

Throughput Analysis of General Network Coding Nodes Based on SW-ARQ Transmission

Yang Qin and Lie-Liang Yang

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

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

Abstract—The steady-state throughput of general network coding node is investigated, when data is transmitted in packets based on the stop-and-wait automatic repeat request (SW-ARQ) error-control scheme. The general network coding node has H number of incoming links that provide packets for forming the coded packets transmitted by one outgoing link. Each of the incoming and outgoing links is assumed to have some buffers for storing the temporary data packets. The state transition of the general network coding node is modelled by a finite state machine, and its steady-state throughput is derived based on the properties of discrete-time Markov chain. Furthermore, the throughput performance of the general network coding node is investigated either by simulations or by evaluation of the expressions derived in this paper. Our performance results show that the analytical expressions obtained can be well justified by the simulation results. Furthermore, it can be shown that the packet error rate, the amount of buffer and the number of incoming links may impose significant impact on the achievable throughput of the general network coding nodes.

I. INTRODUCTION

Network coding deals with the problems of coding over packet networks and it has the potential to improve the capacity of conventional wired or wireless networks [1, 2]. Performance of communication networks employing network coding has been investigated, typically, under the assumption that data packets can be transmitted reliably from one node to another without error [3]. In practical wired or wireless communication networks, however, the communication channels are always non-ideal and transmission errors may occur. Hence, error-control techniques are usually required in order to ensure high-reliability communications [2, 4]. Therefore, network coding has now been studied in association with various error-control techniques, as seen, e.g., in [5–8].

In this contribution, we are interested in the steady-state throughput of the general network coding nodes, where communications between any two nodes are protected by the SW-ARQ data transmission scheme [9, 10]. By generality, we mean our analytical approaches and results are general and independent of the specific network coding schemes employed. Hence, the formulas derived are general and can be used for evaluating the steady-state throughput of network coding nodes employing possibly any specific network coding algorithms. We put our emphasis on the steady-state throughput of the general network coding node, because the achievable throughput of a network coding system is mainly constrained by the throughput achieved by the embedded network coding nodes, when communicating over unreliable communication channels.

We assume that the generalized network coding node has H incoming links and a single outgoing link and, hence, it is referred to as the *HISO* network coding node. Each of the incoming and outgoing links is assumed to have some buffer for storing the temporary data packets. In our analysis, the operations of the general network coding node are modelled by a finite state machine satisfying the principles of discrete-time Markov chain. The state transition of the *HISO* network coding node is analyzed and the approach for obtaining the state transition matrix is proposed. Furthermore, the formulas for evaluating the steady-state throughput of the general *HISO* network coding node is derived. Finally, the throughput performance is investigated using both simulation and numerical approaches. Our performance results show that the dynamic throughput obtained by simulation converges to the steady-state throughput evaluated by formulas, which, hence, justifies the effectiveness of our analytical expressions derived. It

can be shown that the packet error rate, the amount of buffer and the number of incoming links may impose significant impact on the achievable throughput of the network coding node.

II. SYSTEM MODEL, ASSUMPTIONS AND TRANSMISSION OPERATION

The system considered in this contribution is shown in Fig. 1, which is constituted by three nodes, A , B and C . Node A is a source node with H information sources X_1, X_2, \dots, X_H , which generate the packets to be transmitted through the system. Node B is a H -input-single-output (*HISO*) coding node with packet-level network coding. Finally, node C is a sink node receiving the coded packets from node B . As shown in Fig. 1, there is only one link l_0 between nodes B and C . Between nodes A and B , there are H links l_1, l_2, \dots, l_H , each of which connects one of the sources X_1, X_2, \dots, X_H with node B . We assume that packets are transmitted over the links based on the SW-ARQ scheme.

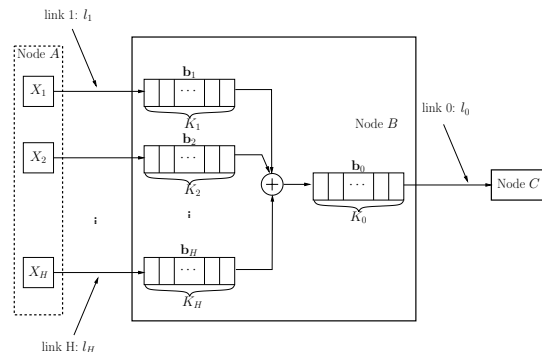


Fig. 1. A network coding node B with H incoming links l_1, l_2, \dots, l_H and one outgoing link l_0 . Each incoming link is attached to a buffer with K_{in} units, and the outgoing link is attached to a buffer with K_{out} units.

