Performance Analysis of Relay-Assisted Network-Coding ARQ with Space-Time Cooperation in Wireless Relay Networks

Jung-Chun Kao, *Member, IEEE*

***Abstract*—Relay-assisted network-coding (RANC) automatic repeat request (ARQ) protocols are ARQ protocols that leverage both opportunistic retransmission and network coding. This paper proposes an RANC ARQ protocol, called the decode-and-cooperate (DC) protocol, which can readily integrate space-time cooperation. The closed-form formulas for saturation throughput, segment delay, and buffer occupancy are derived for DC in slotted wireless relay networks. Extensive analysis and simulation results, which validate each other, confirm that compared to a stop-and-wait cooperative ARQ protocol, DC achieves significant performance gains in terms of throughput and delay.**

***Index Terms*—Cooperative automatic repeat request (ARQ), wireless network, queueing model, opportunistic retransmission, network coding.**

I. Introduction

A COMMON characteristic of various wireless channels is the error-prone nature on data delivery. Because the quality of received signal in wireless environments varies (due to path loss, fading, noise, interference, etc.), packets might fail to reach destination nodes. To achieve efficient and reliable data delivery in wireless networks at the *link layer*, automatic repeat request (ARQ), which perceives packet loss by using acknowledgements and retransmits lost packets on a *per-packet basis*, are enhanced by several important techniques. For example, relaying is widely used in advanced communication networks such as WiMAX and LTE-Advanced to enhance data rate and coverage. More recently, link-layer network coding [1], [2] and opportunistic retransmission [3], [4] have been promising solutions to further improve network performance.

Link-layer network coding techniques (e.g., [1]) are oblivious and resilient to packet loss, while having practically negligible communication overhead caused by acknowledgments (ACKs) themselves and unnecessary packet retransmissions due to ACK loss. A link-layer network coding technique is on a *per-segment basis*, where a *segment* consists of a number of *original blocks*. The source node encodes original blocks into *coded blocks* and keeps sending out the coded blocks without having to wait for per-packet ACKs back. Although blocks could still be lost in transit, as long as the destination receives a sufficient number of coded blocks, the entire segment can be fully retrieved. From the *segment perspective*, link-layer network coding aforementioned is stop-and-wait; whereas it belongs to the continuous ARQ family at the *packet aspect*.

The use of link-layer network coding decreases the number of needed ACKs to *one per segment* from one per packet. According to [1], ACK loss due to poor or varying channel condition may cause the ARQ timeout, which triggers a redundant retransmission of the same data packet and causes a significant communication overhead. By reducing the number of needed ACKs, link-layer network coding can effectively reduce such a communication overhead.

Opportunistic retransmission leverages the broadcast nature of wireless communication and the benefit of multi-path transmission. If the destination does not receive a packet successfully, a close-by node overhearing the packet can act as a *relay node* and retransmit the packet in place of the source. Opportunistic retransmission is beneficial in reliability, coverage, and throughput, particularly when the direct link connecting source and destination nodes is of low quality.

It is beneficial for an ARQ protocol to exploit *either* network coding *or* opportunistic retransmission. Studies have shown that *network coding* [5], [6] brings performance gains from both theoretic and practical aspects. Most of the work on network coding focuses on specific scenarios with multicast traffic [7]–[10], two-way flows [11], [12], or multiple unicast flows [13]; while only few works examine the techniques considering one-way relaying systems.

For *one-way* relaying systems, [1] shows that the MAC-layer random network coding (MRNC) protocol outperforms Hybrid Automatic Repeat reQuest (HARQ) in WiMAX systems. According to [2], the MRNC protocol can be regarded as a special case of the N-in-1 retransmission with network coding scheme (abbreviated as N-in-1 ReTX). N-in-1 ReTX [2] exploits random linear network coding [14], [15] to share retransmission overhead over multiple frames, thus offering improved performance, compared to MRNC and the traditional HARQ. Reference [16] shows that utilizing network coding can reduce end-to-end delay to send a segment along parallel paths, compared to conventional routing. Reference [17] studies transmission efficiency and outage probability in a relay-aided system in which network coding is applied on top of channel coding.

Manuscript received July 20, 2013; revised December 15, 2013 and March 11, 2014; accepted March 31, 2014. Date of publication April 6, 2014; date of current version August 8, 2014. This work was supported in part by the National Science Council Taiwan, under Grant NSC 102-2221-E-007-011. The associate editor coordinating the review of this paper and approving it for publication was W. Gerstacker.

The author is with the Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: jungchuk@cs.nthu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2014.2315793

ARQ protocols exploiting *opportunistic retransmission* are also known as cooperative ARQ protocols or *C-ARQ* protocols. Fundamental concepts, theoretical bounds, and practical protocols of C-ARQ schemes have been extensively studied in the literature. References [18]–[20] study the outage probability and SNR gain of several C-ARQ protocols with single relay. Reference [4] derives the first delay model for C-ARQ protocols with single relay, which shows that C-ARQ protocols gain throughout and delay over hybrid ARQ. Stability region and throughput region for simplified cases are investigated in [21], [22]. All the above works consider a three-node topology that has at most one relay for each source-destination pair.

For the case when each source-destination pair is allowed to be assisted by multiple relays, the C-ARQ protocols presented in [3], [23]–[26] are designed to cope with the cooperation and contention among multiple relays in a distributed fashion. For example, [3] includes a distributed relay selection algorithm that chooses a set of eligible relays among all qualified relays and a relay prioritization mechanism that makes sure high-priority relays transmit with high probability by leveraging the enhanced distributed channel access of IEEE 802.11e.

A considerable amount of works have shown the benefit of leveraging *either* network coding *or* opportunistic relaying; nevertheless, it would be more beneficial to exploit *both* in one-way wireless relay networks. Along this research direction, [27], [28] introduce the framework of relay-assisted network-coding (RANC) ARQ, which is recapped in Section II. Several simple yet efficient RANC ARQ protocols are proposed in [27] for wireless relay networks and one is proposed in [28] for IEEE 802.11 compatible wireless local area networks. Reference [27] derives the closed-form formulas of throughput and delay of the protocols introduced in it.

None of these RANC ARQ protocols exploit *space-time cooperation*. Space-time cooperation is achieved by the use of space-time coding with multiple antennas located at several nodes. The benefit of space-time cooperation is studied in [29]. Moreover, real-world implementations [30]–[32] of distributed space-time codes demonstrate satisfactory time synchronization among nodes (for example, a measured error of within tens of nanoseconds is reported in [30]) and showcase high data rates in distributed MIMO testbeds.

The focus of this paper is on proposing and analyzing a RANC ARQ protocol that can readily take advantage of space-time cooperation in one-way wireless relay networks. We propose the decode-and-cooperate (DC) protocol, the first RANC ARQ protocol that exploits space-time cooperation. The reason that existing RANC ARQ protocols cannot leverage space-time cooperation is because for all of them, a relay node starts to send recoded blocks before retrieving the entire segment. This way, all coded and recoded blocks carry distinct information bits and therefore it is impossible for the existing RANC ARQ protocols to leverage space-time cooperation.

For the DC protocol, we derive the closed-form formulas of the saturation throughput, average segment delay, and average buffer occupancy in one-way wireless relay networks. Our analysis framework transforms the DC protocol to an M/G/1 queue with vacation. The three performance metrics are derived, after obtaining the first and the second moments of the service time distribution of the queue.

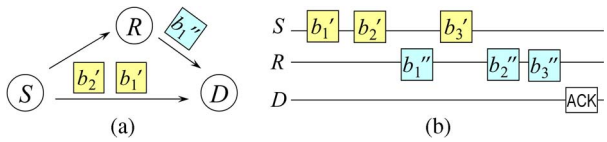

Fig. 1. (a) A three-node network containing source S, destination D, and relay R. The source node and relay node send out coded blocks, denoted by b_i', and recoded blocks, denoted by b_i'', respectively. (b) A high-level overview of RANC ARQ protocols.

The remainder of this paper is organized as follows. Section II outlines the concept of RANC ARQ. Section III introduces the network model we consider. Section IV presents the DC protocol we develop to exploit space-time cooperation. Throughput and delay for the proposed protocol are derived in Section V. Section VI evaluates DC's performance and compares its performance, by both analytical and simulation results, with the performances of the representative C-ARQ protocol, called the opportunistically forwarding (OF) protocol [27], and the optimal single-relay RANC ARQ protocol without space-time cooperation, called the listen-and-supersede (LS) protocol [27]. Section VII, presents concluding remarks.

II. Preliminary: An Introduction to RANC ARQ

With a RANC ARQ protocol, messages are transmitted on a per-segment basis, where a *segment* consists of K*original blocks*, denoted by $b_1, b_2, \ldots, b_K$. Fig. 1 gives a high-level overview of a RANC ARQ protocol. Both source and relay encode data packets by network coding techniques. Instead of transmitting original blocks, a source sends out *coded blocks*, denoted by b_1', b_2', and so forth. Coded blocks are produced, for example, by taking linear combinations of the original blocks over a finite field GF(2^n), where n is a positive integer.

Instead of forwarding overheard coded blocks, the relay node produces a number of *recoded blocks* (denoted by b_1'', b_2'', and so forth) by, for example, taking linear combinations of all or a subset of the overheard coded blocks. After that, the recoded blocks are sent out to the destination node.

Any coded/recoded block may or may not reach the destination node, depending on the varying channel condition. Regardless of which blocks are lost, the destination can retrieve the entire segment from its received (coded and recoded) blocks, as long as the destination has received enough blocks. For a segment, the K original blocks can regarded as K unknowns the destination needs to solve; each coded/recoded block can be regarded as a linear equation involving the K unknowns. Reception of a block at the destination can be regarded as getting a linear equation. The destination can solve all the K unknowns (and equivalently reconstruct the entire segment) if and only if it has collected K linear equations that are linearly independent to each other.

Once the destination node retrieves the entire segment, a *segment transmission* completes. The destination sends out a *per-segment ACK*, notifying the source and relay nodes of the completion of the segment transmission.

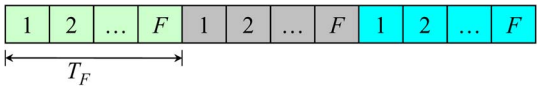

Fig. 2. TDMA scheme with F flows. The frame duration is T_F seconds.

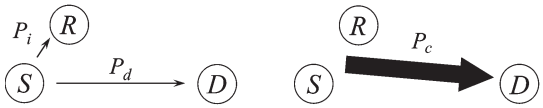

Fig. 3. Link reception probabilities of blocks (denoted by P_d for the direct link from S to D, P_i for the interconnection link from S to R, and P_c for the cooperative link from S/R to D), for a given flow.

