Looking at Large Networks: Coding vs. Queueing

Sandeep Bhadra, Sanjay Shakkottai
Wireless Networking and Communication Group
Department of ECE
University of Texas at Austin, Austin, TX
E-mail : {bhadra, shakkott}@ece.utexas.edu

***Abstract*— Traditionally, network buffer resources have been used at routers to queue transient packets to prevent packet drops. In contrast, we propose a scheme for large multi-hop networks where intermediate routers have *no buffers* for queueing transient packets. In the proposed scheme, network storage resources (memory) are used only at source and destination nodes to encode/decode packets using random linear coding over time.**

Our scheme utilizes the observation that for large networks with many flows through each router, if packet loss occurs in a flow path, it will very likely occur only at only one link in the path. Unfortunately, the location of this congested link varies with time, hence, preventing static buffer allocation strategies from exploiting this observation. We propose network coding as a means of "sharing" memory across links along a flow path. We call this *spatial buffer multiplexing* – where buffering and coding implemented at the source compensates for packet loss at any downstream bufferless link.

In this paper, we consider large spatial multi-hop networks with N nodes and $\Theta(N)$ flows, where the number of flows through each link scales as $\Omega(N^\alpha)$ for some $\alpha \in (0,1)$. Using many-sources large deviations analysis, we show that to obtain comparable packet drop probabilities (QoS), spatial buffer multiplexing provides an order-wise buffer gain of $\Omega(N^\alpha)$ per node over traditional queueing.

I. Introduction

Network coding at intermediate routers in a network (as opposed to switching/routing) was originally proposed with a view of increasing end-to-end throughput in networks [1] and [2]. Network codes have been shown to be throughput optimal (network-wide capacity achieving) for a multicast network by Cai et. al. Furthermore, network coding via Random Linear Coding (RLC) improves network reliability and simplifies network management [3], as well as allows exploiting correlation in sensor data to improve network efficiency [4]. Recent formulations of convex optimization problems [5], [6] to characterize the sum-cost of flows through a network using RLC pose significant reduction of network-wide sum-cost for coding as opposed to routing.

Random Linear Codes applied at intermediate routers effectively spread the information from one flow across multiple flows and hence work well as an error/erasure control scheme. This spreading of information makes RLCs attractive in cases where packet drops or losses are likely to occur, such as in data dissemination over large peer-to-peer networks [7], [8], [9]. Recently, Avalanche [8] has been proposed as an alternative to BitTorrent [10] by using network codes for P2P data dissemination. Further, network codes have been proposed as a means of distributed information dispersal and recovery in large ad-hoc networks via a rumour-spreading(epidemic) model [7], [9].

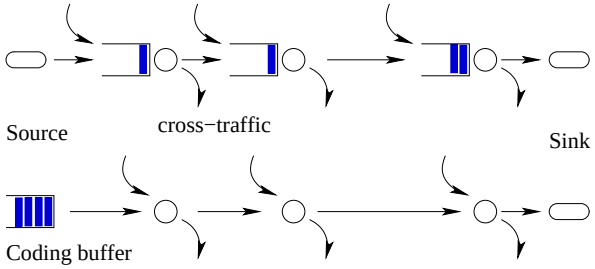

Fig. 1. Buffering at the source versus buffering at nodes: By using network coding, a form of spatial multiplexing gain can be achieved whereby the small buffers at the nodes can be shared across multiple nodes.

The common underlying theme in much of the above work has been that network codes, and specifically RLC's, allow spatial (across the network) stochastic multiplexing across different flows and this feature can be utilized in improving reliability in large networks. Recently however, Lun, Medard and Effros [11], [12] exploit network codes for a capacity-approaching scheme for unicasts or multicasts over large networks. In their scheme, routers perform RLC over packets from different flows as well as over packets transmitted in previous time-slots. Further, for the case of Poisson traffic with i.i.d. losses at intermediate router queues (modeled as M/M/1 queues), they derive the error exponent in the large-delay regime. This is analogous to the use of block codes or convolutional codes for error control in the PHY that spread the information across multiple bits in a block or neighbourhood around each bit. On a related note, error exponents of codes over networks have also been studied by Luby et. al. [13].

The insight from [11]that packets dropped in a particular time-slots can be recovered from RLCs containing the dropped packets in future time-slots motivates us to consider the following questions:

- Can we eliminate buffering at intermediate nodes in favour of coding only at the ends? We consider the scenario where intermediate routers perform no RLCs or buffering, but merely drop packets if the link capacity is exceeded.
- Further, in the event of finite delays, how does network coding at the ends compare with queueing in intermediate routers? Here, we wish to compare QoS parameters such as delay and end-to-end packet loss probability (reliability) with coding as opposed to queueing.

The main idea stems from the fact that in a very large

network with N nodes and $N/2$ unicasts from each source matched to its (randomly chosen) destination, each link in the network carries a large number of flows, say $n = \Omega(N^\alpha)$ for some $\alpha \in (0,1)$ [14], [15], [16], [17]. Naturally, to ensure that the per-flow capacity on each link/edge is an $\Theta(1)$ quantity[1], the aggregate link capacity must scale with n. Stability requirements also enforce the condition that the link capacity should be greater than the mean packet arrival rate at each link. Under these conditions, we have from Chernoff's bound that the probability a link overflows is roughly of the order of $\exp(-N^\alpha \epsilon_0)$ for some $\epsilon_0 > 0$. Assuming good mixing, the probability of a link overflowing anywhere along a path of length $\Omega(N^\beta)$ for some $\beta \in (0,1)$ is approximately $O(N^\beta \exp(N^\alpha))$ which is asymptotically close to $O(\exp(N^\alpha))$ for large N. This can be interpreted as follows – the probability that there is an overflow in a single link is of the same order as the probability that there is an overflow in a path containing a polynomial number of such links. In other words, *"if an overflow occurs in a path, it will very likely occur only at only one link in the path"*. Hence, instead of buffering at each link in the path, it should suffice to buffer only at one link – translating to huge savings in buffer required per-flow for large networks and better scalability in the design of large multi-hop networks. However, the link where the overflow occurs is a function of the sample path of the arrival processes and varies with time. This variation makes it impossible to effectively *multiplex buffers across links on a path for a single flow* using traditional static buffer allocation at each link. Note that this is very different from traditional buffer multiplexing where many flows incident at a single link share buffers across flows [18], [19], [20].

It is in this scenario that we propose network coding as a means of "sharing" memory across links along a flow path. We call this *spatial buffer multiplexing* – where buffering and coding implemented via a sliding window of packets at the source compensates for packet loss at any downstream bufferless link. In addition to the data packets, suppose that the source transmits an additional stream of low-priority packets each of which are independent, random linear combinations of the data packets transmitted over the past d units of time. In other words, each low-priority packet is simply a random weighted sum of all the data packets that were transmitted over the past d units of time. At each of the intermediate nodes in the network, during congestion (i.e., the number of data packets plus the number of coded packets exceeds the link capacity), some of these coded packets are preferentially dropped. In other words, nodes in the network employ a two-level priority scheduling, where data packets are transmitted with higher priority than coded packets. Note that if the total number of data packets arriving in a time-slot exceeds the link capacity, some data packets will be dropped as well. The decoder at the receiver can then recover the *lost data packets* if it receives a suitable number of *random linear coded packets* within an interval of time of d units.

We illustrate this in the context of a path in a network (see Figure 1), where a data flow passes through a sequence of nodes in the presence of cross traffic. In a conventional buffered network, each intermediate node needs packet buffers to temporarily store packets when bursts of data packets arrive. On the other hand, in the network coded case (with zero buffers at intermediate nodes), a coding buffer at the source needs to maintain a window of packets (over a time-interval of d units).