In Fig. 1, node B is a network coding node with the operation represented by \oplus . At node B , \mathbf{b}_0 is the buffer storing the packets to be transmitted to node C . By contrast, $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H$ are the buffers storing the packets received successfully from the source nodes X_1, X_2, \dots, X_H . We assume that the v th, $v = 0, 1, \dots, H$, buffer \mathbf{b}_v has K_v memory units, which are indexed by $\{u_1^v, u_2^v, \dots, u_{K_v}^v\}$; each memory unit can store one packet. Furthermore, we assume that the packets in a buffer forms a First-In-First-Serve (FIFS) queue.

In addition to the above-mentioned assumptions, the other assumptions adopted are summarized as follows.

- The system is operated in a synchronous manner.
- Packets at the sources X_1, X_2, \dots, X_H are always ready to send. The n th packet of the h th source X_h is denoted as $x_h(n)$.
- Each of the H links has two channels: the forward channel and the feedback channel. The forward channel is assumed to be a binary symmetric channel. The probability of (detectable) packet errors is denoted as p_v for the v th forward channel. We assume that the undetectable packet errors can be neglected, which is usually true since, for most error-control codes adopted in practical communication systems, the probability of undetectable errors

is very small, in comparison with the probability of detectable errors. 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 that a node sends a packet and that a confirmation signal is received. 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 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. Alternatively, we can view that the duration of packets and the processing time are included in the RTT.
- When all the packets $x_1(n), x_2(n), \dots, x_H(n)$ are ready and stored in $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H$, then, once \mathbf{b}_0 has a free unit to store, these packets are encoded to form a packet $x_0(n)$, which is immediately stored into \mathbf{b}_0 . At the same time, the memory units storing $x_1(n), x_2(n), \dots, x_H(n)$ are released for storing the other packets. Specifically, the packets in buffers $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H$ are shifted to the right by one unit. In other words, packets in a buffer are always stored in the memory units with the smallest possible subscripts by following the FIFO operation principles.
- Buffers of the sources X_1, X_2, \dots, X_H and node C can store infinite number of packets.

With the aid of the above assumptions, the operations carried out at nodes 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.

- Initially at $t = 0$, the source node A transmits $x_1(0), x_2(0), \dots, x_H(0)$ through, respectively, links l_1, l_2, \dots, l_H to node B . Correspondingly, the outstanding packet is set to $x_h(0)$ for source X_h , where $h = 1, 2, \dots, H$.
 - Assume that a packet $x_h(n)$ 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_h at time $t = (m - 1/2)T$ after half a RTT. Upon receiving this packet, node B checks whether the packet $x_h(n)$ is corrupted during the transmission and whether \mathbf{b}_h has a free memory unit available to store the packet. If the received packet is corrupted or the buffer \mathbf{b}_h is full, a NACK is sent back through the feedback channel of link l_h to X_h of node A . Otherwise, node B sends an ACK to node A and, simultaneously, stores $x_h(n)$ into buffer \mathbf{b}_h .
- For the sink node C , similarly, 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 \mathbf{b}_0 . Upon receiving the packet, node C checks whether the packet is correct. If the packet is received with errors, 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 .
- At $t = mT$, nodes A and B check the feedback channels to see whether the last transmitted packets are successfully conveyed. Specifically, for the source X_h of node A , if an ACK is received from the feedback channel of link l_h while the outstanding packet is $x_h(n)$, then node A transmits the next packet $x_h(n + 1)$ from source X_h . Simultaneously, the outstanding packet of X_h is changed to $x_h(n + 1)$. However, if a NACK is received from the feedback channel of link l_h while the outstanding packet of X_h is $x_h(n)$, then the packet $x_h(n)$ is re-transmitted to node B and the outstanding packet does not change.

For the coding node B , the operations follow the following steps:

- 1) If an ACK is received from the feedback channel of link l_0 while the outstanding packet is $x_0(n)$, then node B sends

the next packet $x_0(n + 1)$, if it is ready. At the same time, the outstanding packet is updated from $x_0(n)$ to $x_0(n + 1)$ by shifting all the packets in \mathbf{b}_0 by one unit to the right. By contrast, if A receives a NACK from node C while the outstanding packet is $x_0(n)$, then node B re-transmits $x_0(n)$ and keeps the outstanding packet unchanged. Explicitly, the outstanding packet of link l_0 is the packet stored in the right-most memory unit of \mathbf{b}_0 .