III. Network Model

Same as [27], we consider a wireless network in which all the nodes are in close proximity. There are F flows sharing the same channel in a TDMA manner illustrated in Fig. 2. A single *data packet* (i.e., a *block*) is transmitted in a time slot. ACKs are sent in the reverse direction over separate channels; or they are short enough to fit in the same time slots allotted to the associated blocks.

A flow in the network involves a source, a destination, and a relay. Any node is allowed to play more than one role simultaneously, acting as source, destination, and/or relay. It is assumed that a relay overhears coded blocks and sends recoded blocks only in the time slots allocated to the corresponding flow. In this way, a node can easily act as a relay for one or multiple flows because each flow uses exclusive time slots.

In reality, it is possible that source and relay cooperate in such a way that they send out blocks simultaneously in some time slots. The DC protocol proposed in this paper is specifically designed to take advantage of space-time cooperation. On the contrary, existing link-layer RANC ARQ protocols assume of no simultaneous transmission due to relay's higher priority.

Original blocks are assumed to be generated at each source node in accordance with a Poisson arrival process with random-sized segments. That is, arrival times of segments follow a Poisson process and the segment size in blocks, K, is random. K can be *arbitrarily distributed* with finite mean and finite variance. The mean of K is denoted by $\mathbf{E}[K]$ or $\overline{K}$.

Error-prone wireless links are modeled by erasure channels in which the reception probabilities of both blocks and ACKs are in [0, 1]. The reception probability of blocks over a link and that of ACKs over the reverse link can be distinct, because these two types of packets may be transmitted over different channels, at different rates, of different lengths, etc.

As to blocks, Fig. 3 illustrates the reception probability of the direct link from source to destination (denoted by P_d), the reception probability of the interconnection link from source to relay (denoted by P_i), and the reception probability of the cooperative link from source/relay to destination (denoted by P_c), respectively. The cooperative link implies space-time cooperation among source and relay; hence $P_c \geq P_d$ and $P_c \geq P_{RD}$, where P_{RD} is the link reception probability of the relay-to-destination link.

The destination node could indicate success or failure of a reception by broadcasting an ACK. For RANC ARQ protocols, per-segment ACKs are assumed to be detected reliably by both source and relay. This is a reasonable assumption; the justification is the same as what is given in [20].[1]

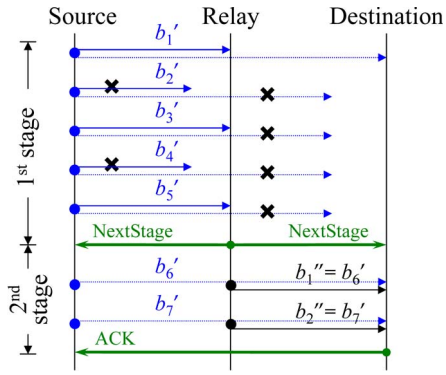

Fig. 4. Example of the DC protocol with the segment size of three. Dotted lines depict direct communications from source to destination. The relay receives $\{b_1', b_3', b_5'\}$ in the first stage. b_6' and b_1'' are sent simultaneously; so are b_7' and b_2''. The destination receives one block $\{b_1'\}$ in the first stage and two distinct blocks $\{b_1'' = b_6', b_2'' = b_7'\}$ in the second stage.

IV. Proposed Protocol

The decode-and-cooperate (DC) protocol is a RANC ARQ protocol we design to take advantage of space-time cooperation. DC belongs to the continuous ARQ family from the packet perspective; whereas it is stop-and-wait at the segment aspect. This section outlines the segment transmission process of DC, explains how DC leverages space-time cooperation, and introduces an algorithm by which DC generates coded/recoded blocks such that all blocks received by the destination are guaranteed innovative.

A. Two-Stage Segment Transmission Process

As illustrated in Fig. 4, the DC protocol consists of two stages. In the first stage, the source sends out coded blocks, $b_1', b_2', \ldots$, in the allocated time slots. Meanwhile, the relay listens to the channel silently and attempts to decode that segment. The first stage continues until either the destination can decode the entire segment, or the relay can decode the entire segment, whichever earlier. In the former case, the destination sends ACK to the source/relay, which finishes the segment transmission process. In the latter case, the relay *decodes* the entire segment and then *cooperates* with the source by using space-time coding with the antennas at the source and relay. This is why we name this protocol decode-and-cooperate. If needed, the *NextStage* signaling can be used to explicitly notify a switch to the second stage.

In the second stage, the relay generates recoded blocks in a way such that the recoded blocks are identical to the coded blocks the source generates in the second stage. Feeding the same sequence number n into (1) makes a recoded block identical to the coded block counterpart. The relay learns the sequence number of the last coded block from TDMA's timing relation or the sequence number field contained in received

[1]For non-RANC ARQ protocols, the source receives per-packet ACKs successfully with a probability of $P_{\mathrm{D,ACK}}$ which is a number in [0, 1], while the appropriately-chosen relay is assumed to overhear per-packet ACKs surely.

TABLE I
TRANSPOSED ALAMOUTI 2×1 CODE

	Antenna 1	Antenna 2
Odd symbol period	x_1	$-x_2^*$
Even symbol period	x_2	x_1^*

blocks. A coded block and the recoded block counterpart are sent concurrently, as shown in Fig. 4. Such a concurrent transmission is inherent in the TDMA scheme (with the help of time synchronization techniques [30]); whereas in general it requires the relay to be informed of the scheduling of the source, or vice versa.

Note that although the information bits of a recoded block and the corresponding coded block are identical to each other, the symbols actually transmitted over the wireless channel(s) are different due to space-time cooperation, which will be explained in Section IV-B in detail. The second stage continues until the destination can retrieve the entire segment. Then, the destination sends a per-segment ACK back to the relay and source, which finishes the segment transmission.

B. Space-Time Cooperation

We explain how the DC protocol achieves space-time cooperation by an example: Consider that the source, relay, and destination nodes have one antenna each, and the source-relay-destination triple uses the *transposed* Alamouti 2×1 code. We present the transposed Alamouti 2×1 code because it is a rate-1 code and it helps to minimize signaling overhead.

As shown in Table I, the source (Antenna 1) simply sends coded blocks in a normal way through the entire segment transmission process: Represent a coded/recoded block by a transmission sequence of modulated symbols, $(x_1, x_2, \ldots, x_n)$. Antenna 1 sends x_1 in the first symbol period, x_2 in the second symbol period, and so on.

Instead of sending the symbols directly, as shown in Table I, the relay (Antenna 2) groups the symbols into groups of two: Antenna 2 sends $-x_2^*$ in the first symbol period, x_1^* in the second symbol period, $-x_4^*$ in the third symbol period, x_3^* in the fourth symbol period, and so on.

Decoding of any space-time code used in the DC protocol can be done by regular methods, for example, the Alamouti code can be decoded by maximum likelihood decoder and zero forcing. Indeed, the destination can decode the symbols locally without any feedback about whether the source and relay are sending symbols concurrently: Two branches (corresponding to DC's two stages) in parallel at the destination can be implemented for decoding the symbols and only one branch output will pass the forward error detection [33].

C. Generation of Coded and Recoded Blocks

For the DC protocol, one way to generate coded/recoded blocks is to use a Reed-Solomon based algorithm over a finite field. The algorithm is designed so that the coding coefficients for different blocks correspond to distinct rows of a Vandermonde matrix. That is, the nth coded block is

$$b'_n = \sum_{i=1}^{K} i^n b_i. \tag{1}$$

A recoded block is the same as its corresponding coded block.

This algorithm has a property—as long as the total number of coded blocks is smaller than the finite field size, any K distinct blocks received at the destination are guaranteed to be innovative. This great property holds true because the K-by-K square matrix formed by the coding coefficients of any K blocks is a Vandermonde matrix with non-zero determinant. Hence, it is guaranteed that relay (and destination) can decode the entire segment once it has received K blocks.

V. PERFORMANCE ANALYSIS

In this section, we derive the following performance metrics for the decode-and-cooperate (DC) protocol:

- η: The saturation throughput, defined as the expected number of blocks successfully decoded at the destination per second, assuming that the source's transmission buffer is always full.
- T: The average segment delay, defined as the expected time elapsed between a segment's arrival at the source node and the end of the segment transmission (when the destination node can decode the entire segment).
- L: The average buffer occupancy (i.e., the expected number of segments buffered at the source node).

These performance metrics for the DC protocol can be obtained with the help of the unified analysis framework developed in our previous work [27], in which any ARQ protocol is transformed into an M/G/1 queue with vacation. Different ARQ protocols have distinct distributions of service times, thereby resulting in different performances. As shown in Theorem 1, the fundamental formulas commonly used in the analysis framework require of knowing $\overline{N}$ and $\overline{N^2}$, where N is the random variable representing the total number of blocks sent by source, relay, or both for a segment (until the segment is fully decoded at destination).

Theorem 1: Given the arrival rate of segments (λ_s) and the first and second moments of N ($\overline{N}$ and $\overline{N^2}$), the average segment delay T, the average buffer occupancy L, and the saturation throughput η for the TDMA system are

$$T = \frac{\lambda_s \overline{N^2} {T_F}^2}{2(1-\lambda_s \overline{N} T_F)} + \left(\overline{N} - \frac{F-1}{F} + \frac{1}{2}\right) T_F$$

$$L = \lambda_s T = \frac{{\lambda_s}^2 \overline{N^2} {T_F}^2}{2(1-\lambda_s \overline{N} T_F)} + \lambda_s \left(\overline{N} - \frac{F-1}{F} + \frac{1}{2}\right) T_F$$

$$\eta = \frac{\overline{K}}{\overline{N} T_F}$$

where F is the number of time slots per time frame; T_F is the length of a time frame; $\overline{K}$ is the average segment size.

The proof of Theorem 1 can be found in [27]. With Theorem 1, what we need to do for obtaining η, T, and L of the DC protocol is to derive $\overline{N}$ and $\overline{N^2}$ for the DC protocol. To facilitate the derivation, we list below the notations:

- N: the number of blocks sent by source, relay, or both for a segment to be fully decoded at destination;
- N_S: the number of blocks sent by node S for a segment in the first stage;

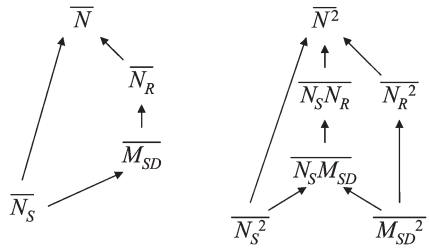

Fig. 5. Dependency charts of the first-order measures ($\overline{N}$, $\overline{N_S}$, $\overline{N_R}$, and $\overline{M_{SD}}$) and of the second-order measures ($\overline{N^2}$, $\overline{N_S{}^2}$, $\overline{N_R{}^2}$, $\overline{N_S N_R}$, $\overline{M_{SD}{}^2}$, and $\overline{N_S M_{SD}}$).

