Steady-State Throughput Analysis of Network Coding Nodes Employing Stop-and-Wait Automatic Repeat Request

Yang Qin, *Student Member, IEEE*, and Lie-Liang Yang, *Senior Member, IEEE*

***Abstract*—This paper analyzes the steady-state throughput of network coding nodes when data is transmitted based on the stop-and-wait automatic repeat request (SW-ARQ) scheme. The state transition of network coding nodes employing SW-ARQ is analyzed, which shows that the operations of network coding nodes can be modeled by a finite state machine. Therefore, the throughput expressions of network coding nodes can be derived based on the properties of finite state machines. Furthermore, the throughput performance of network coding nodes is investigated either by simulations or by evaluation of the expressions obtained. It can be shown that the simulation results converge closely to the numerical results and justify the effectiveness of our analytical expressions obtained.**

***Index Terms*—Automatic repeat request (ARQ), network coding, network coding node, state machine, throughput.**

I. Introduction

NETWORK coding has received a lot of attention, since its invention by Ahlswede *et al.* [1]. Network coding deals with the problems of coding over packet networks, and it has been recognized that the network-coding-assisted routing has the potential to outperform the conventional routing [1], [2]. Performance of communication systems with network coding has been investigated, when assuming that packets are conveyed error-free in the networks [3]. However, in practice, transmission errors always occur, and error-detection or error-correction techniques are often required in order to ensure reliable communications [2], [4]. Therefore, in this paper we motivate to study the steady-state throughput of the network coding nodes, where communications between two nodes are based on the stop-and-wait automatic repeat request (SW-ARQ) data transmission scheme [5]–[7].

Here, the throughput of a network coding node is defined as the rate that coded packets are correctly conveyed by the network coding node to its following node(s). Note that we focus our attention on the steady-state throughput of network coding node since, when communicating over unreliable channels, the achievable throughput of a communication network with network coding nodes is mainly constrained by the throughput of the network coding nodes, as shown in the butterfly network that will be detailed in Section III-A.

More specifically, in this paper, the steady-state throughput of the network coding node is analyzed, where the coding node is associated with multiple-input–single-output (MISO) links connecting the coding node with multiple source nodes and one sink node [8], [9]. Therefore, we refer to it as the MISO network coding node. At our MISO network coding node, we assume that there is a buffer storing at most one packet for each of the incoming and outgoing links, which are supported by the SW-ARQ data transmission scheme. Specifically, we first analyze in detail the operations, properties, and steady-state throughput of a two-input–single-output (2ISO) network coding node supported by the SW-ARQ transmission scheme. It can be shown that the operations of the network coding nodes can be modeled by a state machine working in the principles of discrete-time Markov chain, which is convenient for analyzing the steady-state throughput. Then, the analytical approaches for deriving the steady-state throughput of the 2ISO network coding node is extended to the MISO network coding nodes. Finally, the throughput performance of network coding nodes with different numbers of input links is investigated by both simulation and numerical approaches. Our performance results demonstrate that the simulation results converge well to the numerical results, which justifies the effectiveness of our analytical expressions derived.

It is worthy of mentioning that our analytical approaches proposed and the expressions derived can be applied not only to linear network coding, but also to the other families of network coding schemes, provided that, in these network coding schemes, a network coding node requires one packet from each of the incoming links before encoding a new outgoing packet. Furthermore, our study of the single network coding node may be extended to the networks having multiple network coding nodes. For example, once the throughput values of the network coding nodes in a network are respectively obtained, the achievable throughput of the network, or its approximation, may be found with the aid of the max-flow min-cut principles [10]. However, we should realize that the throughput analysis of the networks with multiple network coding nodes is much more challenging than that of the single network coding node. Hence, the throughput analysis of networks with multiple coding nodes constitutes one of the open problems that require further research.

Manuscript received October 16, 2009; revised October 18, 2010 and May 30, 2011; accepted November 15, 2011; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Shakkottai. Date of publication December 22, 2011; date of current version October 11, 2012. This work was presented in part at the IEEE Vehicular Technology Conference (VTC2010-Spring), Taipei, Taiwan, May 16–19, 2010.

The authors are with the Faculty of Physical and Applied Science, School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: yq06r@ecs.soton.ac.uk; lly@ecs.soton.ac.uk).

Digital Object Identifier 10.1109/TNET.2011.2178860

The remainder of this contribution is organized as follows. Section II provides a review for the work related to this paper. Section III considers the 2ISO network coding system, where system model, representation of coding node's operations, and steady-state throughput are considered. Then, in Section IV, the steady-state throughput of the MISO network coding node is addressed. Performance results are provided in Section V, and finally, in Section VI, we provide the conclusions.

II. Related Work

In this section, we provide a brief summary of the existing references related to our work in this paper. Network coding with feedback has first been considered in [11], where the authors use some simple examples to indicate that using feedback associated with network coding may be beneficial to parameter adaptation, reliability enhancement, and packet acknowledgment in network coding systems. In [12], the authors have introduced buffer into random network coding in order to overcome the problem due to asynchronous arrivals of incoming packets. In [13], an ARQ-assisted network coding scheme, namely the "*drop-when-seen*" algorithm, has been proposed and studied for the sake of reducing queue size. Motivated to reduce the average decoding delay, this ARQ-assisted network coding scheme has also been extended to the three-receiver case [14], where one transmitter communicates with multiple receivers. To be in a little more detail, in the ARQ-assisted *drop-when-seen* network coding, the transmitter broadcasts linear combinations of the packets stored in its buffer. In order to decide which packets are combined for transmission, in [13] and [14], the concept of "*seen*" packets has been introduced, which can be explained as follows. Let us denote a packet by $\boldsymbol{p}$. Then, $\boldsymbol{p}$ is said to be *seen* by a receiver if it is capable of computing the linear combination of $\boldsymbol{p}+\boldsymbol{q}$, where $\boldsymbol{q}$ is a packet linearly combined only by the packets arriving at this receiver later than the packet $\boldsymbol{p}$. In the ARQ-assisted *drop-when-seen* network coding, once a packet $\boldsymbol{p}$ is seen by all the receivers concerned, then, the packet $\boldsymbol{p}$ can be removed from the transmitter's buffer. In this manner, the transmission queue can be kept relatively short [13], while the average decoding delay can be reduced [14].

Additionally, there are several schemes, which have been proposed in order to improve the delay performance of the ARQ-assisted network coding schemes. Specifically, in [15], the authors have suggested to transmit uncoded packets at critical moments, so as to meet the delay requirement. By contrast, in [16], the authors have developed a feedback-based adaptive broadcast coding scheme in order to reduce the delay for the applications where packets must be accepted in order. Furthermore, in contrast to two-hop networks, as the one considered in this paper, some network coding schemes have been proposed and analyzed in the context of single-hop networks. For example, in [17], a random network coding framework employing hybrid ARQ scheme has been proposed for single-hop real-time media broadcast. In [10], a queuing-based dynamic network coding scheme has been studied in the context of the single-hop networks consisting of one source node and multiple receivers. The studies in [10] illustrate that this network coding scheme is capable of asymptotically achieving both the throughput of near the max-flow min-cut bound and the channel erasure probability of near zero.

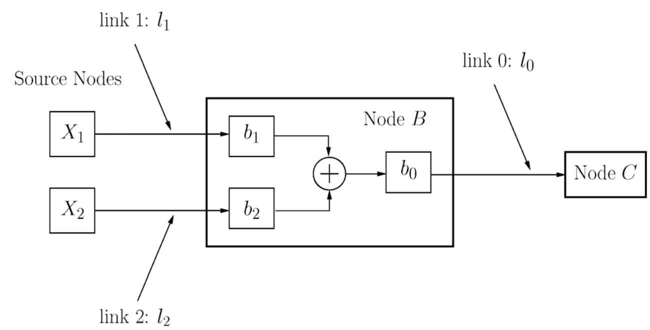

Fig. 1. Network coding node B with two incoming links l_1, l_2 and one outgoing link l_0.