- 2) After updating the outstanding packet for the outgoing link l_0 according to the information fed back from node C , node B checks whether there are any free memory units in buffer \mathbf{b}_0 . If buffer \mathbf{b}_0 is full, node B takes no action until the next period. However, if there are free memory, then node B checks all its input buffers $\mathbf{b}_1, \dots, \mathbf{b}_H$ to see whether the packets for encoding are prepared. If yes, then the packets, say, $x_1(n'), \dots, x_H(n')$, stored in the right-most memory units of $\mathbf{b}_1, \dots, \mathbf{b}_H$ are encoded into a new packet $x_0(n')$, which is then stored into the buffer \mathbf{b}_0 . Simultaneously, the packets $x_1(n'), \dots, x_H(n')$ are removed from the buffers $\mathbf{b}_1, \dots, \mathbf{b}_H$ and the other packets in $\mathbf{b}_1, \dots, \mathbf{b}_H$ are shifted towards the right by one unit.

Let us below focus on the performance analysis of the steady-state throughput.

III. ANALYSIS OF STEADY-STATE THROUGHPUT

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, the steady-state throughput of the network coding system of Fig. 1 is analyzed. The steady-state throughput is defined as the packet rate received by node C , which, at steady-state, equals the rate that the coded packets formed by node B .

A. State Machine Modelling of Network Coding Node's Operations

Let $\mathbf{U}_{k_v}^v = \{0, 1\}$ be a set containing the two states of $u_{k_v}^v$, $v = 0, \dots, H$; $k_v = 1, \dots, K_v$, where $u_{k_v}^v = 1$ corresponds to the state that the memory unit $u_{k_v}^v$ is free to store a new packet, while $u_{k_v}^v = 0$ to the state that $u_{k_v}^v$ is occupied. Then, the state of node B at every $t = mT$ can be determined by

$$S(m) = [u_{K_H}^H \cdots u_1^H; \cdots; u_{K_v}^v; \cdots; u_{K_1}^1 \cdots u_1^1; u_{K_0}^0 \cdots u_1^0] \quad (1)$$

Let $\mathcal{S} = \{S_0, S_1, \dots, S_{2^L-1}\}$ contains all the possible states of node B , representing all the possible combinations of the L -tuple $S(m)$, where $L = \sum_{v=0}^H K_v$ is the total number of memory units in $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_H$, as explained by (1). The mapping of S_l to the L -tuple of (1) is the nature binary mapping, i.e., $l = \sum_{v=0}^H \sum_{k_v=1}^{K_v} u_{k_v}^v 2^{k_v-1+\sum_{i=0}^{v-1} K_i}$.

Note that, there are two types of states - which we refer to as two limitations for convenience - in \mathcal{S} that the coding node B will never enter. The first limitation is that the coding node B never enters the states with discontinuous memory units occupied. This is because we have assumed that the packets in $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_H$ are always stored in the memory units from right to left. The second limitation is that the coding node B never enters the states where the buffer \mathbf{b}_0 has free memory but the buffers $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H$ have packets prepared for encoding. These states will not occur, since, in this case, new coded packets can be formed and stored into \mathbf{b}_0 .

When considering only the first limitation, it can be shown that the buffer \mathbf{b}_v has $(K_v + 1)$ legitimate states, which are collected in $\mathbf{s}_v = \{1 \cdots 11, 1 \cdots 10, \dots, 0 \cdots 00\}$. The total number of states of node B is hence $\prod_{v=0}^H (K_v + 1)$. When considering the second limitation, there are K_0 cases that buffer \mathbf{b}_0 is not fully occupied. For each of these K_0 cases, there are $\prod_{v=1}^H K_v$ different legitimate combinations that any of $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H$ has at least one packet prepared. Therefore, the number of combinations for the second limitation is $\prod_{v=0}^H K_v$. Consequently,

when considering both the limitations, the total number of legitimate states of node B is

$$N_{l_s} = \prod_{v=0}^H (K_v + 1) - \prod_{v=0}^H K_v. \quad (2)$$

When derive the transition matrix $\mathbf{P} = [P_{i,j}]$, where $P_{i,j} = P[S(m+1) = j | S(m) = i]$ denotes the transition probability from state i to state j , we only need to consider these legitimate states. Hence, the dimensions of \mathbf{P} is $(N_{l_s} \times N_{l_s})$. Let all these legitimate states form a set $\tilde{\mathcal{S}} \subset \mathcal{S}$. For convenience of description, all the indices of the legitimate states in $\tilde{\mathcal{S}}$ are collected into the set \mathbf{L} . Let us now analyze the probability of state transition.

