

Throughput and Latency of Acyclic Erasure Networks with Feedback in a Finite Buffer Regime

Nima Torabkhani[†], Badri N. Vellambi^{†‡} Faramarz Fekri[†]

[†] School of Electrical and Computer Engineering, Georgia Institute of Technology

^{†‡} Institute for Telecommunications Research, University of South Australia

E-mail: {nima, fekri}@ece.gatech.edu, badri.vellambi@unisa.edu.au

Abstract—The exact Markov modeling analysis of erasure networks with finite buffers is an extremely hard problem due to the large number of states in the system. In such networks, packets are lost due to either link erasures or blocking by the full buffers. In this paper, we propose a novel method that iteratively estimates the performance parameters of the network and more importantly reduces the computational complexity compared to the exact analysis. This is the first work that analytically studies the effect of finite memory on the throughput and latency in general wired acyclic networks with erasure links. As a case study, a random packet routing scheme with ideal feedback on the links is used. The proposed framework yields a fairly accurate estimate of the probability distribution of buffer occupancies at the intermediate nodes using which we can not only identify the congested and starving nodes but also obtain analytical expressions for throughput and average delay of a packet in the network. The theoretical framework presented here can be applied to many wired networks, from Internet to more futuristic applications such as networks-on-chip under various communication and network coding scenarios.

I. INTRODUCTION

In networks, packets may have to be relayed through a series of intermediate nodes where each may receive packets via many other data streams as well. Hence, the packets may have to be stored at intermediate nodes for transmission at a later time. For infinite buffer case, the intermediate nodes need not have to drop the arriving packets. However, often times, buffers are limited in size. Although a large buffer size is usually affordable and preferred to minimize packet drops, large buffers have an adverse effect on the packet delay. Additionally, as second-order effects, using larger buffer sizes at intermediate nodes would have practical problems such as on-chip board space and increased memory-access latency.

The problem of buffer sizing and congestion control is of paramount interest to router design engineers. Typical routers today route several tens of gigabits of data each second [1]. Realistic studies have shown that, at times, Internet routers handle about ten thousand independent streams/flows of data packets. With a reasonable buffer size of few Gigabytes of data, each stream can only be allocated a few tens of data packets. Therefore, at times when long parallel flows congest a router, the effects of such a small buffer space per flow come to play. Though we are motivated partly by such concerns, our work is far from modeling realistic scenarios. This work modestly aims at providing a theoretical framework to understand the fundamental limits of single information

flow in finite-buffer wired networks and investigate the trade-offs between throughput, average packet delay and buffer size.

The problem of computing capacity and designing efficient coding schemes for lossy networks has been widely studied [2]–[4]. However, the study of capacity of networks with finite buffer sizes has been limited. This can be attributed solely to the fact that finite buffer systems are analytically harder to track in general. In [5], [6], it was shown that min-cut capacity cannot be achieved due to the limited buffer constraint in a line network. Here, we wish to establish a framework to investigate the same for general wired networks.

Recently, [3] outlined various coding strategies for achieving capacity in line erasure networks. However, it only considered infinite buffer in its study. Later, [7] considered the limitations posed by finite memory in a simple two-hop wireline network. Inspired by this work, in [5], [6] authors investigated the information-theoretic capacity of multi-hop wireline networks with varying buffer constraints at each node. However, they only considered upper and lower bounds for the throughput. Further, the problem of finding the packet delay has been visited in the queueing theory literature on the behavior of open tandem queues that are analogous to line networks [8], [9]. However, this view is not applicable directly in general network topologies.

We consider wired acyclic erasure networks with ideal feedback on the links when a directed random packet routing scheme is used. Although the previous works provide some insight into the performance analysis of networks, they are limited to either infinite-buffer cases or a simple two-hop line network with limited memory. Moreover, the interplay of the throughput and latency is not considered in a small buffer regime. Our approach employs a discrete-time model to derive estimates for the buffer occupancy distribution at intermediate nodes. We then obtain analytical expressions for throughput and average packet delay in terms of the estimated buffer occupancy distributions.