Against this background, in this paper, our motivation is to use mathematical and simulation approaches to study the throughput performance of the network coding nodes when the link-level SW-ARQ data transmission scheme is employed. As noted previously in Section I, our analytical approaches as well as the derived expressions may be suitable for the network coding nodes employing possibly different coding strategies. Although our study in this paper was inspired by [11] and [12], however, to the best of the authors' knowledge, the work has not been published in the open literature.

Furthermore, we note that our 2ISO model represents an abstraction of the network coding nodes in a variety of applications, such as the network coding nodes employed in distributed antenna systems [18]. It can be seen that our MISO model and the Fork-Join networks [19], [20] share some similarities if we view each of the incoming links of the coding node in our model as a "server" of the Fork-Join networks. However, the Fork-Join network is a one-hop network model consisting of several parallel queues. By contrast, our MISO network considered is a two-hop network, where the first hop consists of several parallel queues as in the Fork-Join network, while the second hop is a first-input–first-service (FIFS) queue with finite buffer. Let us now turn to consider the throughput analysis of the 2ISO network coding node.

III. Throughput of Two-Input–Single-Output Network Coding Node

A. System Models

The system considered in this contribution is shown in Fig. 1, which is constituted by three types of nodes, namely source nodes X_1 and X_2, network coding node B, as well as sink node C. Packets to be transmitted are generated at the source nodes X_1, X_2. Node B is a 2ISO node employing packet-level network coding, while the sink node C receives packets from node B. As shown in Fig. 1, link l_0 connects nodes B and C, while links l_1 and l_2 connect the source nodes X_1 and X_2 with node B. We assume that packets are transmitted based on the SW-ARQ strategy by the links l_0, l_1, and l_2.

At node B of Fig. 1, b_0 denotes the buffer that stores the outstanding packet being transmitted by l_0. By contrast, b_1 and b_2 are the buffers that store the packets having been successfully received from links l_1 and l_2, respectively. In Fig. 1, the network coding operation is represented by $\oplus$. Note that, although the notation $\oplus$ is used, it does not necessary mean the "XOR" operation. In fact, our study in this contribution is independent of the specific network coding employed and is suitable for any types of network coding operations, provided that the following assumptions adopted in this paper are met.

- The system is operated in synchronous manner.
- The sources X_1 and X_2 always have packets to send. The nth packets generated by X_1 and X_2 are denoted by $x_1(n)$ and $x_2(n)$, respectively.
- Each of the links l_0, l_1, and l_2 is divided into two channels: the forward channel and the feedback channel. The forward channel is assumed to be a binary symmetric channel. The probability that a detectable packet error occurs is denoted as p_0, p_1, or p_2 for the link l_0, l_1, or l_2. We assume that the undetectable packet errors can be neglected, which is usually reasonable since, for most error-control codes adopted in practical communication systems, the probability of undetectable errors is very small, in comparison to the probability of detectable errors. Furthermore, we assume that the feedback channel is perfect without yielding transmission errors.
- Let T denote the round-trip time (RTT), which is the time duration between when a node sends a packet and when it receives a confirmation signal. We assume that half of an RTT, i.e., $T/2$, is required for transmitting a packet from one node to another by the corresponding forward channel. Similarly, half of an RTT is required for sending a confirmation signal from one node to another by the corresponding feedback channel.
- The duration of packets is assumed much shorter than T of the RTT and can be ignored. Furthermore, the processing time of a packet is assumed to be absorbed in the RTT.
- Buffer b_0, b_1, or b_2 can store only one packet. When both $x_1(n)$ and $x_2(n)$ are ready and stored in b_1 and b_2, once b_0 is empty, $x_1(n)$ and $x_2(n)$ are encoded to form $x_0(n)$, which is immediately stored into b_0. At the same time, the buffers b_1 and b_2 are released for receiving the following packets.

Based on the above assumptions, the operations at nodes X_1, X_2, B, and C at time $t = mT$, $m = 0, 1, \ldots$, or $t = (m - (1/2))T$, $m = 1, 2, \ldots$, can be described as follows.

1) First, at $t = 0$, the source nodes X_1 and X_2 transmit $x_1(0)$ and $x_2(0)$ through respective links l_1 and l_2 to node B. Correspondingly, the outstanding packet is set to $x_1(0)$ for source node X_1 and $x_2(0)$ for source node X_2.
2) Assume that a packet $x_1(n)$ (or $x_2(n)$) is transmitted by node X_1 (or X_2) to node B at time $t = (m - 1)T$, $m = 1, 2, \ldots$. This packet arrives at node B through the forward channel of link l_1 (or l_2) at time $t = (m-(1/2))T$ after half of an RTT. Upon receiving this packet, node B checks whether the packet $x_1(n)$ (or $x_2(n)$) is corrupted during the transmission and whether the buffer b_1 (or b_2) is available to store. If the received packet is corrupted or the buffer b_1 (or b_2) is still occupied by the last packet $x_1(n-1)$ (or $x_2(n-1)$), a NACK is sent back through the feedback channel of link l_1 (or l_2) to X_1 (or X_2). Otherwise, node B sends an ACK to node X_1 (or X_2) and simultaneously stores $x_1(n)$ (or $x_2(n)$) into buffer b_1 (or b_2).
   In the context of the sink node C, if there is a packet $x_0(n)$ transmitted by node B at $t = (m - 1)T$, this packet is received by node C from the forward channel of link l_0 at time $t = (m - (1/2))T$. In this case, the outstanding packet of node B is set to $x_0(n)$, which is stored temporarily in buffer b_0. Upon receiving this packet, node C checks whether the packet is corrupted during the transmission. If the packet is corrupted, a NACK is fed back to node B. By contrast, if the packet is assumed correct, then $x_0(n)$ is accepted by node C, and node C sends an ACK to node B using the feedback channel of link l_0.
3) At $t = mT$, source node X_1 (or X_2) and coding node B check the feedback channels to see whether the last transmitted packets are successfully conveyed. Specifically, for the source node X_1 (or X_2), if an ACK is received from the feedback channel of link l_1 (or l_2) while the outstanding packet is $x_1(n)$ (or $x_2(n)$), then source node X_1 (or X_2) transmits the next packet $x_1(n+1)$ (or $x_2(n+1)$). Simultaneously, the outstanding packet for X_1 (or X_2) is set to $x_1(n+1)$ (or $x_2(n+1)$). However, if a NACK is received from the feedback channel of link l_1 (or l_2) while the outstanding packet for source node X_1 (or X_2) is $x_1(n)$ (or $x_2(n)$), then the packet $x_1(n)$ (or $x_2(n)$) is retransmitted to node B through the forward channel of link l_1 (or l_2).
   For the coding node B, if an ACK is received from the feedback channel of link l_0 while the outstanding packet stored in b_0 is $x_0(n)$, then node B updates the outgoing packet from $x_0(n)$ to $x_0(n+1)$ by encoding the contents of b_1 and b_2. Simultaneously, $x_0(n+1)$ is transmitted from node B to node C through the forward channel of link l_0. By contrast, if a NACK is received from the feedback channel of link l_0 while the outstanding packet stored in b_0 is $x_0(n)$, then node B retransmits $x_0(n)$ over the forward channel of link l_0 and retains $x_0(n)$ as its outstanding packet.

According to our above discussion, explicitly, the system model of Fig. 1 fits well the coding path in the butterfly network, as shown by the heavily lined path in Fig. 2, associated with the following assumptions.