B. Probability of State Transitions

It can be shown that a state transition may or may not result in throughput. To be more specific, a state transition can only result in the following cases.

- 1) When a transition does not result in throughput, implying that no new coded packet is formed, and when the output buffer \mathbf{b}_0 is not full, then, at least one of the input buffers $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H$ must be empty in both the previous and current states.
- 2) When a transition results in throughput, implying that a new coded packet is formed during the transition, then, after the transition, the number of packets in any of the buffers that are full in the previous state decreases by one in the current state.
- 3) When a state transition occurs, the difference of the number of packets in a buffer is either zero or one.

Therefore, $P_{i,j} = 0$, if the state transition does not obey any one of the above conditions. Otherwise, we express the state transition probability as $P_{i,j} = P_{i,j}^+ + P_{i,j}^0$, where $P_{i,j}^+$ is the state transition probability yielding throughput, while $P_{i,j}^0$ is the state transition probability without yielding throughput. Let us now consider $P_{i,j}^+$ and $P_{i,j}^0$.

Let us express the state of \mathbf{b}_h at $t = mT$ as $q_h^{k_h}(m)$, $q_h^{k_h}(m) \in \mathbf{s}_h$, $h = 0, \dots, H$; $k_h = 0, \dots, K_h$, which is a K_h -length binary sequence starting with $(K_h - k_h)$ 1's followed by k_h 0's. Explicitly, $q_h^0(m)$ and $q_h^{K_h}(m)$ denote that the buffer \mathbf{b}_h are empty and full, respectively. Then, when assuming that the state transition generates throughput provided by successfully forming a new coded packet, the state transition probability can be expressed as

$$P_{i,j}^+ = \prod_{h=0}^H f_h^+, \quad i, j \in \mathbf{L}, \quad (3)$$

where f_h^+ , $h = 0, 1, \dots, H$, are the corresponding state transition probabilities of \mathbf{b}_h , $h = 0, 1, \dots, H$, from $t = mT$ to $t = (m+1)T$, under the condition that a new coded packet is formed during the transition. These probabilities are analysed as follows.

On the condition that a new coded packet is added to \mathbf{b}_0 at $t = (m+1)T$, upon considering various possible events, the conditional transition probability f_0^+ can be written as

$$f_0^+ = \begin{cases} 1, & \text{if } q_0^0(m) \rightarrow q_0^1(m+1) \\ p_0, & \text{if } q_0^{k_0}(m) \rightarrow q_0^{k_0+1}(m+1), 1 \leq k_0 \leq K_0 - 1 \\ \bar{p}_0, & \text{if } q_0^{k_0}(m) \rightarrow q_0^{k_0}(m+1), 1 \leq k_0 \leq K_0 \\ 0, & \text{else} \end{cases} \quad (4)$$

where p_0 and \bar{p}_0 are respectively the probabilities that a packet sent from node B to node C at $t = mT$ is incorrect and correct. Note that, first, $f_0^+ = 1$, if $q_0^0(m) \rightarrow q_0^1(m+1)$, since there is no transmission from node B to node C at $t = mT$, but a new coded packet is added at $t = (m+1)T$, resulting in that the probability of the transition $q_0^{k_0}(m) \rightarrow q_0^{k_0}(m+1)$ is one. Second, when $1 \leq k_0 \leq K_0 - 1$, the transition of $q_0^{k_0}(m) \rightarrow q_0^{k_0+1}(m+1)$ occurs, only when the packet

sent at $t = mT$ to node C was in error, yielding $f_0^+ = p_0$. Finally, since a new coded packet is added to \mathbf{b}_0 at $t = (m+1)T$, a packet should have been successfully conveyed to node C in order for the states of \mathbf{b}_0 at $t = mT$ and $t = (m+1)T$ to retain the same, satisfying the transition $q_0^{k_0}(m) \rightarrow q_0^{k_0}(m+1)$. Hence, this probability is $f_0^+ = \bar{p}_0$.