Spatial buffer multiplexing can result in significant gains in buffer requirements. Consider, for example, for a rectangular grid network with N nodes which are randomly partitioned into $N/2$ sources matched to $N/2$ destinations. The typical path contains $\Theta(\sqrt{N})$ links and each link carries on an average $\Theta(\sqrt{N})$ flows through it. With a buffer of size b for each flow at each intermediate router, the total number of buffers per-flow is $\Theta(\sqrt{N}b)$. Now, since there are $N/2$ flows, the total number of buffers required across the network is $\Theta(N\sqrt{N}b)$. In contrast, we will show that using network coding with RLCs of $d = \Theta(b)$ time-steps, each source-destination pair requires a (coding) buffer of size $\Theta(b)$ only and no buffers are required at the intermediate nodes. Hence, the total number of buffers required across the network is $\Theta(Nb)$. This comes as an average $\Theta(\sqrt{N})$ buffer-size gain over traditional queueing.

In this paper, we consider a large network with many nodes and many flows through each link(edge) in the network to compare alternate strategies. We employ large deviations based analysis [21], [22] to quantitatively demonstrate that the packet loss probabilities with these two strategies are orderwise similar in the exponent. Large deviations have been used to analyze packet-loss, delay and other QoS parameters in networks with large number of sources (many sources large deviations) [22], [19], [18] or with large buffers (large buffer large deviations) [23]. In the context of many sources large deviations, a rate function indicates that the probability that a QoS parameter is not met decreases *uniformly* in the exponent with the number of sources. Botvich and Duffield [18] show that the queue length Q^n at the head of a link exceeds the buffer size nb is given by the rate

$$\lim_{n\to\infty} \frac{1}{n} \log P(Q^n > nb) = -I(b). \tag{1}$$

Further, the authors show that for uncorrelated arrivals at the queue, $I(b) \approx \delta b + \nu$ for some $\delta > 0$, i.e. the rate function $I(b)$ is linear in b in the large b regime.

A. Main Contributions

In this paper we consider the comparison of buffering at each intermediate link along a path versus network coding at the source and decoding at the destination. We first consider the case of a single *bufferless* link with capacity nC packets per time-slot where n is the number of flows, with mean arrival rate $E[A^m]$ for the m-th flow, through this link, and $C > E[A^m]$, for all $m = 1, 2, \ldots n$, is the capacity per-flow for this edge. We assume that RLCs of packets in d previous time-slots are transmitted as a lower-priority auxiliary coded packet stream. In this context, we obtain the many sources large deviations rate function for packet loss across this edge as an increasing function of d. Subsequently, we generalize this result to the case of a path where the number of edges(links)

[1]We use Knuth's notation $O(n)$, $\Theta(n)$, $\Omega(n)$ to denote functions that scale slower than (upper bounded by), as fast as (upper and lower bounded by positive constants) and faster than (lower bounded by) n respectively.

is a polynomial in n_e, the number of flows through each edge e in the path.

A preliminary overview of large deviations is presented in Section II. Section III presents a detailed system model for the encoder and decoder, a quick overview of Random Linear Coding and describes the *proportional dropping* rule where, in the event of overflow, packets are dropped from flows in proportion to the size of each flow. We also state the conditions under which packets dropped in previous time-slots can be recovered with the aid of coded packets in subsequent time-slots.

Our main contributions are as follows:

(i) Since RLC couples the packet drop in one time-slot with the arrival rates in the past and future time-slots, deriving the exact expression for the probability of packet loss is difficult. In Section IV we upper bound the probability of packet loss over a link with n flows through it by $\exp(-nI_Y(0,d,\bar{B}))$ where $I_Y(0,d,\bar{B}) > 0$ is a increasing function in d. We further derive a lower bound to show that the above bound is orderwise tight in the exponent. Further, in Section VI we show that for i.i.d. Bernoulli arrivals, $I_Y(0,d,\bar{B}) = dK_1$ for some constant $K_1 > 0$. This implies that the probability of a packet loss decreases exponentially with n and d which compares with the many sources queueing result of Botvich and Duffield [18], Equation (1). We plot the packet loss probabilities with network coding in comparison with buffering and show that if the buffer required for coding is orderwise the same as the buffer for queueing, the same QoS (packet loss probability) can be obtained.

(ii) In Section V, we generalize the rate function to the case of a path with multiple links and for coding buffer of $d = \Theta(1)$. We derive an upper bound on the probability of packet drop that decays exponentially in n_Γ, the minimum number of flows through any edge along path Γ. We numerically show in Section VI that the rate function is asymptotically linear in d.
In large networks with N nodes where $n_\Gamma = \Omega(N^\alpha)$, $\alpha \in (0,1)$,(see Section V for networks with this property) we argue that for achieving comparable QoS, `(buffer per node with traditional queueing)/(buffer per node with network coding)` $= \Omega(N^\alpha)$. This orderwise buffer savings makes a case for the use of network coding for *spatial buffer multiplexing* in favour of queueing at intermediate routers for such networks.

Finally as a technical aside, we note that network-wide many-sources large deviations analysis with traditional buffering at intermediate nodes is very difficult due to the correlation of processes in links along a path. However, network coding allows sufficient decoupling that enables our analysis in the network-wide context.

II. Preliminaries

For a large network with many source-destination pairs, under fairly general topology assumptions, each link carries the load of multiple source-destination pairs. Assuming that the link capacities scale orderwise linearly with the number of flows through a link, so as to allow $\Theta(1)$ per-flow capacity at each link, we can quantify various QoS properties of the flows, such as packet drop probability and maximum delay, in terms of large deviations rate functions of the arrival and service processes at the link queues [18], [23], [22], [19].

For a sequence of i.i.d. random variables $X_1, X_2, \ldots$ where $E[X_i] = \hat{X}$, the Strong Law of Large Numbers states that the empirical mean $X^{(n)} = \frac{1}{n}\sum_{i=1}^{n} X_i \rightarrow \hat{X}$ *almost surely* in the limit as $n \rightarrow \infty$. In the pre-limit, for finite n, Chernoff's bound characterizes the rate of convergence of $X^{(n)}$ to the mean $\hat{X}$ as follows,

$$P(|X^{(n)} - \hat{X}| > \delta) \leq 2\exp\left[-n\sup_\theta(\delta\theta - \log M_X(\theta))\right]$$

where $M_X(\theta) = E[\exp(\theta(X_1 - \hat{X}))]$ is the log moment generating function of the zero mean process $X_i - \hat{X}$. Further [22] [21], it can also be shown that the above bound is tight. Thus, for any $\epsilon > 0$, there exists an n_ϵ such that for all $n > n_\epsilon$,

$$P(|X^{(n)} - \hat{X}| > \delta) \geq 2\exp\left[-n\sup_\theta(\delta\theta - \log M_X(\theta) + \epsilon)\right].$$

We can therefore state that the sequence of random variables $X^{(1)}, X^{(2)}, \ldots$, converges to $\hat{X}$ with a *large deviation property* with *rate function*

$$I(x) = \sup_\theta\{\theta x - \log M_X(\theta)\}.$$

The rate function $I(x) \geq 0$ since setting $\theta = 0$, $0.x - \log M_x(0) - 0$.