- The two links from source nodes X_1 and X_2 to node B are two links that employ end-to-end SW-ARQ transmission scheme associated with an RTT of T.
- Node C has an infinite buffer capacity for storing received packets. In other words, node C rejects a received packet only if the packet is detected in error.

As we know from [1] and [3], in the butterfly network of Fig. 2, the data $\{x_1(n)\}$ and $\{x_2(n)\}$ generated at the source nodes X_1 and X_2, respectively, are required to be sent to both the sink nodes S_1 and S_2. Specifically, when we consider the sink node S_1, the data $\{x_1(n)\}$ generated at the source node X_1 can be directly sent to this sink node without the constraint from the network coding node B. Hence, the corresponding

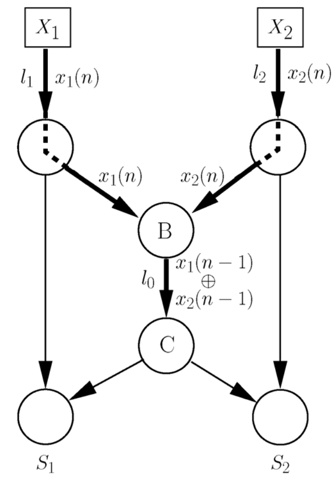

Fig. 2. Schema of butterfly network. It can be seen that the two-input–one-output coding node model shown in Fig. 1 forms the coding path of the butterfly network.

throughput may not be affected by the throughput of the coding node B. However, as seen in Fig. 2, the data $\{x_2(n)\}$ sent from the source node X_2 to the sink node S_1 goes through the network coding node B, and therefore, the resultant throughput is limited by the network coding node. Furthermore, if we assume that all links in the network have a similar reliability, then we have reason to assume that the throughput of the direct link from X_1 to S_1 is higher than that of the coding path containing node B. This is because successful delivery of a packet by the coding node B depends on that it correctly receives two packets from both the source nodes X_1 and X_2. Consequently, if transmitting data from X_1 and X_2 to S_1 is the ultimate objective, the overall throughput at node S_1 is the sum of the direct link's throughput and the coding path's throughput. However, if S_1 is also an intermediate node and if its following operations depend on both $\{x_1(n)\}$ and $\{x_2(n)\}$, then the achievable throughput at node S_1 is limited by the coding path's throughput. Additionally, from Fig. 2, we can easily know that analyzing the throughput of the coding path is much more challenging than analyzing the throughput of the direct links, which is a classic problem of throughput analysis. Therefore, it is highly important for us to study separately the throughput of network coding node.

B. State Machine Modeling of Operations at Network Coding Node

In this section, we first show that a finite state machine can be employed to represent the network coding system shown in Fig. 1. Then, we illustrate that this finite state machine is a stationary Markov process. Therefore, the steady-state throughput of the system is analyzed, where the steady-state throughput is defined as the rate of the packets received by the node C from the coding node B.

Let us first analyze the throughput of node B. Let $Q = \{0, 1\}$ be a set containing the two states of b_i, $i = 0, 1, 2$, where state "1" corresponds to the state that b_i is available to store a new packet, while state "0" indicates that b_i is occupied. Then, the state of node B at every $t = mT$ is determined by the

TABLE I
LIST OF STATES OF THE CODING NODE B

State	$S(m)$	State	$S(m)$
S_0	$(0,0,0)$	S_4	$(1,0,0)$
S_1	$(0,0,1)$	S_5	$(1,0,1)$
S_2	$(0,1,0)$	S_6	$(1,1,0)$
S_3	$(0,1,1)$	S_7	$(1,1,1)$

triple $S(m) = [q_2(m), q_1(m), q_0(m)]$, where $q_i(m) \in Q$ for $i = 0, 1, 2$. Let $S_0, S_1, \ldots, S_7$ denote the states of node B corresponding to all the possible combinations of the triple $S(m) = [q_2(m), q_1(m), q_0(m)]$, which are listed in Table I.

As an example, Fig. 3 demonstrates the state transition of node B with the time, where the number in every box represents the generation number of the corresponding packet, i.e., the number indicating which batch the packet comes from. In more detail, Fig. 3 can be explained as follows.

1) Let us assume that at $t = 0$, the state of node B is $S(0) = S_7$. At $t = 0$, source nodes X_1 and X_2 send packets $x_1(1)$ and $x_2(1)$ to node B.
2) At $t = T/2$, $x_1(1)$ and $x_2(1)$ are received by node B without any errors and, hence, $x_1(1)$ and $x_2(1)$ are stored in b_1 and b_2, and node B sends correspondingly the ACKs to nodes X_1 and X_2. In this case, the state of node B at $t = T/2$ changes to $S(1/2) = S_1$.
3) At $t = T$, node B encodes $x_1(1)$ and $x_2(1)$ into $x_0(1)$, which is stored into b_0 and is also sent to node C. Correspondingly, the state of node B is set to $S(1) = S_6$. At $t = T$, source nodes X_1 and X_2 receive the ACKs for $x_1(1)$ and $x_2(1)$. Hence, they send $x_1(2)$ and $x_2(2)$ to node B at $t = T$.
4) As seen in Fig. 3, at $t = 1.5T$, $x_2(2)$ is received by node B without error, but $x_1(2)$ is received in error. Hence, node B sends node X_2 an ACK for $x_2(2)$, but sends node X_1 a NACK for requesting a retransmission of $x_1(2)$. Correspondingly, the state of node B becomes $S(1.5) = S_2$. Furthermore, it can be implied from Fig. 3 that transmission errors are detected in $x_0(1)$ by node C. Thus, node C sends back node B a NACK to request retransmission of $x_0(1)$ at $t = 1.5T$.
5) At $t = 2T$, node B receives a NACK for $x_0(1)$ from node C and retransmits $x_0(1)$ over the forward channel of l_0. At the same time, node X_2 receives an ACK for $x_2(2)$, while node X_1 receives a NACK for $x_1(2)$ from node B. Hence, node X_2 transmits a new packet $x_2(3)$ over the forward channel of link l_2, and node X_1 retransmits $x_1(2)$ over the forward channel of link l_1. At $t = 2T$, the state of node B remains unchanged and is $S(2) = S_2$.
6) At $t = 2.5T$, $x_2(3)$ arrives at node B without error, but finds b_2 is occupied. Thus, node B sends a NACK along the feedback channel of link l_2. At the same time, $x_1(2)$ is received by node B without error. Therefore, node B sends an ACK to node X_1 and stores $x_1(2)$ into b_1, making the state of b_1 change from 1 to 0. Furthermore, at $t = 2.5T$, $x_0(2)$ arrives at node C without error. Therefore, node C sends an ACK to node B using the feedback channel of l_0. From the above analysis, it can be seen that the state of node B at $t = 2.5T$ is $S(2.5) = S_0$.

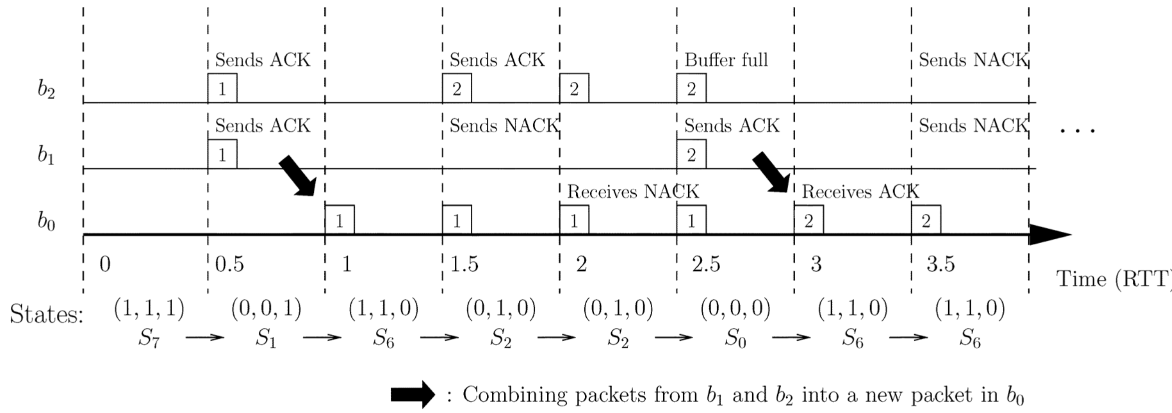

Fig. 3. Illustration of state transition of node B.

