27	0.0835	10	3.13e-6	9	0.0467			9	0.0446
	29	0.1273	30	0.1693			38	0.0535			23	3.12e-6				
	31	0.0259	31	0.0221							24	0.2434				
	58	0.0455	59	0.0394												
$\rho(x)$	3	0.0340	3	0.0592	5	0.2011	6	0.6623	6	0.2923	6	0.5084	7	0.6984	7	0.9667
	4	0.0634	4	0.0497	7	0.6084	7	0.0724	7	0.6073	7	0.3165	8	0.3016	20	0.0333
	6	0.2133	7	0.2295	13	0.1463	20	0.2653	20	0.1003	20	0.1751				
	8	0.4504	8	0.0310	16	0.0442										
	9	0.2091	10	0.5258												
	20	0.0298	20	0.1049												

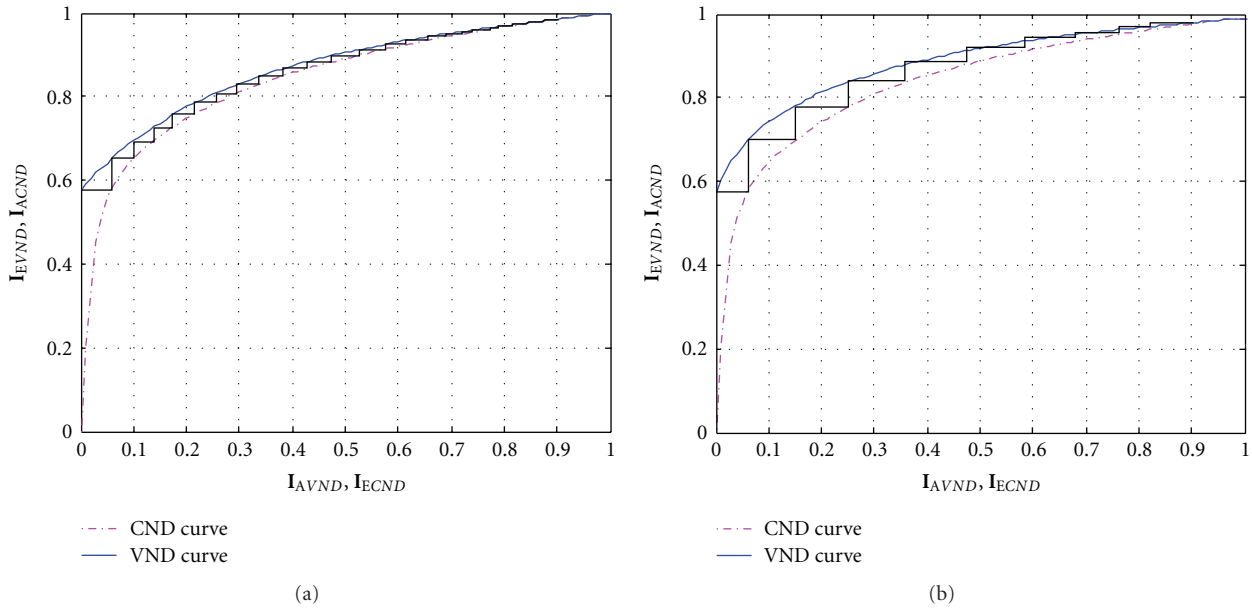


Figure 8: VND-CND decoding trajectory: (a) $C_{S_1}^{t_1}$ or $C_{S_2}^{t_1}$, and (b) C_{SR}.

5.2. BER Performance. Next, we will use an optimized C_{SR} code profile $(\lambda(x), \rho(x))$ in Table 2 to analyze performance in terms of Bit Error Rate (BER) in the AWGN and Rayleigh fading channels, respectively. In simulations, the soft decision information from the demodulator is input into the decoder. The parameters used in simulations are listed in Table 3. The time partition parameter $t_1 = 0.7$ (obtained as in Section 5.2) is chosen to maximize the network coding capacity of the DF MARC model. Codeword length is 10^4.

In a *decode-and-forward* cooperative strategy, R needs to decode information from sources correctly. This requires the entire codeword to be correctly transmitted. Therefore, codes should have excellent frame error ratios (FERs). To ensure the FER performance of LDPC codes, small circles in the parity-check matrix must be removed. Then, parity-check matrices of C_{SR}, $C_{S_1}^{t_1}$, and $C_{S_2}^{t_1}$ are randomly constructed by $\lambda(x)$ and $\rho(x)$), respectively. Accordingly, the girth of length 4 in the bipartite graph has been detected and removed.

Figure 10 shows the BER curve against the SNR at the different $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$) code rate. Obviously, with the help of the cooperative mechanism, the result has a great improvement of performance on BER, because R is near to D and provides almost a 1.2 dB increase in spatial diversity in low SNR in the AWGN channel at a $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$) rate of 0.5. In the AWGN channel at a $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$) rate of 2/3, the performance still improves by 1 dB. In such

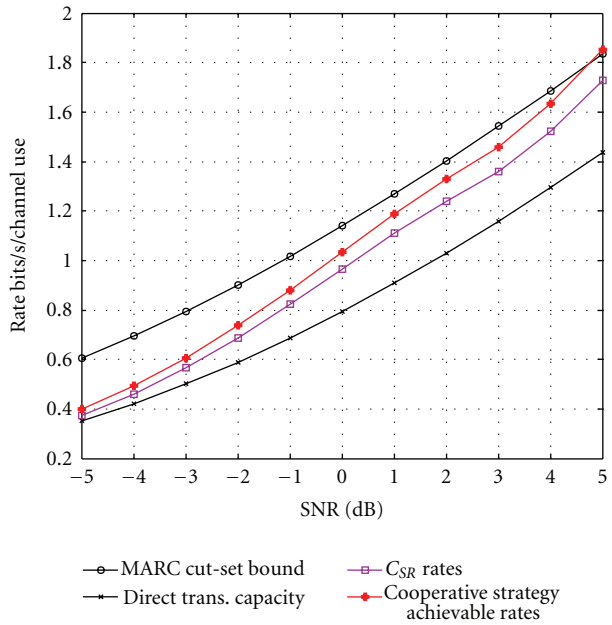


Figure 9: Achievable rates and capacity comparisons.

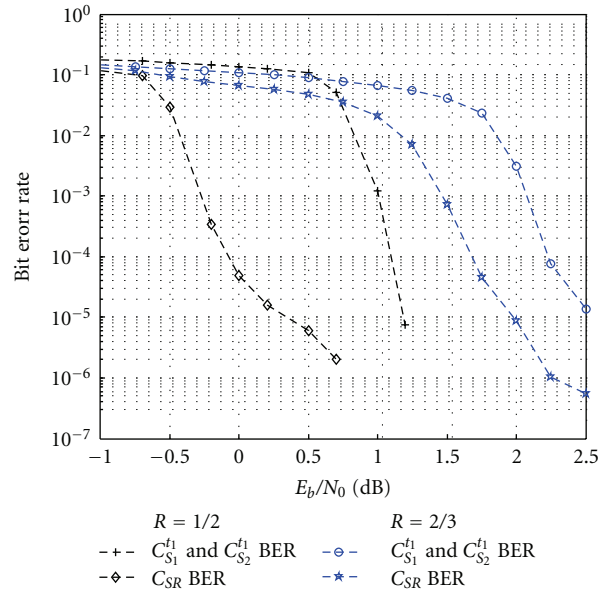


Figure 10: The BER performance under the AWGN channel with BPSK, $d = 0.5$.

Table 3: Parameters in simulation.

Time partition	$t_1 = 0.7$
Codeword length	$N_S = 10^4$
Distance between S and R	Distance = 0.5
Channel model	AWGN, Rayleigh Fasting Fading and Rayleigh Slow Fading
Power	$E[x^2] = 0$ dBw
Max iteration	100
Modulation	BPSK, QPSK
Decoding algorithm	BP

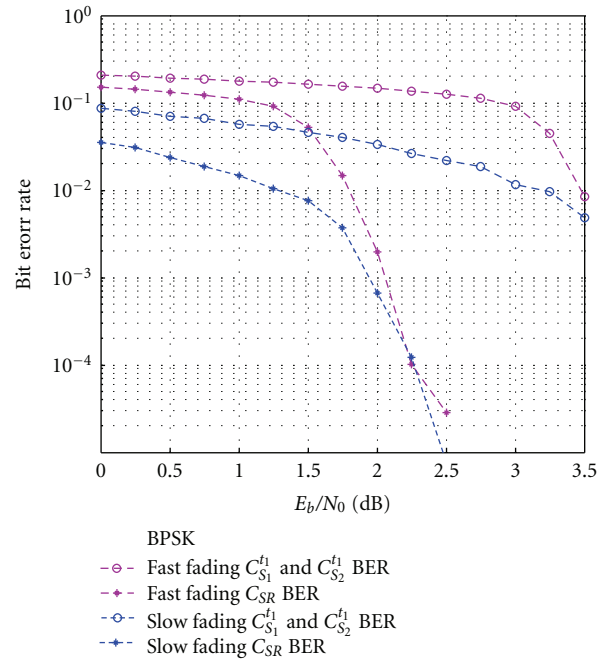


Figure 11: The BER performance under the Rayleigh Fading channel with BPSK, $d = 0.5$, and $R_{S_1R}^{t_1} = R_{S_2R}^{t_1} = 0.5$.

circumstances, the direct link between S and D cannot offer a service-satisfied physical layer QoS transmission, but with the cooperative relaying, the transmission will be employed again. The simulations demonstrate that this cooperative strategy has improved reliability, especially for the cases in low SNR.

In the Rayleigh channel as shown in Figures 11 and 12, the average gain in diversity is larger than 2 dB. Two kinds of modulation schemes are plotted to compare the performance: BPSK and QPSK. The BPSK scheme is shown to have a lower BER performance than the QPSK. Besides, the presented network-coded cooperative strategy has better BER performance under fast fading channel than that under slow fading channel. And it is concluded that in fast fading or mobile environment, the employment of a relay node indeed could provide effective spatial diversity.

We also intend to investigate the effects of the relay position factor d on BER performance. BER curves with $d = 0.1, 0.25$, $d = 0.35$, $d = 0.5$, $d = 0.6$, and $d = 0.8$ are plotted in Figure 13. The comparisons show that increasing d increases the performance of BER versus SNR. This is because the path loss of the RD channel decreases, which is easier to decode $C_R^{t_2}$, resulting in decoding of the extra check bits with a lower error rate. However, from Figure 5 in Section 3.1, when $d = 0.5$, the achievable sum-rate is optimal; the slope of the curve with $d > 0.5$ is larger than that of the curve with $d < 0.5$. In other words, as $d = 0.8$,

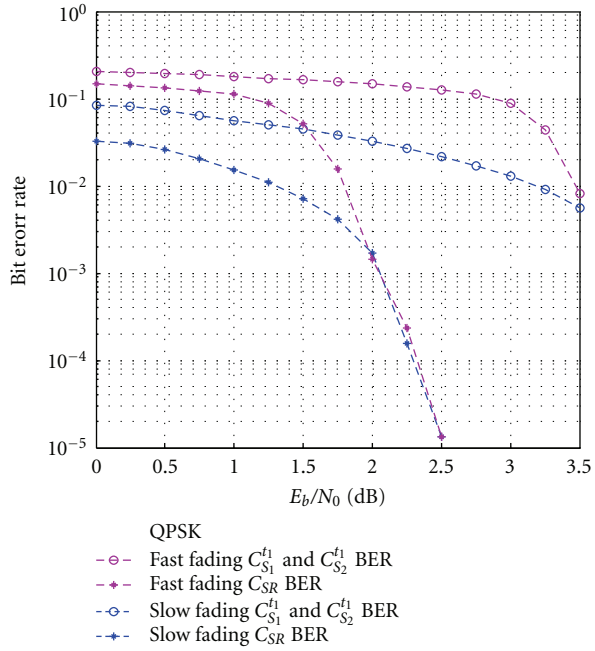


FIGURE 12: The BER performance under the Rayleigh Fading channel with QPSK, $d = 0.5$, and $R_{S_1R}^{t_1} = R_{S_2R}^{t_1} = 0.5$.

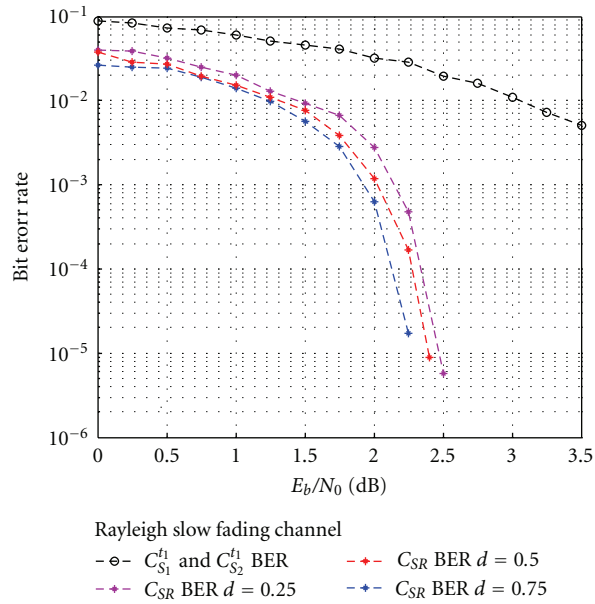


FIGURE 14: The BER performance under the Rayleigh Slow Fading channel with different settings of d, the distance between Source and Relay.

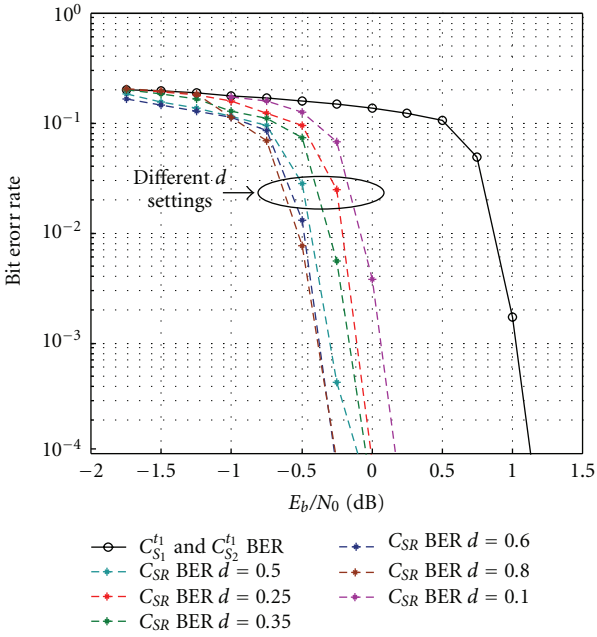


FIGURE 13: The BER performance under the AWGN channel with different settings of d, the distance between Source and Relay.

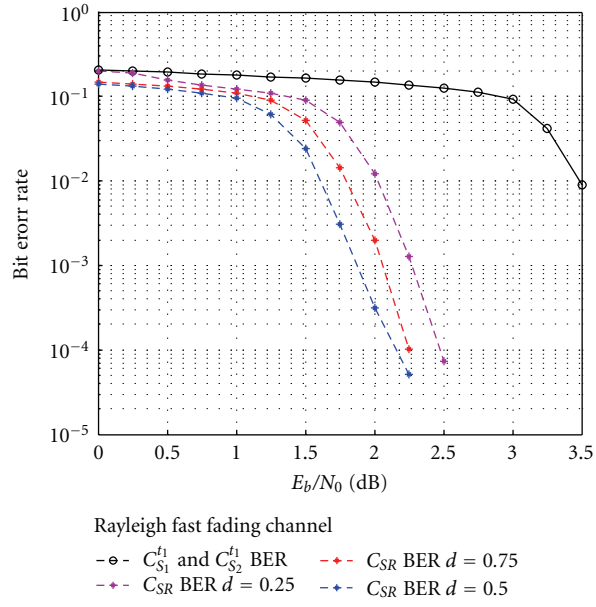


FIGURE 15: The BER performance under the Rayleigh Fast Fading channel with different settings of d, the distance between Source and Relay.

the achievable sum-rate is less than that as $d = 0.2$. Likewise, we also give the BER performance with different settings of the relay position factor d under Rayleigh Slow Fading and Rayleigh Fast Fading channels in Figure 14 and Figure 15. As a result, the spatial diversity and multiplexing can be balanced by the factor d in the parity-check network coding cooperative strategy.

Besides, we also investigate the effects of BER with different numbers of extra check bits under AWGN and Rayleigh Fading channels through Figure 16 to Figure 18. It is valid that the more extra bits are sent, the better the BER performances are, since the rate of cooperative code C_{SR} is reduced. Thereby, the spatial gain obtained by sending more extra check bits is at the cost of throughput of the whole

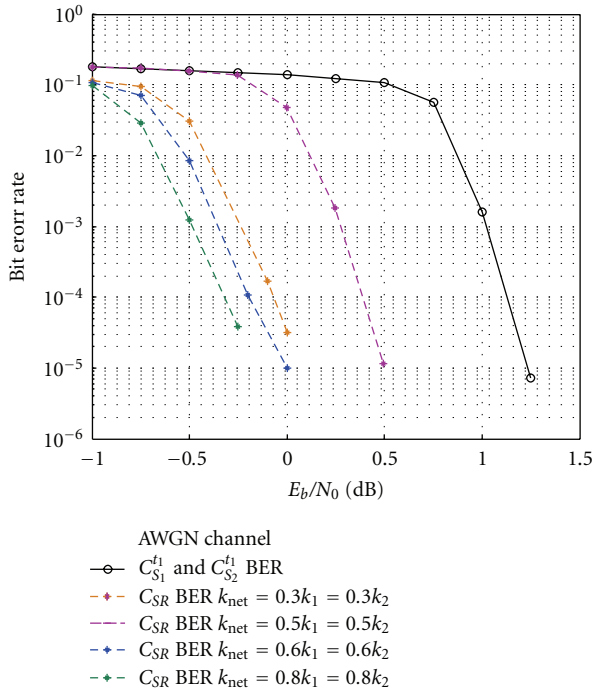


Figure 16: The BER performance under the AWGN channel with different lengths of extra check bits.

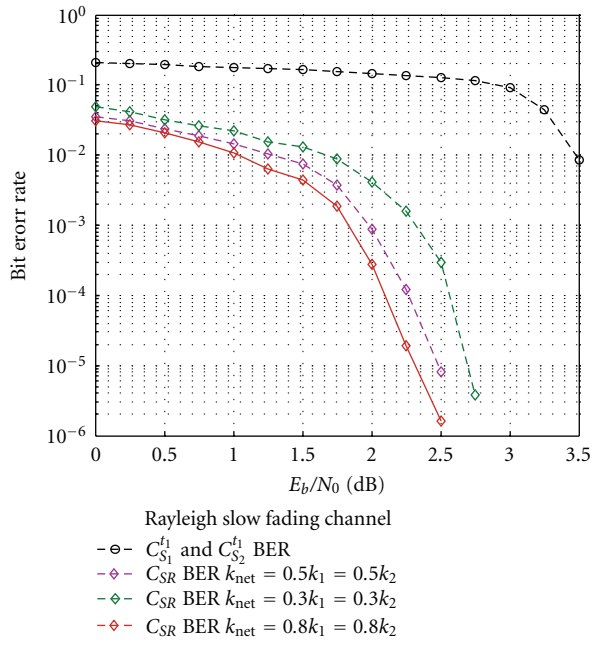


Figure 17: The BER performance under the Rayleigh Slow Fading channel with different lengths of extra check bits.

system. As a result, the spatial diversity and multiplexing can be balanced by maximizing the rate of the cooperative strategy to obtain optimal relay position d and optimal extra check bits length.

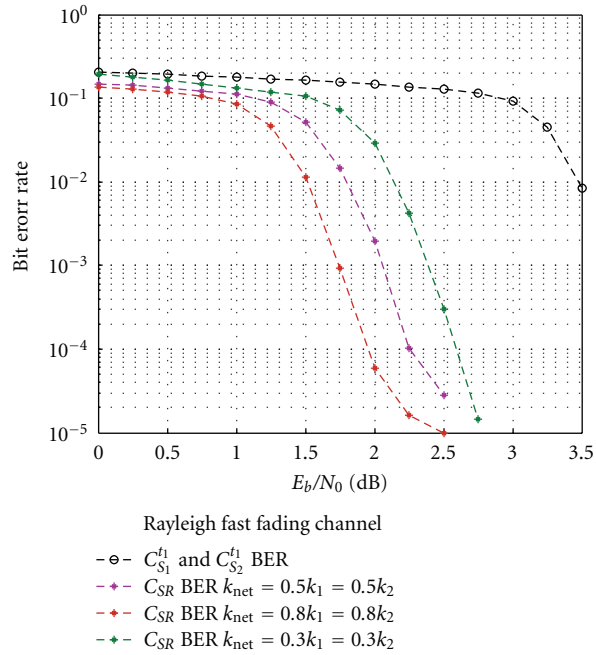


Figure 18: The BER performance under the Rayleigh Fast Fading channel with different lengths of extra check bits.

6. Conclusion

This study investigated a cooperative strategy based on parity-check network coding. The relative performance improvement of the schemes lies in a *decode-and-forward* strategy at the relay node. In particular, this study has revealed that a successful design should (1) employ the most effective extra check bits to make full use of the information contained in x_3 to help decode the messages from S_1 and S_2 and (2) perform linear network coding with the extra check bits. Specifically, we provide an implementation of parity-check network coding based on layered multidimensional LDPC code and a corresponding belief propagation decoding algorithm. The parity-check network coding for both sources removes the bandwidth loss that occurs in relaying, which is only 0.5 dB from the MARC DF "Cut-Set" capacity, and yet the parity-check bits ensures an attractive spatial diversity of cooperative communication. In the future, we would like to extend the proposed scheme to correlated multiple source nodes and conduct further research on network coding in GF(q) fields.

