

Throughput Analysis of Stop-and-Wait Automatic Repeat Request Scheme for Network Coding Nodes

Yang Qin and Lie-Liang Yang

School of ECS, University of Southampton, SO17 1BJ, United Kingdom

Tel: 0044-(0)23-8059 3364, Email: yq06r,lly@ecs.soton.ac.uk; http://www-mobile.ecs.soton.ac.uk

Abstract— This contribution analyzes the steady-state throughput of general network coding nodes, when data is transmitted based on 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 modelled by a finite state machine. Therefore, the throughput expressions of general 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.

I. INTRODUCTION

Network coding has received a lot of attention, since its invention by Ahlswede, Cai, Li and Yeung [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 contribution we motivate to study the steady-state throughput of the general network coding nodes, where communications between two nodes are based on the stop-and-wait automatic repeat request (SW-ARQ) data transmission scheme [5]. Note that, we focus our attention on the steady-state throughput of network coding node, since, when communicating over unreliable communications channels, the achievable throughput of a communications network is mainly constrained by the throughput of the embedded network coding nodes.

Network coding with feedback has first been studied in [6]. In [7], the problems of applying ARQ techniques to network coding have been addressed, suggesting to add an extra network coding layer into the TCP/IP stack in order to introduce network coding into the existing Internet infrastructure. Moreover, in [8] a random network coding framework employing hybrid ARQ scheme has been proposed for real-time media broadcast over single-hop wireless networks.

In this contribution, the steady-state throughput of general network coding node is analyzed, where a coding node is associated with multiple-input-single-output (MISO) links. Therefore, we refer to it as the MISO network coding node. 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 operation of network coding nodes can be modelled 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 are extended to the general MISO network coding nodes. Finally, the throughput performance of network coding nodes with different number 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. Furthermore, it is worthy of mentioning that our analytical approaches and expressions in this contribution are general, which are independent of the specific network coding algorithms employed.

II. THROUGHPUT OF TWO-INPUT-SINGLE-OUTPUT NETWORK CODING NODE

A. System Models

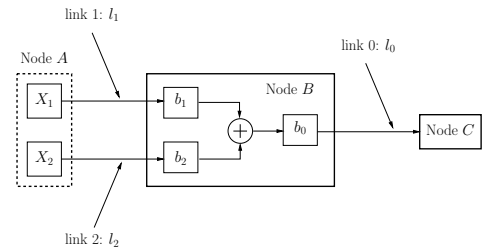


Fig. 1. A network coding node B with two incoming links l_1 , l_2 and one outgoing link l_0 .

The system considered is shown in Fig. 1. Node A is a source node residing the two information sources X_1 and X_2 , which generate the packets to be transmitted. Node B is a two-input-single-output (2ISO) node employing packet-level network coding. Node C is a sink node receiving packets from node B . As shown in Fig. 1, Link l_0 connects node B and node C , while Links l_1 and l_2 are the two links from node A to node B , which connect the sources X_1 and X_2 with node B , respectively. 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 . The buffers b_1 and b_2 store the packets having been successfully received by links l_1 and l_2 , respectively. In Fig. 1 the network coding operation is presented by \oplus . Note that, our study in this contribution is

independent of the specific network coding employed and is suitable for any types of network coding operations.

In order to carry out the analysis, the following assumptions are adopted.

- The system is operated in a synchronous manner.
- The sources X_1 and X_2 always have packets to send. The n th packets generated by X_1 and X_2 are denoted as $x_1(n)$ 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 true since for most error-control codes adopted in practical communications systems the probability of undetectable error is very small, in comparison with the probability of detectable error. Furthermore, we assume that the feedback channel is perfect without yielding transmission error.
- Let T denote the round trip time (RTT), which is the time duration between a node sends a packet and it receives a confirmation signal. We assume that half of a RTT, i.e., $T/2$, is required for transmitting a packet from one node to another by the corresponding forward channel. Similarly, half of a RTT is required for sending a confirmation signal from one node to another by the corresponding feedback channel.
- The duration of packets is much shorter than T of the RTT and can be ignored. Furthermore, the processing time of a packet at a receiving node can also be ignored.
- 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 the nodes of A , B and C at time $t = mT$, $m = 0, 1, \dots$ or $t = (m - 1/2)T$, $m = 1, 2, \dots$ can be described as follows.