Thus the large deviations rate function gives an understanding of how fast a sequence of random variables converges to the typical value of the sequence as we consider increasingly large numbers of these variables. This analysis can be extended to the case a general sequence of random variables as follows. A sequence of random variables $Z_1, Z_2, \ldots$ is said to satisfy a large deviations principle with rate function $I_Z(\cdot)$ if for every Borel set A,

$$\begin{aligned} -\inf_{z\in A^0} I_Z(z) &\leq \liminf_{n\rightarrow\infty} \frac{1}{n}\log P(Z_n \in A) \\ &\leq \limsup_{n\rightarrow\infty} \frac{1}{n}\log P(Z_n \in A) \leq -\inf_{z\in\bar{A}} I_Z(z) \end{aligned} \quad (2)$$

where A^0 and $\bar{A}$ are the interior and closure of set A [22], [21].

In the following sections, we will study the sequence of random variables $f(X^{(1)}), f(X^{(2)}), \ldots$, where each $X^{(n)} \doteq \frac{1}{n}\sum_{m=1}^{n} A^m$ is the empirical average of n independent identically distributed (i.i.d.) random variables A^m, $m = 1, 2, \ldots, n$ and $f()$ is a continuous function. Note that in general, A^m can be either a scalar or a vector random variable.

III. System Model

A. Single source stream

Consider the simplest case of a single user stream over a single zero buffer link of constant capacity C without delays. We assume slotted time. Also, define the window $W_i \doteq \{t : i-d+1 \leq t \leq i\}$ of size d corresponding to the i-th time-slot. In time-slot i, the source (head of the link) generates a random

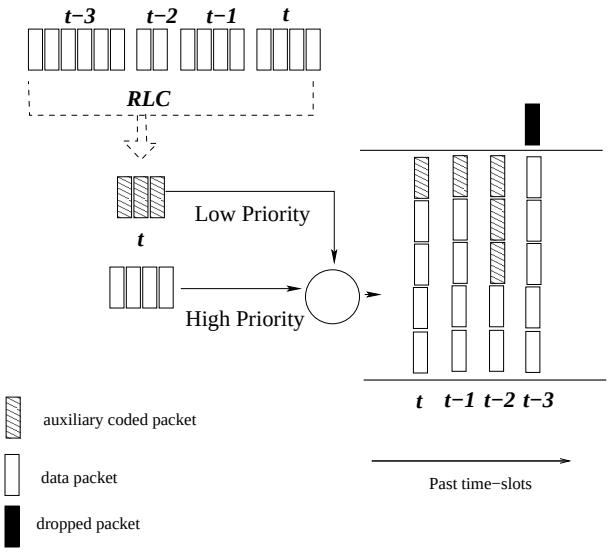

Fig. 2. Illustration of RLC across d time-slots for a particular source for $d = 4$: each small blank rectangular tile represents a data packet. RLC is performed over all the data packets in the previous $d = 4$ time-slots to generate $\bar{B} = 3$ auxiliary coded packets (shaded tiles) each time-slot. Data packets have higher priority in the link with capacity $C = 5$. The auxiliary coded packets have lower priority and are sent when there is spare capacity in the link. The dark tile represents the dropped packet at time-slot $t - 3$ when 6 packets were generated since the link capacity is only 5.

number of *data packets* $\{P_{i,j}\}$, $j = 1, 2, \ldots, A_i$ and transmits them across the link. Here we assume that the data packet arrival process $\{A_i\}$, $i = (-\infty, \infty)$ is a *stationary ergodic* random process taking values chosen from a set $\mathcal{A} \subseteq \mathbb{N}$, with mean strictly less than C.

Each packet $P_{i,j}$ can be assumed to be a vector of size s containing elements $P_{i,j}(m)$, $m = 1, 2, \ldots, s$ chosen from a finite field $\mathbb{F}_q$. In general therefore, each $P_{i,j} \in \mathbb{F}_q^s$. The source also generates a low-priority auxiliary data stream of B *coded packets* $\{P'_{i,j}\}$ by an RLC over all packets in the window W_i according to the rule:

$$P'_{i,j}(m) \doteq \sum_{t \in W_i} \sum_{k=1}^{A_t} \alpha_{t,k} P_{t,k}(m) \qquad (3)$$

for all $j = 1, 2, \ldots, B_i$ and all $m = 1, 2, \ldots, s$ where each $\alpha_{t,k}$ is a random element in $\mathbb{F}_q$ and all arithmetic is over $\mathbb{F}_q$. If $A_i + B_i > C$, priority is given to the data packets $P_{i,j}$ over the coded packets. The purpose of the coded packets is to help recover packets that were lost in any of the past d time-slots. In this sense, the auxiliary data may be thought of being generated by a random linear convolutional encoder with memory d at the source, see Figure 2. Note that the link constraint implies that the number of auxiliary data packets received by the destination (tail of the link) at time t is $\min(\bar{B}, (C - A_t)_+)$.

Denote the number of lost packets in time-slot i by L_i where $x_+ \doteq \max\{x, 0\}$. For the single source case, $L_i \doteq (A_i - C)_+$. When a packet is dropped at, without loss of generality, time-slot 0, the receiver attempts to recover the dropped packets by decoding the coded packets received in future time-slots by solving for the unknown values of $P_{i,j}$ from the set of equations in 3.

The destination receives the the coefficients of the linear equations, $\alpha_{t,k}$, corresponding to each coded packet as header bits within the packet. Alternately, since in most practical considerations, the coefficients $\alpha_{t,k}$ will be generated via a pseudo-random generator, it may be sufficient to initialize the pseudo-random generators at the source and destination to the same state at the beginning of the communication process via some form of handshaking. However, this would require the decoder at the receiver to know the exact number of packets generated in each time-slot so as to maintain both random-number generators at the same state. This information could be encapsulated as part of one or more of the data packets.

Each coded packet, and the corresponding coefficients $\alpha_{t,k}$ represent a linear equation over $P_{i,j}$. The information at the decoder may be represented as a set of linear equations in known and unknown variables. The known variables correspond to the data packets are directly received by the decoder. The unknown variables are the dropped packets. Hence, the decoder requires as many independent linear equations (coded packets) as the number of unknowns to be able to solve for this set of equations. Note that since the field $\mathbb{F}_q$ is finite, in general, two coded packets have a non-zero probability of being linearly dependent. This corresponds to the event where the matrix of coefficients is singular. In the rest of this work we will loosely refer to the set of linear equations as being *invertible* (uninvertible) if this matrix is not invertible (respectively, not invertible).

Since packets that are dropped can be recovered in a future time-slot, we make a distinction between dropping a packet and losing a packet as follows. L_i packets are said to be *dropped* at time-slot i if $A_i > C$. However, some of these dropped packets may be *recovered* by future coded packets. Hence, packets are said to be *lost* if they are dropped and cannot be recovered by solving for the linear equations formed by the coded packets. Observe that the encoding process couples the loss of a packet in one time-slot with losses in the past and the future. This cascading effect implies that a packet that is transmitted at time 0, may be decoded in the distant future (possibly after infinite delay) when the set of linear equations is solvable.

However nearly all practical applications require that all packets must be decoded within finite delay. This motivates an additional QoS condition requiring a packet to be decoded within d time-slots. Conversely, a dropped packet that is not decoded within d time-slots is considered *lost* by the decoder at the destination.

B. Many source streams

In general, a link in a large network transmits packets from a large number of sources. For the subsequent analysis we will assume that the link capacity scales in proportion to the number of sources transmitting over the link. The number of sources transmitting over a link depends, in general, on the total number of nodes N, the topology of the network and the number of simultaneous source-destination pairs transmitting. For simplicity, we will first deal with the abstraction of a link

with n source streams over a single bufferless link of capacity nC packets/time-slot in Section IV.