- N_R: the number of blocks sent by node R for a segment in the second stage;
- M_{SD}: the number of blocks sent by node S that node D receives successfully in the first stage of a segment transmission.

In the second stage of DC, the source and relay transmit exactly the same blocks at the same time slots. A recoded block and its counterpart in coded blocks together are counted once and contribute one to N. So $N = N_S + N_R$ for the DC protocol regardless of what space-time code is used.

For a contained presentation, we derive $\overline{N}$ and $\overline{N^2}$, provided that all segments are of a constant size k. These results can be easily generalized to the case of having random-sized segments, by unconditioning $\mathbf{E}[N|K=k]$ and $\mathbf{E}[N^2|K=k]$ over all possible segment sizes

$$\overline{N} = \sum_{k=1}^{\infty} \Pr(K=k)\mathbf{E}[N|K=k] \tag{2}$$

$$\overline{N^2} = \sum_{k=1}^{\infty} \Pr(K=k)\mathbf{E}[N^2|K=k]. \tag{3}$$

Fig. 5 depicts the dependencies among relevant first-order measures and among relevant second-order measures, cluing how to derive $\overline{N}$ and $\overline{N^2}$ for DC. Take $\overline{N}$ as an example, we first derive $\overline{N_S}$, which makes it possible to derive $\overline{M_{SD}}$, which in turn facilitates us to derive $\overline{N_R}$. With $\overline{N_S}$ and $\overline{N_R}$ available, it is trivial to derive $\overline{N}$ since $\overline{N} = \overline{N_S} + \overline{N_R}$. $\overline{N}$ and $\overline{N^2}$ will be derived in this section according to the dependency charts.

To derive $\overline{N}$, the first step is to obtain a recursive expression for $\overline{N_S}$, as shown in Lemma 1. Then Lemma 2 can help to get $\overline{N_S}$'s closed-form formula which is shown in Corollary 1.

Lemma 1: Given a constant segment size $K = k$, $\overline{N_S}$ for the DC protocol is

$$\overline{N_S} = n_S(k,k)$$

The discrete function $n_S(x,y)$, where x and y are non-negative integers, is defined in a recursive form

$$n_S(x,y) = \frac{1}{p} + \frac{\alpha}{p} n_S(x-1,y) + \frac{\beta}{p} n_S(x,y-1) + \frac{\gamma}{p} n_S(x-1,y-1)$$

with the boundary conditions, $n_S(x,0) = n_S(0,y) = 0, \forall\, x, y \geq 0$. The constants—$p$, α, β, and γ—are $p = 1-(1-P_d)(1-P_i)$, $\alpha = P_d(1-P_i)$, $\beta = (1-P_d)P_i$, and $\gamma = P_d P_i$.

Proof: To derive $\overline{N_S}$ for the DC protocol, we only need to consider the first stage. The ending criterion of the first stage is that either node D or node R (or both) has received k blocks. Consider a certain segment transmission. Let $N_S(x,y)$ be a random variable representing the number of blocks sent by node S in the first stage until the moment when either node D has received x blocks or node R has received y blocks for the segment. In this proof, we put effort into derivation of $\overline{N_S(x,y)}$. Once $\overline{N_S(x,y)}$ is available, $\overline{N_S}$ can be easily obtained by $\overline{N_S} = \overline{N_S(k,k)}$ due to the ending criterion of the first stage of DC.

To derive $\overline{N_S(x,y)}$, let us split a segment transmission into two parts—the first transmission and the remaining transmissions (node S takes in the first stage until either node D has received x blocks or node R has received y blocks). Since the first transmission may or may not reach destination/relay, there are four possible outcomes: The first transmission sent by the source could reach neither node D nor node R, node D only, node R only, or both nodes D and R.

With a probability of $1-p = (1-P_d)(1-P_i)$, the first transmission reaches neither node D nor node R. In this case, for the segment transmission to complete, node D needs to receive x more blocks or node R needs to receive y more blocks. In other words, the source will take $N_S(x,y)$ more transmissions after the first transmission, or equivalently, it takes $N_S(x,y)+1$ transmissions in total. Similarly, the first transmission reaches node D but not node R with probability $\alpha = P_d(1-P_i)$; in this case the source takes $N_S(x-1,y)+1$ transmissions in total. The first transmission reaches node R but not node D with probability $\beta = (1-P_d)P_i$; in this case the source takes $N_S(x,y-1)+1$ transmissions in total. The first transmission reaches both node D and node R with probability $\gamma = P_d P_i$; in this case the source takes $N_S(x-1,y-1)+1$ transmissions in total. Using the law of total expectation, $\overline{N_S(x,y)}$ is obtained

$$\overline{N_S(x,y)} = (1-p)\overline{N_S(x,y)+1} + \alpha\overline{N_S(x-1,y)+1} + \beta\overline{N_S(x,y-1)+1} + \gamma\overline{N_S(x-1,y-1)+1}.$$

After simple algebraic manipulations, the above equation can be simplified as

$$\overline{N_S(x,y)} = \frac{1}{p} + \frac{\alpha}{p}\overline{N_S(x-1,y)} + \frac{\beta}{p}\overline{N_S(x,y-1)} + \frac{\gamma}{p}\overline{N_S(x-1,y-1)}.$$

Comparing the definition of $n_S(x,y)$ with the above equation, it is obvious that the discrete function $n_S(x,y)$ is equal to $\overline{N_S(x,y)}$ for $x > 0$ and $y > 0$. Because $\overline{N_S} = \overline{N_S(k,k)}$, $\overline{N_S} = n_S(k,k)$. So far we have proven this lemma. ■

Lemma 1 gives a recursive expression for $\overline{N_S}$. In case a closed form is preferred, Lemma 2 proven in Appendix A helps in converting a recursive expression to a closed-form formula.

Lemma 2: For any discrete function $f(x,y)$, where x and y are non-negative integers, if $f(x,y)$ satisfies all the following conditions:

1) $f(x,y) = \delta_{x,y} + af(x-1,y) + bf(x,y-1) + cf(x-1,y-1)$ for all positive integers x and y, where $\delta_{x,y}$, a, b, and c are constants;
2) $f(x,0) = 0$ for all non-negative integers x;
3) $f(0,y) = 0$ for all non-negative integers y;

then

$$f(x,y)=\sum_{i=1}^{x}\sum_{j=1}^{y}\sum_{u=0}^{\min(x-i,y-j)}\frac{(x-i+y-j-u)!}{(x-i-u)!(y-j-u)!u!}\times a^{x-i-u}b^{y-j-u}c^{u}\delta_{i,j}.$$

By Lemma 2, $\overline{N_S}$'s recursive expression derived in Lemma 1 can be converted to the closed form shown in Corollary 1.

Corollary 1: Given a constant segment size $K=k$, $\overline{N_S}$ for the DC protocol is

$$\overline{N_S}=\frac{1}{p}\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{u=0}^{\min(k-i,k-j)}\frac{(2k-i-j-u)!}{(k-i-u)!(k-j-u)!u!}\frac{\alpha^{k-i-u}\beta^{k-j-u}\gamma^{u}}{p^{2k-i-j-u}}$$

where $p=1-(1-P_d)(1-P_i)$, $\alpha=P_d(1-P_i)$, $\beta=(1-P_d)P_i$, and $\gamma=P_dP_i$.

Now let us derive $\overline{N}$, with the help of Wald's equation.

Theorem 2: Given a constant segment size $K=k$, $\overline{N}$ for the DC protocol is

$$\overline{N}=\frac{k}{P_c}+\left(1-\frac{P_d}{P_c}\right)\overline{N_S}$$

where $\overline{N_S}$ can be computed using the recursive expression in Lemma 1 or the closed-form formula in Corollary 1.

Proof: $\overline{N}=\overline{N_S}+\overline{N_R}$. Since we have derived $\overline{N_S}$ in Corollary 1, what is left is to derive $\overline{N_R}$.

To derive $\overline{N_R}$, we consider M_{SD} which is defined as the number of blocks sent by node S that are received successfully by node D in the first stage. Since node S sends N_S coded blocks in the first stage,

$$M_{SD}=\sum_{i=1}^{N_S}X_i$$

where X_is, for any positive integer i, are i.i.d. Bernoulli random variables with a success probability of P_d. Note that N_S depends on *past* and *current* X_is (i.e., $X_1,X_2,\ldots,X_{N_S}$) but not *future* X_is (i.e., $X_{N_S+1},X_{N_S+2},\ldots$). By Wald's equation, we get

$$\mathbf{E}[M_{SD}]=\mathbf{E}[N_S]\mathbf{E}[X_i]=\overline{N_S}P_d. \tag{4}$$

Given the value of M_{SD}, the second stage does not finish until node D receives $k-M_{SD}$ (distinct) blocks in the second stage. So N_R is a negative binomial random variable:

$$N_R\sim \mathrm{NB}(P_c,k-M_{SD}).$$

Therefore

$$\overline{N_R}=\mathbf{E}\left[\mathbf{E}[N_R|M_{SD}]\right]=\mathbf{E}\left[\frac{k-M_{SD}}{P_c}\right]=\frac{k-\mathbf{E}[M_{SD}]}{P_c}=\frac{k-\overline{N_S}P_d}{P_c} \tag{5}$$

where the last equality comes from (4).

Having (5), it is straightforward to get $\overline{N}$

$$\overline{N}=\overline{N_S}+\overline{N_R}=\frac{k}{P_c}+\left(1-\frac{P_d}{P_c}\right)\overline{N_S}.$$

So far we have proven this theorem. ■

To derive $\overline{N^2}$, the first step is to get a recursive expression of $\overline{{N_S}^2}$, as shown in Lemma 3. The proof is in the Appendix B.

Lemma 3: Given a constant segment size $K=k$, $\overline{{N_S}^2}$ for the DC protocol is

$$\overline{{N_S}^2}=n_S^{(2)}(k,k).$$

The discrete function $n_S^{(2)}(x,y)$, where x and y are both non-negative integers, is defined in a recursive form

$$n_S^{(2)}(x,y)=\left(\frac{2-p}{p^2}+\frac{2\alpha}{p^2}n_S(x-1,y)+\frac{2\beta}{p^2}n_S(x,y-1)+\frac{2\gamma}{p^2}n_S(x-1,y-1)\right)+\frac{\alpha}{p}n_S^{(2)}(x-1,y)+\frac{\beta}{p}n_S^{(2)}(x,y-1)+\frac{\gamma}{p}n_S^{(2)}(x-1,y-1)$$

with the boundary conditions, $n_S^{(2)}(x,0)=n_S^{(2)}(0,y)=0$, $\forall\, x,\ y\geq 0$. The discrete function $n_S(x,y)$ as well as the constants—p, α, β, and γ—are defined in Lemma 1.