- 1) First, at $t = 0$, the source node A transmits $x_1(0)$ and $x_2(0)$ through respectively links l_1 and l_2 to node B . Correspondingly, the outstanding packet is set to $x_1(0)$ for source X_1 and $x_2(0)$ for source X_2 .
- 2) Assume that a packet $x_1(n)$ (or $x_2(n)$) is transmitted by node A to node B at time $t = (m - 1)T$, $m = 1, 2, \dots$, 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 a 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) of node A . Otherwise, node B sends an ACK to node A 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 temporally 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 to be 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 $t = mT$, nodes A and B check the feedback channels to decide whether the last transmitted packets are successfully conveyed. Specifically, for the source X_1 (or X_2) of node A , 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 node A transmits the next packet $x_1(n + 1)$ (or $x_2(n + 1)$) from source X_1 (or X_2). 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 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 , provided that they have been prepared. 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 maintains $x_0(n)$ as its outstanding packet.

Note that, the system model of Fig. 1 fits well the coding path in the butterfly network [1]. Therefore, it is highly important to study the throughput of the 2ISO network coding node B , since the packet rate of the butterfly network is mainly determined by the throughput of node B [1, 3].

Let $Q_i = \{0, 1\}$, $i = 0, 1, 2$, be a set containing the two states of b_i , $i = 0, 1, 2$, where $q_i = 1$ corresponds to the state that b_i is available to store new packet, while $q_i = 0$ indicates that b_i is occupied. Then, the state of node B at every $t = mT$ is determined by the triple $S(m) = [q_2(m), q_1(m), q_0(m)]$, where $q_i(m) \in Q_i$ for $i = 0, 1, 2$. Let S_0, S_1, \dots, S_7 denote the eight states of node B corresponding to all the possible combinations of the triple $S(m) = [q_2(m), q_1(m), q_0(m)]$, where $S_0 = [0, 0, 0], \dots, S_7 = [1, 1, 1]$. Then, the network coding operations at node B can be described by a state machine with eight states.

Note that, when m is an integer, it can be shown that node B does not enter state $(q_2(m), q_1(m), q_0(m)) = S_1$. This is because, once b_1 and b_2 are prepared, their contents should be encoded and stored to b_0 , making the state of node B be S_6 , instead of the state of S_1 . Let us now analyze the steady-state throughput of the coding node B .

B. Throughput Analysis

The throughput analysis of node B can be carried out by considering only the integer values of m . This is because a packet can possibly be successfully delivered at $t = mT$ only when m is an integer. Furthermore, as mentioned previously, when m is an integer, the state of node B will never enter the state S_1 . 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 $\mathbf{p}(m) = [P_0(m), P_2(m), \dots, P_7(m)]^T$, 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 assume that the system starts with an idle state $S(0) = S_7$. Then we have

$$\mathbf{p}(0) = [0, 0, 0, 0, 0, 0, 1]^T \quad (1)$$

Let $P_{i,j}(m) = P[S(m+1) = S_j | S(m) = S_i]$, $i, j = 0, 2, \dots, 7$, 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 the transition matrix can be expressed as

$$\mathbf{P} = \begin{bmatrix} p_0 & 0 & 0 & 0 & 0 & \bar{p}_0 & 0 \\ p_0\bar{p}_1 & p_0p_1 & \bar{p}_0p_1 & 0 & 0 & \bar{p}_0\bar{p}_1 & 0 \\ 0 & 0 & p_1 & 0 & 0 & \bar{p}_1 & 0 \\ p_0\bar{p}_2 & 0 & 0 & p_0p_2 & \bar{p}_0p_2 & \bar{p}_0\bar{p}_2 & 0 \\ 0 & 0 & 0 & 0 & p_2 & \bar{p}_2 & 0 \\ p_0\bar{p}_1\bar{p}_2 & p_0p_1\bar{p}_2 & \bar{p}_0p_1\bar{p}_2 & p_0\bar{p}_1p_2 & \bar{p}_0\bar{p}_1p_2 & \bar{p}_0\bar{p}_1\bar{p}_2 + p_0p_1p_2 & \bar{p}_0p_1p_2 \\ 0 & 0 & p_1\bar{p}_2 & 0 & \bar{p}_1p_2 & \bar{p}_1\bar{p}_2 & p_1p_2 \end{bmatrix} \quad (2)$$