Each source S_m, $m = 1, 2, \ldots, n$ generates A_t^m packets $P_{t,j}^m$, $j = 1, 2, \ldots, A_t^m$ in time-slot t. A total of $(\sum_{m=1}^n A_t^m - nC)_+$ packets will be dropped in each slot t. However, the distribution of the dropped packets is a function of the dropping rule at the head of the link. We define the *proportional dropping* rule where

$$L_t^m \doteq \frac{A_t^m}{\sum_{m=1}^n A_t^m} \left(\sum_{m=1}^n A_t^m - nC \right)_+ \tag{4}$$

are dropped from the m-th stream at time t. We assume $L_t^m \doteq 0$ for $\sum_{m=1}^n A_t^m = 0$.

If $\sum_{m=1}^n A_t^m < C$, the residual capacity is split equally between coded packets from each source. Thus the number of coded packets from source S_1 received at the tail of the link is

$$B_t^m \doteq \min\left(\bar{B}, \left(C - \frac{1}{n} \sum_{m=1}^n A_t^m \right)_+ \right). \tag{5}$$

IV. Probability of packet loss

Let $\mathcal{E}_T$ be the event that the last window where no packets were dropped from this stream was W_{-T}. Also, let $D_1^{(n)}$ be the random variable denoting the delay within which all packets $P_{0,k}^1$, $k = 1, 2, \ldots, A_0^1$ are successfully received (directly, or via decoding future coded packets). In keeping with the QoS requirement therefore, packets dropped at time 0 (if they are dropped) will be recovered if and only if $D_1^{(n)} \leq d$, i.e. the decoding delay is less than or equal to d. Due to the interdependence of decodability across time-slots, the exact expression for $P(D_1^{(n)} > d)$ is difficult to compute and so we will attempt to bound this value.

For a finite field $\mathbb{F}_q$, a random matrix has a finite probability of not being invertible.

Condition 1: If the number of linear equations is greater than or equal to the number of unknowns, the set of linear equations is solvable for the unknowns if the coefficient matrix of the linear equations is invertible.

We also use $\mathcal{S}_k$ to denote the event that the coefficient matrix corresponding to the RLCs in window W_k is invertible.

For the rest of this paper, we use the notation $\{\mathcal{C}\}$ to denote the event set $\{\omega : \omega \in \Omega,\ \omega \text{ satisfies condition } \mathcal{C}\}$ where $\Omega = \prod_{m=1}^n \Omega_{A^m} \times \Omega_{B^m}$ is the total sample space represented as a product space of the sample path spaces of the packet arrival processes A_t^m and B_t^m. For example $\{D_m^{(n)} > d\} \doteq \{\omega : \omega \in \Omega, D_m^{(n)}(\omega) > d\}$ is the set of sample paths corresponding to the event that the decoding delay for flow from source S_m is greater than d. The complement of an event $\{\mathcal{C}\}$ will be denoted by $\{\neg\mathcal{C}\}$.

Observe that by definition, the sets $\mathcal{E}_T$ are disjoint for different values of T, so if $T \neq T'$, $\mathcal{E}_T \cap \mathcal{E}_{T'} = \emptyset$. Also, since $\mathcal{E}_0$ implies that $L_0^1 = 0$, $P(D_1^{(n)} > d | \mathcal{E}_0) = 0$. Hence, by total probability,

$$P(D_1^{(n)} > d) = \sum_{T=1}^{\infty} P(\{D_1^{(n)} > d\} \cap \mathcal{E}_T). \tag{6}$$

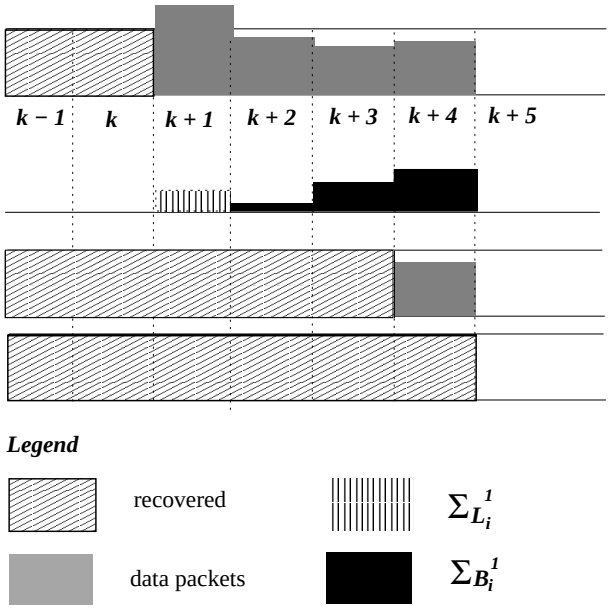

Fig. 3. Progression of the Induction over each $j^* \geq 1$:

A. Upper bound

To obtain the upper bound, we first find a superset of the set of sample paths corresponding to the event $\{D^{(n)} > d | \mathcal{E}_T\}$ in the following lemma.

Lemma 1: $\{D_1^{(n)} > d\} \cap \mathcal{E}_T \subseteq \left[\bigcup_{k=-T}^d \{\{\sum_{i\in W_k} L_i^1 > \sum_{i\in W_k} B_i^1\} \cup \{\neg\mathcal{S}_k\}\}\right] \cap \mathcal{E}_T$.

Proof: We proceed by framing the contrapositive[2].

$$\mathcal{E}_T \cap \bigcap_{k=-T}^d \{\{\sum_{i\in W_k} L_i^1 \leq \sum_{i\in W_k} B_i^1\} \cap \{\mathcal{S}_k\}\} \subseteq \{D_1^{(n)} \leq d\} \cap \mathcal{E}_T. \tag{7}$$

In other words, when $\mathcal{E}_T$ holds, it suffices to show that if for each of the consecutive windows W_{-T} to W_d, the number of losses is less than or equal to the number of coded packets and the RLCs in each window are linearly independent, then packets lost at time-slot 0 can be recovered within d time-slots. We now prove by induction over the sequence of windows $\{W_{-T}, W_{-T+1}, \ldots, W_d\}$. Since $\mathcal{E}_T$ is true, the packets in W_{-T} are all directly received by the destination without requiring any decoding.

Induction Hypothesis: Consider any time-slot $T_0 \geq -T$ such that all packets that were dropped between $-T - d + 1$ and T_0 are recovered by T_0. Then there exists a $1 \leq j^* \leq d$ such that all packets that are dropped between $-T - d + 1$ and $T_0 + j^*$ are recovered by $T_0 + j^*$, i.e. within d time-slots.

We first show that this is true for the base case, i.e. for $T_0 = -T$. Since $\mathcal{E}_T$ holds, no packets are dropped between

[2]We use the following contrapositive argument: Given any sets A, B, C, we have $[A \cap C] \subseteq [B \cap C] \iff [\neg B \cap C] \subseteq [\neg A \cap C]$.