Appendix

Proof of Theorem 1. Rearrange the columns of the C_{SR} parity-check matrix according to the descending sequence of variable node degrees, such as $\{d_{v,SR}, d_{v,SR} - 1, \ldots, 3, 2\}$, and then successively deal with the numbers of variable nodes in each degree. The variable nodes in C_{SR} have two types of degrees, $\lambda_{i,j}^{SR}$, sub-LDPC degree i, $i \geq 2$ and extra degree j, $j \geq 0$. Therefore, the number of variable nodes with a specific degree d in C_{SR}, denoted by $N_d = (\lambda_d/d) \cdot E$, also

has two parts: $N_{d,j=0}$, the number of degree d without extra checks in $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$); $N_{i<d,j\neq 0,i+j=d}$, the number of turning into degree d after extra checks added in $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$).

For the maximum degree $d_{v,SR}$ in cooperative C_{SR}, let $d_{v,S} = \max(d_{v,S_1}, d_{v,S_2})$ and $d_{v,SR} \geq d_{v,S}$. $N_{d_{v,S}j=0}$ represents the number of variable nodes in $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$). Clearly, for the maximum degree $d_{v,SR}$,

$$N_{d_{v,SR}} = N_{d_{v,S},j=0} + \underset{i+j=d_{v,S}}{N_{i<d_{v,S},j\neq 0}}. \tag{A.1}$$

Hence, $N_{d_{v,SR}} \geq N_{d_{v,S}j=0}$ is tenable, and based on $N_d = (\lambda_d/d)\cdot E$, we have

$$\frac{\lambda_{d_{v,SR}}}{d_{v,SR}} E_{SR} \geq \left(\frac{\gamma_{d_{v,S}}}{d_{v,S}} E_{S_1} + \frac{\gamma_{d_{v,S}}}{d_{v,S}} E_{S_2}\right). \tag{A.2}$$

If $d_{v,\mathrm{SR}} = \max(d_{v,S_1}, d_{v,S_2})$, then $\gamma_{d_{v,S}} \neq 0$; if $d_{v,SR} > \max(d_{v,S_1}, d_{v,S_2})$, then $\gamma_{d_{v,S}} = 0$.

Next, considering degree $d_{v,SR}-1$, the number of variable nodes with degrees larger than $d_{v,SR}-1$ is

$$\begin{aligned} &N_{d_{v,SR}} + N_{d_{v,SR}-1} \\ &\quad = \left(N_{d_{v,S},j=0} + N_{d_{v,S}-1,j=0}\right) + \underset{i+j=d_{v,S},(d_{v,S}-1)}{N_{i<(d_{v,S}-1),j\neq 0}}, \end{aligned} \tag{A.3}$$

where $(N_{d_{v,S},j=0} + N_{d_{v,S}-1,j=0})$ is the number of variable nodes with degrees larger than $d_{v,SR}-1$ in $C_{S_1}^{t_1}$ (or $C_{S_2}^{t_1}$). Thus, based on $N_d = (\lambda_d/d)\cdot E$, we obtain

$$\begin{aligned} &\left(\frac{\lambda_{d_{v,SR}}}{d_{v,SR}} + \frac{\lambda_{d_{v,SR}-1}}{d_{v,SR}-1}\right) E_{SR} \\ &\qquad \geq \sum_{i=d_{v,SR},d_{v,SR}-1} \left(\frac{\gamma_{i,S_1}}{i} E_{S_1} + \frac{\gamma_{i,S_2}}{i} E_{S_2}\right). \end{aligned} \tag{A.4}$$

Then, for all degrees in the descending sequence in C_{SR}, it is confirmed that

$$\begin{aligned} &\left(N_{d_{v,SR}} + N_{d_{v,SR}-1} + \cdots + N_2\right) \\ &\quad = \left(\begin{array}{c} \left(N_{d_{v,S},j=0} + N_{d_{v,S}-1,j=0} + \cdots + N_{2,j=0}\right) \\ + \left(N_{\substack{i<(d_{v,S}-1),j\neq 0 \\ i+j=d_{v,S},(d_{v,S}-1)}}\right) + \left(N_{\substack{i<(d_{v,S}-2),j\neq 0 \\ i+j=d_{v,S},(d_{v,S}-1),(d_{v,S}-2)}}\right) \\ + \cdots + \left(N_{\substack{i=2,j\neq 0 \\ i+j=d_{v,S},(d_{v,S}-1),\ldots,3}}\right) \end{array}\right). \end{aligned} \tag{A.5}$$

Using the expression in terms of degree distribution $N_d = (\lambda_d/d)\cdot E_{SR}$ to replace N_d, we have

$$\begin{aligned} \sum_{i=j}^{d_{v,SR}} \frac{\left(\lambda_{i,j}^{SR}\right)}{i} E_{SR} &\geq \sum_{i=j}^{\max(d_{v,S_1},d_{v,S_2})} \left(\frac{\gamma_i^{S_1}}{i} E_{S_1} + \frac{\gamma_i^{S_2}}{i} E_{S_2}\right) \\ &\forall\, j = 2, 3, \ldots, d_{v,SR}. \end{aligned} \tag{A.6}$$

Therefore, the relationships of (13) hold under the cooperative constructions. □

Acknowledgments

This work is supported by the National Nature Science Foundation of China (no. 60532030) and National Outstanding Youth Science Fund (no. 60625102).

References

[1] L. Sankaranarayanan, G. Kramer, and N. B. Mandayam, "Hierarchical sensor networks: capacity bounds and cooperative strategies using the multiple-access relay channel model," in *Proceedings of 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, pp. 191–199, October 2004.

[2] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

[3] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Communications Magazine*, vol. 42, no. 10, pp. 74–80, 2004.

[4] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—part I: system description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.

[5] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[6] J. N. Laneman and G. W. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2415–2425, 2003.

[7] T. E. Hunter and A. Nosratinia, "Cooperation diversity through coding," in *Proceedings of IEEE International Symposium on Information Theory*, p. 220, July 2002.

[8] A. Chakrabarti, A. de Baynast, A. Sabharwal, and B. Aazhang, "Low density parity check codes for the relay channel," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 2, pp. 280–290, 2007.

[9] P. Razaghi and W. Yu, "Bilayer LDPC codes for the relay channel," in *Proceedings of IEEE International Conference on Communications (ICC '06)*, pp. 1574–1579, Istanbul, Turkey, July 2006.

[10] M. A. Khojastepour, *Distributed cooperative communications in wireless networks*, Ph.D. thesis, Rice University, Houston, Tex, USA , 2004.

[11] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[12] E. C. van der Meulen, "Three- terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, pp. 120–154, 1971.

[13] T. M. Cover and A. A. EL Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, 1979.

[14] R. C. King, *Multiple access channels with generalized feedback*, Ph.D. thesis, Stanford University, Stanford, Calif, USA, 1978.

[15] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: physical-layer network coding," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking(MOBICOM '06)*, pp. 358–365, September 2006.

[16] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: analog network coding," in *Proceedings of Conference on Computer Communications (SIGCOMM '07)*, pp. 397–408, August 2007.

[17] C. Hausl, F. Schreckenbach, I. Oikonomidis, and G. Bauch, "Iterative network and channel decoding on a tanner graph," in *Proceedings of the 43rd Annual Allerton Conference on Communications, Control, and Computing (GLOBECOM '05)*, Monticello, Va, USA, September 2005.

[18] X. Bao and J. Li, "Matching code-on-graph with networks-on-graph: adaptive network coding for wireless relay networks," in *Proceedings of the 43rd Annual Allerton Conference on Communications, Control, and Computing (GLOBECOM '05)*, Monticello, Ill, USA, September 2005.

[19] M. Effros, M. Medard, T. Ho, S. Ray, D. R. Karger, and R. Koetter, "Linear network codes: a unified framework for source, channel and network coding," in *Proceedings of DIMACS Workshop on Network Information Theory*, p. 179, Piscataway, NJ, USA, March 2003.

[20] Y. Chen, S. Kishore, and J. Li, "Wireless diversity through network coding," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '06)*, pp. 1681–1686, April 2006.

[21] L. Xiao, T. E. Fuja, J. Kliewer, and D. J. Costello Jr., "A network coding approach to cooperative diversity," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3714–3722, 2007.

[22] S. Yang and R. Koetter, "Network coding over a noisy relay : a belief propagation approach," in *IEEE International Symposium on Information Theory (ISIT '07)*, pp. 801–804, Nice, France, June 2007.

[23] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory," *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 4, pp. 241–329, 2006.

[24] M. A. Khojastepour, A. Sabharwal, and B. Aazhang, "On the capacity of 'cheap' relay networks," in *Proceedings of the 37th Annual Conference on Information Sciences and Systems*, Baltimore, Md, USA, March 2003.

[25] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[26] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[27] E. Sharon, A. Ashikhmin, and S. Litsyn, "Analysis of low-density parity-check codes based on EXIT functions," *IEEE Transactions on Communications*, vol. 54, no. 8, pp. 1407–1414, 2006.

[28] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 657–670, 2001.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 465915, 14 pages
doi:10.1155/2010/465915

Research Article

Data Dissemination in Wireless Sensor Networks with Network Coding

Xiumin Wang,[1,2] Jianping Wang,[2] and Yinlong Xu[1]

[1] *School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, 230027, China*
[2] *Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong*

Correspondence should be addressed to Jianping Wang, jianwang@cityu.edu.hk

Received 28 September 2009; Revised 5 March 2010; Accepted 25 April 2010

Academic Editor: Christina Fragouli

In wireless sensor networks (WSNs), it is often necessary to update the software running on sensors, which requires reliable dissemination of large data objects to each sensor with energy efficiency. During data dissemination, due to sleep scheduling designed for energy efficiency, some sensors may not receive some packets at some time slots. In the meantime, due to the unreliability of wireless communication, a sensor may not successfully receive a packet even when it is in the active mode. Thus, retransmission of such packets to those sensors is necessary, which consumes more energy and increases the delay of data dissemination cycle. In this paper, we propose a network coding-based approach in data dissemination such that data dissemination can be accomplished at the earliest time. Thus, less energy is consumed and the delay can be decreased. The impact of packet loss probability and the sleep probability of sensors on the network coding gain is analyzed. A threshold is also given to decide whether the current sleep scheduling is effective on energy saving in data dissemination process or not. Simulation results demonstrate the effectiveness and scalability of the proposed work.

1. Introduction

Recently, more research attention has been directed towards wireless sensor networks. Once deployed, sensors are expected to operate for extended periods of time, and it is impractical to physically reach all sensors. However, it is quite often necessary to update the software running on those sensors or add new functionality to the sensors [1–3]. Reprogramming the network needs to reliably disseminate large data objects (50–100 KB) to every sensor in the network with energy efficiency [2].

Protocols for reliably disseminating large data objects in WSNs have been developed over years. Protocols in [1–4] achieve data dissemination reliability through different mechanisms such as hop-by-hop recovery, NACKs or ACKs mechanisms, while another requirement of disseminating large objects in WSNs, energy efficiency, has not been well studied.

In WSNs, energy consumption is a critical issue and sleep scheduling has been well studied as a conservative approach to minimize the energy consumption due to *idle listening* [5, 6]. Though sleep scheduling can save energy, sensors in sleep mode cannot receive data packets. In addition, due to the unreliability of wireless communication, a sensor may not receive the packet successfully even when it is in active mode [7]. Hence, a data packet may be transmitted several times in order to be disseminated to all sensors, which wastes energy and increases the delay of the whole data dissemination process. In other words, the data dissemination process consists of sending native data packets and recovering "wanted" packets that each sensor has not received due to sleep scheduling and/or link unreliability. In order to complete the data dissemination process in a timely manner and achieve energy efficiency, it is crucial to assure that the maximum number of "wanted" packets at all sensors can be recovered at each time slot.

Recently, network coding has become a promising approach to improve the system throughput in wireless networks. Network coding with XORs operation in wireless broadcast has been studied in [8], which shows the advantage

of the proposed XORs coding scheme over the traditional wireless broadcast in the bandwidth efficiency through simulations and theoretical analysis. In XORs coding, a coded packet carries both the coding vector information and the encoded data. Thus, upon receiving a coded packet, the receiver knows which packets are encoded together and how to decode the packet with the available packets at the receiver. The work in [9] has proved that optimal XORs encoding decision for wireless broadcast, which decides the coding vector of each coded packet, is an NP-hard problem. Heuristic algorithms of encoding decision problem for wireless broadcast and multicast are proposed in [9, 10]. However, the proposed encoding decision approach can only be applied to the scenario where all receivers remain active during the whole time period of recovery. Such an approach can not be applied to WSNs with sleep scheduling because different sets of active sensors may be available at different time slots.

In this paper, given the sleep scheduling information at the sensors, we aim to determine an effective XORs encoding strategy such that the minimum number of transmissions is required in order for each sensor in the network to successfully receive the whole set of disseminated data packets. Thus, energy consumption can be reduced and the data dissemination process can be accomplished in a timely manner. To achieve such an objective, it is important to maximize the expected number of active sensors that can decode out one "wanted" data packet at each time slot in the recovery process, which is the focus of this paper. The contribution of the proposed work is summarized as follows.

(i) The proposed work takes both link unreliability and sleep scheduling into consideration and proposes an XORs encoding decision algorithm to maximize the expected number of active sensors that can decode out one native packet in their "wanted" data packet sets at each time slot in the recovery process.

(ii) We analyze the impact of each link's packet loss probability and each sensor's sleep probability at each time slot on the network coding gain, which is an extension of the analysis given in [8].

(iii) We also study the effectiveness of sleep scheduling on energy saving, which is offsetted by the total number of active time slots consumed in the data dissemination process. A threshold is derived to decide whether the current sleep scheduling is effective on energy saving or not. The simulation results also confirm the accuracy of our analysis.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. Section 3 introduces the system architecture and data dissemination schemes. The problem description and its complexity is presented in Section 4. Section 5 describes the algorithm design. Theoretical analysis is given in Section 6. Section 7 gives the simulation results. Finally, we conclude the paper in Section 8.

2. Related Work

In this section, we review the related work of network coding in WSNs. Network coding is originally proposed in information theory [11] and recently has become a promising approach to improve the system throughput in wireless networks [11–16]. Adaptive network coding is proposed in [17] to reduce traffic in the process of software updates where linear network coding technique is used. As computation ability and the memory at sensor nodes are very limited, the complexity of linear encoding and decoding introduces extra overhead. Thus, it is more appropriate to use XORs operation in WSNs since both encoding and decoding operations are much simpler. In fact, XORs coding has been widely used in wireless networks to reduce the complexity of linear network coding [8, 10, 18, 19].

COPE proposed in [18] improves the throughput of unicast with XORs coding. By exploiting the broadcast nature of wireless medium, each node buffers overheard packets for a short time and notifies its neighbors which packets it has heard. When a node transmits a packet, it uses its knowledge of what its neighbors have heard to perform *opportunistic coding* and XORs multiple packets to transmit them as a single packet while ensuring that each intended next-hop has enough information to decode the encoded packet.

Network coding with XORs operation in wireless broadcast has also been studied in [8], which shows the advantage of the proposed network coding scheme over traditional wireless broadcast in bandwidth efficiency through simulations and theoretical analysis. However, encoding decision has not been given in [8]. The work in [9] has proved that optimal XORs encoding decision problem for wireless broadcast is an NP-hard problem.

Several heuristic algorithms for encoding decision in wireless broadcast and multicast have been proposed in [9, 10]. With the knowledge of the "wanted" packet set at each receiver, an auxiliary graph is constructed. The encoding decision during the recovery process is then converted to a clique partition problem in the auxiliary graph. However, the proposed encoding decision algorithms can only be applied to the scenario where all receivers remain active during the time period of recovery. Such an approach cannot be applied to WSNs since different set of active sensors may be available at different time slots. Thus, encoding decision in WSNs with sleep scheduling cannot be converted into finding a minimum clique partition in the graph.

The work in [20] proposes a retransmission scheme, which only uses reception estimation to determine the coding set selection. However, the reception estimation at the source node may not be accurate enough, consequently, some receivers may not be able to decode useful information from the coded packet and more retransmissions will be needed. In addition, the coding decision based on reception estimation does not consider the impact of sleep scheduling, which affects the decoding probability at the receivers in low duty-cycled WSNs.

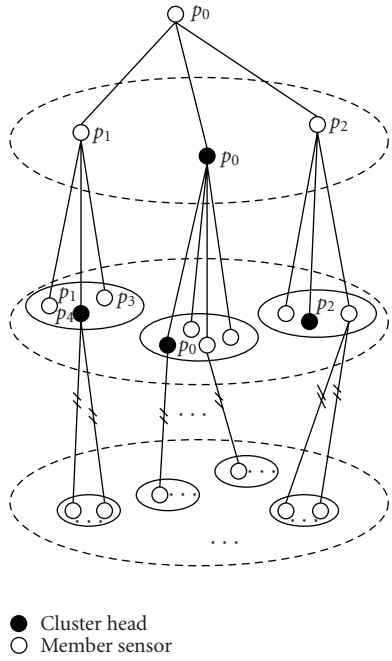


Figure 1: Hierarchical architecture.

In this paper, we propose to use XORs coding in data dissemination in a large scale WSN which is organized as a multihop cluster hierarchy [21]. A multihop cluster hierarchical architecture consists of multiple layers as shown in Figure 1. In the lowest layer, all the nodes in the network are grouped into clusters. In addition, besides being a member in a cluster, a node may act as a cluster head in a down layer cluster, for example, p_2 in the figure. Within each cluster, the cluster head communicates with its member sensors in a one-hop fashion [22]. We also assume that each sensor is aware of its one-hop neighbors' sleep scheduling and the reliability of the wireless links between the sensor to its neighbors. This can be easily accomplished by one-hop information exchange and link loss inference [23].

3. System Architecture and Data Dissemination with Network Coding

Our data dissemination process is conducted at each cluster head so as to make sure that finally all the sensors obtain the updating packets. In a multihop cluster hierarchy, if a cluster head in an intermediate layer starts to transmit the received packet immediately after receiving one fresh packet, the gain of network coding cannot be fully utilized. On the other hand, if a cluster head waits and starts to transmit packets until it receives all packets from the cluster head in the upper layer, it will waste bandwidth and introduce extra delay. In order to achieve the balance between bandwidth efficiency and network coding gain, we propose to use a threshold α to determine when the current cluster head starts to transmit the packets to its member nodes. Specifically, for each cluster head, after obtaining $\alpha M'$ fresh native packets, where $0 < \alpha \leq 1$ and M' is the number of native packets available at its upper-layer cluster head, it will conduct XORs coding scheme to transmit the packets to its member nodes. In the simulation part, we will study the impact of the threshold α on the delay and energy consumption.

In the rest of the paper, we focus on how a cluster head encodes the packets and transmits them to its member sensors. The coding decision at other cluster heads can use the same approach.

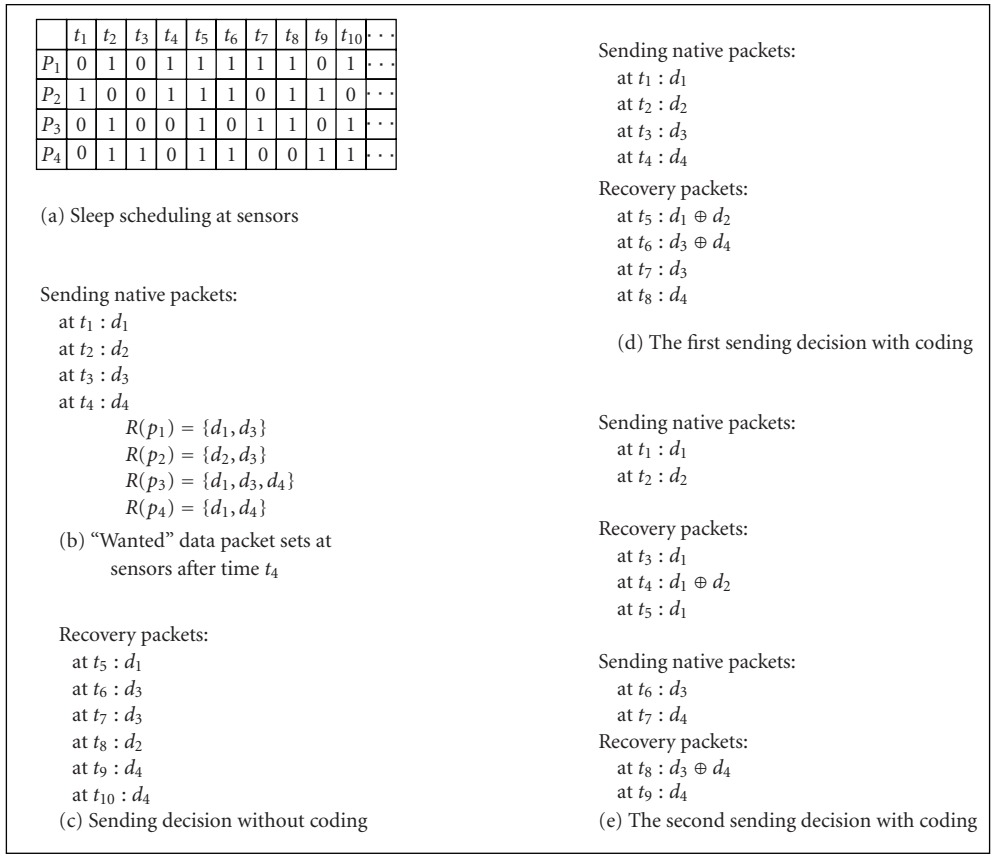
As we mentioned earlier, the data dissemination process consists of sending native data packets and recovering "wanted" packets for each receiver. We now give an example to show that network coding can indeed recover "wanted" packets for all neighbors more efficiently.

Suppose that four packets d_1, d_2, d_3, and d_4 need to be transmitted to sensors p_1, p_2, p_3 and p_4 as shown in Figure 2. The sleep scheduling at each receiver is given in Figure 2(a) where 1 denotes that this sensor is active at the current time slot, otherwise, it is in sleep mode. For the sake of simplicity, in this example, we assume that no packet is lost due to unreliable wireless communication, which means that a sensor can receive a packet successfully when it is in active mode. We also assume that an active sensor can only transmit or receive one packet at each time slot [5]. We show that different data dissemination approaches will lead to different finishing time of data dissemination.