Lemma 3 gives a recursive expression for $\overline{{N_S}^2}$. In case a closed-form formula is preferred, Lemma 2 can be exploited to get the closed-form formula shown in Corollary 2.

Corollary 2: Given a constant segment size $K=k$, $\overline{{N_S}^2}$ for the DC protocol is

$$\overline{{N_S}^2}=\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{u=0}^{\min(k-i,k-j)}\frac{(2k-i-j-u)!}{(k-i-u)!(k-j-u)!u!}\frac{\alpha^{k-i-u}\beta^{k-j-u}\gamma^{u}}{p^{2k-i-j-u}}\delta_{i,j}$$

where the constants—p, α, β, and γ—are defined in Lemma 1 and $\delta_{x,y}$ is defined as (see equation at the bottom of the page).

The next step is to obtain a recursive expression for $\overline{{M_{SD}}^2}$, as shown in Lemma 4. The proof is in the Appendix C.

$$\delta_{x,y}=\frac{2-p}{p^2}+\frac{2}{p^3}\left(\alpha\sum_{i=1}^{x-1}\sum_{j=1}^{y}\sum_{u=0}^{\min(x-1-i,y-j)}\frac{(x-1-i+y-j-u)!}{(x-1-i-u)!(y-j-u)!u!}\frac{\alpha^{x-1-i-u}\beta^{y-j-u}\gamma^{u}}{p^{x-1+y-i-j-u}}+\beta\sum_{i=1}^{x}\sum_{j=1}^{y-1}\sum_{u=0}^{\min(x-i,y-1-j)}\frac{(x-i+y-1-j-u)!}{(x-i-u)!(y-1-j-u)!u!}\frac{\alpha^{x-i-u}\beta^{y-1-j-u}\gamma^{u}}{p^{x+y-1-i-j-u}}+\gamma\sum_{i=1}^{x-1}\sum_{j=1}^{y-1}\sum_{u=0}^{\min(x-1-i,y-1-j)}\frac{(x-1-i+y-1-j-u)!}{(x-1-i-u)!(y-1-j-u)!u!}\frac{\alpha^{x-1-i-u}\beta^{y-1-j-u}\gamma^{u}}{p^{x-1+y-1-i-j-u}}\right)$$

Lemma 4: Given a constant segment size $K = k$, $\overline{M_{SD}}$ and $\overline{{M_{SD}}^2}$ for the DC protocol are

$$\overline{M_{SD}} = m_{SD}(k,k)$$

$$\overline{{M_{SD}}^2} = m^{(2)}_{SD}(k,k).$$

The discrete function $m_{SD}(x,y)$, where x and y are non-negative integers, is defined in a recursive form

$$m_{SD}(x,y) = \frac{\alpha+\gamma}{p} + \frac{\alpha}{p} m_{SD}(x-1,y) + \frac{\beta}{p} m_{SD}(x,y-1) + \frac{\gamma}{p} m_{SD}(x-1,y-1)$$

with the boundary conditions $m_{SD}(x,0) = m_{SD}(0,y) = 0$, $\forall\, x,\ y \geq 0$. The constants—p, α, β, and γ—are defined in Lemma 1.

The discrete function $m^{(2)}_{SD}(x,y)$, where x and y are non-negative integers, is defined in a recursive form

$$m^{(2)}_{SD}(x,y) = \delta_{x,y} + \frac{\alpha}{p} m^{(2)}_{SD}(x-1,y) + \frac{\beta}{p} m^{(2)}_{SD}(x,y-1) + \frac{\gamma}{p} m^{(2)}_{SD}(x-1,y-1)$$

with the boundary conditions $m^{(2)}_{SD}(x,0) = m^{(2)}_{SD}(0,y) = 0$, $\forall\, x, y \geq 0$, where

$$\delta_{x,y} = \frac{P_d}{p} + \frac{2\alpha}{p} m_{SD}(x-1,y) + \frac{2\gamma}{p} m_{SD}(x-1,y-1).$$

Lemma 4 gives a recursive expression for $\overline{{M_{SD}}^2}$. In case a closed-form formula is preferred, we can exploit Lemma 2 to get the closed-form formula shown in Corollary 3.

Corollary 3: Given a constant segment size $K = k$, $\overline{{M_{SD}}^2}$ for the DC protocol is

$$\overline{{M_{SD}}^2} = \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{u=0}^{\min(k-i,k-j)} \frac{(2k-i-j-u)!}{(k-i-u)!(k-j-u)!u!} \frac{\alpha^{k-i-u}\beta^{k-j-u}\gamma^u}{p^{2k-i-j-u}} \delta_{i,j}$$

where $\delta_{x,y}$ is (see equation at the bottom of the page).

Next we derive $\overline{N_S M_{SD}}$ by Wald's second equation.

Lemma 5: Given a constant segment size $K = k$, $\overline{N_S M_{SD}}$ for the DC protocol is

$$\overline{N_S M_{SD}} = \frac{1}{2P_d}\overline{{M_{SD}}^2} + \frac{P_d}{2}\overline{{N_S}^2} - \frac{(1-P_d)}{2}\overline{N_S}$$

where N_S can be computed by either Lemma 1 or Corollary 1; $\overline{{N_S}^2}$ can be computed by either Lemma 3 or Corollary 2; $\overline{{M_{SD}}^2}$ can be computed by either Lemma 4 or Corollary 3.

Proof: Because node S sends N_S coded blocks in the first stage and the reception probability over the direct link is P_d, the number of the coded blocks node D receives in the first stage is $M_{SD} = \sum_{i=1}^{N_S} X_i$ coded blocks. X_is are i.i.d. Bernoulli random variables, each with parameter $\overline{X} = P_d$.

Let Y_i be the centralized X_i, that is $Y_i = X_i - \overline{X} = X_i - P_d$. Using the Wald's second equation, we know

$$\mathbf{E}\left[\left(\sum_{i=1}^{N_S} Y_i\right)^2\right] = \mathbf{E}\left[Y_i^2\right]\mathbf{E}[N_S] = P_d(1-P_d)\overline{N_S} \quad (6)$$

where the last equality holds since the variance of a Bernoulli random variable with parameter P_d equals $P_d(1-P_d)$.

The left-hand side of (6) can be expanded as

$$\begin{aligned}\mathbf{E}\left[\left(\sum_{i=1}^{N_S} Y_i\right)^2\right] &= \mathbf{E}\left[\left(\sum_{i=1}^{N_S}(X_i - P_d)\right)^2\right] \\ &= \mathbf{E}\left[\left(\sum_{i=1}^{N_S} X_i - P_d N_S\right)^2\right] \\ &= \mathbf{E}\left[\left(\sum_{i=1}^{N_S} X_i\right)^2\right] - 2P_d\mathbf{E}\left[N_S\sum_{i=1}^{N_S} X_i\right] \\ &\quad + P_d^2\mathbf{E}\left[{N_S}^2\right] \\ &= \mathbf{E}\left[{M_{SD}}^2\right] - 2P_d\mathbf{E}[N_S M_{SD}] + P_d^2\mathbf{E}\left[{N_S}^2\right].\end{aligned}$$

Substituting the last line above into the left-hand side of (6) and doing simple arithmetic manipulation, we get

$$\mathbf{E}[N_S M_{SD}] = \frac{1}{2P_d}\left(\mathbf{E}\left[{M_{SD}}^2\right] + P_d^2\mathbf{E}\left[{N_S}^2\right] - P_d(1-P_d)\overline{N_S}\right)$$

And we have proven this lemma. ■

Next, $\overline{N_S N_R}$ is derived by the law of total expectation.

$$\delta_{x,y} = \frac{P_d}{p} + \frac{2(\alpha+\gamma)}{p^2}\left(\alpha\sum_{i=1}^{x-1}\sum_{j=1}^{y}\sum_{u=0}^{\min(x-1-i,y-j)} \frac{(x-1-i+y-j-u)!}{(x-1-i-u)!(y-j-u)!u!} \frac{\alpha^{x-1-i-u}\beta^{y-j-u}\gamma^u}{p^{x-1+y-i-j-u}} + \gamma\sum_{i=1}^{x-1}\sum_{j=1}^{y-1}\sum_{u=0}^{\min(x-1-i,y-1-j)} \frac{(x-1-i+y-1-j-u)!}{(x-1-i-u)!(y-1-j-u)!u!} \frac{\alpha^{x-1-i-u}\beta^{y-1-j-u}\gamma^u}{p^{x-1+y-1-i-j-u}}\right)$$

Lemma 6: Given a constant segment size $K = k$, $\overline{N_S N_R}$ for the DC protocol is

$$\overline{N_S N_R} = \frac{k}{P_c}\overline{N_S} - \frac{1}{P_c}\overline{N_S M_{SD}}$$

where $\overline{N_S}$ can be computed using either Lemma 1 or Corollary 1; and $\overline{N_S M_{SD}}$ can be computed using Lemma 5.

Proof: We derive $\overline{N_S N_R}$ by first calculating the conditional expectation given $N_S = n$ and $M_{SD} = m$ and then getting the unconditional expectation.

As explained in Theorem 2, given the value M_{SD}, $N_R \sim \text{NB}(P_c, k - M_{SD})$. Once M_{SD} is given, N_R is independent of N_S. So the conditional expectation of $N_S N_R$ is

$$\begin{aligned}\mathbf{E}[N_S N_R | N_S = n \text{ and } M_{SD} = m] &= \mathbf{E}[nN_R | M_{SD} = m] \\ &= n\mathbf{E}[N_R | M_{SD} = m] = n\frac{k-m}{P_c} = \frac{k}{P_c}n - \frac{1}{P_c}nm.\end{aligned}$$

Using the above result, the unconditional expectation is

$$\begin{aligned}\mathbf{E}[N_S N_R] &= \mathbf{E}\left[\mathbf{E}[N_S N_R | N_S, M_{SD}]\right] \\ &= \mathbf{E}\left[\frac{k}{P_c}N_S\right] - \mathbf{E}\left[\frac{1}{P_c}N_S M_{SD}\right] \\ &= \frac{k}{P_c}\overline{N_S} - \frac{1}{P_c}\overline{N_S M_{SD}}.\end{aligned}$$

So far, we have proven this lemma. ■

Next we derive $\overline{{N_R}^2}$ by the law of total expectation.