$-T-d+1$ and $-T$ and hence all packets dropped between $-T-d+1$ and $-T$ are recovered by $-T$. $\mathcal{E}_T$ also implies that there is a packet lost at time-slot $-T+1$, i.e. $L^1_{-T+1} > 0$. We now need to find a j^* such that all packets dropped before $-T+j^*$ are recovered by $-T+j^*$ to prove that the induction hypothesis is true for the base case. Now consider the time-slots in window W_{-T+d+1} from $-T+1$ to $-T+d$. Also since the LHS of (8) states that condition $\sum_{i\in W_{-T+d}} L^1_i \leq \sum_{i\in W_{-T+d}} B^1_i$ is true, there must be a time-slot

$$-T+j \doteq \arg\min_{t\in W_{-T+d}} \sum_{i=-T+1}^{-T+t} L^1_i \leq \sum_{i=-T+1}^{-T+t} B^1_i. \quad (8)$$

In other words, $-T+j$ indexes the first time-slot after $-T+1$ when the number of auxiliary packets *just overshoots* (i.e. becomes greater than or equal to) the number of lost packets till that time-slot. Since $L^1_{-T+1} > 0$ also implies that $B_{-T+1} = 0$, it must be that $2 \leq j \leq d$. Now, all the auxiliary packets from time-slot $-T+2$ to $-T+j$ are RLCs of data packets generated in the time-slots between $-T-d+3$ to $-T+j$. Since the coefficient of the RLCs are all known at the receiver, each RLC can be considered as a linear equation over the set of known, and unknown symbols, in $\mathbb{F}_q$ corresponding to packets that have not been dropped, and those that have been dropped, respectively. By the definition of j in (8), the number of unknown symbols in this set of linear equations $\sum_{i=-T+1}^{-T+t} L^1_i$ is matched or exceeded by the number of simultaneous linear equations $\sum_{i=-T+1}^{-T+t} B^1_i$. The LHS of (8) also implies that $\mathcal{S}_{-T+j}$ is true, and therefore that these equations are linearly independent, i.e. Condition 1 holds. Consequently, the receiver can solve this set of simultaneous equations (say, by Gaussian elimination), to decode the unknown symbols (dropped packets). Thus, *all* packets that were dropped before $-T+j$ have been recovered at time-slot $-T+j$ for $1 < j \leq d$ demonstrating that the base case holds with $j^* = j$.

In general, assume that the induction hypothesis holds for any arbitrary time-slot $k \geq -T$. This means that all packets from $-T-d+1$ to k are known at the receiver. If there is no loss at time-slot $k+1$, then we set $j^* = 1$ to observe that the induction hypothesis still holds. If otherwise, i.e. $L^{k+1}_i > 0$ (see Figure 3), we consider the window W_{k+d} containing time-slots from $k+1$ to $k+d$. Then, analogous to the base case, we can find a $1 < j' \leq d$ such that

$$k+j' \doteq \arg\min_{t\in W_{k+d}} \sum_{i=k+1}^{k+t} L^1_i \leq \sum_{i=k+1}^{k+t} B^1_i.$$

Again, noting that Condition 1 holds, we have a set of linearly independent simultaneous equations where the number of unknowns is matched or exceeded by the number of equations. Hence, once again, setting $j^* = j'$, we can show that all packets that are dropped between $-T-d+1$ to $k+j^*$ can be recovered at $k+j^*$.

Since j^* is always greater than 1, the induction proceeds forward along the time-steps where packets are recovered all the way to packets lost in time-slot 0. Also, since $j^* < d$, we can easily see that packets dropped at 0 will be decoded within the next d time-slots.

This proves the contrapositive. We are now done.

□

From (6) and Lemma 1, for any fixed $\bar{T} > 0$, we conclude using the union bound that

$$P(D^{(n)}_1 > d) \leq \sum_{T=1}^{\bar{T}} \left(\sum_{k=-T}^{d} P(\sum_{i\in W_k} L^1_i > \sum_{i\in W_k} B^1_i) + \sum_{k=-T}^{d} P(\neg\mathcal{S}_k) \right) + \sum_{T=\bar{T}+1}^{\infty} P(\mathcal{E}_T). \quad (9)$$

We observe that the probability that the matrix of coefficients $\alpha_{t,k}$ will be of non-full rank $P(\neg\mathcal{S}_k)$ depends on the choice of q in $\mathbb{F}_q$. In the sequel, we will first obtain bounds on each $P(\sum_{i\in W_k} L^1_i > \sum_{i\in W_k} B^1_i)$ and then choose q such that $\sum_{k=-T}^{d} P(\neg\mathcal{S}_k)$ is dominated by $\sum_{k=-T}^{d} P(\sum_{i\in W_k} L^1_i > \sum_{i\in W_k} B^1_i)$.

Traditional large deviations analysis applied to queueing systems focuses on events concerning the empirical mean of a growing set of random variables. Similarly, in the present problem, we are interested in the strong properties of the empirical mean

$$X^{(n)}_i \doteq \frac{1}{n}\sum_{m=1}^{n} A^m_i \quad (10)$$

across source inputs. However, the analysis of the probability of decoding failure is complicated by the fact that the expression for L^1_i contains both the empirical mean term and the individual value A^1_i corresponding to the arrivals from the first source. For ease of analysis, we make the practical assumption of a finite support set for the arrival process below.

Assumption 1: $\mathcal{A}$ is a finite (bounded) set in $\mathbb{N}$.

In other words, there is a finite $M \in \mathbb{N}$ such that for all sources S_m and time-slots i, the number of packets from each source is upper bounded, i.e. $A^m_i < M$. This, together with (4) and (5), implies that

$$P(\sum_{i\in W_k} L^1_i > \sum_{i\in W_k} B^1_i) \leq P(\sum_{i\in W_k} \frac{\left(X^{(n)}_i - C\right)_+}{X^{(n)}_i} M - \min(\bar{B}, \left(C - X^{(n)}_i\right)_+) > 0).$$

Further, to characterize rare events, we need to establish regularity properties for the packet arrival process A^m_i. Since the arrival processes at different sources are assumed to be independent, the following assumption suffices.

Assumption 2: Fix any $d \in \mathbb{N}$ and some window W_k of size d. Define the vector $\bar{A}^m = (A^m_i)_{i\in W_k}$. Then for all $m = 1, 2, \ldots, n$, $\bar{\theta} \in \mathbb{R}^d$, the log moment generating function

$$\Lambda_{\bar{A}^m}(\bar{\theta}) \doteq \log E\left[\exp(\bar{\theta}\cdot\bar{A}^m)\right] < \infty$$

exists and is finite.

In addition, we require an assumption on the mixing properties, of the arrival process to bound the value of $P(\mathcal{E}_T)$.

Assumption 3: For each source m, the arrival process $\{A_i^m\}$ is a i.i.d. process.

Remark 1: The assumption of i.i.d. in time made above is not required for the upper bound proof that we present. A sufficient condition is the α mixing condition defined in [24]pp. 363 which is satisfied by Markov chains over finite state spaces. Alternately, if we assume that the arrival process is $\Theta(d) - dependent$ [24]pp. 364, we can bound the value of $P(\mathcal{E}_T)$. However, for notational and computational ease, in this paper we make the i.i.d. assumption throughout.

For $n \in \mathbb{N}$, define

$$\Lambda_{\bar{A}_n}(\bar{\theta}) \doteq \frac{1}{n}\log E\left[\exp(\sum_{m=1}^{n}\bar{\theta}\cdot\bar{A}^m)\right]$$

where $\bar{A}_n \doteq \sum_{m=1}^{n}\bar{A}^m$. Since the arrival processes across different streams are i.i.d. and ergodic, Assumption 2 implies that

$$\Lambda_{\bar{A}}(\bar{\theta}) \doteq \lim_{n\to\infty}\Lambda_{\bar{A}_n}(\bar{\theta}) = \Lambda_{\bar{A}^1}(\bar{\theta})$$

exists for all windows W_k.