which is independent of m , implying that \mathbf{P} is time-invariant. Therefore, in our forthcoming discourse the index m associated with the transition probabilities is dropped for convenience.

The probability $P_j(m)$ can be expressed as

$$P_j(m) = \sum_{i=0,2}^7 P_{i,j} P_i(m-1), \quad j = 0, 2, \dots, 7 \quad (3)$$

which can be expressed in vector form as

$$\mathbf{p}(m) = \mathbf{P}^T \mathbf{p}(m-1) = (\mathbf{P}^T)^m \mathbf{p}(0), \quad m = 0, 1, \dots \quad (4)$$

As shown in (2), the sum of each row of \mathbf{P} equals one. Hence, \mathbf{P}^T is a left stochastic matrix [9], whose limit of $\lim_{m \rightarrow \infty} (\mathbf{P}^T)^m$ exists, according to the *Perron-Frobenius theorem* [9, 10]. Therefore, when $m \rightarrow \infty$, the Markov process becomes stationary and yields

$$\mathbf{p}(m) = \mathbf{p}(m-1) \quad (5)$$

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

$$\boldsymbol{\pi} = \mathbf{P}^T \boldsymbol{\pi} \quad (6)$$

associated with the constraint $\sum_{i=0,2}^7 \pi_i(m) = 1$. Equation (6) shows that $\boldsymbol{\pi}$ is the right eigenvector of matrix \mathbf{P}^T associated

with an eigenvalue one. Therefore, $\boldsymbol{\pi}$ can be evaluated with the aid of the methods for solving the eigenvector problem [9, 10].

Finally, when reaching the steady-state, the throughput of the 2ISO network coding node of Fig. 1 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 node b_1 and b_2 to b_0 .

From the operation principles as described in Section II-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 $\bar{p}_0\bar{p}_1\bar{p}_2$, or if all the packets transmitted at $t = mT$ are detected in errors - which makes node B maintains its previous state S_6 . However, only the former event makes contribution to the throughput. Therefore, when considering all the above events, the normalized throughput of the 2ISO network coding node can be expressed as

$$\begin{aligned} R &= P_0(\infty)P_{0,6} + P_2(\infty)P_{2,6} + P_3(\infty)P_{3,6} + P_4(\infty)P_{4,6} \\ &\quad + P_5(\infty)P_{5,6} + P_6(\infty)\bar{p}_0\bar{p}_1\bar{p}_2 + P_7(\infty)P_{7,6} \\ &= \pi_0P_{0,6} + \pi_2P_{2,6} + \pi_3P_{3,6} + \pi_4P_{4,6} + \pi_5P_{5,6} \\ &\quad + \pi_6\bar{p}_0\bar{p}_1\bar{p}_2 + \pi_7P_{7,6} \end{aligned} \quad (7)$$

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. Let us now consider a general network coding node having multiple incoming channels.

III. THROUGHPUT OF MULTIPLE-INPUT-SINGLE-OUTPUT NETWORK CODING NODE

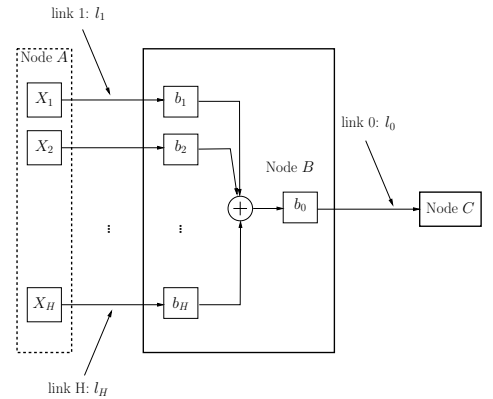


Fig. 2. A network coding node B with H incoming links l_1, l_2, \dots, l_H and one outgoing link l_0 .