7) At $t = 3T$, node X_2 receives a NACK for $x_2(3)$, and node X_1 receives a ACK for $x_1(2)$, hence node X_2 retransmits $x_2(3)$, and node X_1 transmits a new packet $x_1(3)$ over the forward channels of links l_2 and l_1, respectively. At $t = 3T$, node B receives an ACK for $x_0(1)$ from the feedback channel of link l_0. Hence, node B encodes $x_2(2)$ and $x_1(2)$ into $x_0(2)$ and then stores it into b_0 as well as sends it to node C. Based on the above analysis, the state of node B at $t = 3T$ is $S(3) = S_6$.
8) As shown in Fig. 3, at $t = 3.5T$, both $x_2(3)$ and $x_1(3)$ are detected by node B with transmission errors. Hence, both packets are rejected by node B, and the states of b_2 and b_1 do not change. Since b_0 is still occupied by $x_0(2)$ at $t = 3.5T$, the state of node B at $t = 3T$ is still $S(3.5) = S_6$.

The state of node B at other times can be analyzed in the same way as discussed above. Explicitly, the network coding operations at node B can be described by a state machine with eight states. Let us analyze the steady-state throughput of the coding node B. Note that, in this paper, the normalized throughput is considered. Given that $N(m)$ packets are successfully conveyed from node B to node C during $t = 0$ and $t = mT$, the normalized throughput is defined as

$$R(m) = \frac{N(m)}{m}, \qquad m = 1, 2, \ldots. \tag{1}$$

When $m \rightarrow \infty$, $R(m)$ converges to the normalized steady-state throughput expressed by R. Furthermore, it can be readily shown that the steady-state throughput R is equal to the probability that coded packets are formed and forwarded to b_0 by node B.

C. Throughput Analysis

From the analysis in Section III-B, we can know that the throughput analysis of node B can be carried out by considering only the integer values of m. This is because, as seen in Fig. 3 and the corresponding description, a packet can possibly be successfully delivered at $t = mT$ only when m is an integer. Furthermore, as demonstrated in Section III-B, when m is an integer, the state of node B will never enter state S_1, which can be explained in detail as follows. Let us assume that the state of node B at time $t = mT$ is $S(m) = S_1$. Then, according to the operation procedure described in Section III-A, we must have $q_1(m-1/2) = 0$ and $q_2(m-1/2) = 0$. However, $q_0(m-1/2)$ only takes value "0" or "1," which results in the following cases.

1) $q_0(m-(1/2)) = 1$: Since $q_0(m-1/2) = 1$, then at $t = mT$, packets in b_1 and b_2 should be encoded into a new packet and stored into b_0, making $q_0(m) = 0$. Simultaneously, both the buffers of b_1 and b_2 are released for receiving the following packets, yielding $q_1(m) = 1, q_2(m) = 1$. Hence, the state of node B at $t = mT$ should be $S(m) = S_6$, which contradicts the assumption of $S(m) = S_1$.
2) $q_0(m-(1/2)) = 0$, and node B receives an ACK at $t = mT$: In this case, buffer b_0 is released at $t = mT$. The packets in b_1 and b_2 are encoded to form a new packet and stored into b_0, making $q_0(m) = 0$. Since both b_1 and b_2 are available for receiving the following packets, we have $q_1(m) = 1, q_2(m) = 1$. Hence, the state of node B at $t = mT$ is $S(m) = S_6$ instead of $S(m) = S_1$ as assumed.
3) $q_0(m-(1/2)) = 0$, and node B receives a NACK at $t = mT$: In this case, node B needs to retransmit the packet stored in b_0 and $q_0(m) = 0$. Furthermore, $q_1(m) = 0$, $q_2(m) = 0$, since both b_1 and b_2 cannot be released at $t = mT$. Hence, the state of node B in this case is $S(m) = S_0$, which again contradicts the assumption of $S(m) = S_1$.

From the above analysis, it can be implied that, when $q_1(m-1/2) = 0$ and $q_2(m-1/2) = 0$, $q_0(m) = 0$ for any cases, implying that b_0 is always occupied at $t = mT$. Therefore, in our analysis below, we assume that m is an integer and do not consider the state S_1.

Let $P_i(m)$ denote the probability that the state of node B is S_i at time $t = mT$. Let $\boldsymbol{p}(m) = [P_0(m), P_2(m), \cdots, P_7(m)]^T$, which does not contain $P_1(m)$ since $P_1(m)$ is always zero. Explicitly, we have $\sum_{i=0,2}^{7} P_i(m) = 1$. Let us assume that the system starts with an idle state $S(0) = S_7$. Then, we have

$$\boldsymbol{p}(0) = [0, 0, 0, 0, 0, 0, 1]^T. \tag{2}$$

Let $P_{i,j}(m) = P[S(m+1) = S_j | S(m) = S_i]$, $i, j = 0, 2, \cdots, 7$, denote the transition probability from state S_i at $t = mT$ to state S_j at $t = (m+1)T$. $P_{i,j}(m)$ can be derived based on the buffer states in S_i and S_j as well as the probabilities that links l_0, l_1, and l_2 successfully deliver their packets.

Specifically, let $D_i, i = 0, 1, 2$, denote the event that link l_i successfully delivers a packet and $P[D_i] = \bar{p}_i$, while $\bar{D}_i$ denote the event that l_i fails to deliver a packet on link l_i and $P[\bar{D}_i] = p_i$. Explicitly, $p_i = 1 - \bar{p}_i$. Then, it can be shown that we have

$$\begin{aligned}
P_{0,0}(m) &= P\left[S(m+1) = S_0 | S(m) = S_0\right] \\
&= P[\bar{D}_0] = p_0 \\
P_{0,2}(m) &= P\left[S(m+1) = S_2 | S(m) = S_0\right] \\
&= 0 \text{ (illegitimate transition)} \\
&\vdots \\
P_{4,5}(m) &= P\left[S(m+1) = S_5 | S(m) = S_4\right] \\
&= P[D_0]\left(P[D_1] + P[\bar{D}_1]\right) P[\bar{D}_2] = \bar{p}_0 p_2 \\
&\vdots \\
P_{6,6}(m) &= P\left[S(m+1) = S_6 | S(m) = S_6\right] \\
&= P[D_0]P[D_1]P[D_2] + P[\bar{D}_0]P[\bar{D}_1]P[\bar{D}_2] \\
&= \bar{p}_0\bar{p}_1\bar{p}_2 + p_0p_1p_2 \\
&\vdots \\
P_{7,7}(m) &= P\left[S(m+1) = S_7 | S(m) = S_7\right] \\
&= P[\bar{D}_1]P[\bar{D}_2] = p_1p_2.
\end{aligned} \tag{3}$$

As an example, we explain how $P_{4,5}(m)$ is obtained. Note that since $i, j \neq 1$, $P_{4,5}(m)$ is the $(4, 5)$th element of $\mathbf{P}$ seen in (4) at the bottom of the page. For $P_{4,5}(m)$, we have $S(m) = [100]$ and $S(m+1) = [101]$, which means that a packet in b_0 is successfully delivered to node C, yielding a probability $P[D_0] = \bar{p}_0$, while the link l_2 fails to deliver b_2 a packet, corresponding to a probability $P[\bar{D}_2] = p_2$. Furthermore, since there is a packet occupying b_1, it cannot accept a new packet no matter whether this packet is correct or not. Hence, this transition probability is $P[D_1] + P[\bar{D}_1] = 1$. Consequently, we have $P_{4,5}(m) = P[D_0](P[D_1] + P[\bar{D}_1])P[\bar{D}_2] = \bar{p}_0 p_2$. From the above equations in (3), we can see that $\{P_{i,j}(m)\}$'s are independent of m, implying that $\{P_{i,j}(m)\}$ are time-invariant. Therefore, in our forthcoming discourse, the index m associated with the transition probabilities is dropped for simplicity. Finally, with the aid of (3), it can be shown that the transition matrix $\mathbf{P}$ can be expressed as (4).