Lemma 7: Given a constant segment size $K = k$, $\overline{{N_R}^2}$ for the DC protocol is

$$\overline{{N_R}^2} = \frac{k(1-P_c)+k^2}{P_c^2} + \frac{(-1+P_i-2k)P_d}{P_c^2}\overline{N_S} + \frac{1}{P_c^2}\overline{{M_{SD}}^2}$$

where $\overline{N_S}$ is computed by either Lemma 1 or Corollary 1; $\overline{{M_{SD}}^2}$ is computed by either Lemma 4 or Corollary 3.

Proof: As explained in Theorem 2, $N_R \sim \text{NB}(P_c, k - M_{SD})$ given the value of M_{SD}. A negative random variable $X \sim \text{NB}(p, r)$ has a second moment of $(r^2 + r(1-p))/p^2$. So $\overline{{N_R}^2}$'s unconditional expectation can be computed

$$\begin{aligned}\mathbf{E}\left[{N_R}^2\right] &= \mathbf{E}\left[\mathbf{E}\left[{N_R}^2 | M_{SD}\right]\right] \\ &= \mathbf{E}\left[\frac{(k-M_{SD})^2 + (k-M_{SD})(1-P_c)}{P_c^2}\right] \\ &= \frac{k(1-P_c)+k^2}{P_c^2} + \frac{-1+P_i-2k}{P_c^2}\overline{M_{SD}} + \frac{1}{P_c^2}\overline{{M_{SD}}^2}.\end{aligned}$$

Since $\overline{M_{SD}} = P_d\overline{N_S}$ by (4), we have proven this lemma. ■

Because $\overline{N^2} = \overline{{N_S}^2} + \overline{{N_R}^2} + 2\overline{N_S N_R}$, $\overline{N^2}$ can be obtained by combining Lemma 3 (or Corollary 2), Lemma 7, and Lemma 6. The result is shown in Theorem 3.

Theorem 3: Given a constant segment size $K = k$, $\overline{N^2}$ for the DC protocol is

$$\begin{aligned}\overline{N^2} = &\frac{k(1-P_c)+k^2}{P_c^2} + \frac{(-1+P_i-2k)P_d}{P_c^2}\overline{N_S} \\ &+ \frac{P_d - P_c}{P_c^2 P_d}\overline{{M_{SD}}^2} + \frac{2k+1-P_d}{P_c}\overline{N_S} + \frac{P_c - P_d}{P_c}\overline{{N_S}^2}\end{aligned}$$

where $\overline{N_S}$ can be computed by either Lemma 1 or Corollary 1; $\overline{{N_S}^2}$ can be computed by either Lemma 3 or Corollary 2; $\overline{{M_{SD}}^2}$ can be computed by either Lemma 4 or Corollary 3.

The analysis in this section (particularly Theorem 2 and Theorem 3) gives the formulas of $\overline{N}$ and $\overline{N^2}$ for the DC protocol, provided that all segments have equal size. In the general case that segments are randomly sized, $\overline{N}$ and $\overline{N^2}$ can be obtained by using (2) and (3). After substituting the derived $\overline{N}$ and $\overline{N^2}$ into Theorem 1, we obtain the formulas of the saturation throughput, average segment delay, and average buffer occupancy for the DC protocol.

VI. Performance Evaluation

In this section, we evaluate the performance of our proposed protocol, the DC protocol, in terms of saturation throughput η and average segment delay T. There is no need to evaluate the average buffer occupancy L specially, because by Little's formula L is directly proportional to T.

In order to quantify the performance gain over C-ARQ protocols, we compare the performance of the DC protocol with that of the opportunistically forwarding (OF) protocol [27] which is a representative of stop-and-wait C-ARQ protocols without utilizing network coding. In the OF protocol, a relay node, if overhearing a packet, perceives packet loss with the help of the corresponding acknowledgement and retransmits the lost packet on a per-packet basis (in the next time slot allotted to the flow).

In addition, we compare the performance of DC with that of the listen-and-supersede (LS) protocol [27]. Assuming of perfect out-of-band control signaling, LS is optimal in terms of channel usage efficiency for the case without space-time cooperation. In the LS protocol, a relay node silently listens to the wireless channel until the relay and the destination together become able to retrieve the entire segment cooperatively. Then it takes over from the source node since an appropriately-chosen relay has better connectivity to the destination than the source has.

Same as in [27], the throughput gain for a RANC ARQ protocol, denoted by G_η, is defined as the saturation throughput of that protocol divided by OF's saturation throughput:

$$G_\eta(\cdot) = \frac{\eta(\cdot)}{\eta(\text{OF})}$$

where the success probability of the source receiving per-packet ACKs for the OF protocol is set to $P_{\text{D,ACK}} = P_d + P_i - P_d P_i$ in the simulations, in order to reflect the enhancement that the

relay can forward per-packet ACKs to the source. Similarly, the delay gain, denoted by G_T, is defined as

$$G_T(\cdot) = \frac{T(\text{OF})}{T(\cdot)}.$$

Both G_η and G_T are the larger the better. DC is expected to have $G_\eta > 1$ and $G_T > 1$, for two main reasons. First, space-time cooperation decreases the failure probability of the destination receiving blocks. Denote the factor of such a decrease by g. That is, $g = (1 - P_{RD})/(1 - P_c)$, where the parameter $g \geq 1$ represents the gain of space-time cooperation over non-cooperation. Second, the use of network coding minimizes unneeded retransmissions caused by ACK loss. Performance gain caused by the former reason is quantitatively studied in Section VI-A. The overall performance gain is assessed through a case study in Section VI-B.

A. Performance Gain Due to Space-Time Cooperation

Space-time coding is a technique often used to combat fading effects in wireless channels. Denote the outage probabilities of the direct link, the interconnect link, the relay-to-destination (RD) link, and the cooperative link by $\epsilon_d = 1 - P_d$, $\epsilon_i = 1 - P_i$, $\epsilon_{RD} = 1 - P_{RD}$, and $\epsilon_c = 1 - P_c$, respectively. To quantify these outage probabilities, this subsection uses the same model as that used in [33]. Each channel between any two nodes is modeled as Rayleigh block fading. Denote the channel gain of the direct link, the interconnect link, and the RD link by h_d, h_i, and h_{RD}, respectively, where the random variables $|h_d|^2$, $|h_i|^2$, and $|h_{RD}|^2$ follow exponential distributions with means σ_d^2, σ_i^2, and σ_{RD}^2, respectively. Assume that the individual transmit power is E_s and noise is AWGN with zero mean and variance N_0. Denote the individual transmit SNR by $\phi = E_s/N_0$. Given the rate r, the outage probabilities can be obtained by tailoring to our setting the information-theoretic analysis in [33], in which outage occurs if mutual information is smaller than r

$$\epsilon_d = 1 - \exp\left(-\frac{2^r - 1}{\sigma_d^2 \phi}\right)$$

$$\epsilon_i = 1 - \exp\left(-\frac{2^r - 1}{\sigma_i^2 \phi}\right)$$

$$\epsilon_{RD} = 1 - \exp\left(-\frac{2^r - 1}{\sigma_{RD}^2 \phi}\right)$$

$$\epsilon_c = \begin{cases} 1 - \left(1 + \frac{2^r-1}{\sigma_d^2\phi}\right)\exp\left(-\frac{2^r-1}{\sigma_d^2\phi}\right), & \text{if } \sigma_d^2 = \sigma_{RD}^2 \\ 1 - \frac{\sigma_d^2}{\sigma_d^2 - \sigma_{RD}^2}\exp\left(-\frac{2^r-1}{\sigma_d^2\phi}\right) & \\ \quad - \frac{\sigma_{RD}^2}{\sigma_{RD}^2 - \sigma_d^2}\exp\left(-\frac{2^r-1}{\sigma_{RD}^2\phi}\right), & \text{otherwise.} \end{cases}$$

To learn how much the performance gain originates from space-time cooperation, the simulation setup of this subsection intentionally chooses a relay node that cannot provide any benefit by itself. To this end, one assumption is introduced in this setup: $\sigma_d^2 = \sigma_i^2 = \sigma_{RD}^2$. This implies that $\epsilon_d = \epsilon_i = \epsilon_{RD} = \epsilon_1$ and $\epsilon_c = \epsilon_2$, where ϵ_1 and ϵ_2 are the corresponding outage probability values.

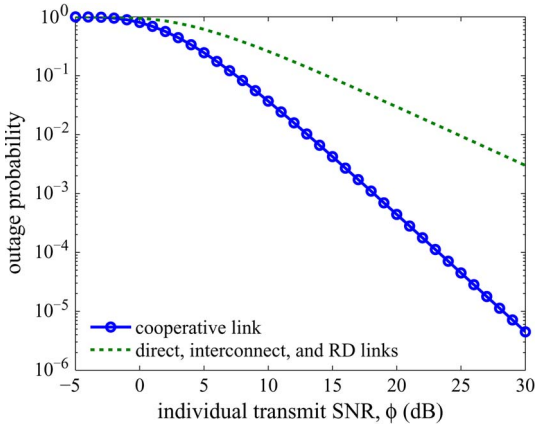

Fig. 6. Outage probability versus individual transmit SNR of the cooperative link (ϵ_c) and of the links without space-time cooperation ($\epsilon_d, \epsilon_i, \epsilon_{RD}$). $r = 2$ bit/s/Hz and $\sigma_d^2 = \sigma_i^2 = \sigma_{RD}^2 = 1$.

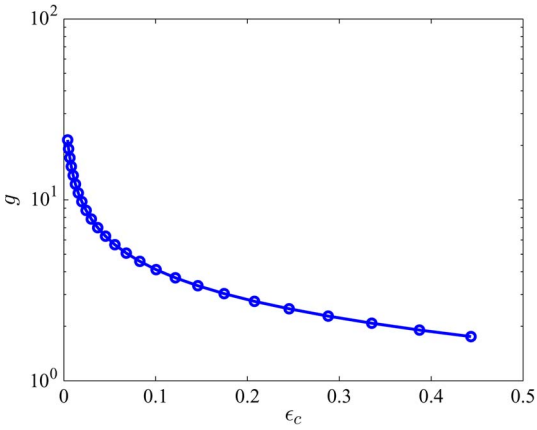

Fig. 7. Outage probability gain g versus ϵ_c. $r = 2$ bit/s/Hz and $\sigma_d^2 = \sigma_i^2 = \sigma_{RD}^2 = 1$.

After using the aforementioned formulas to calculate the outage probabilities, Fig. 6 shows the outage probability performance versus individual transmit SNR of the non-cooperative links (i.e., the direct, interconnect, and RD links) and the cooperative link, with $r = 2$ bit/s/Hz. r is set to 2 bit/s/Hz for consistency with the case study in Section VI-B. The values of other simulation parameters are taken from [33]. It is observed from Fig. 6 that compared with the non-cooperative links, the outage probability of the cooperative link is significantly smaller. In other words, with the same outage probability requirement, space-time cooperation enlarges coverage area or reduces individual transmit power drastically. The outage probability gain $g = (\epsilon_{RD}/\epsilon_c)$ versus ϵ_c is plotted in Fig. 7. In this setup, space-time cooperation makes the gain g in the range between 1.75 times and 21.4 times.