In this section we extend our study in the last section to a general multiple-input-single-output (MISO) network coding node, as shown in Fig. 2, which has three nodes, A , B and C . Node A accommodates H number of information sources, X_1, X_2, \dots, X_H , which are connected with the coding node

B through H links, l_1, l_2, \dots, l_H . Again, we assume that each link consists of a forward channel and a feedback channel. In Fig. 2 node C is a sink node, which is connected with node B by one link l_0 . Finally, in Fig. 2 b_0 and b_1, b_2, \dots, 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, \dots, l_H , respectively. It is assumed that each buffer can store at most one packet.

The operations of the MISO network coding node B in Fig. 2 can be modelled by a Markov chain with $(2^{H+1} - 1)$ states and a corresponding transition matrix \mathbf{P} . Below we first propose an algorithm for generating the state transition matrix \mathbf{P} , which is required for evaluating the throughput.

Let $S(m) = [q_H(m), q_{H-1}(m), \dots, q_0(m)]$ denote the state at time $t = mT$, where $q_h(m) = 1$ or 0 , $h = 0, 1, \dots, H$, means that b_h is empty or occupied at $t = mT$. It can be shown that, for the MISO network coding node, the state $S(m) = [0, 0, \dots, 0, 1]$ 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 $S_0, S_2, \dots, S_{2^{H+1}-1}$, where the subscript i is the integer representation of a corresponding binary representation of $S(m) = [q_H(m), q_{H-1}(m), \dots, q_0(m)]$, such as, $S_0 = [0, 0, \dots, 0]$, $S_{2^{H+1}-1} = [1, 1, \dots, 1]$, etc.

Let the $(2^{H+1} - 1) \times (2^{H+1} - 1)$ -dimensional transition matrix be $\mathbf{P} = [P_{i,j}]$, $i, j = 0, 2, \dots, 2^{H+1} - 1$, where $P_{i,j} = P_{i,j}(m) = P[S(m+1) = S_j | S(m) = S_i]$ expresses the transition probability from the state S_i at $t = mT$ to the state S_j at $t = (m+1)T$. Let $S_j = (1, 1, \dots, 1, 0)$, where $\hat{j} = 2^{H+1} - 2$. Then, the algorithm for computing $P_{i,j}$ can be described as follows.

First, $P_{\hat{j},j}$ corresponding to the transition probability from $S(m) = S_{\hat{j}}$ to $S(m) = S(m+1) = S_j$ is an exceptional case need to be considered specifically. The state changes from $S(m) = S_{\hat{j}}$ to $S(m) = S(m+1) = S_j$, either if all the packets transmitted on the $(H+1)$ links are corrected received - which corresponds a probability of $\prod_{k=0}^H \bar{p}_k$, 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 $P_{\hat{j},j}$ is given by

$$P_{\hat{j},j} = \prod_{k=0}^H p_k + \prod_{k=0}^H \bar{p}_k \quad (8)$$

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

$$P_{i,\hat{j}} = \prod_{h=0}^H f_h, \quad i = 0, 2, \dots, \hat{j} - 1, \hat{j} + 1 \quad (9)$$

where, by definition,

$$f_0 = \begin{cases} \bar{p}_0, & \text{if } q_0(m) = 0 \\ 1, & \text{if } q_0(m) = 1 \end{cases} \quad (10)$$

$$f_h = \begin{cases} 1, & \text{if } q_h(m) = 0 \\ \bar{p}_h, & \text{if } q_h(m) = 1 \end{cases}, \quad h \geq 1$$

Note that, f_0 may be interpreted as the probability of availability for b_0 to store the next new packet, 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 $f_0 = \bar{p}_0$ 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 $f_h = \bar{p}_h$ if $q_h(m) = 1$.