Using the *law of total probability* [24], the probability $P_j(m+1)$ can be expressed as

$$P_j(m+1) = \sum_{i=0,2}^{7} P_{i,j}P_i(m), \qquad j = 0, 2, \ldots, 7 \tag{5}$$

which, when considering all the seven states, can be expressed in vector form as

$$\boldsymbol{p}(m+1) = \boldsymbol{P}^T \boldsymbol{p}(m). \tag{6}$$

Equation (6) represents a recursive equation. Hence, it can be readily shown that $\boldsymbol{p}(m)$ can be expressed as

$$\boldsymbol{p}(m) = \left(\boldsymbol{P}^T\right)^m \boldsymbol{p}(0), \qquad m = 0, 1, \ldots. \tag{7}$$

As shown in (4), the sum of each row of $\mathbf{P}$ is equal to one. Hence, $\mathbf{P}^T$ is a left stochastic matrix [24], whose limit of $\lim_{m\to\infty}(\mathbf{P}^T)^m$ exists, according to the *Perron–Frobenius theorem* [25], [24]. Therefore, when $m \to \infty$, the Markov process becomes stationary [26] and yields

$$\boldsymbol{p}(m+1) = \boldsymbol{p}(m). \tag{8}$$

Let $\boldsymbol{\pi} = [\pi_0, \pi_2, \cdots, \pi_7]^T = \lim_{m\to\infty} \boldsymbol{p}(m)$. Then, the steady-state probabilities in $\boldsymbol{\pi}$ can be obtained by solving the equation

$$\boldsymbol{\pi} = \boldsymbol{P}^T \boldsymbol{\pi} \tag{9}$$

associated with the constraint $\sum_{i=0,2}^{7} \pi_i = 1$. Equation (9) shows that $\boldsymbol{\pi}$ is the right eigenvector of matrix $\boldsymbol{P}^T$ associated with an eigenvalue one. Therefore, $\boldsymbol{\pi}$ can be evaluated with the aid of the methods for solving the eigenvector problem [24], [25].

Finally, when reaching the steady-state, the throughput of the 2ISO network coding node of Fig. 1 or of the 2ISO network coding branch of Fig. 2 is the rate that the encoded packets are successfully conveyed from node B to node C. This rate is also equal to the rate that packets in b_1 and b_2 are encoded and forwarded to b_0. When this rate is normalized by T of the RTT, the throughput is then equal to the probability that the packets are forwarded from nodes b_1 and b_2 to b_0.

From the operation principles described in Section III-A, we can know that the state $S_i, i \neq 6$, say, at $t = mT$ always changes to S_6 at $t = (m+1)T$, when the packets in b_1 and b_2 are encoded and forwarded to b_0 at $t = (m+1)T$. Otherwise, the state at $t = (m+1)T$ will never be S_6. By contrast, when $S(m) = S_6$, then $S(m+1) = S_6$ either if all the packets transmitted at $t = mT$ are correctly conveyed—which results in that the two packets received by node B are encoded and forwarded to b_0 at $t = (m+1)T$, and this probability is $\overline{p}_0\overline{p}_1\overline{p}_2$—or if all the packets transmitted at $t = mT$ are detected in error, which makes node B retains its previous state S_6. However, only the former event makes contribution to the throughput. Therefore,

$$\boldsymbol{P} = \begin{bmatrix}
p_0 & 0 & 0 & 0 & 0 & \overline{p}_0 & 0 \\
p_0\overline{p}_1 & p_0p_1 & \overline{p}_0p_1 & 0 & 0 & \overline{p}_0\overline{p}_1 & 0 \\
0 & 0 & p_1 & 0 & 0 & \overline{p}_1 & 0 \\
p_0\overline{p}_2 & 0 & 0 & p_0p_2 & \overline{p}_0p_2 & \overline{p}_0\overline{p}_2 & 0 \\
0 & 0 & 0 & 0 & p_2 & \overline{p}_2 & 0 \\
p_0\overline{p}_1\overline{p}_2 & p_0p_1\overline{p}_2 & \overline{p}_0p_1\overline{p}_2 & p_0\overline{p}_1p_2 & \overline{p}_0\overline{p}_1p_2 & \overline{p}_0\overline{p}_1\overline{p}_2 + p_0p_1p_2 & \overline{p}_0p_1p_2 \\
0 & 0 & p_1\overline{p}_2 & 0 & \overline{p}_1p_2 & \overline{p}_1\overline{p}_2 & p_1p_2
\end{bmatrix} \tag{4}$$

when considering all the above events, the throughput of the 2ISO network coding node normalized by T of the RTT can be expressed as

$$\begin{aligned} R &= P_0(\infty)P_{0,6} + P_2(\infty)P_{2,6} + P_3(\infty)P_{3,6} + P_4(\infty)P_{4,6} \\ &\quad + P_5(\infty)P_{5,6} + P_6(\infty)\overline{p}_0\overline{p}_1\overline{p}_2 + P_7(\infty)P_{7,6} \\ &= \pi_0 P_{0,6} + \pi_2 P_{2,6} + \pi_3 P_{3,6} + \pi_4 P_{4,6} + \pi_5 P_{5,6} \\ &\quad + \pi_6\overline{p}_0\overline{p}_1\overline{p}_2 + \pi_7 P_{7,6} \end{aligned} \tag{10}$$

where $\pi_i P_{i,6}$, $i \neq 6$, is the probability of the event that the current state is S_i, which transits to S_6 at the next time.

From the discussion and analysis in this section, it can be implied that our analytical approaches and the derived expressions may be applied for the network coding nodes with different network coding schemes, provided that, in these network coding schemes, a network coding node requires one packet from each of the incoming links before encoding a new outgoing packet. Note furthermore that our approaches and derived expressions are also suitable for the node employing random network coding. This is because, in our model, every buffer associated with the network coding node is assumed to store at most one packet. In this case, when operated under the SW-ARQ, the network coding always deals with the packets of the same generation of all the source nodes, regardless of the random network coding scheme. Let us now consider the case of network coding nodes having multiple incoming channels.

IV. Throughput of Multiple-Input–Single-Output Network Coding Node

In this section, we extend our study in Section III to the MISO network coding nodes, as shown, for example, in the diagram of Fig. 4. As with Fig. 1 for the 2ISO system, the MISO network coding node of Fig. 4 is also constituted by three types of nodes, namely, source nodes $X_1, X_2, \ldots, X_H$, network coding node B, and sink node C. As shown in Fig. 4, the source nodes $X_1, X_2, \ldots, X_H$ are connected with the coding node B through the H links, $l_1, l_2, \ldots, l_H$. The sink node C is connected with the coding node B via one link l_0. We assume that each link in Fig. 5 consists of a forward channel and a feedback channel. Furthermore, in Fig. 4, b_0 and $b_1, b_2, \ldots, b_H$ are the buffers for storing the outstanding packet transmitted on link l_0 and the packets received from links $l_1, l_2, \ldots, l_H$, respectively. It is assumed that each buffer can store at most one packet.