(i) Without network coding, 4 native packets will be sent firstly, followed by sending native packets to recover "wanted" packets at sensors. Figure 2(b) gives the "wanted" data packet set at each sensor after 4 native packets are sent out. Without network coding, it will take 10 time slots to finish the data dissemination process as shown in Figure 2(c).

(ii) With network coding, 4 native packets can be sent at first followed by sending encoded packets to recover "wanted" packets at sensors. Assume that our coding strategy at each time slot is to maximize the number of active receivers that can decode the encoded packet. For example, at time t_5, if $d_1 \oplus d_2$ is sent, all four receivers can obtain a "wanted" packet by $d_1 \oplus (d_1 \oplus d_2)$ or $d_2 \oplus (d_1 \oplus d_2)$. Eventually, it will take 8 time slots to finish the data dissemination process as shown in Figure 2(d). Under such a data dissemination approach, as all native packets are sent at first, the available packets at sensors are most diversified. Thus, the best network coding gain can be achieved. This, however, means that each sensor needs to buffer all received native packets in order to decode out "wanted" packets, which might not be feasible in a WSN due to limited memories at sensors.

(iii) An alternative approach will be to divide the data dissemination process into several batches where in each batch, M native packets are sent followed by the recovering process [24]. Once all M native packets are received by all sensors in the cluster, the cluster head proceeds to transmit the following batch of packets. The data dissemination is accomplished when all batches of packets are obtained by all sensor nodes in the network. In Figure 2(e), we send two native



	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	$\cdots$
P_1	0	1	0	1	1	1	1	1	0	1	$\cdots$
P_2	1	0	0	1	1	1	0	1	1	0	$\cdots$
P_3	0	1	0	0	1	0	1	1	0	1	$\cdots$
P_4	0	1	1	0	1	1	0	0	1	1	$\cdots$

Figure 2: Comparison with different dissemination schemes.

packets at first, followed by sending encoded packets to recover "wanted" packets of the first batch at sensors, then send the last two native packets followed by sending encoded packets to recover "wanted" packets of the second batch at sensors. It takes 9 time slots to finish the data dissemination process.

We now discuss how the cluster head can maintain "wanted" packet set at each member sensor. After sending out a packet, the cluster head needs to collect the "wanted" packet set at each member sensor. In order to reduce ACKs implosion, only the active receivers that have received a packet at current time slot successfully and can obtain/decode one "wanted" packet from the received packet will send an ACK message to the cluster head. Thus, according to ACKs from receivers, the cluster head can derive the "wanted" packet set for each active receiver.

With the information of "wanted" packet set of each receiver at each time slot in the recovery process, an encoding decision which aims to maximize the expected number of active sensors that can decode out one "wanted" packet at current time slot will be introduced in the following section.

4. Problem Description and Complexity

In this section, we first describe the encoding decision problem that aims to decide which native packets should be encoded at each time slot t in the recovering process such that the maximum expected number of active sensors at time slot t can decode out one "wanted" native packet. Thus, we limit our discussion to the recovery process of one data dissemination batch in a cluster, which can also be applied to other batches in all other clusters.

Suppose that $D = \{d_1, d_2, \ldots, d_M\}$ is the set of data packets in a batch which need to be disseminated to all the sensors in a cluster. Let $P_t = \{p_{i_1}, p_{i_2}, \ldots, p_{i_l}\}$ be the set of active member sensors in the cluster at tth time slot At each time slot, the cluster head can obtain its neighbor sensors' "wanted" packet set based on ACKs feedback. Let $r_{i,j}$ be 1 if packet d_j is not available at active sensor p_i at current time slot where $d_j \in D$, otherwise, let it be 0. Let $R(p_i) = \{d_j \mid r_{i,j} = 1 \text{ and } p_i \in P_t\}$ be the "wanted" data packet set of active sensor p_i at current time slot t as shown in Figure 2(b). Assume that l_i is the probability that sensor p_i can not successfully receive a packet from the cluster head when p_i is in active mode.

Let a_j be 1 if native packet $d_j \in D$ is combined in current encoded packet, otherwise, let it be 0. Let $c_{i,j}$ be 1 if active sensor p_i can decode out one "wanted" native packet d_j from the current encoded packet where $d_j \in R(p_i)$, otherwise, let it be 0. Considering unreliable wireless communication, the probability that an active sensor p_i can successfully obtain one "wanted" packet at the current time slot is $\sum_{j=1}^{M} c_{i,j}(1 - l_i)$. Thus, at current time slot, the expected

number of sensors that can decode out one "wanted" packet is $\sum_{i\in\{i|p_i\in P_t\}}\sum_{j=1}^{M} c_{i,j}(1-l_i)$, which needs to be maximized in order to save energy.

Still take Figure 2(d) as an example, after t_4, the cluster head starts to recover the "wanted" packets at its member sensors. At t_5, if the cluster head sends an encoded packet $d_1 \oplus d_2$, in an ideal condition where no packet will be lost, active receivers p_1, p_2, p_3, p_4 can decode out one "wanted" packet by $d_1 \oplus (d_1 \oplus d_2)$ or $d_2 \oplus (d_1 \oplus d_2)$. Assume that $l_1 = 0.1$, $l_2 = 0.2$, $l_3 = 0.3$, $l_4 = 0.15$ in a practical wireless network where the probability of successfully receiving a packet at p_1, p_2, p_3, and p_4 is $0.9, 0.8, 0.7$ and 0.85 respectively due to unreliable wireless communication. Thus, the expected number of active receivers that can decode out one "wanted" packet after receiving the current encoded packet $d_1 \oplus d_2$ is $0.9 * 1 + 0.8 * 1 + 0.7 * 1 + 0.85 * 1 = 3.25$, which is maximum at the current time slot. Thus, the cluster head will send out $d_1 \oplus d_2$ at the current time slot. In this paper, such an encoding decision problem using XORs coding is referred to as *network coding based data dissemination* (NCDD) problem.

4.1. Problem Formulation. We can formally formulate the NCDD problem at time slot t in the recovery process as follows:

$$Z = \max \sum_{p_i\in P_t}\sum_{j=1}^{M} c_{i,j} * (1-l_i) \tag{1}$$

subject to

$$\sum_{j=1}^{M} c_{i,j} \le 1, \quad \forall i \in \{i' \mid p_{i'} \in P_t\}. \tag{2}$$

$$c_{i,j} \le a_j * r_{i,j}, \quad \forall i \in \{i' \mid p_{i'} \in P_t\},\ \forall 1 \le j \le M, \tag{3}$$

$$a_{j'} * r_{i,j'} + c_{i,j} \le 1, \quad i \in \{i' \mid p_{i'} \in P_t\},\ \forall 1 \le j \ne j' \le M, \tag{4}$$

$$a_j \le \sum_{i\in\{i'|p_{i'}\in P_t\}} r_{i,j}, \quad \forall 1 \le j \le M, \tag{5}$$

$$a_j, c_{i,j} \in \{0,1\}, \quad \forall i \in \{i' \mid p_{i'} \in P_t\},\ \forall 1 \le j \le M. \tag{6}$$

In the above formulation, the term of the objective represents the expected number of active receivers that can decode out one "wanted" data packet from the encoded packet at the current time slot. Equations (2) and (6) ensure that each receiver can only decode out at most one "wanted" native packet from the encoded packet. Equations (3) and (4) give two requirements that active receiver p_i can decode out one "wanted" packet d_j: (1) packet d_j is in p_i's "wanted" packet set and d_j is participated in the encoded packet; (2) all other combined native packets except d_j in the encoded packet have already been successfully received by receiver p_i. Equation (5) guarantees that if packet d_j is available at all active receivers at current time slot t, d_j must not be combined into the encoded packet.

4.2. Problem Complexity

Theorem 1. *NCDD problem is NP-hard.*

Proof. We prove the theorem by a reduction from MAXIMUM ONE-IN-THREE SAT problem which is a well known NP-hard problem in the strong sense.

MAXIMUM ONE-IN-THREE SAT: We are given a set $U = \{u_1, u_2, \ldots, u_M\}$ of M boolean variables and a collection $C = \{c_1, c_2, \ldots, c_n\}$ of clauses with exactly three literals. Each of these clauses is a boolean formula and it is true if and only if exactly one of its three literals is true. Without loss of generality, we assume that the three literals in c_i are $\{u_{i_1}, u_{i_2}, u_{i_3}\}$. The objective of MAXIMUM ONE-IN-THREE SAT is to find a truth assignment such that the maximum number of clauses is true. We use OPT^s to denote the optimal solution of this problem.

Given an instance of MAXIMUM ONE-IN-THREE SAT, we can construct an instance of the decision version of the NCDD problem in polynomial time as follows. Let there be M data packets needed to be disseminated from the cluster head to n receiver nodes. If $u_j = 1$, packet d_j is participated in encoding, otherwise, d_j is not participated in encoding. For each clause c_i, if u_j is a literal of c_i, then d_j is a "wanted" packet at p_i. In other words, each sensor p_i has lost exactly three packets and has all other packets. Let the probability that an active sensor can successfully receive a packet be 100%. Then, our objective is to maximize $\sum_{i\in\{i|p_i\in P_t\}}\sum_{j=1}^{M} c_{i,j}$ For a given encoded packet, p_i can decode a new native packet if and only if exactly one native packet in R_i is encoded into the new encoded one. The problem is to find an encoding strategy to maximize the number of receivers which can decode out one "wanted" packet from the encoded packet. We use OPT^p to refer to the result of this objective.

(i) Suppose that there is a true assignment for MAXIMUM ONE-IN-THREE SAT with the maximum number of clauses. If c_i is true, there must be exactly one true assignment for $\{u_{i_1}, u_{i_2}, u_{i_3}\}$. Without loss of generality, we assume that u_{i_2} is true while u_{i_1}, u_{i_3} are both false. According to the construction of the instance, only d_{i_2} is participated in encoding while neither d_{i_1} nor d_{i_3} is participated in encoding. In other words, only one lost packet of p_i is participated in encoding and p_i has all other packets involved in encoding, thus, p_i can decode out one "wanted" native packet d_{i_2}. Therefore, if there is a clause which is true in the MAXIMUM ONE-IN-THREE SAT problem, there must be a receiver which can obtain a "wanted" native packet. Then, we have $OPT^s \le OPT^p$.

(ii) Suppose that there is an encoding strategy such that the maximum number of receivers can decode the new native packet. Assume that p_i can decode a new native packet d_{i_2} from the encoded one. According to the decoding strategy, the other two "wanted" packets d_{i_1}, d_{i_3} must not be encoded into the new one, that is, u_{i_1}, u_{i_3} both have false assignment while u_{i_2} is true.

In this assignment, c_i also has a true value. So, we have $OPT^p \leq OPT^s$.

The above analysis shows that $OPT^p = OPT^s$. Thus NCDD problem is NP-hard. □

5. Algorithm for NCDD Problem

In this section, we first introduce an auxiliary graph in which each vertex is assigned a weight. We then show that the proposed NCDD problem can be converted into finding a maximum weight clique problem in the auxiliary graph, based on which we develop a heuristic algorithm for the NCDD problem.

5.1. Model Design. At any tth time slot, let $R(p_i) \subseteq D$ be the set of packets "wanted" by p_i and $H(p_i) \subseteq D$ be the set of packets received by p_i. We can construct an auxiliary graph $G(V, E)$ similar to [9] where $V = \{v_{i,j} \mid d_j \in R(p_i)$ and $p_i \in P_t\}$, which means that every "wanted" packet of each active sensor has a vertex in G. Considering two receivers p_{i_1} and p_{i_2}, if they have lost the same packet d_j, then they can both recover d_j if only native packet d_j is encoded at current time slot. We use a link $e \in E$ between $v_{i_1,j}$ and $v_{i_2,j}$ to denote such recoverability. If d_{j_1} is a "wanted" packet of p_{i_1} and $d_{j_1} \in H(p_{i_2})$, while d_{j_2} is a "wanted" packet of p_{i_2} and $d_{j_2} \in H(p_{i_1})$, then p_{i_1} can recover d_{j_1} when it receives $d_{j_1} \oplus d_{j_2}$ and p_{i_2} can recover d_{j_2} when it receives $d_{j_1} \oplus d_{j_2}$. We use a link $e \in E$ between v_{i_1,j_1} and v_{i_2,j_2} to denote such recoverability. In other words, $E = \{(v_{i_1,j_1}, v_{i_2,j_2}) \mid d_{j_2} \in H(p_{i_1})$, $d_{j_1} \in H(p_{i_2})$ or $j_1 = j_2, i_1 \neq i_2\}$ where $p_{i_1}, p_{i_2} \in P_t$.

For a clique $Q = \{v_{i_1,j_1}, v_{i_2,j_2}, \ldots, v_{i_k,j_k}\}$ in the graph, let $P' = \{p_i | v_{i,j} \in Q, 1 \leq j \leq M\}$ be the sensors which have "wanted" packets in Q and $D' = \{d_j \mid v_{i,j} \in Q, 1 \leq i \leq n\}$ be the set of "wanted" packets of those sensors in Q. Suppose that there are m' packets in D'. For any vertex $v_{i,j} \in Q$, according to the edge assignment of G, p_i must have already successfully obtained the packets in $D' - \{v_j\}$ but still requires packet v_j. Thus, if $d_{j_1} \oplus d_{j_2} \oplus \cdots \oplus d_{j_{m'}}$ where $d_{j_1}, d_{j_2}, \ldots, d_{j_{m'}} \in D'$ are encoded and sent at tth time slot, each sensor in P' will be able to decode out one "wanted" packet if the encoded packet can be successfully received by all sensors in P'. To consider the unreliability of wireless communication, we assign weight $w_{i,j} = 1 - l_i$ in the vertex $v_{i,j}$ for any $j \in \{j \mid v_{i,j} \in V\}$. Then the weight for clique Q which is defined in

$$w(Q) = \sum_{(i,j) \in \{(i,j) | v_{i,j} \in Q\}} w_{i,j}, \tag{7}$$

is equivalent to the expected number of active sensors which can successfully decode out one "wanted" packet if all packets in D' are encoded together. Thus, our NCDD problem which aims to maximize the expected number of active sensors that can decode out one "wanted" packet is converted into finding a maximum weight clique in graph G.

For example, after the whole 4 native packets are sent, the "wanted" packet set in Figure 2(b) can be constructed into Figure 3. Thus, the encoding decision for recovery process at t_5 is then converted into finding a maximum weight clique in such a graph. As shown in Figure 3, the clique that consists of $\{v_{1,1}, v_{2,2}, v_{3,1}, v_{4,1}\}$ is the clique with the maximum weight $0.9 + 0.8 + 0.7 + 0.85 = 3.25$. After the encoded packet $d_1 \oplus d_2$ is sent, active receivers p_1, p_2, p_3, p_4 can decode out d_1, d_2, d_1, d_1, respectively, if all sensors successfully receive $d_1 \oplus d_2$.

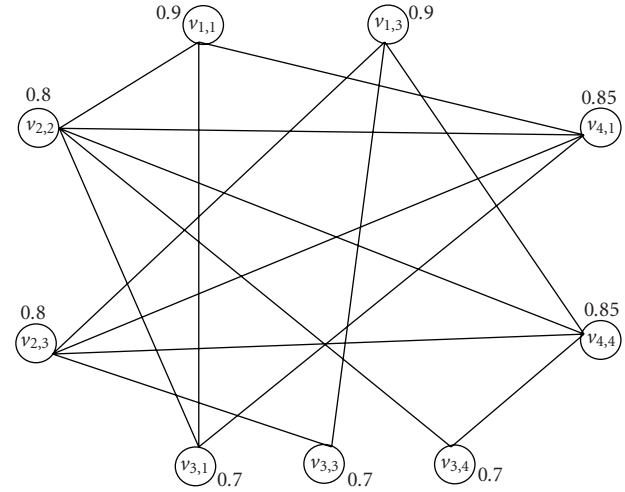


Figure 3: Graph model.

5.2. Algorithm Design. Assume that the total number of vertices in $G(V, E)$ is N. We first sort all vertices into nonincreasing order according to $w_{i,j}$. For the example given in Figure 3, vertices in G will be sorted into $V = \{v_{1,1}, v_{1,3}, v_{4,1}, v_{4,4}, v_{2,2}, v_{2,3}, v_{3,1}, v_{3,3}, v_{3,4}\}$.

For the simplicity of presentation, we abuse the notation a little bit and assign a unique id v_k for each vertex in G, which uses one-dimensional subscript for vertices in G instead of using two-dimensional subscripts. Correspondingly, we use w_k to denote the weight of v_k. Thus, for the example given in Figure 3, we have $V = \{v_1(v_{1,1}), v_2(v_{1,3}), v_3(v_{4,1}), v_4(v_{4,4}), v_5(v_{2,2}), v_6(v_{2,3}), v_7(v_{3,1}), v_8(v_{3,3}), v_9(v_{3,4})\}$. Without loss of generality, we assume that $V = \{v_1, v_2, \ldots, v_N\}$ where $w_1 \geq w_2 \geq \cdots \geq w_N$.

Let Q_i be the clique with maximum weight in the subgraph which only contains vertices of $S_i = \{v_i, v_{i+1}, \ldots, v_N\}$ and let $C(Q_i)$ be the weight of clique Q_i. In other words, Q_i represents the maximum weight clique the algorithm has found considering of the subgraph consisting of vertices $\{v_i, v_{i_1}, \ldots, v_N\}$. The algorithm starts with $i = N$ and iteratively considers more vertices until all vertices in G are considered. The algorithm stops when Q_1 is found.

When we consider vertex v_{i-1}, there are two cases. If $Q_i \cup \{v_{i-1}\}$ is also a clique, then $Q_{i-1} = Q_i \cup \{v_{i-1}\}$ and $C(Q_{i-1}) = C(Q_i) + w_{i-1}$, otherwise, if $Q_i \cup \{v_{i-1}\}$ is not a clique, we need to find out a clique Q_{i-1} that includes v_{i-1} in the subgraph consisting of $S_{i-1} = \{v_{i-1}, v_i, \ldots, v_N\}$. Let $N(v_{i-1})$ be the set of neighbors of vertex v_{i-1}. Initially, $Q_{i-1} = \{v_{i-1}\}$ and $S_{i-1} = N(v_{i-1}) \cap S_{i-1}$. If S_{i-1} is not $\emptyset$, let j be the smallest j such that $v_j \in S_{i-1}$. We add v_j to the clique, that is, $Q_{i-1} = Q_{i-1} \cup \{v_j\}$,

```
Function wclique(S_i, C(Q_i), i)
  if |S_i| = ∅
    if C(Q_i) > max
      max = C(Q_i);
      mc = i;
    return
  while S_i ≠ ∅
    j = min{j | v_j ∈ S_i};
    Q_i = Q_i ∪ {v_j};
    C(Q_i) = C(Q_i) + w_j;
    S_i = (S_i − {v_j}) ∩ N(v_j);
    if S_i = ∅
      if C(Q_i) > max
        max = C(Q_i); mc = i;
  return
Function MWC
  Q_i = ∅, i ∈ {1,...,N − 1}; Q_N = {v_N};
  C(Q_i) = 0, i ∈ {1,2,...,N − 1};
  C(Q_N) = w_N;
  max = C(Q_N);
  S_i = {v_i, v_{i+1},...,v_N}, i ∈ {1,2,...,N};
  for i = N − 1 down to 1
    if {v_i} ∪ Q_{i+1} is also a clique;
      Q_i = Q_{i+1} ∪ {v_i};
      C(Q_i) = C(Q_{i+1}) + w_i;
    else
      Q_i = {v_i};
      C(Q_i) = w_i;
      wclique(S_i ⋂ N(v_i), w_i, i);
      if mc ≠ i
        Q_i = Q_{i+1}; C(Q_i) = C(Q_{i+1});
  return
```

Algorithm 1: Maximum weight clique algorithm.

and update S_{i-1}, that is, $S_{i-1} = S_{i-1} \cap N(v_j)$. If S_{i-1} is still not $\emptyset$, we then add another vertex whose index is the smallest in S_{i-1} into the clique Q_{i-1}. We repeat this process until there is no vertex in S_{i-1}, that is, $S_{i-1} = \emptyset$. By comparing the weight of the clique Q_i without including v_{i-1} and the weight of the clique Q_{i-1} including v_{i-1}, the clique Q_{i-1} with maximum weight in the subgraph including vertices in $\{v_{i-1}, v_i, \ldots, v_N\}$ is set to be the one with the larger weight. The detail of the algorithm is given in Algorithm 1. After this algorithm, Q_1 gives all vertices in the found maximum weight clique. All native packets involved in Q_1 will be encoded together and be sent out at current time slot.

We now show how to find the maximum weight clique of the graph shown in Figure 3. Assume that Q_2 has been found, which consists of $\{v_2, v_4, v_6\}$. Next, we will consider Q_1. Since $\{v_1\} \cup Q_2$ is not a clique, we need to find Q_1 which includes vertex v_1 in the subgraph consisting of $S_1 = \{v_1, v_2, \ldots, v_9\}$. The corresponding steps for finding such Q_1 is given in Algorithm 2 where $v_k(v_{i_1,j_1})$ in V denotes that we use a unique id v_k in the algorithm to replace the original vertex v_{i_1,j_1}. After Q_1 is found, we compare it with Q_2 which has the weight $C(Q_2) = 2.55$. Since $C(Q_1)$ is larger than $C(Q_2)$, the clique $Q_1 = \{v_1, v_3, v_5, v_7\}$ is the maximum weight clique found in graph G. Vertices in Q_1 indicate that p_1, p_3 and p_4 lost packet d_1 and p_2 lost packet d_2. The encoding decision will be to send $d_1 \oplus d_2$.

6. Analysis

In this section, we firstly analyze the impact of packet loss probability and sleep probability on network coding gain. Then, we derive a threshold to decide whether the current sleep scheduling can save energy compared with no sleep scheduling. We only limit the analysis to one cluster in the multihop cluster hierarchy.

6.1. Impact of Packet Loss Probability and Sleep Probability on Network Coding Gain. Suppose that N_a is the number of transmissions that the data dissemination process requires without coding and N_b is the number of transmissions required with XORs coding. Assume that the probability that receiver p_i is in sleep mode is s_i at each time slot, and l_i is the probability that receiver p_i can not successfully receive a packet even when it is in active mode due to unreliable wireless communication. We have the following two lemmas.