The motivation behind this work is twofold: From a practical point of view, it analyzes the performance parameters of the random routing protocol in wired directed networks which is more of an interest to the network community. On the other hand, from a theoretical point of view, our work develops a framework that not only can be adapted to estimate the performance parameters of various communication schemes such as random linear coding but also can provide insight

into the interplay among the buffer size, throughput and delay in general. The later is impossible to be exactly analyzed due to the exponential growth in the number of states and the complexity of the equations [5] but our work develops an approximation to it. Further, the work establishes a lower bound on the information theoretic capacity of finite buffer networks.

This paper is organized as follows. First, we present a formal definition of the problem and the network model in Section II. Next, we investigate the tools and steps for finite-buffer analysis in Section III. We then obtain expressions for throughput and average delay in Section IV. Finally, Section V presents our analytical results compared to the simulations.

II. PROBLEM STATEMENT AND NETWORK MODEL

Throughout this work, we model the network by an acyclic directed graph $\vec{G}(V, \vec{E})$, where packets can be transmitted over a link $\vec{e} = (u, v)$ only from the node u to v . The system is analyzed using a discrete-time model, where each node can transmit at most a single packet over a link in an epoch. The links are assumed to be unidirectional, memoryless and lossy, i.e., packets transmitted on a link $\vec{e} = (u, v) \in \vec{E}$ are lost randomly with a probability of $\varepsilon_{\vec{e}} = \varepsilon_{(u,v)}$. Note that the erasures are due to the quality of links (e.g., noise, interference) and do not represent packet losses due to finite buffers. Clearly, erased packets are disregarded. Moreover, the packet losses on different links are assumed to be independent. Each node $v \in V$ has a buffer size of m_v packets with each packet having a fixed size. Source and destination pairs are assumed to have sufficient memory to store any data packets. The unicast information theoretic throughput between a pair of nodes is defined to be the transmission rate of information packets (in packets per epoch) between them. The delay of a packet is also defined as the time taken from the instant when the source starts transmitting a packet to the instant when the destination receives it. Note that the source node can generate innovative packets during each epoch.

Throughout this paper, node s and node d represent the source and destination nodes respectively. Also, for any $x \in [0, 1]$, $\bar{x} \triangleq 1 - x$.

III. UNDERSTANDING FINITE-BUFFER ANALYSIS

Here, we study the tools and steps that enable our framework for analyzing finite-buffer wired erasure networks.

A. Communication Scheme and Buffer States

Thoroughly investigated in [5] for the exact analysis of a finite-buffer line network, the problem of identifying the throughput capacity is equivalent to the problem of finding the buffer occupancy distribution of the intermediate nodes as a result of ergodicity of the corresponding Markov chain. Hence, in order to approach our problem properly it is necessary to clearly define the buffer states in a manner that first of all, the buffer states construct an irreducible ergodic Markov chain and second the steady state distribution of the buffer states could be helpful to obtain expressions for the performance parameters of the network. Thus, a proper definition for the

buffer states cannot be proposed unless the communication scheme is known. For example, since Random Linear Coding (RLC) is used as the communication scheme in [5], the states are defined as the number of packets stored in the buffer of a node whose information has not yet been conveyed to the next-hop node. Analyzing RLC in a general network (which achieves the capacity of finite buffer regime) is very involved and is the subject of the future work. Instead, as a first step towards estimating the performance parameters we consider a directed random routing scheme for packets together with lossless zero-delay feedback on the links.

To be more precise, the nodes operate using the following rules, one after another.

1. At each epoch, every node u selects a random ordering of the outgoing edges and transmits the packets it houses one by one. If the packet is successfully received and stored at a neighbor, u deletes the packet from its buffer and transmits the next packet (if any) on the next edge in the selected order. Else, it tries to transmit the same packet on the next (in the selected order) outgoing edge. This process is continued until all packets are transmitted or a transmission is attempted on each link. Therefore, a node with d_o outgoing links transmits at most d_o packets per epoch.
2. After the transmission attempts are made, the node attempts to accept the arriving packets. If more packets are received than it can store, it selects a random subset of the arriving packets whose size equals the amount of space available and stores the selected packets. Consistent with step 1, appropriate acknowledgment messages are then sent.

Note that under such a mode of operation, at any epoch when two nodes receive two packets from a particular node, the received packets have no common information. Equivalently, no replication is performed at any node and it is possible to define a concept of state or occupancy for each node individually. The buffer state of a node is simply defined to be the number of packets it stores. It can be seen that this concept of occupancy follows a Markov chain behavior and can be studied thus.