Finally, the other entries of \mathbf{P} corresponding to $j \neq \hat{j}$ can be expressed in the form of

$$P_{i,j} = \prod_{h=0}^H f_h, \quad i = 0, 2, \dots, \hat{j} + 1, \quad j = 0, 2, \dots, \hat{j} - 1, \hat{j} + 1 \quad (11)$$

where, for $h = 0$, f_0 is defined as

$$f_0 = \begin{cases} p_0, & \text{if } q_0(m) = 0 \text{ and } q_0(m+1) = 0 \\ \bar{p}_0, & \text{if } q_0(m) = 0 \text{ and } q_0(m+1) = 1 \\ 0, & \text{if } q_0(m) = 1 \text{ and } q_0(m+1) = 0 \\ 1, & \text{if } q_0(m) = 1 \text{ and } q_0(m+1) = 1 \end{cases} \quad (12)$$

and, for $h = 1, 2, \dots, H$, we have

$$f_h = \begin{cases} 1, & \text{if } q_h(m) = 0 \text{ and } q_h(m+1) = 0 \\ 0, & \text{if } q_h(m) = 0 \text{ and } q_h(m+1) = 1 \\ \bar{p}_h, & \text{if } q_h(m) = 1 \text{ and } q_h(m+1) = 0 \\ p_h, & \text{if } q_h(m) = 1 \text{ and } q_h(m+1) = 1 \end{cases} \quad (13)$$

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 (12) and (13). 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 \mathbf{P} is determined, the steady-state probabilities π can be obtained following the equations from (4)-(6). Finally, the normalized throughput of the general MISO network coding node 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 $S(m+1) = S_{\hat{j}}$, which can be expressed as

$$R = \sum_{i=0,2}^{2^{(H+1)}-1} \pi_i P_{i,\hat{j}} - \pi_{\hat{j}} \prod_{h=0}^H p_h$$

$$= \sum_{i=0,2;i \neq \hat{j}}^{2^{(H+1)}-1} \pi_i P_{i,\hat{j}} + \pi_{\hat{j}} \prod_{h=0}^H \bar{p}_h. \quad (14)$$

IV. PERFORMANCE RESULTS

In our simulation and numerical evaluation examples, we assume that the packet error rates of links l_0, l_1 and l_2 are the same and equal p_e .

Figs. 3 and 4 depict the normalized throughput versus the time normalized by RTT, when the network coding node employs $H = 2$ or 3 incoming links, respectively. In our simulations, the normalized throughput at $t = mT$ was obtained as

$$R(m) = \frac{N(m)}{mT}, \quad m = 1, 2, \dots \quad (15)$$

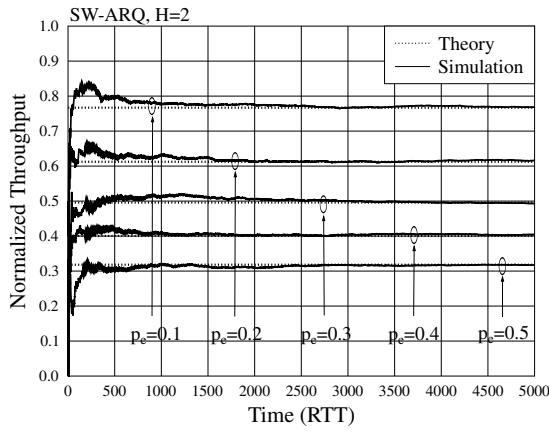


Fig. 3. The simulated throughput of the system shown in Fig. 2 with $H = 2$ converges to the theoretical throughput as time passes by.