Lemma 1. *The total number of transmissions without coding required for transmitting sufficient large M packets to n receivers is*

$$N_a = \sum_{i_1,i_2,\ldots,i_n} \frac{(-1)^{i_1+i_2+\cdots+i_n-1}}{1-(l_1')^{i_1}(l_2')^{i_2}\cdots(l_n')^{i_n}} M, \tag{8}$$

where $i_1, i_2, \ldots, i_n \in \{0,1\}$ and $\exists i_j \neq 0$, $l_i' = 1-(1-l_i)(1-s_i)$.

Proof. See Appendix A. □

Lemma 2. *The total number of transmissions with XORs coding for transmitting sufficient large M packets to n receivers is*

$$N_b = \frac{M}{\min_{i\in\{1,2,\ldots,n\}}\{(1-l_i)(1-s_i)\}}. \tag{9}$$

Proof. See Appendix B. □

With the analytical result of N_a and N_b, we can define analytical network coding gain as

$$\gamma = \frac{N_a - N_b}{N_a}. \tag{10}$$

Take two receivers as an example, assume that $l_1 = 0.1$, $l_2 = 0.25$, $s_1 = 0.15$, $s_2 = 0.05$ and M is sufficient large. According to (8) and (9), we can calculate that $N_a = 1.6382\,\mathrm{M}$, $N_b = 1.4035\,\mathrm{M}$. Then, the analytical network coding gain is $\gamma = 0.1433$.

From Lemmas 1 and 2, we can also obtain the following corollary.

Corollary 1. *With two receivers, the maximum network coding gain γ can be achieved if $l_1' = l_2'$, that is, $(1-l_1)(1-s_1) = (1-l_2)(1-s_2)$.*

Proof. See Appendix C. □

$V = \{v_1(v_{1,1}), v_2(v_{1,3}), v_3(v_{4,1}), v_4(v_{4,4}), v_5(v_{2,2}), v_6(v_{2,3}), v_7(v_{3,1}), v_8(v_{3,3}), v_9(v_{3,4})\}$
Step 1:
$Q_1 = \{v_1\}$, $S_1 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$, $N(v_1) = \{v_4, v_5, v_7\}$ $C(Q_1) = 0.9$
$S_1 = S_1 \cap N(v_1) = \{v_3, v_5, v_7\}$ $v_j = v_3$
$Q_1 = Q_1 \cup \{v_3\} = \{v_1, v_3\}$ $C(Q_1) = 0.9 + 0.85 = 1.75$
Step 2:
$N(v_3) = \{v_1, v_5, v_6, v_7\}$
$S_1 = S_1 \cap N(v_3) = \{v_5, v_7\}$ $v_j = v_5$
$Q_1 = Q_1 \cup \{v_5\} = \{v_1, v_3, v_5\}$ $C(Q_1) = 1.75 + 0.8 = 2.55$
Step 3:
$N(v_5) = \{v_1, v_3, v_4, v_7, v_9\}$
$S_1 = S_1 \cap N(v_5) = \{v_7\}$ $v_j = v_7$
$Q_1 = Q_1 \cup \{v_7\} = \{v_1, v_3, v_5, v_7\}$ $C(Q_1) = 2.55 + 0.7 = 3.25$
Step 4:
$N(v_7) = \{v_1, v_3, v_5\}$
$S_1 = S_1 \cap N(v_5) = \Phi$
Terminate;

Algorithm 2: The steps of finding Q_1.

6.2. Impact of Sleep Probability on Energy Consumption. Though sleep scheduling can save energy consumption due to *idle listening*, sensors in sleep mode cannot receive data packets, which imposes retransmission and may consume more energy. If sensor p_i is active at tth time slot, we say that tth time slot is an active time slot for sensor p_i. We know that only at its active time slot, sensor p_i consumes its energy. Thus, we can use the total number of active time slots consumed for the sensors to successfully receive the whole set of packets as the energy consumption for data dissemination.

We define a threshold as follows:

$$\varepsilon = \sum_{i=1}^{n}(1 - s_i) - \frac{n}{l_{\min}} \min_{i\in\{1,2,\ldots,n\}} \{(1 - l_i)(1 - s_i)\}, \quad (11)$$

where $l_{\min} = 1 - \max_{i\in\{1,2,\ldots,n\}}\{l_i\}$.

Then, we have the following lemma.

Lemma 3. *In XORs coding, if $\varepsilon < 0$, the current sleep scheduling can save energy consumed by idle listening; otherwise, the current sleep scheduling has no contribution to energy saving.*

Proof. See Appendix D. □

Take two receivers with $l_1 = 0.23$, $s_1 = 0.15$, $l_2 = 0.27$, $s_2 = 0.18$ as an example, according to (11), we have $\varepsilon > 0$. Thus, the energy saving with sleep scheduling is offsetted by more retransmissions. In this case, the cluster head should wake up more sensors. An interesting problem is how to design an optimal sleep scheduling such that energy saving of sleep scheduling will not be offsetted by more retransmission, which is out of the scope of this paper.

7. Simulation Results

In this section, we demonstrate the effectiveness of our dissemination schemes through simulations using C++ simulator. In our simulations, a multihop cluster hierarchical WSN is randomly generated with the fixed value of the number of sensors if without specification. We group the packets required to send into batches, and each batch has M packets. Recovery process with network coding starts after every M native packets are transmitted. In a cluster, we randomly generate sensor p_i's sleep scheduling according to its sleeping probability s_i.

To demonstrate the advantage of our coding scheme, we introduce two baseline algorithms, namely, *dissemination without coding* algorithm and *dissemination with random_coding* algorithm. *Dissemination without coding* algorithm randomly transmits a native "wanted" packet at each time slot until all receivers obtain their "wanted" data packets while *dissemination with random_coding* algorithm transmits an XORs packet which is randomly generated at each time slot until all receivers obtain their "wanted" packets.

In the simulation, we are interested in evaluating the performance of our coding schemes from the following perspectives.

(i) The number of active receivers that can obtain a new "wanted" packet at one time slot and the total number of transmissions required in one batch data dissemination within one cluster.

(ii) The impact of the number of receiver sensors n, batch size M, sleep probability and packet loss probability on the network coding gain under different dissemination schemes within one cluster.

(iii) How close the performance of our proposed algorithms is to the derived analytical results within one cluster.

(iv) The impact of the threshold α on the delay and the total number of transmissions required in a multihop cluster hierarchy.

For each setting, we simulate 150 instances and report the average performance.

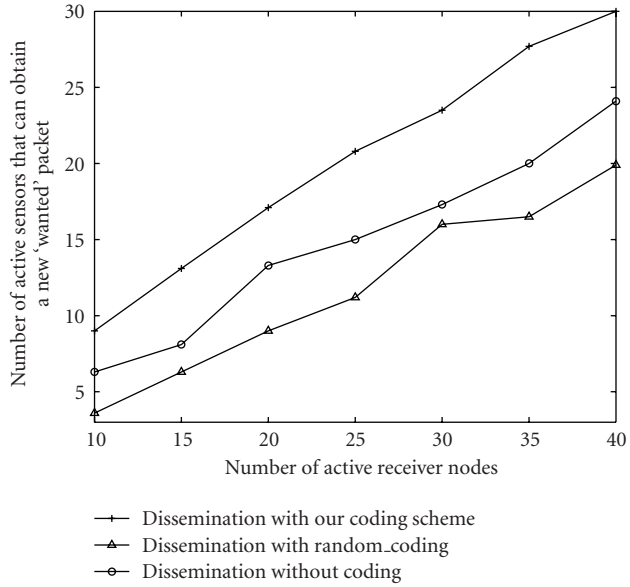


Figure 4: The number of receivers that can obtain a new "wanted" packet at a time slot versus the number of receiver nodes.

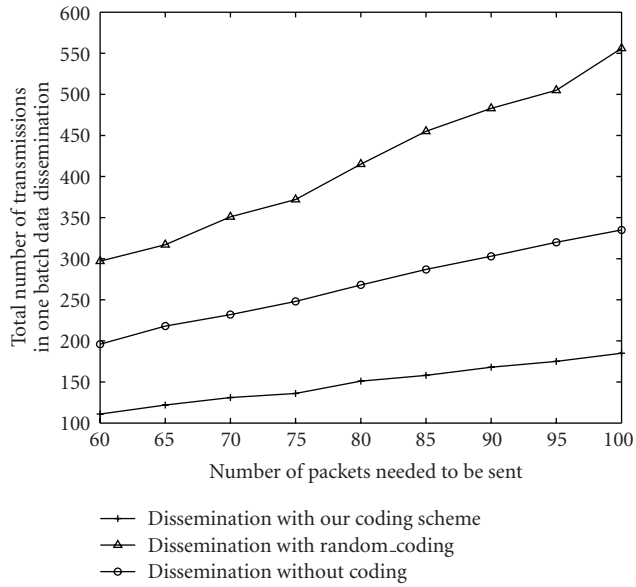


Figure 5: Total number of transmissions versus number of packets needed to be sent.

7.1. Comparison with Different Data Dissemination Schemes. The effectiveness of our coding scheme for maximizing the expected number of sensors that can obtain one "wanted" packet at one time slot is demonstrated by comparing with *dissemination without coding* algorithm and *dissemination with random_coding* algorithm.

We evaluate the performance of our algorithms by varying the number of active sensors within a cluster at one time slot in the range of $[10, 40]$ for $M = 50$, and $l_i = 0.2$. As shown in Figure 4, the number of active sensors that can obtain one "wanted" packet by our coding scheme is much more than that by *dissemination without coding* algorithm and *dissemination with random_coding* algorithm.

For one batch data dissemination process within a cluster, to demonstrate the performance of our coding scheme, the total number of transmissions required is also compared with the other two baseline algorithms: *dissemination without coding* and *dissemination with random_coding* algorithms. We vary the number of packets needed to be sent in the range of $[60, 100]$ for $n = 10$, $s_i = 0.3$, $l_i = 0.2$. As shown in Figure 5, the total number of transmissions required in one batch dissemination by our coding scheme is much less than that by *dissemination without coding* and *dissemination with random_coding* algorithms. Hence, for data dissemination with a large set of packets, our XORs coding scheme can efficiently decrease the number of transmissions required. Thus, more energy can be saved.

7.2. Network Coding Gain Comparison with Analytical Results. We demonstrate the effectiveness of the proposed network coding algorithm by comparing the network coding gain obtained through simulation with the analytical network coding gain.

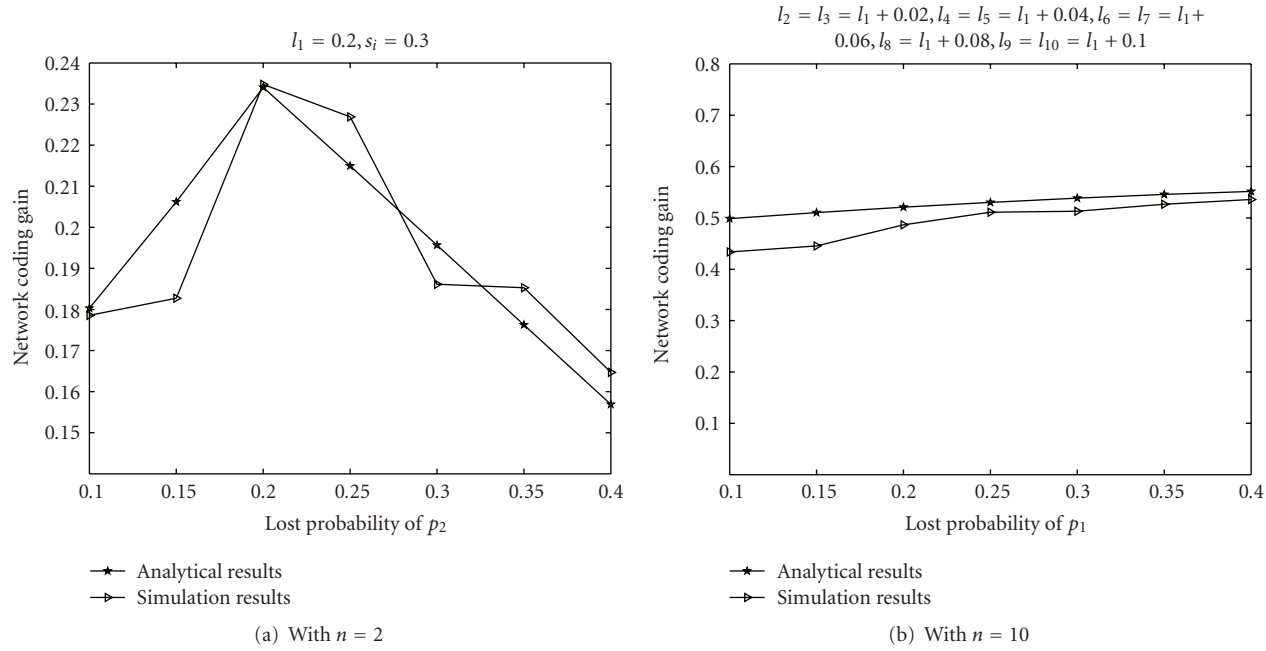
We start with a simple experiment where there are only two members sensors in a cluster. We fix l_1 to 0.2 and vary l_2 in the range of $[0.1, 0.4]$ for $n = 2, s_i = 0.3, M = 100$. As shown in Figure 6(a), the network coding gain obtained by our simulation follows the same trend as the analytical results. In addition, the maximum network coding gain is achieved when $l_1' = l_2' = 1-(1-0.2)(1-0.3) = 0.44$ with both our simulation results and analytical results, which verifies Corollary 1. When $l_1' = l_2'$, most likely the "wanted" packets at one receiver are the packets available at another receiver, thus, coding opportunity is high, which achieves maximum network coding gain.

We also extend the simulation to 10 receivers in a cluster. The loss probability of p_1 is varied along the x-axis for $M = 100$; $s_i = 0.2$, $1 \le i \le 5$; $s_i = 0.3$, $6 \le i \le 10$ and $l_2 = l_3 = l_1 + 0.02$, $l_4 = l_5 = l_1 + 0.04$, $l_6 = l_7 = l_1 + 0.06$, $l_8 = l_1 + 0.08$, $l_9 = l_{10} = l_1 + 0.1$. As shown in Figure 6(b), the simulation results are very close to the analytical results. In addition, Figure 6 verifies that network coding indeed can bring gains on reducing the number of transmissions required.

In Figure 7, we vary the sleep probability at sensors, similar results as Figure 6 can be observed and the network coding gain obtained through simulations is quite close to the analytical results.

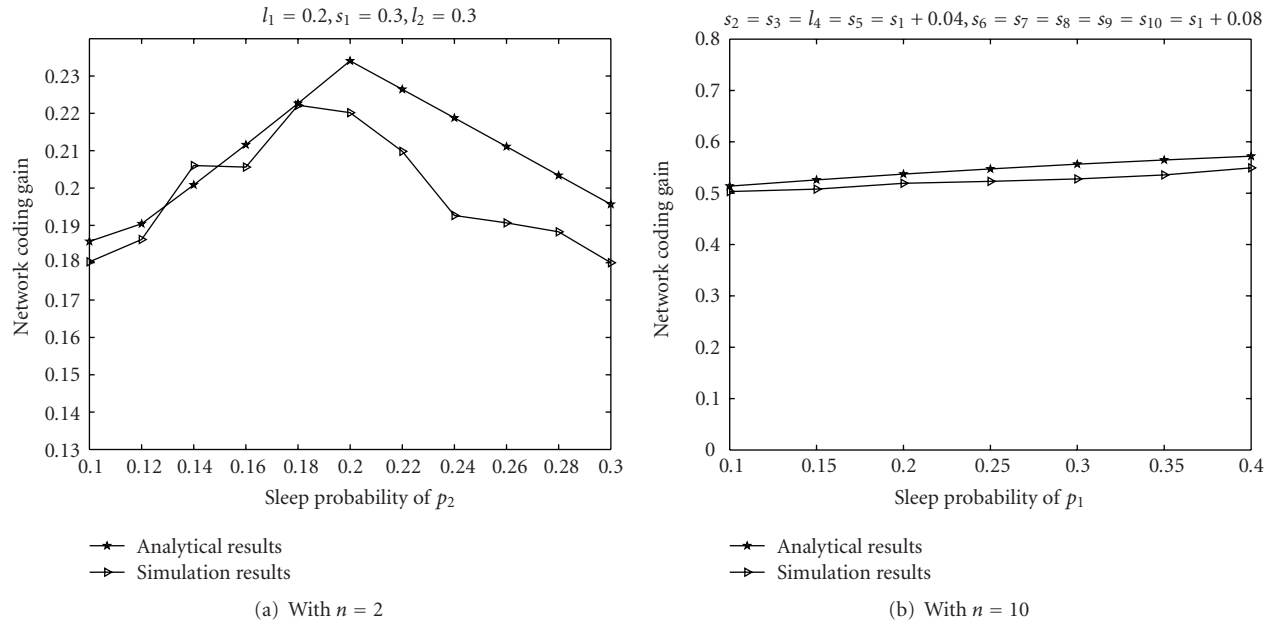
7.3. The Impact of Sleep Scheduling on Energy Saving. We now study the impact of sleep scheduling on the energy consumption. Our simulation is conducted within one cluster. We use the total number of active time slots consumed to denote the energy consumption in data dissemination process.

Suppose that XORs coding is applied. Let η_s be the total number of active time slots consumed for data dissemination with sleep scheduling and η_{ns} be the total number of transmissions for data dissemination without sleep scheduling.



(a) With $n = 2$ (b) With $n = 10$

Figure 6: Network coding gain versus packet loss probability of sensor.



(a) With $n = 2$ (b) With $n = 10$

Figure 7: Network coding gain versus sleep probability of sensor.

The energy saving in XORs coding with sleep scheduling over that without sleep scheduling is

$$\delta = \frac{\eta_{ns} - \eta_s}{\eta_{ns}}. \tag{12}$$

For data dissemination without coding, we can define energy saving with sleep scheduling over that without sleep scheduling in a similar way.

We evaluate the performance of our algorithm by varying s_2 in $[0, 0.3]$ for $n = 2$, $l_1 = 0.1$, $l_2 = 0.25$, $s_1 = 0.15$, $M = 50$. As shown in Figure 8, the simulation results are very close to the analytical results.

For our XORs coding, from the figure, we know that the energy consumption with sleep scheduling is less than that without sleep scheduling when s_2 is less than 0.15. When $s_2 = 0.15$, the energy consumption with sleep scheduling is equal to that without sleep scheduling. When s_2 is larger than 0.15, sleep scheduling has no contribution to the energy saving, it even incurs more energy consumption than that without sleep scheduling. This interesting result is plausible

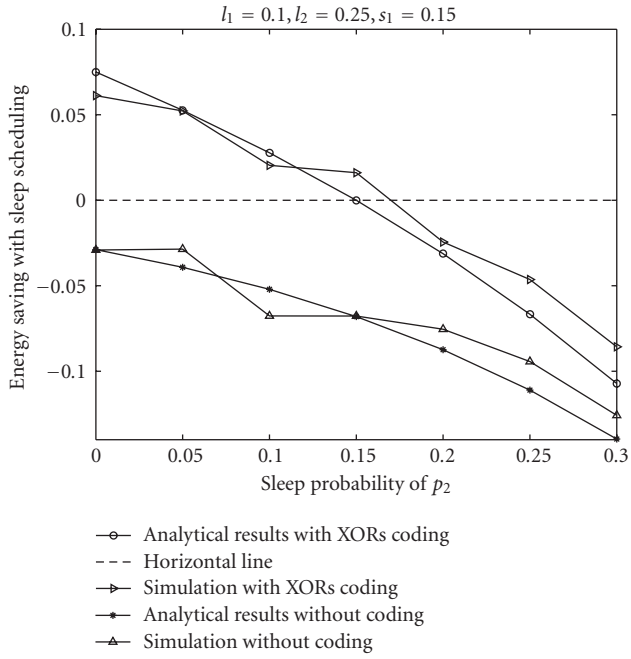


Figure 8: Energy saving versus sleep probability of sensor p_2.

since when the number of sleep sensors becomes larger, more retransmissions are required, which imposes more energy consumption. In this case, the energy saving with sleep scheduling is offsetted by more retransmissions, which means that the threshold $\varepsilon > 0$ and the cluster head should wake up more sensors to receive packets in order to save energy.

7.4. The Impact of Threshold α on the Delay and the Total Number of Transmissions Required. We now study the impact of threshold α on the delay of the data dissemination process in a multihop cluster hierarchical WSN. The threshold α is varied in the range of $[0.2, 1.0]$ for $M = 30, 40, 50$. Figure 9 gives the delay required for data dissemination when the number of layers is 5 and 6, respectively. We can see that the delay increases with the threshold α. This is because the cluster heads need to wait more time before they can transmit their available packets to their members with the increasing of α. Thus, the cluster heads in down layers can do nothing for a long time. Specifically, when $\alpha = 1$, each cluster head cannot transmit its available packets until receiving all M packets. In this case, concurrent transmissions cannot be allowed even if there is no collision between them, which thus increases the delay. From Figure 9, we can also see that the delay increases with the number of layers, because the number of receivers increases with the number of layers.

We further study the impact of the threshold α on the total number of transmissions required under a multihop cluster hierarchical WSN. The threshold α is also varied in the range of $[0.2, 1.0]$ for $M = 30, 40, 50$. As shown in Figure 10, the total number of transmissions required decreases with the threshold α. When α is small, the cluster heads transmit the packets to their members more quickly. Therefore, the number of fresh packets available at cluster heads is small, which can not fully utilize the network coding gain. Hence, the total number of transmissions required is more than with larger threshold α.

8. Conclusion

This paper studies data dissemination in wireless sensor networks with network coding to achieve energy efficiency. In order to quickly complete the whole process of data dissemination, at each time slot in the recovery process, we aim to transmit an encoded packet such that the expected number of active sensors that can decode out one "wanted" packet is maximized. A maximum weight clique model is proposed here to achieve such an objective. We further study the impact of packet loss probability and sleep probability on network coding gain. We also analyze the impact of sleep probability on energy saving gain and derive a threshold which can be used to decide whether the current sleep scheduling is effective on energy saving or not. The simulation results verify the work proposed in the paper.