B. Approximate Markov Chain for an intermediate Node

Consider a node $u \in V$ in a network $\vec{G}(V, \vec{E})$ with d_i incoming and d_o outgoing edges and a buffer size of m_u as depicted in Fig. 1. Let the nodes that can send packets

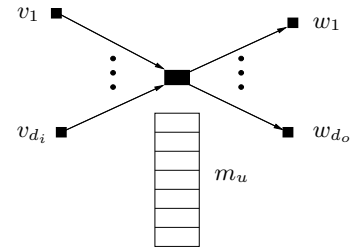


Fig. 1. A Node in a general wired network.

to u be denoted by $\mathcal{N}^+(u) \triangleq \{v_i, \dots, v_{d_i}\}$. Similarly,

let the nodes to which u can send packets be denoted by $\mathcal{N}^-(u) \triangleq \{w_i, \dots, w_{d_o}\}$. We assume that the following assumptions hold in the network regarding the arrival and departure processes.

1. For each $k = 1, \dots, d_i$, suppose that the packets arrive on (v_k, u) in a memoryless fashion with a rate of λ_k packets/epoch *i.e.*, with inter-arrival times having a geometric distribution with mean $\frac{1}{\lambda_k}$. Also, the processes on different incoming links are statistically independent.
2. At any instant, for every $k = 1, \dots, d_o$, a packet is sent on (u, w_k) it is successfully received and stored at w_k with a probability ω_k independent of the past and future events on the edge.

Note that this is hypothetical since in any realistic model of a network, the probability that a packet is successfully transmitted and stored at the next hop depends not only on the channel conditions, but also state of the next-hop node. Since the state of the next-hop node has dependence on its past, the probability of successful receipt can also be expected to have a dependence on its past. In fact this mode of node operation can be replaced by any other scheme that fits into the Markovian set-up of the assumptions above.

At any instant, the number of packets arriving can range from 0 up to d_i and the number of packets departing can range from 0 to d_o . Hence, at each epoch, the state n_u can change to any other in the set $\{n_u - d_o, \dots, n_u + d_i\} \cap \{0, \dots, m_u\}$. At any epoch, the probability a_k with which k packets arrive and the probability e_k with which k packets depart are given by

$$A(x) = \sum_{k=0}^{d_i} a_k x^k = \prod_{j=1}^{d_i} (\bar{\lambda}_j + \lambda_j x) \quad (1)$$

$$E(x) = \sum_{k=0}^{d_o} e_k x^k = \prod_{j=1}^{d_o} (\bar{\omega}_j + \omega_j x). \quad (2)$$

The dynamics of the number of packets stored at u at the l^{th} epoch is a Markov chain that is similar to the one depicted in Fig. 2.

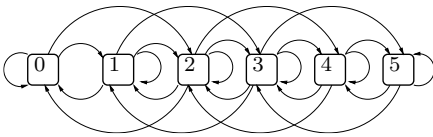


Fig. 2. The dynamics of a node u with $m_u = 5$ and $d_i = d_o = 2$.

For all input parameters, the Markov chain can be shown to be aperiodic, irreducible and ergodic. Therefore, it possesses a unique steady-state distribution. Letting $\Lambda = (\lambda_1, \dots, \lambda_{d_i})$ to denote the vector of arrival rates and $\Omega = (\omega_1, \dots, \omega_{d_o})$ to denote the vector of departure rates, the unique steady-state distribution $\vartheta(\cdot, \Lambda, \Omega, m_u)$ for the chain can be computed using a pair of probability transition matrices T_E and T_A ¹ that correspond to the transitions between states that are effected

¹For notational consistency, we can extend $e_k = 0$ for $k > d_o$ and $a_k = 0$ for $k > d_i$. Also, for notational convenience, we use $\vartheta(\cdot)$ as a short-hand $\vartheta(\cdot, \Lambda, \Omega, m_u)$.

by the departure and arrival of packets, respectively. Note that ϑ is the steady-state distribution after the arriving packets are processed. These transition matrices are defined as follows.