Remembering that one packet from each of the buffers $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H$ is removed after forming a new coded packet, the conditional transition probability f_h^+ , $h = 1, 2, \dots, H$, can be expressed as

$$f_h^+ = \begin{cases} 1, & \text{if } q_h^{K_h}(m) \rightarrow q_h^{K_h-1}(m+1) \\ p_h, & \text{if } q_h^{k_h}(m) \rightarrow q_h^{k_h-1}(m+1), 1 \leq k_h \leq K_h - 1 \\ \bar{p}_h, & \text{if } q_h^{k_h}(m) \rightarrow q_h^{k_h}(m+1), 0 \leq k_h \leq K_h - 1 \\ 0, & \text{else} \end{cases} \quad (5)$$

where $h = 1, 2, \dots, H$ and the related terms can be explained as follows. First, if \mathbf{b}_h is full at $t = mT$ corresponding to the state $q_h^{K_h}(m)$, then, no new packet is able to be added to \mathbf{b}_h at $t = (m+1)T$, no matter the received packet is correct or incorrect. In this case, after forming a new coded packet, the right-most packet in \mathbf{b}_h is deleted, making the transition from the state $q_h^{K_h}(m)$ to the state $q_h^{K_h-1}(m+1)$ certain. Hence, the probability of this transition is $f_h^+ = 1$. When $1 \leq k_h \leq K_h - 1$, the transition from the state $q_h^{k_h}(m)$ to the state $q_h^{k_h-1}(m+1)$ occurs, only when the packet received from link h is in error. Therefore, the corresponding transition probability is $f_h^+ = p_h$. Similarly, when $0 \leq k_h \leq K_h - 1$, the transition from the state $q_h^{k_h}(m)$ to $q_h^{k_h}(m+1)$ occurs, only when a new packet is received correctly from link h . Correspondingly, the transition probability is $f_h^+ = \bar{p}_h$. For all the other cases, $f_h^+ = 0$.

The transition probability $P_{i,j}^0$ can be expressed as

$$P_{i,j}^0 = \prod_{h=0}^H f_h^0, \quad i, j \in \mathbf{L} \quad (6)$$

where f_h^0 , $h = 0, 1, \dots, H$, represents the corresponding transition probability occurred with the buffer \mathbf{b}_h , $h = 0, 1, \dots, H$ from $t = mT$ to $t = (m+1)T$, under the condition that no new coded packet is formed during the transition. The conditional transition probability f_0^0 can be expressed as

$$f_0^0 = \begin{cases} 1, & \text{if } q_0^0(m) \rightarrow q_0^0(m+1) \\ p_0, & \text{if } q_0^{k_0}(m) \rightarrow q_0^{k_0}(m+1), 1 \leq k_0 \leq K_0 \\ \bar{p}_0, & \text{if } q_0^{k_0}(m) \rightarrow q_0^{k_0-1}(m+1), 1 \leq k_0 \leq K_0 \\ 0, & \text{else} \end{cases} \quad (7)$$

which can be explained similarly as for (4). The conditional transition probability f_h^0 can be expressed as

$$f_h^0 = \begin{cases} 1, & \text{if } q_h^{K_h}(m) \rightarrow q_h^{K_h}(m+1) \\ p_h, & \text{if } q_h^{k_h}(m) \rightarrow q_h^{k_h}(m+1), 0 \leq k_h \leq K_h - 1 \\ \bar{p}_h, & \text{if } q_h^{k_h}(m) \rightarrow q_h^{k_h+1}(m+1), 0 \leq k_h \leq K_h - 1 \\ 0, & \text{else} \end{cases} \quad (8)$$

where $h = 1, 2, \dots, H$. The explanation for (8) is similar as that for (5).

Finally, with $P_{i,j}^+$ and $P_{i,j}^0$ for all $i, j \in \mathbf{L}$ at hand, the probability transition matrix $\mathbf{P} = (P_{i,j})$ can be formed as $\mathbf{P} = (P_{i,j} = P_{i,j}^+ + P_{i,j}^0)$. The probability transition matrix \mathbf{P} can also be decomposed into $\mathbf{P} = \mathbf{P}^+ + \mathbf{P}^0$, where $\mathbf{P}^+ = (P_{i,j}^+)$ and $\mathbf{P}^0 = (P_{i,j}^0)$, respectively. Let us now analyze the steady-state throughput of the network coding node shown in Fig. 1.