As for the 2ISO network coding node B in Fig. 4, the operations of the MISO network coding node B in Fig. 4 can also be modeled by a Markov chain with $(2^{H+1} - 1)$ states and a corresponding transition matrix $\boldsymbol{P}$. The throughput of the MISO network coding node can also be analyzed in the same way as that for the 2ISO network coding node, once the state transition matrix $\boldsymbol{P}$ is available. Hence, in order to analyze the throughput of the MISO network coding node, it is essential to determine first the state transition matrix $\boldsymbol{P}$. In this paper, an algorithm is proposed for generating the state transition matrix $\boldsymbol{P}$, which is described in detail as follows.

Let $S(m) = [q_H(m), q_{H-1}(m), \cdots, q_0(m)]$ denote the state at time $t = mT$, where $q_h(m) = 1$ or 0, and $h = 0, 1, \ldots, H$, means that b_h is empty ($q_h(m) = 1$) or occupied ($q_h(m) = 0$) at $t = mT$. It can be shown that, for the MISO network coding node, the state $S(m) = [0, 0, \ldots, 0, 1]$ does not exist, which explains that the Markov chain has $(2^{H+1} - 1)$ states. Let the $(2^{H+1} - 1)$ possible states of $S(m)$ be expressed as $S_0, S_2, \ldots, S_{2^{H+1}-1}$, where the subscript i is the integer representation of a corresponding binary representation of $S(m) = [q_H(m), q_{H-1}(m), \cdots, q_0(m)]$, such as $S_0 = [0, 0, \ldots, 0]$, $S_{2^{H+1}-1} = [1, 1, \ldots, 1]$, etc.

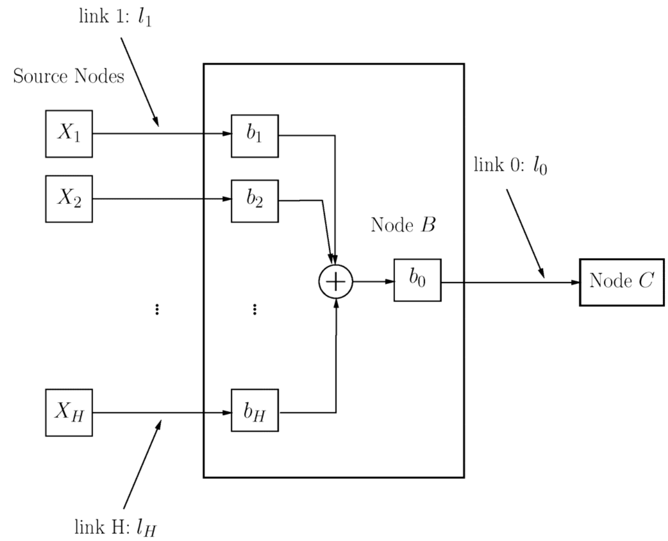

Fig. 4. Network coding node B with H incoming links $l_1, l_2, \ldots, l_H$ and one outgoing link l_0.

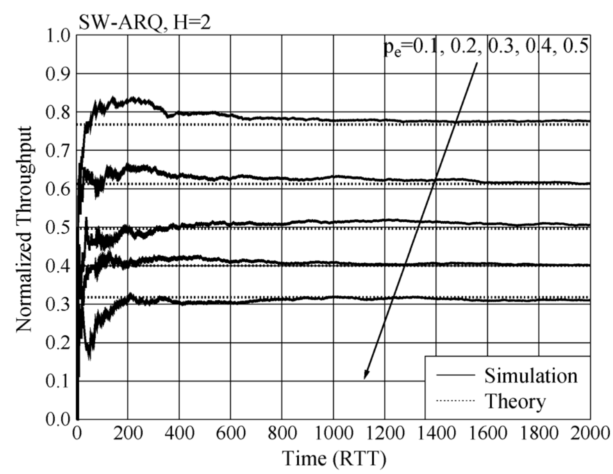

Fig. 5. Steady-state throughput and simulated throughput for the network coding node shown in Fig. 4 with $H = 2$. Explicitly, the simulated throughput converges to the theoretical steady-state throughput.

Let the $(2^{H+1} - 1) \times (2^{H+1} - 1)$-dimensional transition matrix be $\boldsymbol{P} = [P_{i,j}]$, $i, j = 0, 2, \ldots, 2^{H+1} - 1$, where $P_{i,j} = P_{i,j}(m) = P[S(m+1) = S_j | S(m) = S_i]$ expresses the transition probability from state S_i at $t = mT$ to state S_j at $t = (m+1)$. Let $S_{\hat{J}} = (1, 1, \cdots, 1, 0)$, where $\hat{J} = 2^{(H+1)} - 2$. Then, the calculation of $P_{i,j}$ can be described as follows.

First, $P_{\hat{J},\hat{J}}$ corresponding to the transition probability from $S(m) = S_{\hat{J}}$ to $S(m+1) = S(m) = S_{\hat{J}}$ is an exceptional case that needs to be considered specifically. The state changes from $S(m) = S_{\hat{J}}$ to $S(m+1) = S(m) = S_{\hat{J}}$, either if all the packets transmitted on the $(H+1)$ links are correctly received, which corresponds a probability of $\prod_{k=0}^{H} \overline{p}_k$, or if all these packets are failed to deliver, which has a probability of $\prod_{k=0}^{H} p_k$. Hence, the transition probability of $P_{\hat{J},\hat{J}}$ is given by

$$P_{\hat{J},\hat{J}} = \prod_{k=0}^{H} p_k + \prod_{k=0}^{H} \overline{p}_k. \tag{11}$$

Second, the transition probability $P_{i,\hat{J}}, i = 0,2,\ldots,\hat{J}-1,\hat{J}+1$, can be expressed in the form of

$$P_{i,\hat{J}} = \prod_{h=0}^{H} f_h, \quad i = 0,2,\ldots,\hat{J}-1,\hat{J}+1 \tag{12}$$

where, by definition

$$f_0 = \begin{cases} \overline{p}_0, & \text{if } q_0(m) = 0 \\ 1, & \text{if } q_0(m) = 1 \end{cases}$$
$$f_h = \begin{cases} 1, & \text{if } q_h(m) = 0 \\ \overline{p}_h, & \text{if } q_h(m) = 1 \end{cases} \quad h \geq 1. \tag{13}$$

Note that f_0 may be interpreted as the probability of availability for b_0 to store the next new packet, and b_0 is always available ($f_0 = 1$) at $t = (m+1)T$ if $q_0(m) = 1$ and is available with a probability of $f_0 = \overline{p}_0$ at $t = (m+1)T$ if $q_0(m) = 0$. By contrast, $f_h, h \geq 1$, may be interpreted as the probability of availability of a packet at b_h that can be encoded and forwarded to b_0 at $t = (m+1)T$. Explicitly, $f_h = 1$ if $q_h(m) = 0$ and $f_h = \overline{p}_h$ if $q_h(m) = 1$.