$$T_E = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \sum_{k=1}^{d_i} e_k & e_0 & 0 & \cdots & 0 & 0 \\ \sum_{k=2}^{d_i} e_k & e_1 & e_0 & \cdots & 0 & 0 \\ \sum_{k=3}^{d_i} e_k & e_2 & e_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \sum_{k=m_u}^{d_i} e_k & e_{m_u-1} & e_{m_u-2} & \cdots & e_1 & e_0 \end{bmatrix} \quad (3)$$

$$T_A = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots & a_{m_u-1} & \sum_{k=m_u}^{d_i} a_k \\ 0 & a_0 & a_1 & a_2 & \cdots & a_{m_u-2} & \sum_{k=m_u-1}^{d_i} a_k \\ 0 & 0 & a_0 & a_1 & \cdots & a_{m_u-3} & \sum_{k=m_u-2}^{d_i} a_k \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_0 & \sum_{k=1}^{d_i} a_k \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \quad (4)$$

Note that, the i, j^{th} entry in T_E corresponds to the transition of the occupancy from $i-1$ to $j-1$ with the departure of $i-j$ packets. Similarly, the i, j^{th} entry in T_A corresponds to the transition from $i-1$ to $j-1$ with the arrival of $i-j$ packets. The actual transition matrix for the Markov chain is then seen to be $T_E T_A$. The steady-state distribution ϑ of the occupancy just after the arriving packets are accepted and the steady-state distribution ϑ^\dagger of the occupancy just after the packets have been sent but before arriving packets are accommodated are given by

$$\vartheta T_E T_A = \vartheta \quad \text{and} \quad \vartheta^\dagger T_A T_E = \vartheta^\dagger. \quad (5)$$

However, these two steady-state distributions are related by $\vartheta^\dagger = \vartheta T_E$ and $\vartheta = \vartheta^\dagger T_A$. To evaluate the rate of information on the link (u, w_i) , one must investigate the rule for packet departure. If at an epoch, more packets are stored than the number of links that allow successful transmission, then each link conveys a packet of information to its neighbors. However, if the occupancy n_u at an epoch l is smaller than the number h of outgoing links that allow for transmission, each link can be assumed to equally receive $\frac{n_u}{h}$ packets on the average – a consequence of the random selection of ordering for outgoing links. Then, the time average of the information rate on the edge (u, w_i) can be seen as

$$I(\{(u, w_i)\}, \Lambda, \Omega, m_u) = \sum_{i \in H} \left(\prod_{k \in H} \omega_k \right) \times \quad (6)$$

$$\left(\prod_{k' \in H^c} \bar{\omega}_{k'} \right) \left(\sum_{j \geq |H|} \vartheta(j) + \sum_{j < |H|} \frac{j}{|H|} \vartheta(j) \right).$$

In a similar argument, we notice that some of the arriving packets get randomly blocked if all the arriving packets cannot be stored. We can evaluate the probability with which a packet arriving on the edge (v_i, u) is blocked from

$$p_b(\{(v_i, u)\}, \Lambda, \Omega, m_u) = \sum_{i \in H} \left(\prod_{k \in H \setminus \{i\}} \lambda_k \right) \times \quad (7)$$

$$\left(\prod_{k' \in H^c} \bar{\lambda}_{k'} \right) \left(\sum_{m_u - j < |H|} \frac{|H| - m_u + j}{|H|} \vartheta^\dagger(j) \right).$$

C. Iterative Estimation of the Buffer Occupancies

In this section, we discuss our iterative estimation technique in details based on the approximate Markov chain model introduced in section III-B. Considering that blocking will introduce dependence of the packet incoming/outgoing process over each edge on its past, in order to use the results of Section III-B, we have to make certain simplifying assumptions on the blocking phenomenon. We model the blocking on every edge $\vec{e} = (u, v)$ of the network as follows.

- Every packet that arrives at v successfully (without getting erased) is blocked in a memoryless fashion with probability q_{uv} . Also, at any epoch, the blocking of packets on any subset of incoming edges of v is assumed to be independent of one another.