C. Analysis of Steady-State Throughput

As discussed in Section III-A, at the times of $t = mT$, where m is an integer, node B can only be in one of the states in the set $\tilde{\mathbf{S}}$. Therefore, in our throughput analysis, we only need to consider the states in $\tilde{\mathbf{S}}$. Let $P_i(m)$ denote the probability that the state of node B is S_i at time $t = mT$, where $i \in \mathbf{L}$. Let $\mathbf{p}(m) = [P_{l_1}(m), P_{l_2}(m), \dots, P_{l_{N_{ls}}}(m)]^T$, where $l_1, l_2, \dots, l_{N_{ls}}$ are respectively the 1st, 2nd, \dots , N_{ls} th indices in \mathbf{L} . Explicitly, we have

$$\sum_{i \in \mathbf{L}} P_i(m) = 1. \quad (9)$$

Let us assume that the system is activated from an idle state $S(0) = S_{2L-1}$. Then we have

$$\mathbf{p}(0) = [0, \dots, 0, 1]^T. \quad (10)$$

Then, using the *law of total probability*, we can express the probability $P_j(m+1)$ as

$$P_j(m+1) = \sum_{i \in \mathbf{L}} P_{i,j} P_i(m), \quad j \in \mathbf{L} \quad (11)$$

where $P_{i,j} = P_{i,j}(m)$ denotes the transition probability from state S_i at $t = mT$ to state S_j at $t = (m+1)T$, which has been analyzed in Section III-B. Note that, the transition probabilities are time-invariant and independent of m . Alternatively, (11) with all $j \in \mathbf{L}$ can be expressed in vector form as

$$\mathbf{p}(m+1) = \mathbf{P}^T \mathbf{p}(m) \quad (12)$$

which is a recursive equation. Hence, given $\mathbf{p}(0)$, $\mathbf{p}(m)$ can be expressed as

$$\mathbf{p}(m) = (\mathbf{P}^T)^m \mathbf{p}(0), \quad m = 1, 2, \dots \quad (13)$$

As shown in (9), the sum of each row of \mathbf{P} equals one. Hence, \mathbf{P}^T is a left stochastic matrix [11], whose limit of $\lim_{m \rightarrow \infty} (\mathbf{P}^T)^m$ exists, according to the *Perron-Frobenius theorem* [11, 12]. Therefore, when $m \rightarrow \infty$, the Markov process becomes stationary [13] and yields

$$\mathbf{p}(m+1) = \mathbf{p}(m) = \boldsymbol{\pi} \quad (14)$$

where $\boldsymbol{\pi} = [\pi_{l_1}, \pi_{l_2}, \dots, \pi_{l_{N_{ls}}}]^T = \lim_{m \rightarrow \infty} \mathbf{p}(m)$ and $\sum_{i \in \mathbf{L}} \pi_i(m) = 1$. Therefore, letting $m \rightarrow \infty$ and applying (14) into (12), we have

$$\boldsymbol{\pi} = \mathbf{P}^T \boldsymbol{\pi} \quad (15)$$

under the constraint of $\sum_{i \in \mathbf{L}} \pi_i(m) = 1$. Equation (15) shows that $\boldsymbol{\pi}$ is a right eigenvector of matrix \mathbf{P}^T corresponding to the eigenvalue one. Therefore, the solution to $\boldsymbol{\pi}$ can be obtained with the aid of the methods for solving the eigenvector problem [11, 12].

Finally, when reaching the steady-state, the throughput of the *HISO* network coding system as shown in Fig. 1 can be measured by the rate that packets in buffers $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H$ are successfully encoded and forwarded to buffer \mathbf{b}_0 . When this rate is normalized by T of the RTT, the throughput of the network coding system is simply equal to the probability that new coded packets are formed and forwarded to \mathbf{b}_0 . According to the operation principles as detailed in Section III-B, throughput is generated only when the state transitions of node B result in the non-zero transition probabilities of $\{P_{i,j}^+\}$, which are the non-zero elements in matrix \mathbf{P}^+ . Therefore, the steady-state throughput of the *HISO* network coding system can be expressed as

$$R = \sum_{i \in \mathbf{L}} \pi_i \sum_{j \in \mathbf{L}} P_{i,j}^+. \quad (16)$$

Let us now provide a range of performance results.

IV. PERFORMANCE RESULTS

In this section we provide both numerical and simulation results in order to characterize the throughput performance of the *HISO* network coding system as shown in Fig. 1 and to justify our analytical results obtained in the previous section. In our considered examples, we assumed that the packet error rate of links l_0, l_1, \dots, l_H is the same and equal to p_e . Furthermore, we assumed that all the input buffers had the same amount of memory, expressed by K_{in} . The amount of memory of the output buffer was expressed by K_{out} . In our simulations, the normalized throughput at $t = mT$ was obtained as

$$R(m) = \frac{N(m)}{mT}, \quad m = 1, 2, \dots \quad (17)$$

where $N(m)$ represents the number of packets successfully transmitted from node B to node C during the period from $t = 0$ to $t = mT$.