Except for the above cases, all the other entries $\{P_{i,j}, j \neq \hat{J}\}$ in $\boldsymbol{P}$ correspond to the transitions that do not yield throughput. In these cases, the state of b_0 is not affected by the states of $b_1, b_2, \ldots, b_H$, implying that there is no correlation between the inputs and output of the network coding node. Furthermore, from the operation of the network coding node as described in the previous sections, the operations of the input links of the network coding node are also independent. Therefore, the probability of a transition of this type can be obtained by the product of the individual state transition probabilities of $b_0, b_1, \ldots, b_H$, due to the fact that the operations associated with their corresponding links are independent. Specifically, the transition probability $P_{i,j}$ with $j \neq \hat{J}$ can be expressed in the form of

$$P_{i,j} = \prod_{h=0}^{H} f_h, \qquad i = 0,2,\ldots,\hat{J}+1;$$
$$j = 0,2,\ldots,\hat{J}-1,\hat{J}+1 \tag{14}$$

where f_0 is the corresponding transition probability of b_0 defined as

$$f_0 = \begin{cases} p_0, & \text{if } q_0(m) = 0 \text{ and } q_0(m+1) = 0 \\ \overline{p}_0, & \text{if } q_0(m) = 0 \text{ and } q_0(m+1) = 1 \\ 0, & \text{if } q_0(m) = 1 \text{ and } q_0(m+1) = 0 \\ 1, & \text{if } q_0(m) = 1 \text{ and } q_0(m+1) = 1 \end{cases} \tag{15}$$

and, for $h = 1,2,\ldots,H$, the corresponding transition probability of b_h can be expressed as

$$f_h = \begin{cases} 1, & \text{if } q_h(m) = 0 \text{ and } q_h(m+1) = 0 \\ 0, & \text{if } q_h(m) = 0 \text{ and } q_h(m+1) = 1 \\ \overline{p}_h, & \text{if } q_h(m) = 1 \text{ and } q_h(m+1) = 0 \\ p_h, & \text{if } q_h(m) = 1 \text{ and } q_h(m+1) = 1. \end{cases} \tag{16}$$

Note that since the state of node B always changes to $S_{\hat{J}}$ after a new packet is formed and forwarded to b_0, hence the transition from S_i to S_j with $j \neq \hat{J}$ means no new packet is formed and forwarded to b_0. When keeping this in mind, it is then not difficult to follow the equations in (15) and (16). For example, when $q_0(m) = 0$ (occupied) and $q_0(m+1) = 0$ (occupied), this event occurs only when the packet is not successfully delivered at $t = (m+1)T$. Hence, we have $f_0 = p_0$. As another example, for the case of $h \geq 1$, the event of $q_h(m) = 1$ (empty) and $q_h(m+1) = 0$ (occupied) means that b_h is filled by a new packet at $t = (m+1)T$. Hence, the probability is $f_h = \bar{p}_h$.

After the transition probability matrix $\boldsymbol{P}$ is determined, the steady-state probabilities in $\boldsymbol{\pi}$ can be obtained by following the equations from (6)–(9). Finally, the throughput of the MISO network coding node normalized by T of the RTT is the total probability of entering the state $S_{\hat{J}}$ excluding the probability of the erroneous event from the state $S(m) = S_{\hat{J}}$ to the state $S(m+1) = S_{\hat{J}}$. Hence, the normalized throughput can be expressed as

$$\begin{aligned} R &= \sum_{i=0,2}^{2^{(H+1)}-1} \pi_i P_{i,\hat{J}} - \pi_{\hat{J}} \prod_{h=0}^{H} p_h \\ &= \sum_{i=0,2;i\neq\hat{J}}^{2^{(H+1)}-1} \pi_i P_{i,\hat{J}} + \pi_{\hat{J}} \prod_{h=0}^{H} \overline{p}_h. \end{aligned} \tag{17}$$

It can be readily shown that, when $H = 2$, (17) is reduced to (10), the throughput expression of 2ISO network coding nodes.

Let us demonstrate a range of performance results for characterizing the throughput performance of the network coding node.

V. Performance Results

In this section, we illustrate a range of numerical and simulation results in order to characterize the throughput performance of the MISO network coding node and to justify our analytical results obtained in the previous sections. In our simulation and numerical evaluation examples, we assume that the packet error rates of links $l_0, l_1, \ldots, l_H$ are the same and equal to p_e.

As an example, Table II shows the steady-state probabilities $\boldsymbol{\pi}$ as well as the normalized throughput, which were evaluated using (10), for different packet error rate values. As the results in Table II show, the normalized throughput decreases as the packet error rate of the communication links increases.

Figs. 5–8 depict the normalized throughput versus the time normalized by RTT, when the network coding node employs $H = 2,3,4,5$ incoming links, respectively. In these figures, the corresponding steady-state throughput evaluated by (17) are depicted for all the packet error rate values considered.

From the results of Figs. 5–8, we can observe that the simulated throughput starts at $R(0) = 0$ since there are no packets

TABLE II
STEADY-STATE PROBABILITIES $\boldsymbol{\pi}$ AND CORRESPONDING STEADY-STATE THROUGHPUT EVALUATED BY (10)

p_e	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
π_0	0.0705	0.1070	0.1282	0.1418	0.1515	0.1591	0.1656	0.1714
π_2	0.0070	0.0206	0.0353	0.0489	0.0606	0.0702	0.0778	0.0836
π_3	0.0705	0.1070	0.1282	0.1418	0.1515	0.1591	0.1656	0.1714
π_4	0.0070	0.0206	0.0353	0.0489	0.0606	0.0702	0.0778	0.0836
π_5	0.0705	0.1070	0.1282	0.1418	0.1515	0.1591	0.1656	0.1714
π_6	0.7676	0.6173	0.5096	0.4279	0.3636	0.3120	0.2699	0.2351
π_7	0.0070	0.0206	0.0353	0.0489	0.0606	0.0702	0.0778	0.0836
R	0.7669	0.6123	0.4958	0.4005	0.3182	0.2446	0.1773	0.1147

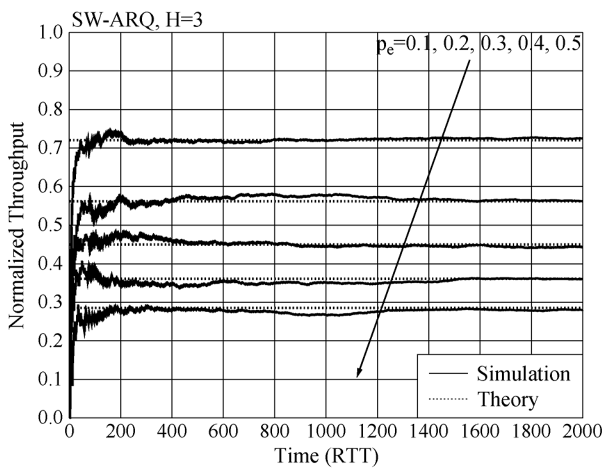

Fig. 6. Steady-state throughput and simulated throughput for the network coding node shown in Fig. 4 with $H = 3$. Explicitly, the simulated throughput converges to the theoretical steady-state throughput.

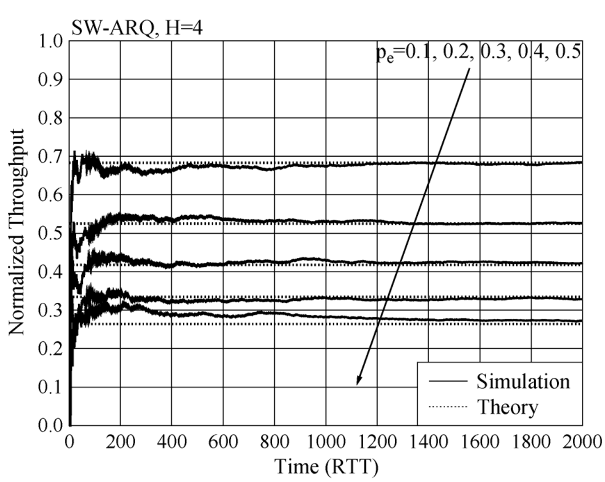

Fig. 7. Steady-state throughput and simulated throughput for the network coding node shown in Fig. 4 with $H = 4$. Explicitly, the simulated throughput converges to the theoretical steady-state throughput.