Under the above assumption, the blocking process and hence the departure process on every link of the network is modeled as a memoryless process. Since each packet arriving on an edge $\vec{e} = (u, v)$ is blocked with a probability of q_{uv} , a packet arriving on \vec{e} is accepted only if both the channel allows the packet and the node accepts it. Therefore, the effective departure rate on the edge (u, v) is seen to be $\bar{\varepsilon}_{uv}\bar{q}_{uv}$. Assuming that the node operates in the mode described in Section III-A, we can use (6) and (7) to identify both the rate of information flow and the blocking probabilities on every edge of the network. Thus, the problem reduces to finding a solution $(\varrho_{uv}, q_{uv})_{(u,v) \in \vec{E}}$ that satisfies the following system of non-linear equations for each $(u, v) \in \vec{E}$.

$$\varrho_{uv} = \begin{cases} \frac{\bar{\varepsilon}_{uv}}{I(\{(u,v)\}, (\varrho_{wu})_{w \in \mathcal{N}^+(u)}, (\bar{\varepsilon}_{uu'}\bar{q}_{uu'})_{u' \in \mathcal{N}^-(u)}, m_u)} \bar{q}_{uv}} & u = s \\ 0 & u \neq s \end{cases},$$

$$q_{uv} = \begin{cases} p_b(\{(u,v)\}; (\varrho_{wv})_{w \in \mathcal{N}^+(v)}, (\bar{\varepsilon}_{vv'}\bar{q}_{vv'})_{v' \in \mathcal{N}^-(v)}, m_v) & v \neq d \\ 0 & v = d \end{cases}$$

Note that in the above equations ϱ_{uv} represents the fraction of time at which packets will be delivered to v . However, the actual rate of information flow is equal to $\rho_{uv} = \bar{q}_{uv}\varrho_{uv}$.

Since the above set of equations are an approximation to the actual dynamics, it is not clear as to whether there even exists a solution to the above system. However, the proof of existence and uniqueness of the solution is detailed in [10] for the case of line networks.

Finally, the solution to the system of equations can be found by identifying the limit of the sequence defined by the following iterative procedure².

1. Set $i = 1$ and for each edge $(u, v) \in \vec{E}$, set $q_{uv} = 0$ and $\varrho_{uv}^{(1)} = \begin{cases} 0 & u \neq s \\ \bar{\varepsilon}_{uv} & u = s \end{cases}$.
2. Compute $\varrho_{uv}^{(i+1)}, q_{uv}^{(i+1)}$ by using $\varrho_{uv}^{(i)}, q_{uv}^{(i)}$ on the right-hand side of the above system of nonlinear equations and increment i by 1.
3. If $i < L + 1$, perform step 2.

²In practice, the number of iterations L which suffice to converge to the solution within reasonable accuracy depends on the structure of the network. Alternatively, one may use $|\vartheta^{(i+1)} - \vartheta^{(i)}| + |\vartheta^{\dagger(i+1)} - \vartheta^{\dagger(i)}| < \epsilon$ for the convergence criteria.

IV. ESTIMATION OF THE THROUGHPUT AND AVERAGE PACKET DELAY

In this section, we exploit the results of the iterative estimation method for buffer occupancy distributions and obtain analytical expressions for throughput and average delay.

Since the routing scheme is such that information is not replicated at any node, the estimate of the total information that arrives at the destination is the sum total of the information rate arriving on each of its incoming edges. Hence,

$$\hat{\mathcal{C}}(s, d, \vec{G}) = \sum_{v \in \mathcal{N}^+(d)} \varrho_{vd}^* (1 - q_{vd}^*) = \sum_{v \in \mathcal{N}^+(d)} \varrho_{vd}^*, \quad (8)$$

where we let $(\varrho_{uv}^*, q_{uv}^*)$ to be either the component-wise limit of the sequence $\{\varrho_{uv}^{(i)}, q_{uv}^{(i)}\}_{i \in \mathbb{N}}$ when $L = \infty$, or $(\varrho_{uv}^{(L)}, q_{uv}^{(L)})$ when $L < \infty$. Additionally, by the conservation of information flow, the above estimate can be obtained by computing the rate of flow of information through any cut F using the following.

$$\hat{\mathcal{C}}(s, d, \vec{G}) = \sum_{uv \in F} \varrho_{uv}^* (1 - q_{uv}^*). \quad (9)$$

As it is defined in Section III-A, the operation scheme is chosen to have feedback on all the links and we treat packets in a First-Come First-Serve (FCFS) fashion at the buffers. Also, the absence of directed cycles allows us to assign an order v_1, v_2, \dots, v_n to all the nodes of the network in a manner that we have $i < j$ for every link $(v_i, v_j) \in \vec{E}$.