Fig. 8 plots the throughput gain caused by space-time cooperation versus ϵ_c. In terms of saturation throughput, it can be observed that DC outperforms OF significantly. DC performs slightly better than LS; more precisely, in this setup in which an inappropriate relay is chosen, DC outperforms LS in G_η by a factor of 2% on average. As shown later in Section VI-B, the improvement of DC over LS is more significant in the case of having an appropriately-chosen relay. The throughput gains of

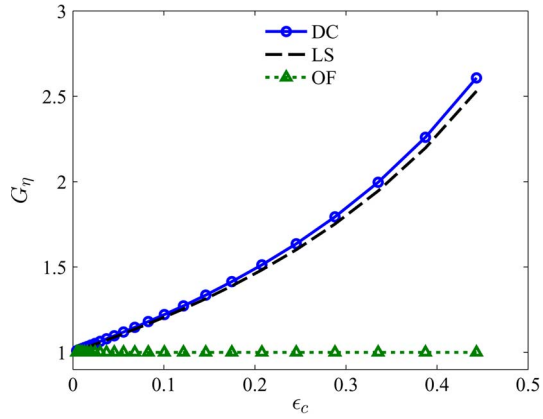

Fig. 8. Throughput gain versus ϵ_c. This figure shows the relative throughput of DC and LS, compared to OF's throughput which is normalized to one.

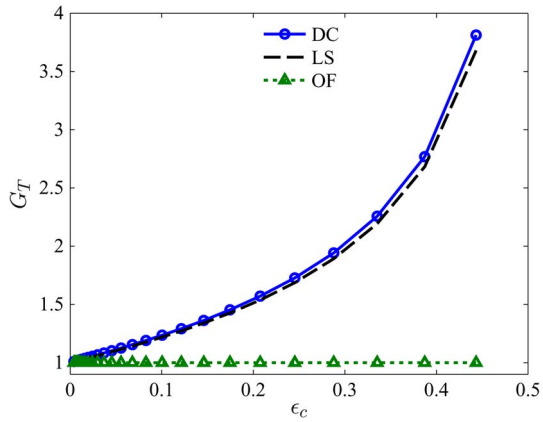

Fig. 9. Delay gain versus ϵ_c. This figure shows the relative performance of DC and LS in terms of the inverse of delay, compared to OF.

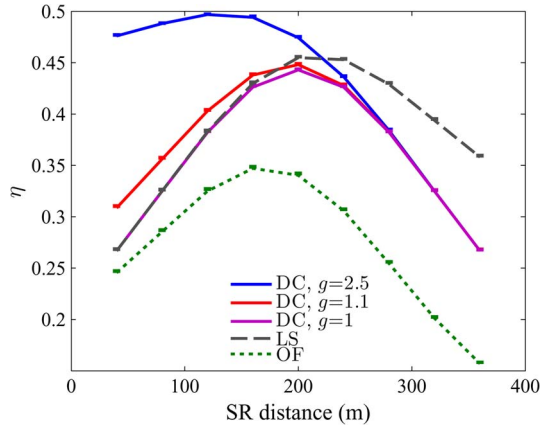

Fig. 10. Saturation throughput versus source-relay distance in the case study. Error bars depict 95% confidence intervals.

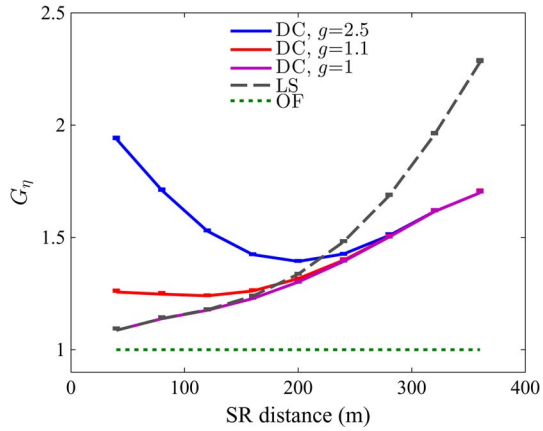

Fig. 11. Throughput gain versus source-relay distance in the case study. Error bars depict 95% confidence intervals.

DC and LS increase as ϵ_c increases. Observing Fig. 9 which plots the delay gain caused by space-time cooperation versus ϵ_c, the same qualitative conclusions can be drawn in terms of segment delay and delay gain.

B. Case Study

In this subsection, a case study is presented to evaluate the performance gains when using DC with segment size $K = 100$ in a WiMAX system[2] Suppose there are a base station (source), a subscriber station (destination), and a relay station (relay) in-between the base station and the subscriber station. The distance between source and destination is 400 meters.

The channel parameters of the WiMAX system are set according to pages 75–76 in [34], which take path loss, fading, and additive white Gaussian noise into account. Assume the 16-QAM modulation and a convolution code of rate 1/2 is employed. In such a situation, a received signal-to-noise ratio (SNR) of 14.7 dB is required. The link reception probability over a wireless link from node i to node j (before the use of space-time cooperation) are computed as $P_{ij} = Q(-12.2 + 5\log_{10} d_{ij})$, where d_{ij} is the link length and $Q(\cdot)$ is the Gaussian Q-function.

The performances of DC, LS, and OF in terms of saturation throughput and segment delay are extensively evaluated in this case study, by theoretical formulas and through simulations. In all figures in this subsection, theoretical results are drawn by lines in different colors; while simulation results are averaged over 10 000 segments and are depicted by error bars with 95% confidence intervals. Comparing theoretical results and simulation counterparts helps to ensure the validity of our mathematical analysis.

Figs. 10 and 12 plot absolute results for $g = 1, 1.1, 2.5$[3]; Figs. 11 and 13 plot relative results. As observed in Figs. 10 and 12, saturation throughput and segment delay for all of DC, LS, and OF get substantially improved when the relay is located appropriately. For LS and OF, the best location for the relay node is often around the center of the source and the

[2]We use WiMAX's parameters to conduct this case study, but the study can apply to LTE-Advanced (and other systems) by replacing the WiMAX's parameters with the parameters in LTE-Advanced systems.

[3]Observing Fig. 7, the setting of $g = 1, 1.1, 2.5$ is quite conservative, since g is often larger. (Indeed, $g = 1$ represents the extreme case where the use of space-time cooperation has no benefit.) These conservative values are taken from [4] in which these values are used as the type II over type I probability gain to successfully decode data frames received at the destination.

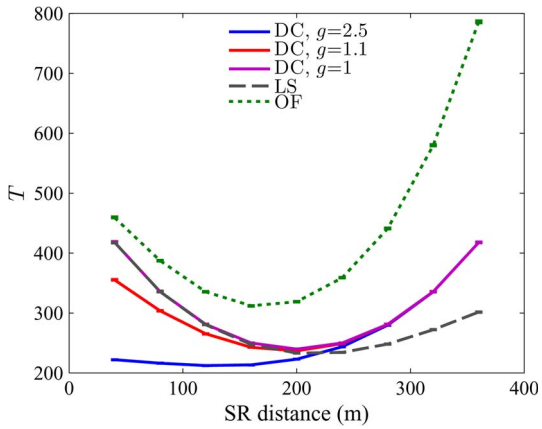

Fig. 12. Segment delay versus source-relay distance in the case study. Error bars depict 95% confidence intervals.

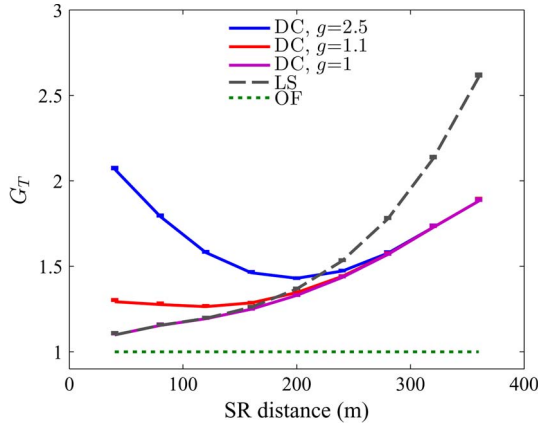

Fig. 13. Delay gain versus source-relay distance in the case study. Error bars depict 95% confidence intervals.

destination. For DC, relay's best position is often closer to the source, particularly when g is large, because it makes the relay to cooperate with the source sooner.

For each protocol, locating the relay at the best location achieves maximum saturation throughput and minimum segment delay. When each protocol locates the relay at its own best position, it is observed in Figs. 10 and 12 that each RANC ARQ protocol surpasses the C-ARQ protocol (i.e., OF) significantly in terms of both maximum saturation throughput and minimum segment delay. Compared with OF, the RANC ARQ protocols increase maximum saturation throughput by a factor between 28% and 43% and decrease minimum segment delay by 23% to 32%. Both the best saturation throughput value and the best segment delay value are achieved by DC with $g = 2.5$. When $g = 2.5$, DC outperforms LS in maximum saturation throughput by 9% and in minimum segment delay by 9%. Considering that LS is optimal in the single-relay case without space-time cooperation and DC incurs simpler control signaling and smaller communication overhead than LS does, DC performs very well.

Regardless where the relay is located, DC surpasses OF consistently and significantly in terms of both saturation throughput and segment delay. This can be easily observed from Figs. 11 and 13, since the gains obviously are greater than one.

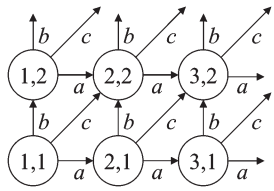

Fig. 14. This figure illustrates the recursive relationship about $f(3, 2)$.

While DC outperforms OF significantly, DC and LS appear almost on a par. DC excels LS in situations when the relay is at an appropriate location and a high-g space-time coding is adopted. LS outperforms DC when the source-destination channel has a low SNR and the relay-destination channel has a high SNR. In other situations, DC and LS performs comparably well. Considering that DC with an appropriate relay node can achieve a performance better than or comparable to LS (whose channel usage efficiency is optimal if space-time cooperation is not allowed) without the need of letting a relay node contribute its own allotted bandwidth, DC really performs very well.

VII. Conclusion

In this paper, the framework of the relay-assisted network-coding (RANC) ARQ protocols for one-way relaying systems has been addressed. In particular, the first RANC ARQ protocol that takes advantage of space-time cooperation has been proposed.