Appendices

A. Proof of Lemma 1

According to [8], we can obtain that the total number of transmissions without coding to successfully deliver sufficient large M packets to n receivers is $\sum_{i_1,i_2,\ldots,i_n} ((-1)^{i_1+i_2+\cdots+i_n-1}/(1-(l_1)^{i_1}(l_2)^{i_2}\cdots(l_n)^{i_n}))M$, where each receiver keeps in active mode during data transmission process.

However, in the data dissemination process, receiver sensor p_i may be in sleep mode and can not successfully receive a packet. Therefore, the probability that sensor p_i can successfully receive the packet at any time slot is $(1-l_i)(1-s_i)$. In other words, the probability that sensor p_i will lose the packet is $1-(1-l_i)(1-s_i)$. Thus, considering sleep scheduling, the total number of transmissions required without coding is

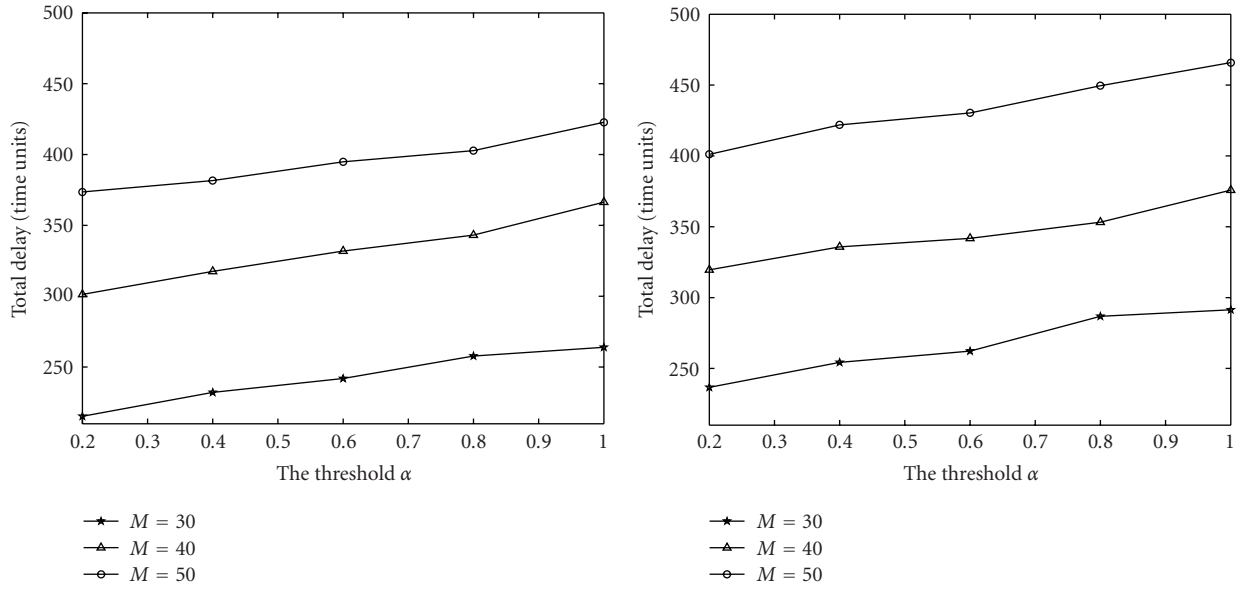
$$N_a = \sum_{i_1,i_2,\ldots,i_n} \frac{(-1)^{i_1+i_2+\cdots+i_n-1}}{1-(l_1')^{i_1}(l_2')^{i_2}\cdots(l_n')^{i_n}} M, \tag{A.1}$$

where $i_1, i_2, \ldots, i_n \in \{0,1\}$ and $\exists i_j \neq 0, l_i' = 1-(1-l_i)(1-s_i)$.

B. Proof of Lemma 2

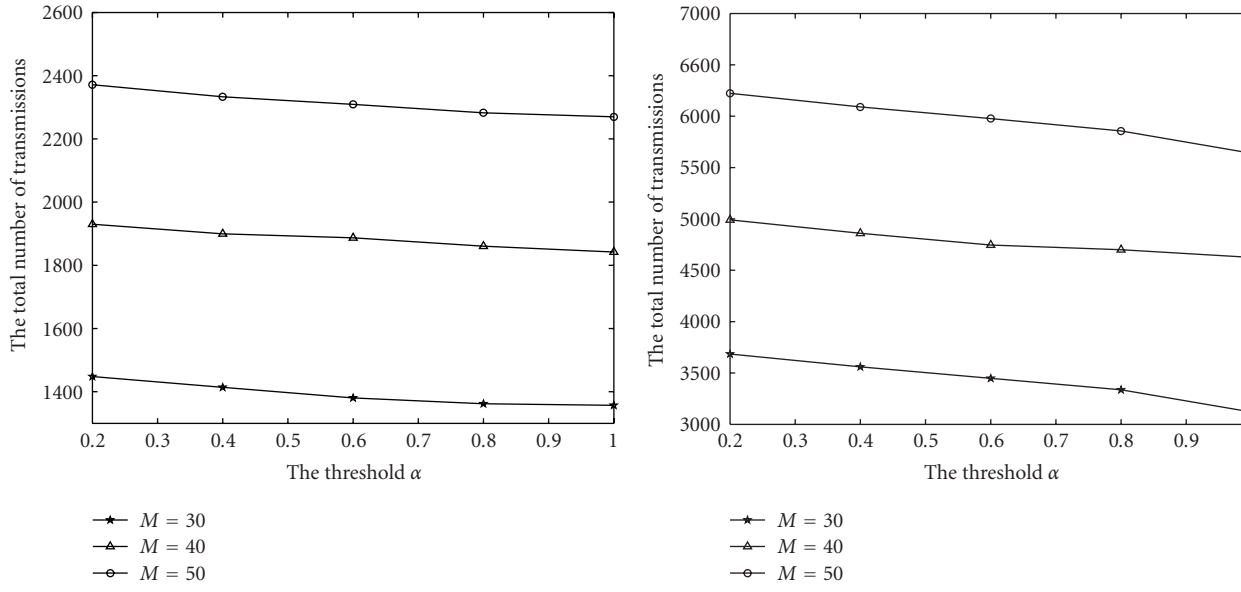
From [8], we know that the total number of transmissions with XORs coding to successfully deliver sufficient large M packets to n receivers is $M/(1-\max_{i\in\{1,2,\ldots,n\}}\{l_i\})$, where each receiver keeps in active mode during the data transmission process.

As in Appendix A , the probability that sensor p_i can not successfully receive the packet with sleep scheduling is changed into $1-(1-l_i)(1-s_i)$. Thus, the total number



(a) Hierarchy with 5 layers (b) Hierarchy with 6 layers

Figure 9: The total delay (time units) versus the threshold α.



(a) Hierarchy with 5 layers (b) Hierarchy with 6 layers

Figure 10: The total number of transmissions required versus the threshold α.

of transmissions required with XORs coding to transmit sufficient large M packets to n receivers is

$$N_b = \frac{M}{1 - \max_{i\in\{1,2,\ldots,n\}}\{1-(1-l_i)(1-s_i)\}} = \frac{M}{\min_{i\in\{1,2,\ldots,n\}}\{(1-l_i)(1-s_i)\}}, \quad \text{(B.2)}$$

C. Proof of Corollary 1

With two receivers, from Lemma 1, the total number of transmissions required for M packets without coding is $N_a = M/(1-l_1') + M/(1-l_2') - M/(1-l_1'l_2')$, and from Lemma 2, the total number of transmissions with XORs coding is $N_b = M/\min\{1-l_1', 1-l_2'\}$, where $l_i' = 1-(1-l_i)(1-s_i)$.

Without loss of generality, suppose that $l_1' \geq l_2'$ and $l_2' = \beta l_1'$, $0 \leq \beta \leq 1$. We have

$$\begin{aligned}\gamma &= \frac{N_a - N_b}{N_a} \\ &= 1 - \frac{N_b}{N_a} \\ &= 1 - \frac{1/(1-l_1')}{1/(1-l_1') + 1/(1-l_2') - 1/(1-l_1'l_2')} \\ &= 1 - \frac{1}{1 + (1-l_1')/(1-\beta l_1') - (1-l_1')/(1-\beta l_1'^2)},\end{aligned} \tag{C.3}$$

Define a function $f(\beta) = \gamma$ with β being the variable. We can easily prove that $f(\beta)$ is an increasing function. Thus, when β is 1, the value of function $f(\beta)$ is maximum. That is when $l_1' = l_2'$, the network coding gain γ is maximum, which proves our Corollary 1.

D. Proof of Lemma 3

From the analysis in the previous section, we can see that the total number of active time slots consumed for data dissemination with XORs coding is

$$\begin{aligned}\eta_s &= N_b \sum_{i=1}^{n} (1 - s_i) \\ &= \frac{M \sum_{i=1}^{n} (1 - s_i)}{\min_{i \in \{1,2,\ldots,n\}} \{(1 - l_i)(1 - s_i)\}},\end{aligned} \tag{D.4}$$

where s_i is the probability that sensor p_i is in sleep mode at each time slot.

However, if there is no sleep scheduling at sensors, that is $s_i = 0$, the total number of transmissions for disseminating sufficient large M packets to n receivers with XORs coding is

$$N_b' = \frac{M}{1 - \max_{i \in \{1,2,\ldots,n\}} \{l_i\}}. \tag{D.5}$$

Since no sensors are in sleep mode, the total number of active time slots consumed for disseminating packets with XORs coding is

$$\begin{aligned}\eta_{ns} &= N_b' n \\ &= \frac{Mn}{1 - \max_{i \in \{1,2,\ldots,n\}} \{l_i\}}.\end{aligned} \tag{D.6}$$

From the above formulation, we know that only if $\eta_s < \eta_{ns}$, sleep scheduling has contribution to save energy consumed by *idle listening*, otherwise, the retransmission due to sleep scheduling in sensors imposes more energy consumption. The above $\eta_s < \eta_{ns}$ changes into

$$\frac{M \sum_{i=1}^{n} (1 - s_i)}{\min_{i \in \{1,2,\ldots,n\}} \{(1 - l_i)(1 - s_i)\}} < \frac{Mn}{1 - \max_{i \in \{1,2,\ldots,n\}} \{l_i\}}. \tag{D.7}$$

That is

$$\sum_{i=1}^{n} (1 - s_i) < \frac{n}{l_{\min}} \min_{i \in \{1,2,\ldots,n\}} \{(1 - l_i)(1 - s_i)\}, \tag{D.8}$$

where $l_{\min} = 1 - \max_{i \in \{1,2,\ldots,n\}} \{l_i\}$.

Thus, if (D.8) can be satisfied, the current sleep scheduling must have contribution to save energy compared with no sleep scheduling.

Acknowledgment

This paper is supported by the National Science Foundation of China under Grant no. 60773036.

References

[1] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 81–94, Baltimore, Md, USA, November 2004.

[2] C.-J. M. Liang, R. Musăloiu-E., and A. Terzis, "Typhoon: a reliable data dissemination protocol for wireless sensor networks," in *Proceedings of the 5th European Conference on Wireless Sensor Networks (EWSN '08)*, vol. 4913 of *Lecture Notes in Computer Science*, pp. 268–285, January 2008.

[3] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: a reliable transport protocol for wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 1–11, Atlanta, Ga, USA, September 2002.

[4] S. S. Kulkarni and L. Wang, "MNP: multihop network reprogramming service for sensor networks," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*, pp. 7–16, June 2005.

[5] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *Proceedings of the 24th IEEE Annual Joint Conference of the Computer and Communications Societies (INFOCOM '05)*, vol. 4, pp. 2470–2481, Miami, Fla, USA, 2005.

[6] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st IEEE Annual Joint Conference of the Computer and Communications Societies (INFOCOM '02)*, vol. 3, pp. 1567–1576, 2002.

[7] A. Cerpa, J. L. Wong, L. Kuang, M. Potkonjak, and D. Estrin, "Statistical model of lossy links in wireless sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 81–88, Los Angeles, Calif, USA, 2005.

[8] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, 2009.

[9] S. Y. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in single-hop wireless coding networks," in *Proceedings of the IEEE Information Theory Workshop (ITW '07)*, pp. 120–125, Tahoe City, Calif, USA, September 2007.

[10] C. Zhan, Y. Xu, J. Wang, and V. Lee, "Reliable multicast in wireless networks using network coding," in *Proceedings of the 6th International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, pp. 506–515, Macau, China, 2009.

[11] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[12] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 829–848, 2006.

[13] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *Computer Communication Review*, vol. 36, no. 1, pp. 63–68.

[14] S. Deb, M. Effros, and T. Ho, "Network coding for wireless applications: a brief tutorial," in *Proceedings of the International Workshop on Wireless Adhoc Networks (IWWAN '05)*, May 2005.

[15] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 2171–2179, Phoenix, Ariz, USA, April 2008.

[16] Y. Lin, B. Liang, and B. Li, "Data persistence in large-scale sensor networks with decentralized fountain codes," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1658–1666, Anchorage, Alaska, USA, May 2007.

[17] I.-H. Hou, Y.-E. Tsai, T. F. Abdelzaher, and I. Gupta, "AdapCode: adaptive network coding for code updates in wireless sensor networks," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 1517–1525, Phoenix, Aariz, USA, 2008.

[18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.

[19] S. Rayanchu, S. Sen, J. Wu, S. Baneriee, and S. Senqupta, "Loss-aware network coding for unicast wireless sessions: design, implementation, and performance evaluation," in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '08)*, pp. 85–96, Annapolis, Md, USA, 2008.

[20] F.-C. Kuo, K. Tan, X. Y. Li, J. Zhang, and X. Fu, "XOR rescue: exploiting network coding in lossy wireless networks," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '09)*, pp. 1–9, Rome, Italy, June 2009.

[21] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in wireless networks," in *Proceedings of the 20th IEEE Annual Joint Conference of the Computer and Communications Societies (INFOCOM '01)*, pp. 1028–1037, 2001.

[22] K. Iwanicki and M. Van Steen, "Multi-hop cluster hierarchy maintenance in wireless sensor networks: a case for gossip-based protocols," in *Proceedings of the 6th European Conference on Wireless Sensor Networks*, pp. 102–117, Springer, Cork, Ireland, February 2009.

[23] G. Hartl and B. Li, "Loss inference in wireless sensor networks based on data aggregation," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 396–404, April 2004.

[24] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '07)*, pp. 169–180, Kyoto, Japan, 2007.

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 618016, 14 pages
doi:10.1155/2010/618016

Research Article

An Optimal Adaptive Network Coding Scheme for Minimizing Decoding Delay in Broadcast Erasure Channels

Parastoo Sadeghi,[1] Ramtin Shams,[1] and Danail Traskov[2]

[1] *Research School of Information Sciences and Engineering, The Australian National University, Canberra ACT 0200, Australia*
[2] *Institute for Communications Engineering, Technische Universität München, D-80290 München, Germany*

Correspondence should be addressed to Parastoo Sadeghi, parastoo.sadeghi@anu.edu.au

Received 31 August 2009; Accepted 3 March 2010

Academic Editor: Heung-No Lee

We are concerned with designing feedback-based adaptive network coding schemes with the aim of minimizing decoding delay in each transmission in packet-based erasure networks. We study systems where each packet brings new information to the destination regardless of its order and require the packets to be instantaneously decodable. We first formulate the decoding delay minimization problem as an integer linear program and then propose efficient algorithms for finding its optimal solution(s). We show that our problem formulation is applicable to memoryless erasures as well as Gilbert-Elliott erasures with memory. We then propose a number of heuristic algorithms with worst case linear execution complexity that can be used when an optimal solution cannot be found in a reasonable time. We verify the delay and speed performance of our techniques through numerical analysis. This analysis reveals that by taking channel memory into account in network coding decisions, one can considerably reduce decoding delays.

1. Introduction

In this paper, we are concerned with designing feedback-based adaptive network coding schemes that can deliver high throughputs and low decoding delays in packet erasure networks. We first present some background on existing work and emphasize that the notion of delay and the choice of a suitable network coding strategy are highly entangled with the underlying application.

1.1. Motivation and Background. Consider a broadcast packet-based transmission from one source to many destinations where erasures can occur in the links between the source and destinations. Two main *throughput optimal* schemes to deal with such erasures are fountain codes [1] and random linear network codes (RLNC) [2]. In the latter scheme, for example, the source transmits random linear mixtures of all the packets to be delivered. It is well-known that if the random coefficients are chosen from a finite field with a sufficiently large size, each coded packet will almost surely become linearly independent of all previously received coded packets and hence, *innovative* for every destination [2]. The scheme is therefore almost surely throughput optimal. Another benefit of fountain codes and RLNC is that they do not require feedback about erasures in individual links in order to operate.

However in these schemes, throughput optimality comes at the cost of large decoding delays, as the receiver needs, in general, to collect all coded packets in a block before being able to decode. Despite this drawback, there are applications which are insensitive to such delays. Consider, for example, a simple software update (file download). The update only starts to work when the whole file is downloaded. In this case, the main desired properties are throughput optimality and the mean completion time and there is often little or no incentive to aim for partial "premature" decoding. The completion time performance of RLNC for rateless file download applications has been considered in [3]. In [3], the mean completion time of RLNC is shown to be much shorter than scheduling. Reference [4] considers time division duplex systems with large round-trip link latencies and proposes solutions for the number of coded packet

transmissions before waiting for acknowledgement on the received number of degrees of freedom.

There are applications where partial decoding can crucially influence the end user's experience. Consider, for example, broadcasting a continuous stream of video or audio in live or playback modes. Even though fountain codes and RLNC are throughput optimal, having to wait for the entire coded block to arrive can result in unacceptable delays in the application layer. But, we also note that partial decoding of packets out of their natural temporal order does not necessarily translate into low delivery delays desired by the application layer. The authors in [5, 6] have proposed feedback-based throughput-optimal schemes to deal with the transmitter queue size, as well as decoding and delivery delays at the destinations. When the traffic load approaches system capacity, their methods are shown to behave "gracefully" and meet the delay performance benchmark of single-receiver automatic repeat request (ARQ) schemes.

There is yet another set of applications for which partial decoding is beneficial and can result in lower delays irrespective of the order in which packets are being decoded. Consider, for example, a wireless sensor network in which there is a fusion/command center together with numerous sensors/agents scattered in a region. Each sensor/agent has to execute or process one or more complex commands. Each command and its associated data is dispatched from the center in a packet. For coordination purposes, each agent needs to know its own and other agents' commands. Therefore, commands are broadcast to everyone in the network. In this application, in-order processing/execution of commands may not be a real issue. However, fast command execution may be crucial and therefore, it is imperative that innovative packets arrive and get decoded at the destinations as quickly as possible regardless of their order. As another example, consider emergency operations in a large geographical region where emergency-related updates of the map of the area need to be dispatched to all emergency crew members. In such situations too, updates of different parts of the map can be decoded in any order and still be useful for handling the emergency.

Finally, some applications may be designed in such a way that they are insensitive to in-order delivery. This can be particularly useful where the transport medium is unreliable. In such a case, it may be natural to use *multiple-description* source coding techniques [7], in which every decoded packet brings new information to the destination, irrespective of its order. In light of the emergency applications described above, one can perform multiple-description coding for map updates, so that updates of different subregions can be divided into multiple packets and each packet can provide an improved view of one region in a truly order-insensitive fashion.

1.2. Contributions. In this paper, we are inspired by the last set of order-insensitive packet delivery applications and hence, focus on designing network coding schemes that, with the help of feedback, can deliver innovative packets in any order to the destination and also guarantee fast decoding of such packets. As a first step towards such goal, we limit ourselves to broadcast erasure channels, but emphasize that the ideas can be extended to other more complicated scenarios. We also consider the class of *instantaneously decodable* network coding schemes, in which each coded transmission contains at most one new source packet that a receiver has not decoded yet. The rationale is that in an order-insensitive application, any innovative packet that cannot be decoded immediately incurs a unit of delay. Obviously, one other source of delay is when a coded packet does not contain any new information for a receiver and hence, is not innovative. A similar definition of the decoding delay was first considered in [8], where the authors presented a number of heuristic algorithms to reduce order-insensitive decoding delay. In this context, our main contributions are the following.

(i) In Section 1.1, we have motivated the problem in light of possible applications in sensor and ad hoc networks. To the best of our knowledge, such application-dependent classification of network coding delays did not previously exist in the literature.

(ii) In Section 3.1, we present a systematic framework for the minimization of decoding delay in each transmission subject to the instantaneous decodability constraint. We show that this problem can be cast into a special integer linear programming (ILP) framework, where instantaneously decodable packet transmission corresponds to a *set packing* problem [9] on an appropriately defined set structure.

(iii) In Section 3.2, we provide a customized and efficient method for finding the optimal solution to the set packing problem (which is in general NP-hard). Our numerical results in Section 6 show that for reasonably sized number of receivers, the optimum solution(s) can be found in a time that is linearly proportional to the total number of packets.

(iv) In Section 4, we discuss decoding delay minimization for an important class of erasure channels with memory, which can occur in wireless communication systems due to deep fades and shadowing [10]. We show that the general set packing framework in Section 3 can be easily modified to account for the erasure memory. Our results in Section 6 reveal that by adapting network coding decisions based on channel erasure conditions, significant improvements in delay are possible compared to when decisions are taken irrespective of channel states.

(v) In Section 5, we provide a number of heuristic variations of the optimal search for finding (possibly suboptimal) solutions faster, if needed. Our results in Section 6 show that such heuristics work very well and often provide solutions that are very close to the search algorithm. Moreover, they improve on the proposed random opportunistic method in [8].

2. Network Model

Consider a single source that wants to broadcast some data to N receivers, denoted by R_i for $i = 1, \ldots, N$. The data to be broadcast is divided into K packets, denoted by m_j for $j = 1, \ldots, K$. Time is slotted and the source can transmit one (possibly coded) packet per slot.