Hence, from the Gartner-Ellis Theorem, for any $\bar{x} \in \mathbb{R}^d$, $\bar{X}^{(n)} \doteq \frac{1}{n}\bar{A}_n$ satisfies a large deviation property (LDP) with good rate function [21], [22], [23] that is a convex dual[3] of $\Lambda_{\bar{A}}$

$$I_{\bar{X}}(\bar{x}) = \sup_{\bar{\theta}}\left(\bar{x}\cdot\bar{\theta} - \Lambda_{\bar{A}}(\bar{\theta})\right). \quad (11)$$

Observe that the function $f : \mathbb{R}^d \to \mathbb{R}$ defined as

$$f(\bar{x}) \doteq \sum_{i=1}^{d}\left[\frac{(x_i - C)_+}{x_i}M - \min(\bar{B}, (C - x_i)_+)\right] \quad (12)$$

is a continuous function defined on $\mathbb{R}^d$. Figure 5, plots f for the case of $d = 1$. Now, using the contraction principle [21], [22], the sequence of random variables,

$$Y_k^{(n)} \doteq \sum_{i\in W_k}\left[\frac{(X_i^{(n)} - C)_+}{X_i^{(n)}}M - \min(\bar{B}, (C - X_i^{(n)})_+)\right]$$

satisfies an LDP with rate function,

$$I_{Y_k}(y, d, \bar{B}) = \inf\{I_{\bar{X}}(\bar{x}) : f(\bar{x}) = y\}. \quad (13)$$

where the $\inf$ of an empty set is defined in the usual manner as ∞. We include d as an argument to the rate function since the rate function varies with d. Subsequently, in Section VI we will show that $I_{Y_k}(y, d, \bar{B})$ increases linearly in d for arrival processes satisfying Assumption 6.

Lemma 2: If $E(A_0^m) < C$ for all $m = 1, 2, \ldots, n$, there exists a fixed $\epsilon > 0$ such that for all $T > 0$ and $n > N_\epsilon$

$$P(\mathcal{E}_T) \le e^{-n\epsilon\lfloor T/d\rfloor} \quad (14)$$

[3]The convex dual is otherwise known as the Legendre-Frenchel transform.

Proof: Define $\mathcal{R}_k$ to be the event that window W_k has no packet drops for packets from source 1 with probability

$$P(\mathcal{R}_k) = P(\{\bigcap_{i\in W_k}(L_i^1 = 0)\}).$$

Therefore,

$$P(\neg\mathcal{R}_k) = P(\{\bigcup_{i\in W_k}(L_i^1 > 0)\}) \quad (15)$$
$$\le dP(L_i^1 > 0) \quad (16)$$
$$= dP(\sum_{m=1}^{n}A_0^m > nC) \quad (17)$$

where (16) follows from the union bound and (17) follows from the ergodicity of the arrival process. Also, since $E(A_0^m) < C$, for some $\epsilon_1 > 0$, the Chernoff bound on $P(\sum_{m=1}^{n}A_0^m > nC)$, together with (17) yields the following inequality,

$$P(\neg\mathcal{R}_k) \le de^{-n\epsilon_1}. \quad (18)$$

From the definition of $\mathcal{E}_T$, we have that

$$\mathcal{E}_T = \left[\bigcap_{k=-1}^{-T}\{\neg\mathcal{R}_k\}\right]\cap\mathcal{R}_{-T} \subseteq \bigcap_{k=-1}^{-T}\{\neg\mathcal{R}_k\} \quad (19)$$
$$\subseteq \bigcap_{j=0}^{\lfloor T/d\rfloor}\{\neg\mathcal{R}_{-1-dj}\} \quad (20)$$

Further, from Assumption 3, the events $\neg\mathcal{R}_{-1-dj}$, $d = 1, 2, \ldots, \lfloor\frac{T}{d}\rfloor$ are independent since they are functions of non-overlapping windows of length d between $-T$ and -1. Now, using (18),

$$P(\mathcal{E}_T) \le \exp(-n[\epsilon_1 - \frac{\log d}{n}]\lfloor T/d\rfloor). \quad (21)$$

Since d is finite, we can choose N_ϵ appropriately such that $\epsilon_1 - \frac{\log d}{n} < \epsilon$ for all $n > N_\epsilon$ to obtain (14).

□

Note that $P(\mathcal{S}_k)$ is a function of the size of $\mathbb{F}_q$. Most recent work on network coding [11], [6], [5] assumes that the field size is large enough to consider that the coefficient matrix at the receiver is completely invertible. For a $k \times k$ matrix with elements taken from $\mathbb{F}_q$, the probability that the matrix will not be invertible is $1 - \prod_{l=1}^{k}(1 - q^{-l})$. The size of the matrix to be inverted depends on the congestion at the link. For instance, if there is no congestion in the link – an event with high probability, since $E[A] < C$ and n is large – none of the auxiliary coded packets need to be decoded since there are no packet drops. Hence, the size of the matrix that needs to be inverted is equal to the number of drops in d consecutive time-slots. Trivially, the number of auxiliary packets in d slots is bounded by $\bar{B}d$. Hence, for the purposes of our analysis, it is sufficient to bound the field size from below as follows such that Condition 1 always holds.

Assumption 4: We consider that the field $\mathbb{F}_q$ is large enough so that

$$1 - \prod_{l=1}^{\bar{B}d}(1 - q^{-l}) \le P(\sum_{i\in W_0}\left(L_i^1 - B_i^1\right) > 0).$$

Remark 2: Assumption 4 is easily satisfied in most practical cases. We note that with $\bar{B}, d = 10$, and $P(\sum_{i\in W_0}\left(L_i^1 - B_i^1\right) > 0)$ of the order of 10^{-6} (or 10^{-8}, respectively), this implies that q must be approximately 20 (30, respectively) bits long.

We are now ready to state our first result.

Theorem 1: If the average arrival rate for each of $m = 1, 2, \ldots, n$ sources $E(A_0^m) < C$, and $D^{(n)}$ be the delay within which all dropped packets must be recovered, then under the condition that Assumption 4 is satisfied for field size $\mathbb{F}_q$, for any finite $d > 0$,

$$\lim_{n\to\infty}\frac{1}{n}\log P(D_1^{(n)} > d) \quad \leq \quad -I_Y(0, d, \bar{B})$$

where

$$I_Y(y, d, \bar{B}) \quad = \quad \inf\{I_{\bar{X}}(\bar{x}) : f(\bar{x}) = y\}. \tag{22}$$

for the mapping $f(\cdot)$ defined in (12).

Proof: Since the processes A_i^m, $m = 1, 2, \ldots, n$ are ergodic and identically distributed, from (13) and the definition of the rate function in (2),

$$P(\sum_{i\in W_k}\{L_i^1 - B_i^1\} > 0) \quad \leq \quad \exp\left(-nI_{Y_k}(0, d, \bar{B})\right)$$

for all $k = -1, -2, \ldots - \infty$. So, defining $I_Y \doteq I_{Y_k}$ and using the upper bound in (9), we have for n large enough and for some finite $K > 0$,

$$\begin{aligned}
&P(D_1^{(n)} > d)\\
&\leq \sum_{T=1}^{\bar{T}} 2(T+d)\exp\left(-nI_{Y_k}(0, d, \bar{B})\right)\\
&\quad + \sum_{T=\bar{T}+1}^{\infty} P(\mathcal{E}_T) &(23)\\
&\leq \bar{T}(\bar{T}+2d+1)\exp\left(-nI_{Y_k}(0, d, \bar{B})\right)\\
&\quad + \sum_{T=\bar{T}+1}^{\infty} e^{-n\epsilon\lfloor T/d\rfloor} &(24)\\
&\leq \bar{T}(\bar{T}+2d+1)\exp\left(-nI_{Y_k}(0, d, \bar{B})\right)\\
&\quad + \sum_{T=\bar{T}+1}^{\infty} e^{-n\epsilon(T/d-1)} &(25)\\
&\leq \bar{T}(\bar{T}+2d+1)\exp\left(-nI_{Y_k}(0, d, \bar{B})\right)\\
&\quad + 2e^{-n(\epsilon/d)(\bar{T}-d)} &(26)\\
&\leq K\exp\left(-nI_{Y_k}(0, d, \bar{B})\right) &(27)
\end{aligned}$$

where (26) follows from Lemma 2 and from standard results on the convergence of series. (27) follows by choosing a fixed $\bar{T}$ to satisfy $(\epsilon/d)(\bar{T} - d) > I_{Y_k}(0, d, \bar{B})$. We are now done. The 2 in (23) stems from Assumption 4 and the consequent bound $P(\neg\mathcal{S}_k) \leq P(\sum_{i\in W_0}\left(L_i^1 - B_i^1\right) > 0)$.