We have derived the saturation throughput, average segment delay, and average buffer occupancy for the proposed RANC ARQ protocol, DC. In order to quantify the performance gain of DC over C-ARQ protocols, we have compared the performance of DC with that of a representative of C-ARQ protocol and demonstrated significant performance gains. We have also compared the performance of DC with that of LS, which is the optimal RANC ARQ protocol in terms of channel usage efficiency if space-time cooperation is not allowed. Extensive analysis and simulation results show that DC performs very well in both saturation throughput and segment delay.

Appendix A
Proof of Lemma 2

Step 1—(Superposition principle for the recursion): For exposition purpose, we illustrate the recursive relationship of $f(x, y)$ in a plane (see Fig. 14 as an example); we call the constant $\delta_{x,y}$ as the *stimulus* at vertex (x, y) and call $f(x, y)$ as the *response* at vertex (x, y). Because the recursive expression $f(x, y) = \delta_{x,y} + af(x - 1, y) + bf(x, y - 1) + cf(x - 1, y - 1)$ is linear, the response at any vertex can be computed by the *superposition principle*: That is, the (net) response at any vertex caused by multiple stimuli is the sum of the responses which would have been caused by each stimulus acting alone. Therefore, $f(x, y)$ can be expressed as $f(x, y) = \sum_i \sum_j c_{i,j}^{x,y} \delta_{i,j}$, where $c_{i,j}^{x,y}$ is the amplification factor of the stimulus $\delta_{i,j}$ on the response at vertex (x, y).

Take $c_{1,1}^{3,2}$ as an example, which corresponds to the response at vertex (3, 2) caused by the stimulus at vertex (1, 1) acting

alone. The value of $c_{1,1}^{3,2}$ can be obtained by expanding the recursive expression of $f(x,y)$ step by step and substituting $\delta_{1,1}$ by one and all other $\delta_{x,y}$s by zeros

$$\begin{aligned} c_{1,1}^{3,2} &= f(3,2) = af(2,2) + bf(3,1) + cf(2,1) \\ &= a\left(af(1,2) + bf(2,1) + c\cdot 1\right) + b\left(af(2,1)\right) + c(a\cdot 1) \\ &= a\left(a(b\cdot 1) + b(a\cdot 1) + c\cdot 1\right) + b\left(a(a\cdot 1)\right) + c(a\cdot 1) \\ &= (3a^2b + 2ac)\cdot 1. \end{aligned} \tag{7}$$

Observing the recursive relationship, we notice two properties. First, the response at vertex (x,y) is caused by only the stimuli at vertices lower-left to vertex (x,y). That is, $f(x,y)$ depends only on $\delta_{i,j}$s, $1 \le i \le x$ and $1 \le j \le y$. In other words, a stimulus causes responses at the vertices upper-right to it (but has no impact on the vertices that are strictly below or strictly left to it). This implies that $f(x,y) = \sum_{i=1}^{x}\sum_{j=1}^{y} c_{i,j}^{x,y}\delta_{i,j}$.

The second property is that (due to symmetry of the recursive relationship) shifting the locations of a stimulus and a target vertex with the same displacement does not affect the response at the target vertex caused by the stimulus. That is, $c_{i,j}^{x,y} = c_{i+m,j+n}^{x+m,y+n}$ where m and n are integers. In other words, all $c_{i,j}^{x,y}$s can be computed by a common function with two arguments, $\Delta_x = x-i$ and $\Delta_y = y-j$. Denote this binary function by $g(\Delta_x,\Delta_y)$. Since $c_{i,j}^{x,y} = g(x-i,y-j)$, $f(x,y)$ can be expressed as

$$f(x,y) = \sum_{i=1}^{x}\sum_{j=1}^{y} g(x-i,y-j)\delta_{i,j}.$$

To prove Lemma 2, what is left is to prove that for $\Delta_x \ge 0$ and $\Delta_y \ge 0$, the amplification factor

$$g(\Delta_x,\Delta_y) = \sum_{u=0}^{\min(\Delta_x,\Delta_y)} \frac{(\Delta_x+\Delta_y-u)!}{(\Delta_x-u)!(\Delta_y-u)!u!} a^{\Delta_x-u}b^{\Delta_y-u}c^u. \tag{8}$$

Note that in this step, a single stimulus is assumed to act alone at the lower-left corner (i.e., one end) and we pay attention on the response caused by the stimulus at the target vertex located at the upper-right corner (i.e., the other end).

Recall that $c_{1,1}^{3,2} = g(2,1)$ is computed in (7) by expanding the recursive expression step by step. (Keep Fig. 14 in your mind.) Expanding the recursive expression step by step is equivalent to traversing from one end to the other end along all possible paths connecting the two ends; in (7), the two ends are vertices (1, 1) and (3, 2). Each time of passing through a horizonal edge, a factor of a is multiplied; while passing through a vertical edge and passing through a diagonal edge (i.e., an edge from lower left to upper right) amplify the stimulus by a factor of b and c, respectively. From this perspective, $c_{1,1}^{3,2} = 3a^2b^1 + 2a^1c^1$ is because from vertex (1, 1) to vertex (3, 2), there are three paths that pass through two horizontal edges and one vertical edge [as shown in Fig. 15(a)] and there are two paths that pass through one horizontal edges and one diagonal edge [as shown in Fig. 15(b)].

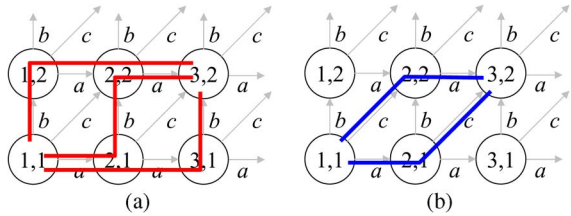

Fig. 15. (a) The class of paths connecting vertices (1, 1) and (3, 2) that do not pass through any diagonal edge. (b) The class of paths passing through exactly one diagonal edge.

What follows extends the idea of computing the special case $g(2,1)$ to the general case $g(\Delta_x,\Delta_y)$ where Δ_x and Δ_y can be any non-negative integer. Same as the special case, $g(\Delta_x,\Delta_y)$ is obtained by expanding the recursive expression step by step (and substituting one stimulus by one and all other stimuli by zeros), which is equivalent to traversing along all possible paths that connect two ends with x-displacement equal to Δ_x and y-displacement equal to Δ_y.

To facilitate derivation of $g(\Delta_x,\Delta_y)$, the paths connecting the two ends are categorized into classes, depending on the number of diagonal edges passed through. Because the number of diagonal edges along any path can exceed neither Δ_x nor Δ_y, its range is $0,1,\ldots,\min(\Delta_x,\Delta_y)$. We categorize the paths connecting the two ends into $\min(\Delta_x,\Delta_y)+1$ classes.

Let us focus on one specific class of paths, say the paths passing through exactly u diagonal edges; our goal is to derive the sum of the responses at the target vertex that propagate along this class of paths. We notice that each of the paths in this class pass through exactly $\Delta_x - u$ horizontal edges and $\Delta_y - u$ vertical edges. (This is true because each path must reach the other end in both the x-direction and the y-direction.) As a result, each path in this class amplify the stimulus by a factor of $a^{\Delta_x-u}b^{\Delta_y-u}c^u$. Each path in this class consists of $\Delta_x+\Delta_y-u$ edges, including Δ_x-u horizontal edges, Δ_y-u vertical edges, and u diagonal edges. Therefore, this class contains $(\Delta_x+\Delta_y-u)!/((\Delta_x-u)!(\Delta_y-u)!u!)$ paths in total. As a result, the class of paths with exactly u diagonal edges aggregately amplify the stimulus by a factor of $(\Delta_x+\Delta_y-u)!/((\Delta_x-u)!(\Delta_y-u)!u!)a^{\Delta_x-u}b^{\Delta_y-u}c^u$.

So far, we have derived the amplification factor over a single class of paths, the amplification factor over all classes of paths, $g(\Delta_x,\Delta_y)$, can be computed by summing up the amplification factor over each class of paths. Since the sum is equal to (8), we have proven this lemma. ■

APPENDIX B
PROOF OF LEMMA 3

To derive $\overline{N_S{}^2}$, we use a recursive technique similar to that used in Lemma 1. The major difference is that instead of splitting a segment transmission into the first transmission and the remaining transmissions, now the observation point becomes the moment when either relay or destination (or both) receive a block in the first stage. Consider the number of transmissions the source takes until the observation point, which is a random variable denoted by Z. The probability mass function of Z is $\Pr(Z{=}n) = (1-p)^{n-1}p$, where n is a positive integer.

At the observation point, there are three possible outcomes:

1) D (but not node R) receives the block with probability α/p. In this case, the source will take $N_S(x-1,y)$ more transmissions after the observation point.
2) R (but not node D) receives the block with probability β/p. In this case, the source will take $N_S(x,y-1)$ more transmissions after the observation point.
3) D and R receive the block with probability γ/p. In this case, the source will take $N_S(x-1,y-1)$ more transmissions after the observation point.

Given $Z=n$, the number of transmissions node S takes until the observation point is equal to n, for all types of outcomes. By the law of total expectation, we get the recursive expression of the conditional expectation

$$\mathbf{E}\left[{N_S}^2(x,y)|Z=n\right]=\frac{\alpha}{p}\overline{(N_S(x-1,y)+n)^2}+\frac{\beta}{p}\overline{(N_S(x,y-1)+n)^2}+\frac{\gamma}{p}\overline{(N_S(x-1,y-1)+n)^2}.$$

With the conditional expectation shown above, we can get the unconditional expectation by unconditioning the conditional expectation over all possible values of Z; that is $\mathbf{E}[{N_S}^2(x,y)]=\mathbf{E}[\mathbf{E}[{N_S}^2(x,y)|Z]]$. And we get

$$\overline{{N_S}^2(x,y)}=\sum_{n=1}^{\infty}(1-p)^{n-1}p\left(\frac{\alpha}{p}\overline{(N_S(x-1,y)+n)^2}+\frac{\beta}{p}\overline{(N_S(x,y-1)+n)^2}+\frac{\gamma}{p}\overline{(N_S(x-1,y-1)+n)^2}\right).$$

After some arithmetic manipulations, we get

$$\overline{{N_S}^2(x,y)}=\left(\frac{2-p}{p^2}+\frac{2\alpha}{p^2}\overline{N_S(x-1,y)}+\frac{2\beta}{p^2}\overline{N_S(x,y-1)}+\frac{2\gamma}{p^2}\overline{N_S(x-1,y-1)}\right)+\frac{\alpha}{p}\overline{{N_S}^2(x-1,y)}+\frac{\beta}{p}\overline{{N_S}^2(x,y-1)}+\frac{\gamma}{p}\overline{{N_S}^2(x-1,y-1)}.$$