A packet erasure link L_i connects the source to each individual receiver R_i. Erasures in different links can be independent or correlated with each other. Different erasures in a single link can be independent (memoryless) or correlated with each other (with memory) over time.

For memoryless erasures, an erasure in link L_i can occur with a probability of $p_{e,i}$ in each packet transmission round independent of previous erasures.

For correlated erasures, we consider the well-known Gilbert-Elliott channel (GEC) [11], which is a Markov model with a *good* and a *bad* state. If the channel is in the good state, packets can be successfully received, while in the bad state packets are lost (e.g., due to deep fades or shadowing in the channel). The probability of moving from the good state G to the bad state B in link L_i is $b_i \triangleq \Pr(C_{i,\ell} = B \mid C_{i,\ell-1} = G)$ and the probability of moving from the bad state B to the good state G is $g_i \triangleq \Pr(C_{i,\ell} = G \mid C_{i,\ell-1} = B)$, where ℓ is the time slot index. Steady-state probabilities are given by $P_{G,i} \triangleq \Pr(C_i = G) = g_i/(b_i + g_i)$ and $P_{B,i} \triangleq \Pr(C_i = B) = b_i/(b_i + g_i)$. Following [12], we define the memory content of the GEC in link L_i as $0 \leq \mu_i = 1 - b_i - g_i < 1$, which signifies the persistence of the channel in remaining in the same state. A small μ means a channel with little memory and a large μ means a channel with large memory.

Before transmission of the next packet, the source collects error-free and delay-free 1-bit feedback from each destination indicating if the packet was successfully received or not. A successful reception generates an acknowledgement (ACK) and an erasure generates a negative acknowledgement (NAK). This feedback is used for optimizing network coding decisions at the source for the next packet transmission round, as described in future sections.

In this work, we consider linear network coding [2] in which coded packets are formed by taking linear combinations of the original source packets. Packets are vectors of fixed size over a finite field $\mathbb{F}_q$. The coefficient vector used for linear network coding is sent in the packet header so that each destination can at some point recover the original packets. Since in this paper we are only dealing with instantaneously decodable packet transmission, it suffices to consider linear network coding over $\mathbb{F}_2$. That is, coded packets are formed using binary XOR of the original source packets. Thus, network coding is performed in a similar manner as in [13].

Definition 1. A transmitted packet is instantaneously decodable for receiver R_i if it is a linear combination of source packets containing at most one source packet that R_i has not decoded yet. A scheme is called instantaneously decodable if all transmissions have this property for all receivers.

Definition 2. At the end of transmission round ℓ in an instantaneously decodable scheme, the knowledge of receiver R_i is the set consisting of all packets that the receiver has decoded so far. The receiver can therefore, compute any linear combination of the packets that it has decoded for decoding future packets.

Definition 3. In an instantaneously decodable scheme, a coded packet is called non-innovative for receiver R_i if it only contains source packets that the receiver has decoded so far. Otherwise, the packet is innovative.

Definition 4. A scheme is called rate or throughput optimal if all transmissions are innovative for the entire set of receivers.

Definition 5. In time slot ℓ, receiver R_i experiences one unit of delay if it successfully receives a packet that is either non-innovative or not instantaneously decodable. If we impose instantaneous decodability on the scheme, a delay can only occur if the received packet is not innovative.

Note that in the last definition, we do not count channel inflicted delays due to erasures. The delay only counts "algorithmic" overhead delays when we are not able to provide innovative and instantaneously decodable packets to a receiver.

As an example, if the knowledge of R_1 is $\{m_1, m_2, m_3\}$, receiving $m_1 \oplus m_2$ will cause R_1 to experience one unit of delay, whereas $m_1 \oplus m_2 \oplus m_5$ is innovative and instantaneously decodable, hence does not incur any delay.

We note that a packet that is not transmitted yet or transmitted but not received by any receiver can be transmitted in an uncoded manner at any transmission slot without incurring any algorithmic delay. In fact, this is how the transmission starts: by sending m_1 uncoded, for example.

A zero-delay scheme would require all packets to be both innovative and instantaneously decodable to all receivers. Thus zero-delay implies rate optimality, but not vice versa. As the authors show in [8, Theorem 1] for the case of $N = 2$ and $N = 3$ receivers, there exists an *offline* algorithm that is both rate optimal and delay-free. For $N \geq 4$ the authors prove that a zero-delay algorithm does not exist. By offline we mean that the algorithm needs to know future realizations of erasures in broadcast links. In contrast, an *online* algorithm decides on what to send in the next time slot based on the information received in the past and in the current slot. In this paper, we focus on designing online algorithms.

3. Optimization Framework

3.1. Problem Formulation Based on Integer Linear Programming. Instantaneous decodability can be naturally cast into the framework of integer optimization. To this end, let us fix the packet transmission round to ℓ and consider the knowledge of all receivers, which is also available at the source because of the feedback. The state of the entire system at time index ℓ (in terms of packets that are still needed by

the receivers) can be described by an $N \times K$ binary *receiver-packet incidence* matrix A with elements

$$a_{ij} = \begin{cases} 1 & \text{if } R_i \text{ needs } m_j, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Columns of matrix A are denoted by $\mathbf{a}_1$ to $\mathbf{a}_K$. We assume that packets received by all receivers are removed from the receiver-packet incidence matrix. Hence, A does not contain any all-zero columns.

Example 1. Consider $N = 2$ receivers and $K = 3$ packets. Before the transmission begins, the receiver-packet incidence matrix A is an all-one 2×3 matrix. If we send packet m_1 in the first transmission round $\ell = 1$ and assuming that only receiver R_2 successfully receives it, A will become

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \quad (2)$$

If we send packet m_2 in the next transmission round $\ell = 2$ and assuming that only receiver R_1 successfully receives it, A will then be

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \quad (3)$$

The condition of instantaneous decodability means that at any transmission round we cannot choose more than one packet which is still unknown to a receiver R_i. In the example above, at $\ell = 3$, we cannot send $m_1 \oplus m_3$ because it contains more than one packet unknown to R_1.

Let $\mathbf{x}$ represent a binary decision vector of length K that determines which packets are being coded together. The transmitted packet consists of the binary XOR of the source packets for which $x_j = 1$. More formally, we can define the instantaneous decodability constraint for all receivers as $A\mathbf{x} \leq \mathbf{1}_N$, where $\mathbf{1}_N$ represents an all-one vector of length N and the inequality is examined on an element-by-element basis (Note that although $\mathbf{x}$ is a binary or Boolean vector, $A\mathbf{x}$ is calculated in real domain. Hence, $A\mathbf{x} \leq \mathbf{1}_N$ is in fact a pseudo-Boolean constraint.). This condition ensures that a transmitted coded packet contains at most one unknown source packet for each receiver. A vector $\mathbf{x}$ is called *infeasible* if it does not satisfy the instantaneous decodability condition. In other words, $\mathbf{x}$ is called *infeasible* if and only if there exists at least one p for which $b_p > 1$ in $A\mathbf{x} = \mathbf{b} = [b_1, \ldots, b_p, \ldots, b_N]^T$. A vector $\mathbf{x}$ is called a *solution* if and only if it satisfies $A\mathbf{x} \leq \mathbf{1}_N$. In the rest of this paper, "$A\mathbf{x} \leq \mathbf{1}_N$" and "$\mathbf{x}$ is a solution" are used interchangeably.

Now consider sets $M_1, \ldots, M_K \subset \{R_1, \ldots, R_N\}$, where M_j is the nonempty set of receivers that still *need* source packet m_j. Note that these sets can be easily determined by looking at the columns of matrix A. The "importance" of packet m_j can be, for example, taken to be the size of set M_j, which is the number of receivers that still need m_j.

We now formally describe the optimization procedure that should be performed at the transmitter. Maximizing the number of receivers for which a transmission is innovative, subject to the constraint of instantaneous decodability, can be posed as the following (binary-valued) integer linear program (ILP):

$$\begin{aligned} \max \quad & \mathbf{w}^T\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{1}_N, \quad \mathbf{x} \in \{0,1\}^K, \end{aligned} \quad (4)$$

where $\mathbf{w}^T = (|M_1|, \ldots, |M_K|)$. This is a standard problem in combinatorial optimization, usually called *set packing* [9]. Here the universe is the set of all receivers and we need to find disjoint (due to instantaneous decodability condition) subsets M_j with the largest total size. In the (most desirable) case when equality holds in $A\mathbf{x} \leq \mathbf{1}_N$ for every receiver, we also speak of a *set partition*. This is equivalent to a zero-delay transmission.

In Section 4, we will consider other measures of packet importance and discuss the role of $\mathbf{w}$ in tailoring the optimization problem according to the application requirements or channel conditions, such as memory in erasure links.

We assume that elements of $\mathbf{w}$, which signify packet importance, are all positive. If one has already found a solution such as $\mathbf{x}_1 = [x_1, \ldots, x_{p-1}, 1, x_{p+1}, \ldots, x_K]$ with $\mathbf{w}^T\mathbf{x}_1 = v_1$, then changing this solution into $\mathbf{x}_0 = [x_1, \ldots, x_{p-1}, 0, x_{p+1}, \ldots, x_K]$ by changing $x_p = 1$ into $x_p = 0$ can only result in a $\mathbf{w}^T\mathbf{x}_0 = v_0$ strictly smaller than v_1. We say that *given* solution $\mathbf{x}_1$, $\mathbf{x}_0$ is clearly *suboptimal* and hence, can be discarded in an algorithm that searches for the optimal solution(s).

3.2. Efficient Search Methods for Finding the Optimal Solution of (4). It is well known that the set packing problem is NP-hard [9]. Here, we present an efficient ILP solver designed to take advantage of the specific problem structure. Later, we will see that for many practical situations of interest, our method performs well empirically. Based on this framework, we will also present some heuristics in Section 5 to deal with more complicated and time-consuming problem instances.

We begin presenting our method by first defining constrained and unconstrained variables.

Definition 6. Two binary-valued variables are said to be *constrained* if they cannot be simultaneously 1 in a solution. Or formally, x_i and x_j are constrained if for any $\mathbf{x}$ satisfying $A\mathbf{x} \leq \mathbf{1}_N$, $x_i + x_j \leq 1$ (Again, note that the addition of variables takes place in real domain.). We also say that x_j is constrained to x_i and vice versa. It can be proven that x_i and x_j are constrained if and only if there exits at least one row index p in A for which $a_{pi} = a_{pj} = 1$.

Definition 7. The set of all variables constrained to x_i is called the *constrained set* of x_i and is denoted by $\mathcal{C}_i$. That is,

$$\mathcal{C}_i = \left\{ x_j \mid j \neq i, A\mathbf{x} \leq \mathbf{1}_N \Longrightarrow x_i + x_j \leq 1 \right\}. \quad (5)$$

If x_i and x_j are not constrained to each other ($x_i \notin \mathcal{C}_j$ and $x_j \notin \mathcal{C}_i$), then columns $\mathbf{a}_i$ and $\mathbf{a}_j$ in A cannot have nonzero elements in the same row position. That is, for each row index p, $a_{pi} = 1 \Rightarrow a_{pj} = 0$ and $a_{pj} = 1 \Rightarrow a_{pi} = 0$.

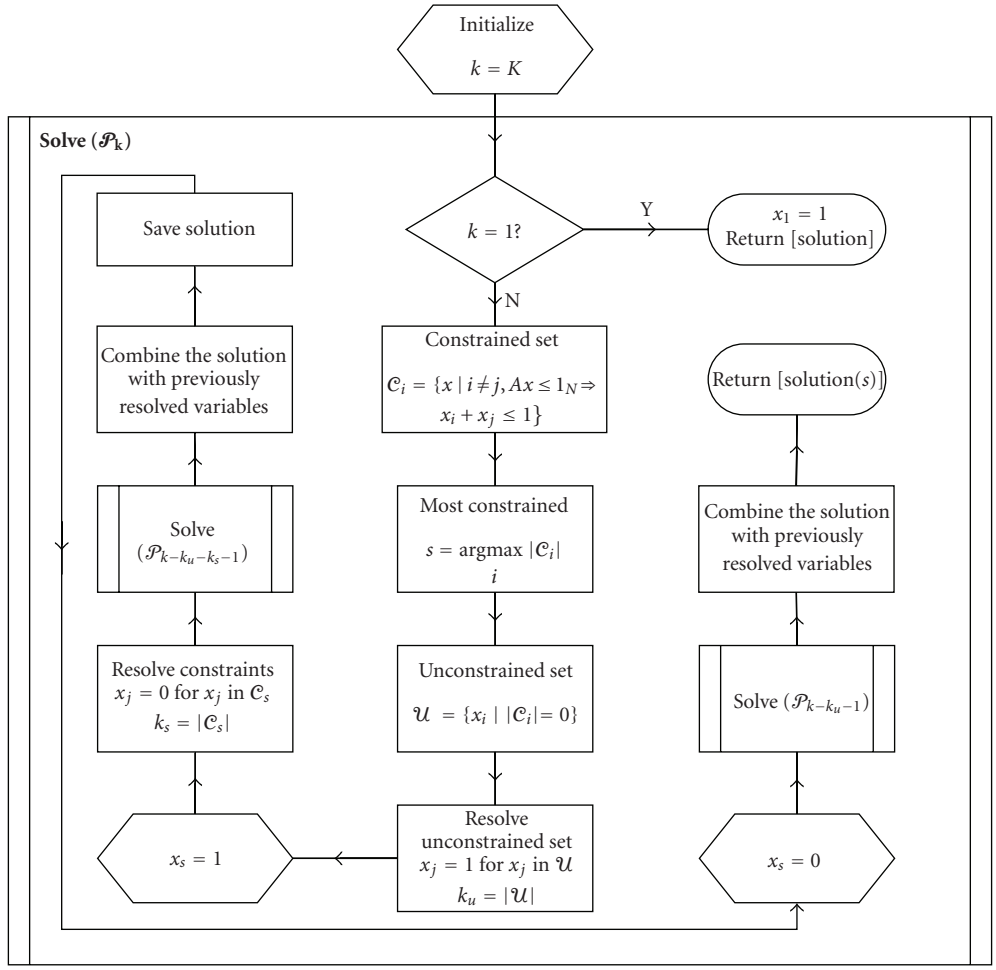


Figure 1: A schematic of Algorithm 1 with greedy pruning for finding the optimal network coding solution of (4). Note that the algorithm is recursive as it calls $\mathcal{P}_{k-k_u-k_s-1}$ and $\mathcal{P}_{k-k_u-1}$ within itself.

Definition 8. A variable x_i is said to be *unconstrained* if $\mathcal{C}_i = \emptyset$. The set of all unconstrained variables is denoted by $\mathcal{U}$ and is referred to as the *unconstrained set*.

If x_i is an unconstrained variable, then for each row index p, $a_{pi} = 1 \Rightarrow a_{pj} = 0$ for all $j \neq i$ (otherwise, x_i and x_j would become constrained).

Example 2. Consider the following receiver-packet incidence matrix A

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

One can easily verify the relations defined above. For example, variables x_1 and x_3 are constrained because for $p = 1$, $a_{p1} = a_{p3} = 1$. Variables x_1 and x_4 are not constrained to each other because columns $\mathbf{a}_1$ and $\mathbf{a}_4$ do not have a nonzero element in the same row position. Variable x_6 is unconstrained because no other column has a nonzero element in rows 6 or 7. In summary, $\mathcal{C}_1 = \{x_2, x_3\}$, $\mathcal{C}_2 = \{x_1\}$, $\mathcal{C}_3 = \{x_1, x_4\}$, $\mathcal{C}_4 = \{x_3\}$ and $\mathcal{C}_5 = \mathcal{C}_6 = \emptyset$.

To design an efficient search algorithm, one needs to efficiently prune the parameter space and reduce the problem size. We make the following observations for pruning of the parameter space.

(1) Unconstrained variables must be set to 1. In other words, setting those variables to 0 does not contribute to the optimal solution (note that the elements in $\mathbf{w}$ are positive). In the above example, x_5 and x_6 must be set to 1 because no other variable is constrained to them (we will make this statement formal in the optimality proof of the algorithm in the appendix).

(2) If a constrained variable is set to 1 all members of its constrained set must be set to 0. In the above example, setting $x_1 = 1$ forces x_2 and x_3 to zero.

(3) At a given step, the parameter space can be pruned most by resolving the variable with the largest constrained set.

Application of the third observation, in a search algorithm results in greedy pruning of the parameter space. We note that greedy pruning is only optimal for a given step of the algorithm and is not guaranteed to result in the optimal reduction of the overall complexity of the search.

We now make a final remark before presenting the search algorithm. In particular, we have observed that finding constrained sets for each variable in each step of the algorithm can be somewhat time consuming. A very effective alternative is to first sort matrix A, column-wise, in descending order of the number of 1's in each column. Setting the "most important" head variable x_1 (with the highest $|M_1|$) to 1 is likely to result in the largest constrained set (because it potentially overlaps with many other variables) and hence, many variables will be resolved in the next recursion. We will refer to the approach based on finding the largest constrained set as the *greedy pruning* strategy and to the alterative approach as the *sorted pruning* search strategy.

The greedy pruning search strategy is shown in Figure 1, which with appropriate modifications can also represent the sorted pruning variation. Let $\mathcal{P}_k$ denote the problem of size k whose input is an $N \times k$ receiver-packet incidence matrix A_k and whose output is a set of solutions of the form $\mathbf{x}$ of length k which satisfy the instantaneous decodability condition $A_k\mathbf{x} \leq \mathbf{1}_N$. The algorithms can be described as shown in Algorithm 1.

In the appendix, we prove by structural induction that Algorithm 1 is guaranteed to return all optimal solutions of (4). However, we note that not every solution returned by Algorithm 1 is optimal. The nonoptimal solutions can be easily discarded by testing against the objective function (4) at the end of the algorithm. We also note that in Algorithm 1, we can simply remove those packets received by every receiver from the problem. If there are K_0 such variables, we can start step (1) above from $k = K - K_0$ instead of K. The Matlab code for both the greedy and sorted pruning algorithms can be found at http://users.rsise.anu.edu.au/~parastoo/netcod/.

We conclude this section by a brief note on the computational complexity of Algorithm 1. Let us denote the number of recursions required to solve the problem of size k by $\mathfrak{C}_k$. According to Algorithm 1, this problem is always broken into two smaller problems of size $k - k_u - k_s - 1$ and $k - k_u - 1$. Therefore, one can find the number of recursions required to solve $\mathcal{P}_k$ by recursively computing $\mathfrak{C}_k = \mathfrak{C}_{k-k_u-k_s-1} + \mathfrak{C}_{k-k_u-1}$. The recursion stops when one reaches a problem of size 1 (only one packet to transmit) where $\mathfrak{C}_1 = 1$.

4. Adaptive Network Coding in the Presence of Erasure Memory

Here, we present a generalization of the set packing approach for coded transmission in erasure channels with memory. The idea is that the importance of a packet m_j is no longer determined by how many receivers need m_j, but by the probability that m_j will be successfully decoded by the receivers that need it. In computing this probability, one can use the fact that successive channel erasures in a link are usually correlated with each other and hence, their history can be used to make predictions about whether a receiver is going to experience erasure or not in the next time slot. To present the idea, we focus on the GEC model for representing channel erasures. More general memory models for erasure can also be incorporated into our framework.

We define the reward p_i of sending a packet to receiver R_i as the probability of successful reception by R_i in the next time slot: $p_i = \Pr(C_{i,\ell} = G \mid C_{i,\ell-1})$, where $C_{i,\ell-1}$ is the state of R_i in the previous transmission round (Statements like "state of R_i" should be interpreted as the state of the physical link L_i connecting the source to R_i.). The total reward or importance of sending packet m_j is then

$$w_j = \sum_{i \in M_j} p_i. \tag{7}$$

The above weight vector gives higher priority to a packet m_j for which there is a higher chance of successful reception, because the receivers that need m_j are more likely to be in good state in the next time slot. With this newly defined weight vector, one can try to solve the optimization problem given in (4) under the same instantaneous decodability condition.

Remark 1. We conclude this section by emphasizing that the optimization framework in (4) is very flexible in accommodating other possibilities for the weight vector $\mathbf{w}$, which can be appropriately determined based on the application. For example, instead of allocating the same weight to a packet needed by a subset of receivers, one can allocate different weights to the same packet (looking column-wise at A) depending on the priorities or demands of each user. In the map update example described in the Introduction, different emergency units can adaptively flag to the base station different parts of the map as more or less important depending on their distance from a certain disaster zone. The task of the base station is then to send a packet combination that satisfies the largest total priority. One can also combine user-dependent packet weights with the channel state prediction outcomes in a GEC. One possibility is to multiply the probabilities p_i by the receiver priority. It could then turn out that although a receiver is more likely to be in erasure in the next transmission round, it may be served because of a high priority request.

5. Heuristic Search Algorithms

In Section 3.2, we proposed efficient search algorithms for finding the optimal solution(s) of (4). However, there may be situations where one would like to obtain a (possibly suboptimal) solution much more quickly. This may be the case, for example, when the total number of packets to be transmitted is very large. Therefore, designing efficient heuristic algorithms to complement the optimal search is

(1) *Start with the original problem of size* $k = K$.
(2) ***if*** *sorted pruning strategy is desired* ***then***
(3) *Rearrange the variables in* A_k *in descending order of packet importance (number of* 1*'s in each column).*
(4) ***end if***
(5) *Solve* $(\mathcal{P}_k)$*:*
(6) ***if*** $k = 1$ ***then***
(7) *Return* $x_1 = 1$ *(since the variable is not constrained).*
(8) ***else***
(9) ***if*** *greedy pruning strategy is desired* ***then***
(10) *Determine the constrained set for all variables* x_1 *to* x_k.
(11) *Denote the index of the variable with the largest constrained set by* s *and the cardinality of its constrained set by* k_s.
(12) ***else***
(13) *Determine the constrained set for the head variable* x_1 *with cardinality* k_1 *and also the set of unconstrained variables (Note that we have overused index 1 to refer to the head variable in the reordered matrix at each recursion.). Set* $s = 1$.
(14) ***end if***
(15) *Denote the cardinality of the unconstrained set* $\mathcal{U}$ *by* k_u.
(16) *Set all the unconstrained variables to 1.*
(17) *Set* $x_s = 1$ *and the variables in its corresponding constrained set* $\mathcal{C}_s$ *to 0.*
(18) *Reduce the problem by removing resolved variables. Reduce* A_k *accordingly.*
(19) *Solve* $(\mathcal{P}_{k-k_u-k_s-1})$ *(Note that* k_u *unconstrained variables are set to one,* $x_s = 1$ *and* k_s *variables constrained by* x_s *are set to zero, hence a total of* $k_s + k_u + 1$ *variables are resolved.).*
(20) *Combine the solution with previously resolved variables. Save solution.*
(21) *Set* $x_s = 0$.
(22) *Reduce the problem by removing resolved variables. Reduce* A_k *accordingly.*
(23) *Solve* $(\mathcal{P}_{k-k_u-1})$ *(Note that* k_u *unconstrained variables are set to one and* $x_s = 0$*, hence a total of* $k_u + 1$ *variables are resolved.).*
(24) *Combine the solution with previously resolved variables. Return solution(s).*
(25) ***end if***

Algorithm 1: Recursive search for the optimal solution(s) of (4).

important. In this section, we propose a number of such heuristics.