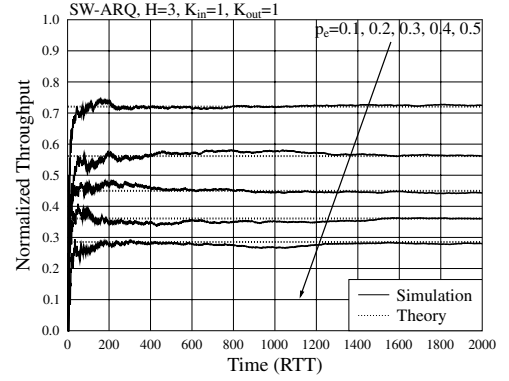


Fig. 2. Simulated time-variant throughput and numerically evaluated steady-state throughput for the 3ISO network coding system with the buffer size $K_{in} = K_{out} = 1$.

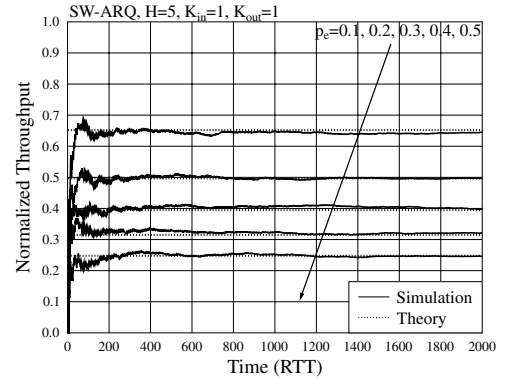


Fig. 3. Simulated time-variant throughput and numerically evaluated (theory) steady-state throughput for the 5ISO network coding system with the buffer size $K_{in} = K_{out} = 1$.

Figs. 2 and 3 depict the normalized throughput performance of the network coding system having $H = 3$ or 5 incoming links. Each of the buffers is assumed to have only $K_{in} = K_{out} = 1$ memory unit for storing maximum one packet. Furthermore, in these figures, the corresponding steady-state throughput evaluated from formula (16) is depicted for the sake of comparison. From the results of Figs. 2 and 3, we can have the following observations. First, the throughput starts from $R = 0$ at $t = 0$ and converges to the steady-state throughput as more packets are transmitted. The throughput obtained by simulations fluctuates around its corresponding steady-state throughput obtained from evaluation of (16), due to an insufficient number of packets transmitted. Second, the simulation results shown in Figs. 2 and 3 justify our analytical results obtained in Section III, which is effective for evaluation of the steady-state throughput of the general *HISO* network coding systems. Third, the normalized throughput decreases

significantly, as the packet error rate p_e increases, which requires more frequent re-transmissions. Finally, when comparing Fig. 2 with Fig. 3, we can find that, for a given p_e , the throughput in Fig. 2 is higher than that in Fig. 3. In other words, the throughput decreases, as the number of incoming links of H increases. The reason for this observation is obvious, since more incoming links requires in average more waiting time to prepare the packets for a new coded packet.

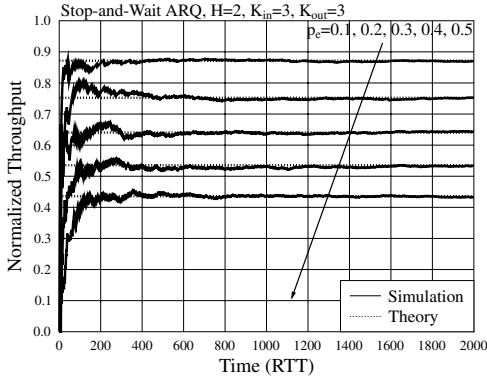


Fig. 4. Simulated time-variant throughput and numerically evaluated steady-state throughput for the 2ISO network coding system with the buffer size $K_{in} = K_{out} = 3$.

In Fig. 4 we illustrate the throughput performance of the network coding node B with two incoming links, where the memory size for each incoming and outgoing link is $K_{in} = K_{out} = 3$. The results of Fig. 4 show that the simulated dynamic throughput converges to the steady-state throughput evaluated by (16), which hence justifies our analytical results. The throughput of the network coding node decreases significantly, if the packet error rate increases.