□

B. Lower Bound

In this section, we lower bound $P(D_1^{(n)} < d)$ to study the tightness of the upper bound in the previous subsection. We define $\mathcal{E}_i'$ as the event where data packet drops occur in all time-slots in window W_i. Therefore, if $\{L_0 > 0 \cap \mathcal{E}_d'\}$ occurs, then no auxiliary coded packets containing information about the packets lost at time-slot 0 arrive at the destination. Hence, none of the dropped packets can be recovered. Since $\mathcal{E}_d' = \bigcap_{i=1}^{d}\{\sum_{m=1}^{n} A_i^m > nC\}$,

$$P\left(\bigcap_{i=1}^{d}\left\{\sum_{m=1}^{n} A_i^m > nC\right\}\right) \quad \leq \quad P(D^{(n)} > d). \tag{28}$$

In particular, if the arrival process $\{A_i^m\}$ is i.i.d. across time, the lower bound in the above expression can be evaluated exactly in terms of the rate function of A_m^1 as follows,

$$P\left(\bigcap_{i=1}^{d}\left\{\sum_{m=1}^{n} A_i^m > nC\right\}\right) \quad = \quad \left[P\left(\sum_{m=1}^{n} A_i^m > nC\right)\right]^d \tag{29}$$

Let

$$\Lambda_A(\theta) \quad \doteq \quad \log E[\exp(\theta A_m^i)] \tag{30}$$

be the log moment generating function of the random variable $\{A_m^i\}$, $i = -\infty, \ldots, -1, 0, 1, \ldots, \infty$, $m = 1, 2, \ldots, n$. Note that since the sources have i.i.d. arrival processes, we do not index the expression for log MGF by time-slot i or source m, and will use the same expression for the arrival process from any source at any time.

Since $E[A_i^m] < C$, we have from [21],

$$P\left(\sum_{m=1}^{n} A_i^m > nC\right) \quad \geq \quad \exp(-n\Lambda_A^*(C) + o(n))$$

where

$$\Lambda_A^*(x) \doteq \sup_{\theta}(\theta x - \Lambda_A(\theta)) \tag{31}$$

and a function $f(n) = o(n)$ if $\lim_{n\to\infty} f(n)/n = 0$.

Then, from (28), (29) and (31), we arrive at the following result.

Lemma 3: If each of the sources $m = 1, 2, \ldots, n$ has i.i.d. arrival process $\{A_i^m\}$, with $E(A_0^m) < C$, and $D^{(n)}$ be the delay within which all dropped packets must be recovered, then for any finite $d > 0$,

$$\lim_{n\to\infty}\frac{1}{n}\log P(D^{(n)} > d) \quad \geq \quad -d\Lambda_A^*(C)$$

□

We note that the determining lower bound on the limit of $\frac{1}{n}P(D^{(n)} > d)$ as $n \to \infty$ for Markov arrival process at each source in general remains an open problem. Further, we conjecture that the upper and the lower bounds of the limit above are identical in the order of d for the general Markov arrival process case.

V. MULTI-HOP NETWORKS

In this section, we extend the large deviations results of the previous section from a single link to a general multi-hop network. Recall that we had selected the $\bar{B}$ as the constant rate at which auxiliary data packets are generated by the the source for the single link case. However, in a multi-link path Γ of length $|\Gamma|$ from source N_0 to destination N_L where intermediate nodes $N_1, N_2, \ldots, N_{|\Gamma|-1}$ function as either sources or sinks for their respective streams and well as routing packets destined for other hosts, the rate of auxiliary packets arriving at destination $N_{|\Gamma|}$ is a function of the aggregate traffic flow across all intermediate links. This coupling of the sample paths of each individual source process motivates an approach based on decoupling flows to obtain an appropriate bound on the end-to-end probability that a packet transmitted at time-slot 0 will be *lost*.

We also note that the number of paths n_e crossing a link(edge) e is a function of the topology of the network and the source-destination partition of the nodes in the network. We will assume that at each edge, the capacity of the edge scales as $n_e C$ to ensure that no source-destination paths a completely blocked. For a path Γ defined as a set of edges $e_{N_k,N_{l+1}}$, along the path, we define

$$n_\Gamma \doteq \min_{e\in\Gamma} n_e. \tag{32}$$

Assumption 5: We consider networks where for each edge e in the network $n_e = \Omega(N^\alpha)$ where N is the number of nodes in the network uniformly for some fixed $\alpha \in (0,1)$. Also the path length $|\Gamma| = \Omega(N^\beta)$ for some $\beta \in (0,1)$.

This assumption is motivated by the spate of recent results in scaling laws over large networks [14], [17], [16], [15] such as ad-hoc networks or in server grids. The authors in [16] prove that if N nodes are scattered uniformly over a unit area, divided into sure tiles of area $a(N)$ each, and under a relaxation of the Protocol Model for wireless ad-hoc networks proposed in Gupta and Kumar [14], the number of paths crossing each tile is $O(N/\sqrt{a(N)})$ with high probability when the propagation occurs along a straight line path. Further, for direction based routing with errors but with a *progressive routing* assumption where the distance between the source and destination is reduced by at least $\delta\sqrt{a(N)}$ for some $\delta > 0$ and $a(N) = \frac{\log N}{N}$, Subramanian and Shakkottai [15] show that the total number of tiles $|\Gamma|$, that a path can touch is upper bounded by $\frac{1}{\delta K a(N)}$ for some $K \in \Theta(1)$. Thus, since the mean Euclidean path length is $\Theta(1)$, by symmetry the probability that a path crosses a given tile is lower bounded by $\delta K a(N)$.

For a symmetric rectangular grid of N computers, ignoring edge effects (or assuming a wrap-around at the edges to form a torus) and source-destination pairs chosen uniformly at random from among the nodes, the expected number of paths through any edge is $\sqrt{N}$. This, again points to the validity of Assumption 5.

Assumption 5 together with the definition in (32) implies that $n_\Gamma = \Omega(N^\alpha)$ for any path Γ in the network.

Further, by Assumption 4, we will consider the field size (packet size) is large enough such that a lost packet can be decoded simply if the number of auxiliary packets is greater than the number of lost packets in window.

Let $A^m_{i,e}$ is the flow from source m through edge e at time i. Then we define $X^{(n)}_{i,e} \doteq \frac{1}{n}\sum_{m=1}^{n} A^m_{i,e}$ as the normalized cumulative flow of data packets through e at time i.

Further, we use $L^m_{\Gamma,i}$ to denote the number of packets from source S^m *dropped* in time-slot i along path P. Recall that we assume that there are no packet transmissions delays and that we treat each link as a pipe that instantaneously transfers the packet from source to destination in case there is sufficient capacity, else the packet is dropped at the first edge where there is a congestion. In general, link propagation delays can be handled easily by the appropriate indexing of time at each link along the propagation path. However, we skip the details since it does not affect our analysis in any way.