In order to estimate the average delay that a packet experiences in the network, one can proceed in a recursive fashion. The average delay that an arriving packet (at time l) at node $u \in V$ experiences depends on the buffer occupancy of the node u and its outgoing links. For example, suppose at epoch l (packet arrival time), node u has already k packets where $k \leq m_u - 1$. Then, the arriving packet has to wait for the first k packets to leave node u before it can be transmitted. We define $\mathcal{D}_u(k)$ as the average time it takes from the instant that node u receives a packet given that it has already stored k packets, until the time that the destination node receives that packet. We compute the average delay function $\mathcal{D}_u(\cdot)$ for all the intermediate nodes $u \in V$ using the following proposition.

Proposition 1: Let $r_{uv} = \bar{\varepsilon}_{uv}\bar{q}_{uv}$ be the average packet transfer rate on link $(u, v) \in \vec{E}$ and r_u^- be the sum of the rates on all outgoing edges (i.e., $r_u^- = \sum_{v \in \mathcal{N}^-(u)} r_{uv}$). Also, let $\pi_v(j)$ ($j = 0, 1, \dots, m_v - 1$) be the steady state probability of the buffer of node $v \in V$ storing already j packets right before a new packet arrives and is stored in the buffer. For every intermediate node $u \in V$, given the average delay functions of all its next-hop neighbors ($\mathcal{D}_v(j)$ for all $v \in \mathcal{N}^-(u)$ and $j = 0, 1, \dots, m_v - 1$), $\mathcal{D}_u(\cdot)$ can be obtained by

$$\mathcal{D}_u(k) = \frac{k+1}{r_u^-} + \sum_{w \in \mathcal{N}^-(u)} \frac{r_{uw}}{r_u^-} \left(\sum_{j=0}^{m_w-1} \pi_w(j) \mathcal{D}_w(j) \right) \quad (10)$$

for $k = 0, 1, \dots, m_u - 1$.

Proof: Due to space limitation, we just provide a brief sketch of the proof. Equation (10) is consisted of two terms:

1. The first term represents the average time it takes for a total of $k + 1$ packets (counting our selected packet) to leave node u successfully.
2. The second term relates to the average delay due to the rest of the network. The probability of conveying a packet from node u to node v can be estimated by $\frac{r_{uv}}{r_u}$. An arriving packet at node v finds its buffer already occupied by j packets with probability $\pi_v(j)$. Thus, the packet will experience an average delay of $\mathcal{D}_v(j)$ from this node to the destination. Hence, the average packet delay from node v to the destination is equal to $\sum_{j=0}^{m_v-1} \pi_v(j) \mathcal{D}_v(j)$.

It is easy to see that $\pi_v(j)$ can be calculated using

$$\pi_v(j) = \begin{cases} \frac{\vartheta_v^j(j)}{1-\vartheta_v^j(m_v)} & j = 1, 2, \dots, m_v - 1 \\ 0 & j = m_v \end{cases} \quad (11)$$

To obtain the average packet delay from the source to the destination, the average delay function $\mathcal{D}_u(\cdot)$ is computed for all the nodes in the reverse order³ (i.e., $\{v_n, \dots, v_2, v_1\}$). Then, the total average packet delay ($\mathcal{D}_s(0)$) is computed by applying Proposition 1 to the source node.

V. SIMULATION RESULTS

In this section, we present the results of actual network simulations in comparison with our analysis and will show that our framework gives accurate estimates of buffer occupancy distributions as well as throughput and average delay.