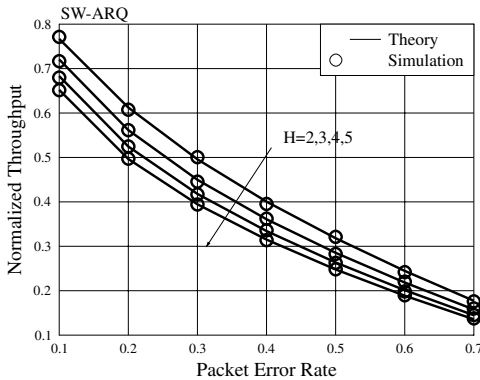


Fig. 5. Steady-state throughput versus packet error rate performance of the network coding system with $H = 2, 3, 4$ or 5 incoming links and the memory space $K_{in} = K_{out} = 1$.

In Fig. 5 we compare the normalized steady-state throughput against packet error rate for the network coding node B , when it has $H = 2, 3, 4$ or 5 incoming links. Explicitly, at a given packet error rate p_e , the normalized throughput decreases, when the network coding node has more incoming links. This is because the chance of forming a new coded packet and forwarding it to buffer b_0 becomes smaller, as the new coded packet is depended on correctly receiving more packets from the incoming links. Furthermore, 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.

Finally, in Fig. 6, we investigate the impact of the buffers allocated for the incoming and outgoing links of node B on the achievable throughput, when the total buffer retains constant of $K_{total} = 25$. From Fig. 6, we can observe that, given the total memory space, initially, increasing the buffers allocated to the incoming links yields

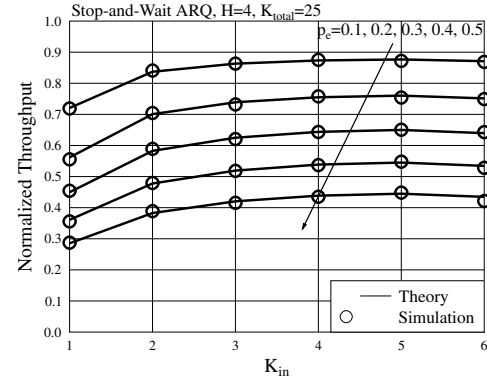


Fig. 6. Effect of buffers on the steady-state throughput performance of the network coding system with $H = 4$. The total buffer is $K_{total} = K_{in} \times 2 + K_{out} = 25$.

higher throughput. However, when the incoming links have sufficient buffers but the outgoing link has insufficient buffer, then, the achievable throughput is dominated by the outgoing link and decreases as the incoming links are allocated more buffer. Therefore, when given the total buffer of a coding node, there is an optimum buffer allocation to the incoming and outgoing links, which results in the highest throughput. As shown in Figs. 6, this argument becomes more declared, when the frame error rate p_e decreases or/and the coding node has more incoming links.

In **Conclusion**, we have investigated the steady-state throughput of general network coding nodes with on-node buffers, when the SW-ARQ error-control scheme is employed. Both analytical and simulation approaches have been considered. It can be shown that the simulation results justify our analytical expressions derived. The performance results show that the throughput of a network coding node decreases, as the number of incoming links increases. Therefore, in a network coding system, the coding nodes may form the bottleneck for information delivery. However, the throughput performance of network coding nodes may be improved by increasing the buffers of the links involved. Furthermore, given the total buffer of a network coding node, there exists an optimum buffer allocation, which results in the highest throughput.

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 Conf. on Commun., Cont., and Comput.*, 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. Inter. Symp. on Infor. Theory '05*, 4–9 Sept. 2005, pp. 1848–1852.
- [5] C. Fragouli, D. Lun, M. Medard, and P. Pakzad, "On feedback for network coding," in *Proc. 41st Annual Conf. on Infor. Sci. and Systems 2007*, 14–16 Mar. 2007, pp. 248–252.
- [6] J. Kumar Sundararajan, D. Shah, and M. Medard, "Arq for network coding," in *Proc. IEEE Inter. Symp. on Infor. Theory 2008*, 6–11 July 2008, pp. 1651–1655.
- [7] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Hybrid arq-random network coding for wireless media streaming," in *Proc. 2nd Inter. Conf. on Commun. and Elec. 2008*, 4–6 June 2008, pp. 115–120.
- [8] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Annual Allerton Conf. on Commun., Cont., and Comput.*, 2003.
- [9] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 2004.
- [10] 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.
- [11] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, 4th ed. Academic Press, 2009.
- [12] R. A. Horn, *Matrix Analysis*. Cambridge University Press, 1990.
- [13] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Upper Saddle River, New Jersey: Prentice Hall, 1992.