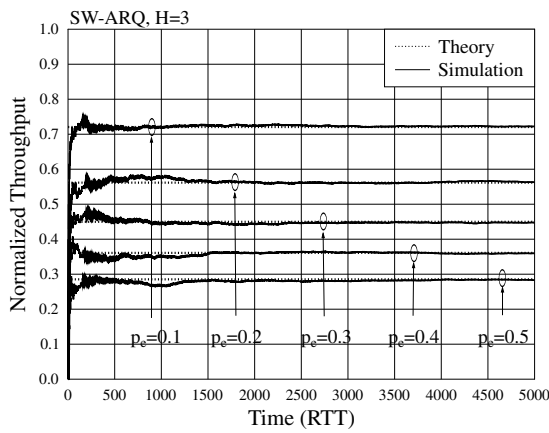


Fig. 4. The simulated throughput of the system shown in Fig. 2 with $H = 3$ converges to the theoretical throughput as time passes by.

where $N(m)$ represents the number of packets transmitted successfully from node B to node C during $t = 0$ and $t = mT$. Furthermore, in Figs. 3 and 4 the corresponding steady-state throughput evaluated by formula (14) are depicted for all the packet error rate values considered.

From the results of Figs. 3 and 4 we can observe that the simulated throughput starts at $R(0) = 0$, since there are no packets received by node C at $t = 0$. Then, the throughput obtained by simulations fluctuates around its corresponding steady-state throughput obtained from evaluation of (14), due to insufficient number of samples. Finally, the throughput obtained by simulations converges to the theoretical steady-state throughput. The results of Figs. 3 and 4 demonstrate that our analytical results obtained in Sections II and III are effective for evaluation of the steady-state throughput of general MISO network coding node.

Finally, in Fig. 5 we compare the normalized steady-state throughput versus packet error rate performance of the network coding node, when it has $H = 2, 3, 4$ and 5 incoming links. The results of Fig. 5 show that, at a given packet error rate p_e , the normalized throughput decreases, when the coding

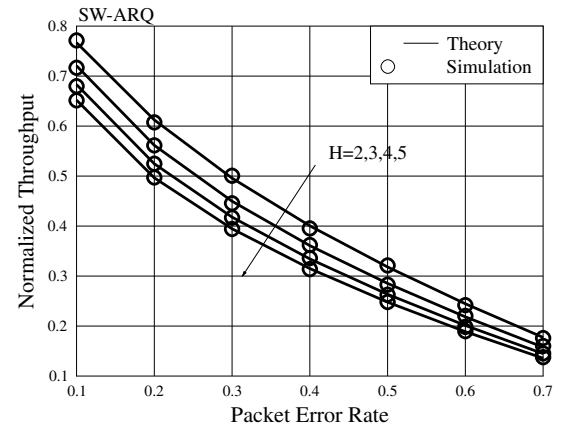


Fig. 5. Steady-state throughput versus packet error rate performance of the network coding node associated with $H = 2, 3, 4$ or 5 incoming links.

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. 5 we observe that the difference of the normalized throughput corresponding to different number of incoming links becomes smaller, as the packet error rate increases.

In **conclusion**, this contribution has investigated the steady-state throughput of general network coding nodes, when the simple SW-ARQ transmission scheme is employed. Expressions for computing the steady-state throughput have been obtained. It can be shown that the simulation results justify our analytical expressions derived. Furthermore, performance results show that the throughput of a coding node decreases, as the number of incoming links of the coding node increases. This property implies that in a network coding system the coding nodes may form the bottleneck for information delivery.

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, July 2000.
- [2] D. S. Lun, M. Medard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, Sept./Oct. 2004.
- [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. International Symposium on Information Theory '05*, 4–9 Sept. 2005, pp. 1848–1852.
- [5] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 2004.
- [6] C. Fragouli, D. Lun, M. Medard, and P. Pakzad, "On feedback for network coding," in *Proc. 41st Annual Conference on Information Sciences and Systems*, 14–16 March 2007, pp. 248–252.
- [7] J. Kumar Sundararajan, D. Shah, and M. Medard, "Arq for network coding," in *Proc. IEEE International Symposium on Information Theory 2008*, 6–11 July 2008, pp. 1651–1655.
- [8] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Hybrid arq-random network coding for wireless media streaming," in *Proc. Second International Conference on Communications and Electronics 2008*, 4–6 June 2008, pp. 115–120.
- [9] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, 4th ed. Academic Press, 2009.
- [10] R. A. Horn, *Matrix Analysis*. Cambridge University Press, 1990.