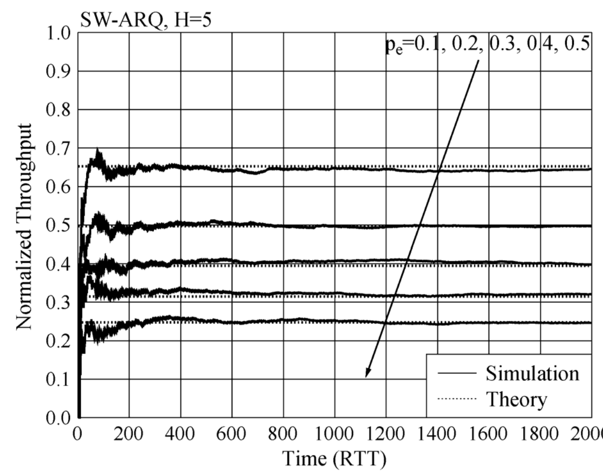

Fig. 8. Steady-state throughput and simulated throughput for the network coding node shown in Fig. 4 with $H = 5$. Explicitly, the simulated throughput converges to the theoretical steady-state throughput.

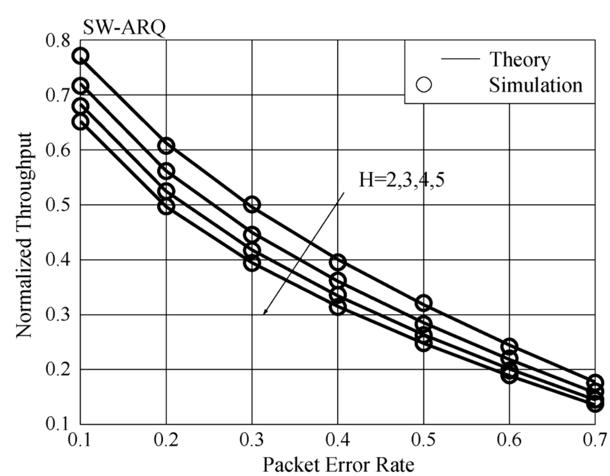

Fig. 9. Steady-state throughput versus packet error rate performance of the network coding node associated with $H = 2, 3, 4, 5$ incoming links.

received by node C at $t = 0$. Then, the throughput obtained by simulations fluctuates around its corresponding steady-state throughput obtained from evaluation of (17) due to insufficient number of samples. Finally, the throughput obtained by simulations converges to the theoretical steady-state throughput. The results of Figs. 5–8 demonstrate that our analytical results obtained in Sections III and IV are effective for evaluation of the steady-state throughput of MISO network coding nodes.

Finally, in Fig. 9, we compare the normalized steady-state throughput versus packet error rate performance of the network coding node, when it has $H = 2, 3, 4, 5$ incoming links. The results of Fig. 9 show that at a given packet error rate p_e, the normalized throughput decreases when the coding node has more incoming links. This is because, explicitly, the chance of forming a new packet and forwarding it to buffer b_0 becomes smaller, as the new packet depends on correctly receiving more packets. Additionally, from Fig. 9 we observe that the difference of the normalized throughput corresponding to different number of incoming links becomes smaller as the packet error rate increases.

VI. Conclusion

In this paper, we have investigated the steady-state throughput of network coding nodes when the simple SW-ARQ transmission scheme is employed. Expressions for computing the steady-state throughput have been obtained by first considering a specific 2ISO network coding node and then extending it to the more general MISO network coding nodes. The steady-state throughput performance is investigated by both simulation and numerical approaches. It can be shown that the simulation results justify our analytical expressions derived. Furthermore, our performance results show that the throughput of a coding node decreases as the number of source nodes feeding packets into the coding node increases. This property implies that, in a network coding system, the coding nodes may form the bottlenecks for information delivery. The method presented in this paper is versatile in the sense that the steady-state throughput of a network coding system can be evaluated by following the similar steps of our method, provided that a corresponding transition probability matrix $\boldsymbol{P}$ can be obtained. Our future research in this area will consider extensions to the network coding systems using other types ARQ schemes, where every buffer may store a number of packets, and to the network coding systems with multiple coding nodes.

Acknowledgment

The authors would like to thank the anonymous reviewers for their fruitful suggestions.

References

[1] 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.
[2] D. S. Lun, M. Medard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. 42nd Annu. Allerton Conf. Commun., Control, Comput.*, Sept.–Oct. 2004, pp. 20–29.
[3] 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.
[4] D. S. Lun, M. Medard, R. Koetter, and M. Effros, "Further results on coding for reliable communication over packet networks," in *Proc. IEEE Int. Symp. Inf. Theory*, Sept. 4–9, 2005, pp. 1848–1852.
[5] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2004.
[6] A. S. Tanenbaum, *Computer Networks*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2003.
[7] Y. Hayashida, "Throughput analysis of tandem-type go-back-n ARQ scheme for satellite communications," *IEEE Trans. Commun.*, vol. 41, no. 10, pp. 1517–1524, Oct. 1993.
[8] Y. Qin and L.-L. Yang, "Throughput analysis of stop-and-wait automatic repeat request scheme for network coding nodes," in *Proc. IEEE VTC2010-Spring*, Taipei, Taiwan, May 16–19, 2010, pp. 1–5.
[9] Y. Qin and L.-L. Yang, "Throughput analysis of general network coding nodes based on SW-ARQ transmission," in *Proc. IEEE VTC2010-Fall*, Ottawa, ON, Canada, Sep. 6–9, 2010, pp. 1–5.
[10] Y. Sagduyu and A. Ephremides, "On broadcast stability of queue-based dynamic network coding over erasure channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5463–5478, Dec. 2009.
[11] C. Fragouli, D. Lun, M. Medard, and P. Pakzad, "On feedback for network coding," in *Proc. 41st Annu. Conf. Inf. Sci. Syst.*, Mar. 14–16, 2007, pp. 248–252.
[12] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Annu. Allerton Conf. Commun., Control, Comput.*, 2003.
[13] J. K. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 6–11, 2008, pp. 1651–1655.
[14] J. K. Sundararajan, D. Shah, and M. Médard, "Online network coding for optimal throughput and delay-the two-receiver case," *Comput. Res. Repository*, vol. abs/0806.4264, 2008.
[15] J. Barros, R. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 208–216.
[16] J. K. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *Proc. Workshop Netw. Coding, Theory, Appl.*, Jun. 2009, pp. 1–6.
[17] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Hybrid ARQ-random network coding for wireless media streaming," in *Proc. 2nd Int. Conf. Commun. Electron.*, Jun. 4–6, 2008, pp. 115–120.
[18] Y. Chen, S. Kishore, and J. Li, "Wireless diversity through network coding," in *Proc. IEEE WCNC*, 2006, vol. 3, pp. 1681–1686.
[19] Y. Dallery, Z. Liu, and D. Towsley, "Equivalence, reversibility, symmetry and concavity properties in fork-join queuing networks with blocking," *J. ACM*, vol. 41, no. 5, pp. 903–942, 1994.
[20] Y. Dallery, Z. Liu, and D. Towsley, "Properties of fork/join queuing networks with blocking under various operating mechanisms," *IEEE Trans. Robot. Autom.*, vol. 13, no. 4, pp. 503–518, Aug. 1997.
[21] R. Nelson and A. Tantawi, "Approximate analysis of fork/join synchronization in parallel queues," *IEEE Trans. Comput.*, vol. 37, no. 6, pp. 739–743, Jun. 1988.
[22] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 829–848, Mar. 2006.
[23] A. Lebrecht and W. J. Knottenbelt, "Response time approximations in Fork-Join queues," in *Proc. 23rd Annu. UKPEW*, Jul. 2007 [Online]. Available: http://pubs.doc.ic.ac.uk/forkjoin
[24] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, 4th ed. New York: Academic, 2009.
[25] R. A. Horn, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.
[26] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Upper Saddle River, NJ: Prentice-Hall, 1992.

Yang Qin, photograph and biography not available at the time of publication.

Lie-Liang Yang photograph and biography not available at the time of publication.