5.1. Heuristic 1—Weight Sorted Heuristic Algorithm. The idea behind this recursive algorithm is very simple. As in Algorithm 1, we start with the original problem of size $k = K$. We then rearrange the columns of the matrix A in descending order of $|w_j|$ (starting from the packet with the highest weight). Note that this is different from the sorted pruning version of the Algorithm 1, in which the columns of A were sorted in descending order of $|M_j|$ to potentially result in large constrained sets. We then set the head variable $x_1 = 1$ and find its corresponding constrained set $\mathcal{C}_1$ to resolve $k_1 = |\mathcal{C}_1|$ variables that are to be set to zero. We then solve the smaller problem of size $\mathcal{P}_{k-k_1}$ and continue until the problem cannot be further reduced. One main difference between Heuristic 1 and Algorithm 1 is that at each recursion, the head variable is only set to one; the other possibility of $x_1 = 0$ is not pursued at all. In a sense, this heuristic algorithm finds *greedy* solutions to the problem at each recursion by serving the highest priority packet. In this heuristic algorithm, all k_u unconstrained variables are naturally set to 1 in the course of the algorithm. The computational complexity of this method is at worst proportional to K, which can happen when there is no constraint between packets.

5.2. Heuristic 2—Search Algorithm 1 with Maximum Recursions/Elapsed Time. It is possible to terminate the recursive search Algorithm 1 prematurely once it reaches a maximum number of allowed recursions/elapsed time. If the algorithm reaches this value and the search is not complete, it performs a *termination procedure* whereby it heuristically resolves the remaining unresolved packets in the current incomplete solution. That is, it performs Heuristic 1 on a smaller problem, which is yet to be solved. It then returns the best solution that has been found so far. We note that due the extra termination procedure, the actual number of recursions/elapsed time can be (slightly) higher than the preset value.

Two comments are in order here. Firstly, Algorithm 1 is designed to sort the matrix A based on the number of receivers that need a packet. It only reverts to sorting the unresolved variables based on the vector $\mathbf{w}$ in the termination process. Secondly, if the maximum number of recursions is set to one, Algorithm 1 just performs the termination process and becomes identical to Heuristic 1.

5.3. Heuristic 3—Dynamic Number of Recursions. This heuristic is based on Heuristic 2, where we dynamically increase the number of allowed recursions as needed. At each transmission round, we start with only one allowed

recursion (effectively run Heuristic 1). If the throughput (Let $\mathcal{Q} \subset \{1, \ldots, N\}$ denote the index of receivers that still need at least one packet and $\mathcal{R}_{\mathcal{Q}}$ denote such receivers. The achieved throughput at time slot ℓ is defined as $\mathbf{w}^T\mathbf{x}/f(\mathcal{R}_{\mathcal{Q}})$, where $\mathbf{x}$ is the found solution and $f(\mathcal{R}_{\mathcal{Q}})$ is an appropriate function of receivers' needs. For memoryless erasures $f(\mathcal{R}_{\mathcal{Q}}) = |\mathcal{R}_{\mathcal{Q}}|$ and for GEC's $f(\mathcal{R}_{\mathcal{Q}}) = \sum_{q\in\mathcal{Q}} |p_q|$ (refer to Section 4 and (7)).) is higher than a desired value, there is no need to proceed any further. Otherwise, we can gradually increase the number of recursions by an appropriate step size. This heuristic stops when it either reaches the maximum allowed recursions or when increasing the number of recursions does not result in a noticeable improvement in the throughput.

6. Numerical Results and Secondary Coding Considerations

We start this section by presenting *end-to-end* decoding delay results for memoryless erasure channels. We then specialize to erasure channels with memory. The end-to-end problem is the complete transmission of K packets. End-to-end decoding delay of a receiver is the sum of decoding delays for the receiver in each transmission step. In the following, when we say "the delay performance of method X", we are referring to the delay performance of the end-to-end transmission, where method X is applied at each step.

In the course of presenting the results and based on the observed trends, we will discuss some secondary coding techniques and post processing considerations that can improve the decoding delay. Throughout the analysis of this section, we assume independent erasures in different links with identical probabilities. Hence, we can drop subscript i when referring to link erasure probabilities.

Figure 2 shows the median of decoding delay for the transmission of $K = 100$ packets to $N = 3$ to $N = 100$ receivers. Channel erasures are memoryless and occur with a high probability of $p = 0.5$ independently in every link. The median of delay is computed across all receivers and is, in fact, also the median across many stochastic runs of the algorithms. The first curve from below shows the delay obtained from Algorithm 1 (Throughout the numerical evaluations, we used the sorted pruning version of Algorithm 1.). The middle curve is the delay obtained by performing Heuristic 1. The top curve shows a reproduction of delay results reported in [8] which are based on a *random* opportunistic instantaneous network coding strategy. In this case, the transmitter first selects a packet needed by at least one receiver at random. Then, it goes over other packets in some order and adds a packet to the current choice only if their addition still results in instantaneous decodability. In comparison, Heuristic 1 performs noticeably better than that in [8] and more importantly, is not much far away from the results of Algorithm 1. This is specially important since for some number of receivers, Heuristic 1 can run considerably faster than Algorithm 1, which will be shown in the coming figures shortly.

Figure 3 compares the mean delay performance of different heuristics presented in Section 5 with that of

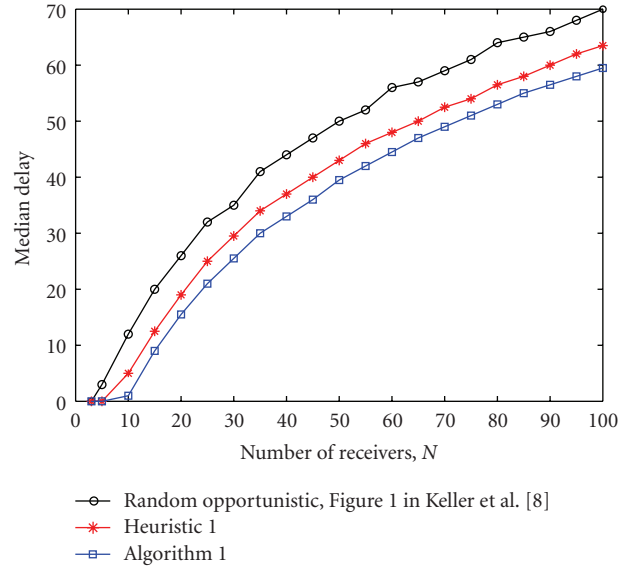


Figure 2: Median of decoding delay for the transmission of $K = 100$ packets to $N = 3$ to $N = 100$ receivers. Channel erasures are memoryless and occur with a high probability of $p = 0.5$ independently in every link. Algorithm 1, Heuristic 1 and random Heuristic [8] are compared with each other.

Algorithm 1. Similar to the previous figure, mean delay is computed across all receivers. The delay performance of Heuristic 2, Heuristic 3, and Algorithm 1 are close, whereas Heuristic 1 results in the largest delay. A careful reader may notice that the end-to-end performance of Heuristic 2 is at times better than Algorithm 1. While the difference is practically insignificant, this deserves some explanation. The end-to-end transmission problem involves making packet transmission decisions at each step. While all algorithms start with the same packet incidence matrix (all-ones), due to packet erasures and as they make decisions about transmission of packets at each step, they take diverging paths in the solution space. As a result, they end up with different packet incidence matrices to solve over time. Hence, it is conceivable for an algorithm to make suboptimal decisions at one or more steps and yet end up with a better end-to-end delay than Algorithm 1 that strictly makes optimal decisions at every step. Intuition suggests that an algorithm such as Heuristic 1 that consistently makes suboptimal decisions is unlikely to outperform Algorithm 1 end-to-end, which is confirmed by the numerical results. However, an algorithm such as Heuristic 2 which almost always makes optimal decisions with only infrequent exceptions, may outperform Algorithm 1. According to Figure 3, these perturbations in end-to-end performance are practically insignificant and the intuitive choice of the optimal or a largely optimal algorithm at each step will result in the best end-to-end performance.

We note that the delays presented here (and also in the following figures) are, in fact, *excess median or mean delays* beyond the minimum required number of transmissions, which is K. For example, a mean delay of 10 slots for $K = 100$ packets signifies on average 10% overhead, which is the

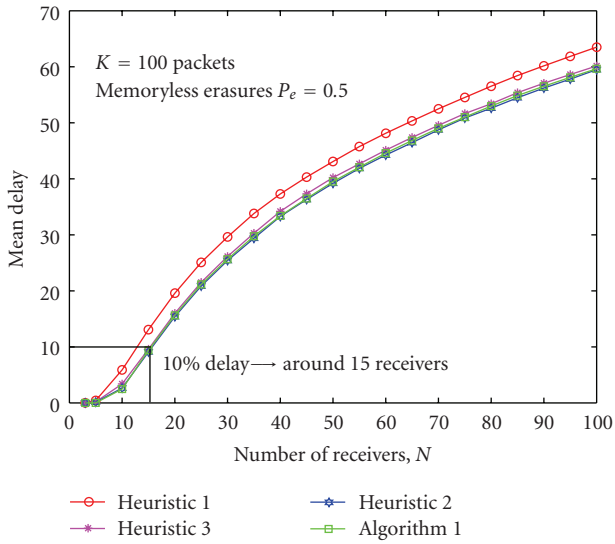


Figure 3: Mean decoding delay for the transmission of $K = 100$ packets to $N = 3$ to $N = 100$ receivers. Algorithm 1 is compared with Heuristics 1–3. Both Heuristics 2 and 3 perform very closely to Algorithm 1. The maximum number of recursions for both Heuristic 2 and 3 is set to 100.

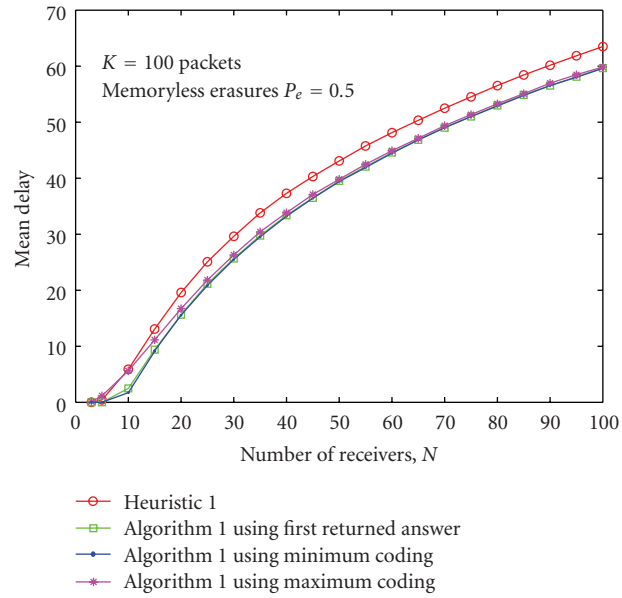


Figure 4: The effect of post processing on mean delay. Whenever Algorithm 1 returns multiple solutions, minimum amount of coding should be chosen. Heuristic 1 is shown for reference.

price for guaranteeing instantaneous decodability. In other words, one measure of throughput is $\text{th}_1 = K/(K + \overline{d})$, where $\overline{d}$ is the mean delay across all receivers. An example is shown in Figure 3. For up to around 15 receivers in the system, Algorithm 1, Heuristics 2, and 3 ensure an average throughput loss of 10%.

It is quite possible that Algorithm 1 returns multiple network coding solutions all of which have the same objective value $\mathbf{w}^T\mathbf{x}$. A natural question that arises is whether systematic selection of a solution with a particular property is better than others in the presence of erasures in the channel. Our experiments verify that indeed some secondary post processing on the solutions can improve the end-to-end delay. In particular, we compare two post processing techniques: (1) selecting a solution which involves minimum amount of coding (lowest number of 1's in the solution vector $\mathbf{x}$) and (2) selecting a solution with maximum amount of coding (highest number of 1's in the solution vector $\mathbf{x}$). Figure 4 shows the effects of such processing on the overall decoding delays. It is clear that *maximum coding* is not a reasonable choice and results in worse delays compared with *minimum coding*. We attempt to explain this behavior by means of an example and intuitive reasoning. Let us assume that there are $K = 3$ packets to be transmitted to $N = 3$ receivers and at the beginning of the third transmission round, matrix A is given as follows

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{8}$$

It is clear that there are two optimal solutions: we can either send packets $m_1 \oplus m_2$ or packet m_3 by itself, where the former involves coding and latter is uncoded. Now let us assume that we select the maximum coding strategy and send $m_1 \oplus m_2$. If in the third transmission round only R_2 successfully receives, A will become

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \tag{9}$$

and clearly the optimal solution is sending packet m_3. If in the fourth transmission round only R_1 successfully receives, A will become

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \tag{10}$$

where it is evident that in the fifth transmission round, we cannot find a packet which is innovative and instantaneously decodable for all the three receivers. On the other hand, one can verify that if we adopt a minimum coding strategy and send packet m_3 in the third transmission round, we can always find innovative and instantaneously decodable packets for all three receivers in the future regardless of erasures in the channel. In summary, solutions with less coding tend to cause less constrains on the problem in the future.

It is noted in Figure 4 that the first solution returned by Algorithm 1 performs almost the same as the minimum coding solution. The reason for this is that Algorithm 1 first ranks the packets based on the number of receivers that need them. Therefore, the first solution picked by the algorithm is likely to contain packets with largest constrained sets and hence, many resolved packets are set to zero, which often translates into small amount of coding. Throughout this

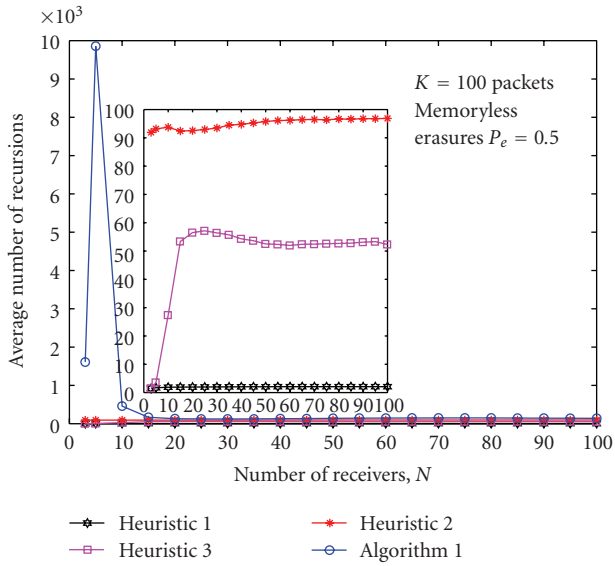


Figure 5: Average number of recursions in Algorithm 1 and Heuristics 1–3. The maximum number of recursions for both Heuristic 2 and 3 is set to 100. By referring to Figure 3, we observe that for small number of receivers, Heuristics 2-3 can provide same decoding delays at a fraction of computational complexity.

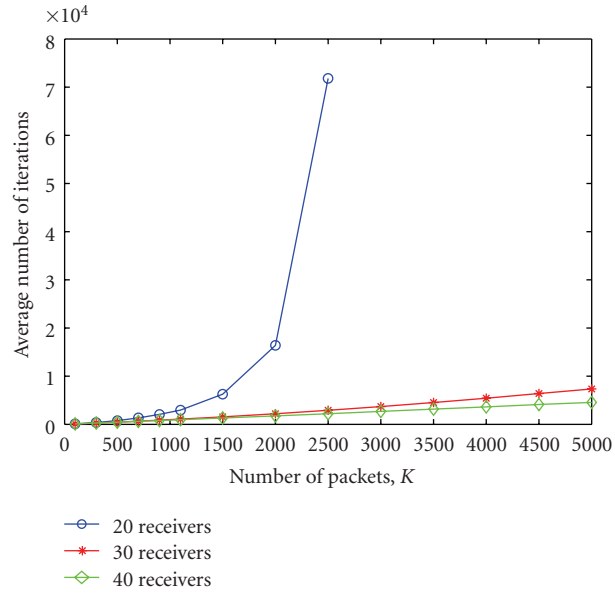


Figure 6: The effect of increasing the number of packets on the computational complexity of Algorithm 1 in terms of number of recursions. The complexity remains linear with the number of packets for well-sized receiver populations (30 and 40 receivers).

section, unless otherwise stated, we have shown the delay results based on the first returned solution of Algorithm 1.

It is interesting to analyze the actual number of recursions that the search in Algorithm 1 takes to find the optimum solution. This is shown in Figure 5 for $K = 100$ packets along with the number of recursions required in Heuristics 1, 2, and 3. Algorithm 1 shows three modes of behavior: low, medium, and high number of recursions. When the number of receivers is larger than $N = 20$, Algorithm 1 finds the optimal solution very quickly and the number of recursions is very close to the number of packets K. However, when the number of receivers is lower, the constraints that each receiver imposes on the network coding decisions cannot limit the search space enough and hence, a large number of combinations have to be tested. Obviously, Heuristic 1 has the lowest number of recursions. Compared to Heuristic 2 with 100 fixed recursions, dynamic Heuristic 3 can almost halve the number of recursions with negligible effect on delay performance (see Figure 3). By referring to Figure 3, we conclude that for the system under consideration, the excessive number of recursions in Algorithm 1 is not warranted as it does not result in any noticeable delay improvement compared to Heuristics 2 or 3.

Figure 6 shows the effect of increasing the number of packets on the computational complexity of Algorithm 1 in terms of number of recursions to complete the search. Three different numbers of receivers $N = 20$, $N = 30$, and $N = 40$ are considered. The complexity remains linear with the number of packets for well-sized receiver populations (30 and 40 receivers). This is in agreement with observations in Figure 5. When the number of receivers is not so large (see the blue curve in Figure 6 for $N = 20$), we see a sudden growth in complexity, in terms of number of recursions, when $K \gtrsim 700$ packets. In such situations, truncating the number of recursion to be linear with the number of packets (Heuristic 2) is a good alternative.

Figure 7 shows the impact of the number of packets and also erasure probability on the decoding delay. The normalized mean delay versus number of packets K is plotted for three different erasure probabilities $P_e = 0.5$, $P_e = 0.4$, and $P_e = 0.2$, which are still high erasure probabilities. The number of receivers is fixed to $N = 20$. The delay performance of Heuristics 1 and 3 are shown. A few observations are made. Firstly, as expected, the delay (both absolute and normalized measures) decreases as the erasure probability decreases. Secondly, the difference in the delay performance between Heuristics 1 and 3 decreases as the erasure probability decreases. This trend has also been observed for other number of receivers. Moreover, the difference between heuristics and Algorithm 1 decreases with erasure probability, which is not shown here for clarity of figure. Finally, the normalized delay decreases as the number of packets increases. We noted, however, that the absolute delay may increase or decrease depending on the number of receivers in the system. We attribute possible decrease in the normalized delay to the fact that when there are more packets to transmit, the transmitter has more options to choose from and hence, encounters delays less often in a normalized sense.

An important question that may arise in practical situations is how to choose the "block size" or the number of packets that are taken into account for making network coding decisions. If one has a total of K packets to transmit, does it make sense to divide them into subblocks of smaller

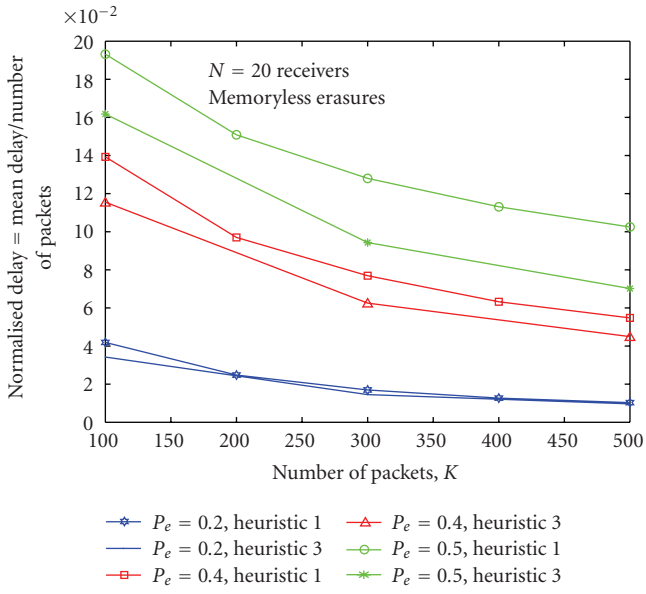


FIGURE 7: The effect of number of packets and erasure probabilities on the normalized delay. The maximum number of recursions for Heuristic 3 is set to 100. As the erasure probability decreases, the delay decreases as expected. The normalized delay decreases with K for this particular N (this is not always the case).