We consider the network shown in Fig. 3 to compare the results of the simulation and inferences. In this network, all

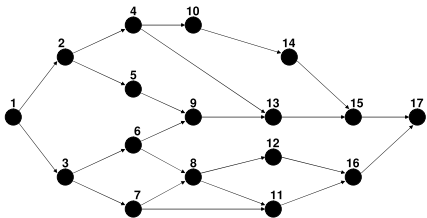


Fig. 3. A general wired acyclic directed network chosen for simulation).

the edges have $\varepsilon = 0.5$ (erasure probability) except the edges $\{(1, 2), (1, 3), (15, 17), (16, 17)\}$ for which $\varepsilon = 0.05$. All the intermediate nodes are assumed to have the same buffer size. In order to measure the exact performance parameters of this network, millions of packets are sent from the source (Node 1) to the destination (Node 17). Fig. 4 presents a comparison between the actual buffer occupancy distributions and our iterative estimates for four of the nodes in the network of Fig. 3. Also, Fig. 5 presents the variations of the actual throughput and average packet delay and our analytical results versus the buffer size. Note that, the throughput is presented in *packets/epoch* and average packet delay is presented in *epochs*. As it can be observed, our estimation is very close to the actual simulation results.

³Note that we have $\mathcal{D}_d(k) = 0$ for every k where d denotes the destination node.

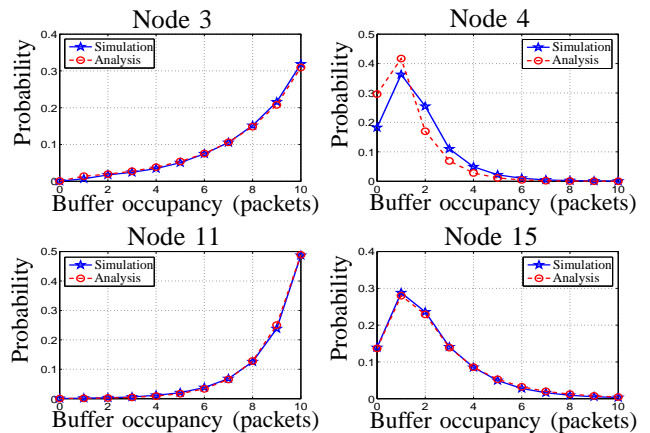


Fig. 4. buffer occupancy distributions for nodes 3, 4, 11 and 15.

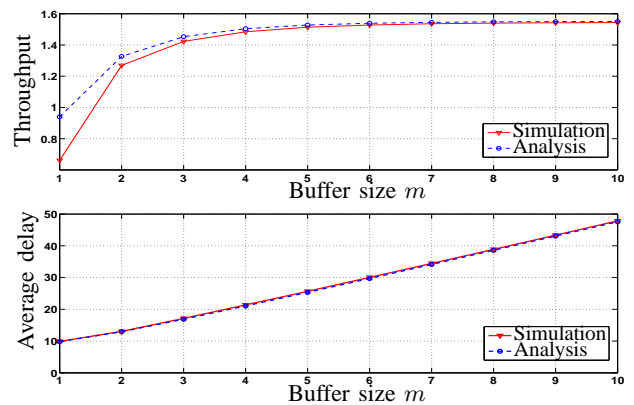


Fig. 5. Performance parameters of the network for different buffer sizes.

REFERENCES

- [1] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 281–292, 2004.
- [2] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 789–804, 2006.
- [3] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," *IEEE Int'l Symposium on Information Theory*, Sept. 2005.
- [4] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Inform. Theory*, vol. 49, pp. 371–381, February 2003.
- [5] B. N. Vellambi, N. Rahnavard, and F. Fekri, "The effect of finite memory on throughput of wireline packet networks," in *Proc. of Information Theory Workshop (ITW 2007), Lake Tahoe, CA*, Sept. 2007.
- [6] B. N. Vellambi and F. Fekri, "On the throughput of acyclic wired packet networks with finite buffers," in *2008 Information Theory and Applications Workshop (ITA 2008), San Diego, CA*, Jan. 2008.
- [7] D. S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter, "An analysis of finite-memory random linear coding on packet streams," in *Proc. of the 2nd Workshop on Network Coding, Theory, and Applications (NetCod 2006), Boston, MA, April 3-7, 2006*.
- [8] T. Altiok, "Approximate analysis of queues in series with phase-type service times and blocking," *Operations Research*, vol. 37, pp. 301–310, July 1989.
- [9] T. Altiok, "Approximate analysis of exponential tandem queues with blocking," *European Journal of Operational Research*, vol. 11, no. 4, pp. 390–398, 1982.
- [10] B. N. Vellambi, N. Torabkhani, and F. Fekri, "Throughput and latency in finite buffer line networks," *accepted in IEEE Transaction on Information Theory*, June 2010.