Comparing the definitions of $n_S^{(2)}(x,y)$ and $n_S(x,y)$ with the above equation, it is obvious that $\overline{{N_S}^2}=\overline{{N_S}^2(k,k)}=n_S^{(2)}(k,k)$. So far we have proven this lemma. ■

Appendix C
Proof of Lemma 4

Step 1—(Derivation of $\overline{M_{SD}}$): $\overline{M_{SD}}$ can be derived in a way very similar to $\overline{N_S}$'s derivation in Lemma 1. Consider a segment transmission. Define $M_{SD}(x,y)$ as the number of blocks sent by the source in the first stage that successfully reach the destination, until the moment when either node D has received x blocks or node R has received y blocks for the segment. Same as $\overline{N_S}$'s derivation in Lemma 1, a segment transmission is split into two parts—the first transmission and the remaining transmissions. There is only one major difference between the derivations of $\overline{M_{SD}(x,y)}$ and $\overline{N_S(x,y)}$: The first transmission sent by the source node always contributes one to $N_S(x,y)$, but it sometimes contributes zero to $M_{SD}(x,y)$. The first transmission contributes *one* to $M_{SD}(x,y)$ only when node D receives the first transmission successfully, corresponding to the second term and the last term at the right-hand side of (9). The first transmission contributes *zero* to $M_{SD}(x,y)$ when the first transmission fails to reach node D, corresponding to the first term and the third term of (9). The rest of the derivation of $\overline{M_{SD}(x,y)}$ is the same as the derivation of $\overline{N_S(x,y)}$. Therefore

$$\overline{M_{SD}(x,y)}=(1-p)\overline{M_{SD}(x,y)+0}+\alpha\overline{M_{SD}(x-1,y)+1}+\beta\overline{M_{SD}(x,y-1)+0}+\gamma\overline{M_{SD}(x-1,y-1)+1}. \quad (9)$$

After simple arithmetic manipulations, we get the recursive expression of $\overline{M_{SD}(x,y)}$

$$\overline{M_{SD}(x,y)}=\frac{\alpha+\gamma}{p}+\frac{\alpha}{p}\overline{M_{SD}(x-1,y)}+\frac{\beta}{p}\overline{M_{SD}(x,y-1)}+\frac{\gamma}{p}\overline{M_{SD}(x-1,y-1)}.$$

Comparing the above formula with the definition of $m_{SD}(x,y)$, it is trivial that $\overline{M_{SD}(x,y)}=m_{SD}(x,y)$. Because $\overline{M_{SD}}=\overline{M_{SD}(k,k)}$ by definition, $\overline{M_{SD}}=m_{SD}(k,k)$.

Step 2—(Derivation of $\overline{{M_{SD}}^2}$): Similar to Step 1, $\overline{{M_{SD}}^2(x,y)}$ can be derived as follows:

$$\overline{{M_{SD}}^2(x,y)}=(1-p)\overline{(M_{SD}(x,y)+0)^2}+\alpha\overline{(M_{SD}(x-1,y)+1)^2}+\beta\overline{(M_{SD}(x,y-1)+0)^2}+\gamma\overline{(M_{SD}(x-1,y-1)+1)^2}.$$

After simple arithmetic manipulations, we get the recursive expression of $\overline{{M_{SD}}^2(x,y)}$

$$\overline{{M_{SD}}^2(x,y)}=\left(\frac{P_d}{p}+\frac{2\alpha}{p}\overline{M_{SD}(x-1,y)}+\frac{2\gamma}{p}\overline{M_{SD}(x-1,y-1)}\right)+\frac{\alpha}{p}\overline{{M_{SD}}^2(x-1,y)}+\frac{\beta}{p}\overline{{M_{SD}}^2(x,y-1)}+\frac{\gamma}{p}\overline{{M_{SD}}^2(x-1,y-1)}.$$

Comparing the above formula with $m_{SD}^{(2)}(x,y)$'s definition and using the fact that $\overline{M_{SD}(x,y)}=m_{SD}(x,y)$, it is trivial that

$$\overline{{M_{SD}}^2(x,y)}=m_{SD}^{(2)}(x,y)$$

for $x>0$ and $y>0$. So $\overline{{M_{SD}}^2}=\overline{{M_{SD}}^2(k,k)}=m_{SD}^{(2)}(k,k)$.

So far we have proven this lemma. ■

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers and the editor for their many helpful suggestions.

REFERENCES

[1] J. Jin, B. Li, and T. Kong, "Is random network coding helpful in WiMAX?" in *Proc. IEEE INFOCOM*, Phoenix, AZ, USA, Apr. 2008, pp. 191–195.

[2] Z. Li, Q. Luo, and W. Featherstone, "N-in-1 retransmission with network coding," *IEEE Trans. Wireless Commun.*, vol. 9, no. 9, pp. 2689–2694, Sep. 2010.

[3] M.-H. Lu, P. Steenkiste, and T. Chen, "Design, implementation and evaluation of an efficient opportunistic retransmission protocol," in *Proc. ACM MobiCom*, Beijing, China, Sep. 2009, pp. 73–94.

[4] I. Cerutti, A. Fumagalli, and P. Gupta, "Delay models of single-source single-relay cooperative ARQ protocols in slotted radio networks with Poisson frame arrivals," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 371–382, Apr. 2008.

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

[6] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[7] S. Bhadra, S. Shakkottai, and P. Gupta, "Min-cost selfish multicast with network coding," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 5077–5087, Nov. 2006.

[8] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, USA, Apr. 2008, pp. 196–200.

[9] X.-B. Liang, "Matrix games in the multicast networks: Maximum information flows with network switching," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2433–2466, Jun. 2006.

[10] B. Shrader and T. C. Royster, "Cooperative multicast strategies under heterogeneous link loss rates," in *Proc. IEEE GLOBECOM*, Houston, TX, USA, Dec. 2011, pp. 1–5.

[11] Q.-T. Vien, L.-N. Tran, and H. X. Nguyen, "Network coding-based ARQ retransmission strategies for two-way wireless relay networks," in *Proc. Int. Conf. SoftCOM Netw.*, Split, Croatia, Sep. 2010, pp. 180–184.

[12] S. Fu, K. Lu, Y. Qian, and M. Varanasi, "Cooperative network coding for wireless ad-hoc networks," in *Proc. IEEE GLOBECOM*, Washington, DC, USA, Nov. 2007, pp. 812–816.

[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. ACM SIGCOMM*, Pisa, Italy, Sep. 2006, pp. 243–254.

[14] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[15] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[16] T. K. Dikaliotis, A. Dimakis, T. Ho, and M. Effros, "On the delay advantage of coding in packet erasure networks," in *Proc. IEEE ITW*, Dublin, Ireland, Aug./Sep. 2010, pp. 1–5.

[17] F. Sun, "Two-layer coding rate optimization in relay-aided systems," in *Proc. IEEE VTC*, San Francisco, CA, USA, Sep. 2011, pp. 1–5.

[18] E. Zimmermann, P. Herhold, and G. Fettweis, "The impact of cooperation on diversity-exploiting protocols," in *Proc. IEEE VTC*, Milan, Italy, May 2004, pp. 410–414.

[19] E. Zimmermann, P. Herhold, and G. Fettweis, "On the performance of cooperative relaying protocols in wireless networks," *Eur. Trans. Telecommun.*, vol. 16, no. 1, pp. 5–16, Jan./Feb. 2005.

[20] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[21] B. Rong and A. Ephremides, "Cooperation above the physical layer: The case of a simple network," in *Proc. IEEE ISIT*, Seoul, Korea, Jun. 2009, pp. 1789–1793.

[22] A. K. Sadek, K. J. R. Liu, and A. Ephremides, "Cognitive multiple access via cooperation: Protocol design and performance analysis," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3677–3696, Oct. 2007.

[23] X. Wang and C. Yang, "A MAC protocol supporting cooperative diversity for distributed wireless *ad hoc* networks," in *Proc. IEEE PIMRC*, Berlin, Germany, Sep. 2005, pp. 1396–1400.

[24] J. Alonso-Zarate, E. Kartsakli, C. Verikoukis, and L. Alonso, "Persistent RCSMA: A MAC protocol for a distributed cooperative ARQ scheme in wireless networks," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, p. 817 401, Apr. 2008.

[25] S. S. N, C.-T. Chou, and M. Ghosh, "Cooperative communication MAC (CMAC)—A new MAC protocol for next generation wireless LANs," in *Proc. Int. Conf. Wireless Netw., Commun. Mobile Comput.*, Briarcliff Manor, NY, USA, Jun. 2005, pp. 1–6.

[26] N. Agarwal, N. Agarwal, L. N. Kannan, M. Tacca, and A. Fumagalli, "IEEE 802.11b cooperative protocols: A performance study," in *Proc. IFIP/TX6 Netw.*, Atlanta, GA, USA, May 2007, pp. 415–426.

[27] J.-C. Kao and F.-W. Chen, "On RANC ARQ for wireless relay networks: From the transmission perspective," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2962–2976, Jun. 2013.

[28] J.-C. Kao, F.-W. Chen, M.-H. Yang, and T.-L. Wang, "Coverage enhancement to IEEE 802.11p using work-based opportunistic relay-assisted network-coding ARQ," in *Proc. IEEE Asia Pac. Wireless Commun. Symp.*, Singapore, Aug. 2011, pp. 1–5.

[29] M. Janani, A. Hedayat, T. E. Hunter, and A. Nosratinia, "Coded cooperation in wireless communications: Space-time transmission and iterative decoding," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 362–371, Feb. 2004.

[30] H. Rahul, H. Hassanieh, and D. Katabi, "Sourcesync: A distributed wireless architecture for exploiting sender diversity," in *Proc. ACM SIGCOMM*, New Delhi, India, Aug./Sep. 2010, pp. 171–182.

[31] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained channel access in wireless LAN," in *Proc. ACM SIGCOMM*, New Delhi, India, Aug./Sep. 2010, pp. 147–158.

[32] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "Achieving high data rates in a distributed MIMO system," in *Proc. ACM MobiCom*, Istanbul, Turkey, Aug. 2012, pp. 41–52.

[33] Y. Zou, Y.-D. Yao, and B. Zheng, "Opportunistic distributed space-time coding for decode-and-forward cooperation systems," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1766–1781, Apr. 2012.

[34] J. G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*. Upper Saddle River, NJ, USA: Prentice-Hall, 2007.

Jung-Chun Kao (S'04–M'09) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1999, the M.S. degree in electrical engineering from University of Southern California, Los Angeles, CA, USA, in 2003, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2008. He joined the faculty of the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, in August 2008. His research interests include wireless ad-hoc and sensor networks, wireless access networks, cooperative communications, network coding, cyber-physical systems, and design and analysis of communication/network protocols.