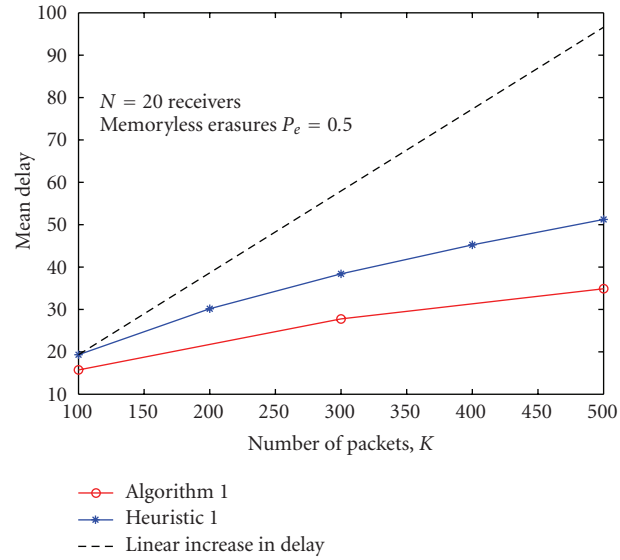


FIGURE 8: The effect of block size on the mean delay. If the delay of transmitting $K = 100$ packets in Heuristic 1, d_{100}, is taken as the reference, we can see that the delay of including $\lambda \times 100$ packets in transmission is less than λd_{100}. The same observation applies to the delay of Algorithm 1. In general, it is recommended to include all "order-insensitive" packets in making transmission decisions and only break the packets into subblocks when the assumption of order insensitivity between subblocks breaks down.

sizes or does it make sense to treat them as one single block of packets? The short answer is to include all "order-insensitive" packets in making transmission decisions and only break the packets into subblocks when the assumption of order insensitivity between subblocks breaks down. In the extreme case, an infinite number of order-insensitive packets provides an infinite pool of packets to choose from that can satisfy the demands of all receivers and are instantaneously decodable. Figure 8 shows the end-to-end delay when the number of packets in a block is finite and $K = 100$ packets is chosen as the reference for comparison. We can see that although the delay of transmitting λK packets, $d_{\lambda K}$, can be larger than that of transmitting K packets d_K, the delay does not increase by a factor of λ. That is $d_{\lambda K} < \lambda d_K$ and one does not benefit from breaking λK packets into λ subblocks of size K packets each. By treating λ subblocks of size K as one block of size λK, we add more degrees of freedom in making decisions.

Now we turn our attention to the delay performance of our algorithms in channels with memory. Figure 9 shows the mean delay of different algorithms for $K = 100$ packets and $N = 3$ receivers. The GEC parameters for all links are identical with $b = g$. The horizontal axis shows the memory content $\mu = 1 - 2b$. The first curve from above shows the performance of Algorithm 1 when the transmitter does not take channel conditions into account in making coding decisions. In other words, $w_j = |M_j|$ is used in Algorithm 1 as if the channel states were memoryless. For relatively large memory contents, this method results in the largest mean delay. The next curve shows the delay performance of Heuristic 1. The next two curves, which are almost indistinguishable, show the performance of Algorithm 1 which takes channel states into account (using (7)) and Heuristic 2 with 100 recursions. The last curve shows the best delay that can be achieved by occasionally violating the instantaneous decodability rule for one receiver in favor of the other two receivers that are predicted to be in good state in the next transmission round. More details can be found in [14].

Figure 10 shows the delay performance of Algorithm 1 using packet weights according to (7) for $N = 3$ to $N = 15$ receivers. Both the mean delay and mean delay plus one standard deviation of delay (across 1000 stochastic runs of the transmission) are shown. As expected, the delay increases as the number of receivers increases. Comparing the delay's standard deviation with its mean, we observe that when the number of receivers is 3–5, the delay is relatively more variant than when the number of receivers is 10–15. For example, for $N = 3$ and $\mu = 0.984$, the ratio of standard deviation to mean delay is around $3.225/0.8183 \approxeq 4$, whereas for $N = 15$ and $\mu = 0.94$ this ratio reduce to only $7.35/22.49 \approxeq 0.33$. One should keep these variations in mind when designing the transmission system.

We conclude this section with a brief look at the effect of post processing on the delay performance in channels with memory. Figure 11 shows different delays for $N = 15$ receivers and $K = 100$ packets. The figure confirms our earlier finding that selecting the maximum amount of coding among the optimal solutions provided by Algorithm 1 can result in larger end-to-end delays. We also note that serving the maximum number of receivers can have an adverse effect

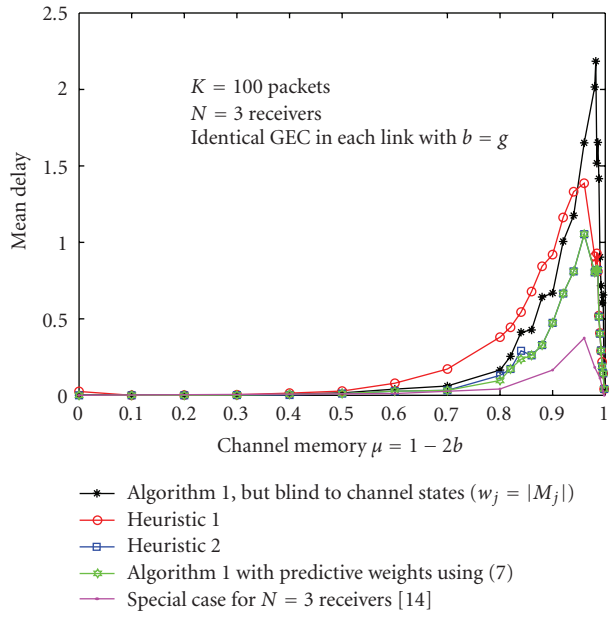


Figure 9: Delay performance of different algorithms in Gilbert-Elliott channels. The maximum number of recursions for Heuristic 2 is set to 100. By predicting next channel states and defining packet weights accordingly (see (7)), one can achieve considerably lower delays.

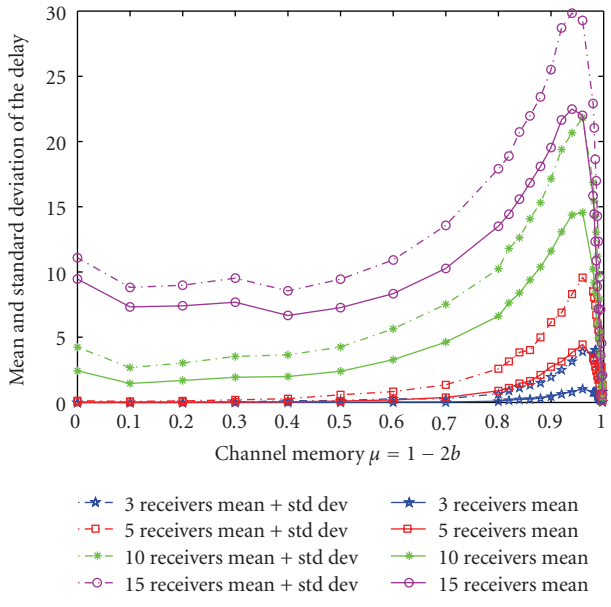


Figure 10: Delay performance of Algorithm 1 with weights defined using (7) for different number of receivers. As expected, the delay increases with the number of receivers. Both the mean delay (solid curves) and mean delay plus one standard deviation of delay (dashed-dotted curves) across 1000 stochastic runs of the transmission are shown.

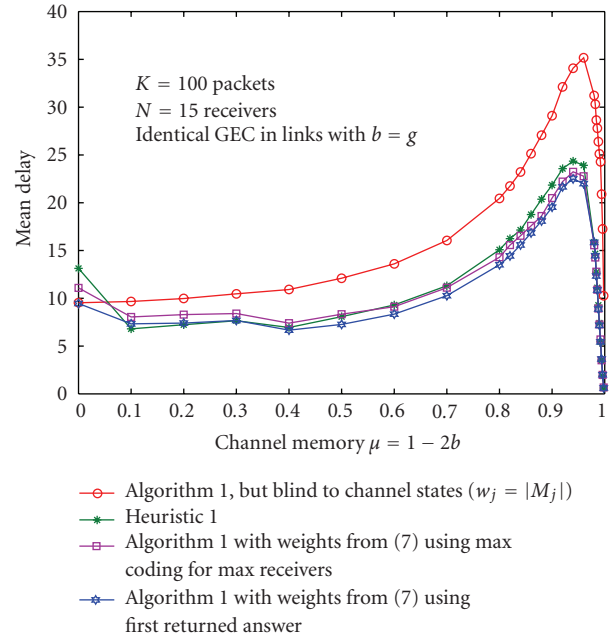


Figure 11: The effect of post processing on mean delay. As explained in the main text, whenever Algorithm 1 returns multiple solutions, choosing the maximum amount of coding and serving maximum number of receivers can often have adverse effects on the delay.

on the delay in GEC's. To explain this, consider an example where there are $K = 2$ left packets to be transmitted to $N = 100$ receivers. Packet 1 is needed by R_1 to R_{99} and packet 2 is needed by R_{99} and R_{100}. Since both packets are needed by R_{99}, we can either send packet 1 or 2, but not both. Now assume that R_1 to R_{99} are all predicted to be in good state with probability 0.01 and R_{100} is predicted to be in good state with probability 0.98, so that $w_1 = w_2 = 0.99$ according to (7). Therefore, transmission of either packet seems to be equally optimal. However, one can easily verify that the probability of at least one receiver among R_1 to R_{99} receiving packet 1 is only $1 - 0.99^{99} = 0.63$, whereas the probability of either R_{99} or R_{100} receiving packet 2 is $1 - 0.99 * 0.02 = 0.9802$. Therefore, it makes sense to satisfy only two receivers, one of which has a high priority due its good channel conditions.

7. Conclusions

In this paper, we provided an online optimal network coding scheme with feedback to minimize decoding delay in each transmission round in erasure broadcast channels. Efficient search algorithms for the optimal network coding solution, as well as heuristic methods were presented and their delay and computational performance were tested in several system scenarios. We found that adopting an optimized approach using as much information about the channel as possible, such as memory, leads to a significantly better decoding delay. An interesting problem for future research is

to relax the instantaneous decodability condition to L-step decodability and investigate the delay-throughput tradeoff.

Appendix

Here we prove by structural induction that (a) every result returned by Algorithm 1 is a solution of (4) and (b) the set of solutions returned by the algorithm contains all the optimal solutions. We note that the algorithm is designed to discard infeasible vectors and those solutions that are clearly suboptimal at each recursion to improve performance. The latter is based on positiveness of the elements of $\mathbf{w}$ as explained below.

The algorithm generates a binary tree. Each node represents a problem of size k and $\mathcal{P}_k$, and branches into two subproblems of size $\mathcal{P}_{k-k_u-k_s-1}$ and $\mathcal{P}_{k-k_u-1}$. The former subproblem is a result of setting $x_s = 1$ and the latter a result of setting $x_s = 0$. A leaf is reached when we need to solve $\mathcal{P}_1$. Without loss of generality let us assume that the variable to be examined is the first variable ($s = 1$) which is followed by k_s variables (x_2 to x_{k_s+1}) that are constrained to x_1, $k-k_u-k_s-1$ variables (x_{k_s+2} to x_{k-k_u}) that are constrained but not to x_1, and finally k_u unconstrained variables x_{k-k_u+1} to x_k. This can be easily accomplished by rearranging the columns of A.

For $k = 1$, it is clear that the only optimal solution to $\mathcal{P}_1$ is $x_1 = 1$ which is returned by the algorithm. Hence, the *minimal structure* of the algorithm returns the optimal solution and our claim is true for $k = 1$. The induction hypothesis is that the two subproblems $\mathcal{P}_{k-k_u-k_s-1}$ and $\mathcal{P}_{k-k_u-1}$ have only discarded infeasible vectors and some suboptimal solutions. We need to prove that the same statement applies to the parent problem $\mathcal{P}_k$.

We first look at the left branch where $x_1 = 1$. According to the construction of the algorithm, any solution such as $\mathbf{x}_1$ of length $k-k_s-k_u-1$ provided by the left branch $\mathcal{P}_{k-k_s-k_u-1}$ is appended by the parent problem $\mathcal{P}_k$ to form

$$\mathbf{x} = \left[1, \underbrace{0, 0, \ldots, 0}_{k_s}, \mathbf{x}_1, \underbrace{1, 1, \ldots, 1}_{k_u}\right], \tag{A.1}$$

where the head variable x_1 is set to one, variables constrained to x_1 are set to zero and all unconstrained variables are set to one. We first prove that $\mathbf{x}$ is indeed a solution and then show that changing any element of $\mathbf{x}$ results in either an infeasible or a clearly suboptimal $\mathbf{x}$. We use Definitions 6–8.

(i) For

$$\mathbf{x} = \left[1, \underbrace{0, 0, \ldots, 0}_{k_s}, \mathbf{x}_1, \underbrace{1, 1, \ldots, 1}_{k_u}\right], \tag{A.2}$$

we write the condition $A\mathbf{x}$ as a weighted sum of columns of A. That is, $A\mathbf{x} = 1\mathbf{a}_1 + A'\mathbf{x}_1 + 1\mathbf{a}_{k-k_u+1} + \cdots + 1\mathbf{a}_k$, where A' is a submatrix of A of size $N \times (k-k_s-k_u-1)$, which is input to $\mathcal{P}_{k-k_s-k_u-1}$, and according to the induction hypothesis $A'\mathbf{x}_1 \leq \mathbf{1}_N$. But since no variable in $\mathcal{P}_{k-k_u-k_s-1}$ is constrained to x_1, no column in A' and $\mathbf{a}_1$ can have ones in the same row position. Therefore, $1\mathbf{a}_1 + A'\mathbf{x}_1 \leq \mathbf{1}_N$.

(ii) Since x_{k-k_u+1} to x_k are unconstrained, no column $\mathbf{a}_{k-k_u+1}$ to $\mathbf{a}_k$ can have ones in the same row position. Hence, $1\mathbf{a}_{k-k_u+1} + \cdots + 1\mathbf{a}_k \leq \mathbf{1}_N$.

(iii) Using similar arguments, we can assert that no column in A' or $\mathbf{a}_1$ can have ones in the same row positions as $\mathbf{a}_{k-k_u+1}$ to $\mathbf{a}_k$ do. Therefore, $1\mathbf{a}_1 + A'\mathbf{x}_1 + 1\mathbf{a}_{k-k_u+1} + \cdots + 1\mathbf{a}_k \leq \mathbf{1}_N$ and $\mathbf{x}$ is a solution.

(iv) We now argue that variables x_2 to x_{k_s+1} cannot be anything other than zero. This directly follows from the fact that x_1 is constrained with x_i for $2 \leq i \leq k_s+1$ and hence, in any given solution they cannot be simultaneously one.

(v) Since we have already found a solution $\mathbf{x}$ where the first and last k_u variables are one, we know that any other solution such as $\mathbf{x}'$ with one or more zeros in these positions becomes suboptimal and can be discarded. That is, $\mathbf{w}^T\mathbf{x}' < \mathbf{w}^T\mathbf{x}$ due to positiveness of elements of $\mathbf{w}$.

(vi) Finally, according to induction hypothesis, we know that $\mathbf{x}_1$ cannot be changed into anything other than what $\mathcal{P}_{k-k_s-k_u-1}$ provides without making it either infeasible or suboptimal.

In summary, for each solution $\mathbf{x}_1$ provided by the left branch $\mathcal{P}_{k-k_s-k_u-1}$, the constructed vector

$$\mathbf{x} = \left[1, \underbrace{0, 0, \ldots, 0}_{k_s}, \mathbf{x}_1, \underbrace{1, 1, \ldots, 1}_{k_u}\right], \tag{A.3}$$

is the only solution that is not trivially suboptimal.

Now we look at the right branch where $x_1 = 0$. According to the construction of the algorithm, a given solution such as $\mathbf{x}_0$ of length $k-k_u-1$ provided by the right branch $\mathcal{P}_{k-k_u-1}$ is appended by the parent problem $\mathcal{P}_k$ to form

$$\mathbf{x} = \left[0, \mathbf{x}_0, \underbrace{1, 1, \ldots, 1}_{k_u}\right], \tag{A.4}$$

where the head variable is set to zero and all unconstrained variables are set to one. We need to show that for a given $\mathbf{x}_0$ this is indeed a solution. We then show that changing any element of $\mathbf{x}$ can only result in an infeasible vector, a clearly suboptimal solution, or a duplicate solution already provided by the left branch and hence, can be discarded. We use Definitions 6–8.

(i) We write $A\mathbf{x}$ as $A\mathbf{x} = 0\mathbf{a}_1 + A''\mathbf{x}_0 + 1\mathbf{a}_{k_s+2} + \cdots + 1\mathbf{a}_k$, where A'' is a submatrix of A of size $N \times (k-k_u-1)$, which is input to $\mathcal{P}_{k-k_u-1}$, and according to the induction hypothesis $A''\mathbf{x}_0 \leq \mathbf{1}_N$. Similar to the arguments for the left branch, we can assert that no column $\mathbf{a}_{k-k_u+1}$ to $\mathbf{a}_k$ corresponding to unconstrained variables can have ones in the same row position. Hence, $1\mathbf{a}_{k-k_u+1} + \cdots + 1\mathbf{a}_k \leq \mathbf{1}_N$. Furthermore, that no column in A'' can have ones in the same row positions as $\mathbf{a}_{k-k_u+1}$ to $\mathbf{a}_k$. Therefore, $A''\mathbf{x}_0 + 1\mathbf{a}_{k-k_u+1} + \cdots + 1\mathbf{a}_k \leq \mathbf{1}_N$ and $\mathbf{x}$ is a solution.

(ii) Since we have already found a solution $\mathbf{x}$ where the last k_u variables are one, we know that any other solution such as $\mathbf{x}'$ with one or more zeros in these positions becomes suboptimal and can be discarded.

(iii) Finally, we show that any vector of the form

$$\mathbf{x}' = \left[1, \mathbf{x}_0, \underbrace{1, 1, \ldots, 1}_{k_u}\right] \tag{A.5}$$

with a one in the first variable is either infeasible or is already constructed based on solutions from the left branch and hence, need not be considered twice. We consider two possibilities for $\mathbf{x}_0 = [x_2, \ldots, x_{k_s+1}, x_{k_s+2}, \ldots, x_{k-k_u}]$. If $x_i = 1$ for any $2 \leq i \leq k_s + 1$, then we have already shown in the analysis of the left branch that

$$\mathbf{x}' = \left[1, \mathbf{x}_0, \underbrace{1, 1, \ldots, 1}_{k_u}\right] \tag{A.6}$$

is infeasible because x_1 and x_i are constrained to each other. If none of x_2 to x_{k_s+1} are one, then $\mathbf{x}'$ will be of the form

$$\mathbf{x}' = \left[1, \underbrace{0, 0, \ldots, 0, \mathbf{x}_1}_{\mathbf{x}_0}, \underbrace{1, 1, \ldots, 1}_{k_u}\right] \tag{A.7}$$

for some $\mathbf{x}_1$. But, $\mathbf{x}_1$ has to be a solution of $\mathcal{P}_{k-k_s-k_u-1}$. Hence, considering vectors of the form

$$\mathbf{x}' = \left[1, \underbrace{0, 0, \ldots, 0, \mathbf{x}_1}_{\mathbf{x}_0}, \underbrace{1, 1, \ldots, 1}_{k_u}\right] \tag{A.8}$$

does not lead to any new solution.

In summary, for each solution $\mathbf{x}_0$ provided by the right branch $\mathcal{P}_{k-k_u-1}$, the constructed vector

$$\mathbf{x} = \left[0, \mathbf{x}_0, \underbrace{1, 1, \ldots, 1}_{k_u}\right] \tag{A.9}$$

is the only novel solution that is not trivially suboptimal. By combining the arguments of left and right branch, the induction claim is proven.

Acknowledgments

The authors wish to thank anonymous reviewers for their valuable comments which helped to improve the presentation of this paper. In the early stages of this work, the authors benefited from fruitful discussions with Ralf Koetter. This paper is dedicated to his memory. Preliminary results of this paper were presented in the 2009 Workshop on Network Coding, Theory and Applications (NetCod 2009), Lausanne, Switzerland. The work of P. Sadeghi was supported under ARC Discovery Projects funding scheme (Project no. DP0984950). The work of D. Traskov was supported by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (Contract no. 216715).

References

[1] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[2] T. Ho, M. Medard, R. Koetter, et al., "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[3] A. Eryilmaz, A. Ozdaglar, and M. Medard, "On delay performance gains from network coding," in *Proceedings of the IEEE Annual Conference on Information Sciences and Systems (CISS '06)*, pp. 864–870, Princeton, NJ, USA, March 2006.

[4] D. E. Lucani, M. Stojanovic, and M. Medard, "Random linear network coding for time division duplexing: when to stop talking and start listening," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 1800–1808, April 2009.

[5] J.-K. Sundararajan, D. Shah, and M. Medard, "Feedback-based online network coding," Submitted to *IEEE Transactions on Information Theory*, http://arxiv.org/pdf/0904.1730v1.

[6] J.-K. Sundararajan, P. Sadeghi, and M. Medard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *Proceedings of the Workshop on Network Coding, Theory, and Applications (NetCod '09)*, Lausanne, Switzerland, June 2009.

[7] V. K. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74–93, 2001.

[8] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Proceedings of the 4th Workshop on Network Coding, Theory, and Applications (NetCod '08)*, Hong kong, January 2008.

[9] D. Bertsimas and R. Weissmantel, *Optimization Over Integers*, Dynamic Ideas, Belmont, Mass, USA, 2005.

[10] T. S. Rappaport, *Wireless Communications, Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.

[11] P. Sadeghi, R. A. Kennedy, P. B. Rapajic, and R. Shams, "Finite-state Markov modeling of fading channels: a survey of principles and applications," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 57–80, 2008.

[12] M. Mushkin and I. Bar-David, "Capacity and coding for the Gilbert-Elliot channels," *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1277–1290, 1989.

[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *Proceedings of the ACM Computer Communication Review (SIGCOMM '06)*, vol. 36, pp. 243–254, ACM Press, October 2006.

[14] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *Proceedings of the Workshop on Network Coding, Theory, and Applications (NetCod '09)*, pp. 80–85, Lausanne, Switzerland, June 2009.