Also, let $B^m_{\Gamma,i}$ be the number of auxiliary packets from source S_m that reach the destination at the end of path P in time-slot i. Also fix any $\bar{T} > 0$. Assuming that the field size $\mathbb{F}_q$ is large enough as before we can bound the term $P(\mathcal{S}_k)$ corresponding to decoding failure in (9) (using Assumption 4) to write the probability that packet loss of a packet from source S_m dropped in time-slot 0 on path Γ as

$$\begin{aligned} P(D^{(n_\Gamma)}_{\Gamma,m} > d) \quad &\leq \quad \sum_{T=1}^{\bar{T}} \left(\sum_{k=-T}^{d} 2P\Big(\sum_{i\in W_k} L^m_{\Gamma,i} - B^m_{\Gamma,i} > 0\Big)\right) \\ &\quad + \sum_{T=\bar{T}+1}^{\infty} P(\mathcal{E}_T). \end{aligned} \tag{33}$$

Observe that the path packet drop term $L^m_{\Gamma,i}$ is a sum of the edge losses at each edge. However, the edge losses are not independent a each link. Therefore, we bound $L^m_{\Gamma,i}$ by

$$L^m_{\Gamma,i} \leq \bar{L}^m_{\Gamma,i} \doteq M \max_{e\in\Gamma} I_{\{X^{(n_\Gamma)}_{i,e} > C\}} \tag{34}$$

where $I_{\{\mathcal{A}\}}$ is the identity function for event $\{\mathcal{A}\}$. The intuition behind the above bound is simple – if the most congested link e along path Γ has $X^{(n_\Gamma)}_{i,e} > C$, then $\bar{L}^m_{\Gamma,i}$ corresponds to the case where the *entire* set of data packets from S_m, which is bounded by M following Assumption 1, is dropped along path Γ.

Note that using the bound in (34), we can write the following inequality

$$P\Big(\sum_{i\in W_k} L^m_{\Gamma,i} - B^m_{\Gamma,i} > 0\Big) \leq P\Big(\sum_{i\in W_k} \bar{L}^m_{\Gamma,i} - B^m_{\Gamma,i} > 0\Big). \tag{35}$$

Now, assuming that the source generates auxiliary coded packets at a maximum data rate of $\bar{B}$, and using the model of a data-pipe along with packets can be dropped, the rate at which auxiliary packets can reach the destination is determined by the (normalized) cumulative packet $X^{(n_\Gamma)}_{i,e}$ on the most congested link $e \in \Gamma$, see Figure 4. Thus,

$$B^m_{\Gamma,i} = \min\left(\bar{B},\ \min_{e\in\Gamma}\ (C - X^{(n_\Gamma)}_{i,e})_+\right). \tag{36}$$

Next, it follows that

$$\bar{L}^m_{\Gamma,i} - B^m_{\Gamma,i} = \max_{e\in\Gamma}\left(M I_{\{X^{(n_\Gamma)}_{i,e} > C\}} - \min(\bar{B}, (C - X^{(n_\Gamma)}_{i,e})_+)\right).$$

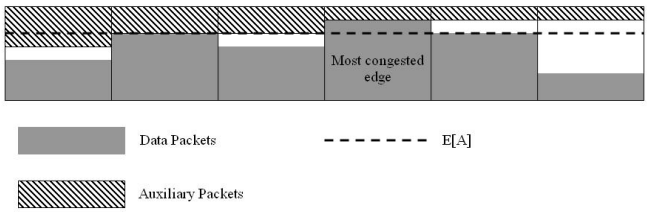

Fig. 4. The rate of auxiliary packets received at the destination of path Γ is equal to the rate at the tail of the most congested link along P as shown here.

We show this by considering the following two cases. Case (i) occurs when there is no overflow in any link on the entire path, i.e., $\bar{L}^m_{\Gamma,i} = 0$, and thus LHS in the equation above is $-B^m_{\Gamma,i}$. The RHS of the equation above, in this case, is

$$\max_{e\in\Gamma}\left(-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e})_+)\right)$$
$$=-\min\left(\bar{B},\min_{e\in\Gamma}(C-X^{(n_\Gamma)}_{i,e})_+\right)=-B^m_{\Gamma,i}=LHS,$$

and we are done. On the other hand in Case (ii), there is a loss on one (or more) link $e \in \Gamma$ (i.e., $X^{(n_\Gamma)}_{i,e} > C$). In this case, $(C - X^{(n_\Gamma)}_{i,e})_+ = 0$ and hence $B^m_{\Gamma,i} = 0$. Then, we have

$$LHS \;=\; \bar{L}^m_{\Gamma,i} \;=\; M \;=\; RHS,$$

for Case (ii) as well.

This implies that

$$P(\sum_{i\in W_k}\bar{L}^m_{\Gamma,i}-B^m_{\Gamma,i}>0)=$$
$$P(\sum_{i=1}^{d}\max_{e\in\Gamma}\left(MI_{\{X^{(n_\Gamma)}_{i,e}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e})_+)\right)>0)$$

Assumption 6: The packet arrival process at each source S_m, $\{A^m_i\}$ is i.i.d. in time, i.e. for two time-slots i, j: $i \neq j$ A^m_i is independent of A^m_j and the two random variables are identically distributed.

Let $\mathcal{L} = \{(e_1, e_2, \ldots, e_d)\}$, $e_i \in \Gamma$, $i = 1, 2, \ldots, d$. Note that $|\mathcal{L}| = |\Gamma|^d$. Then we have that

$$\{\sum_{i=1}^{d}\max_{e\in\Gamma}\left(MI_{\{X^{(n_\Gamma)}_{i,e}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e})_+)\right)>0\}$$
$$=\bigcup_{(e_1\ldots e_d)\in\mathcal{L}}(\sum_{i=1}^{d}\left(MI_{\{X^{(n_\Gamma)}_{i,e_i}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e_i})_+)\right)>0)$$

To see this, consider any four random variables Z_1, Z_2, Z_3, Z_4. Then, observe that the event $\{\max(Z_1, Z_2) + \max(Z_3, Z_4) > 0\}$ is the same as $\{(Z_1 + Z_3 > 0) \cup (Z_1 + Z_4 > 0) \cup (Z_2 + Z_3 > 0) \cup (Z_2 + Z_4 > 0)\}$ The above statement is merely an extension of this result.

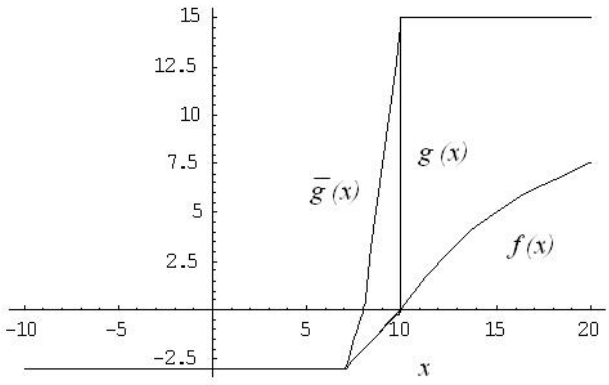

Fig. 5. Contraction mapping functions f, g and $\bar{g}$ plotted for the case of $M = 15, C = 10, \bar{B} = 3, \beta = 2, E[A] = 8$. Note that the large β is merely for purposes of illustration. A small $\beta > 0$ leads to tighter bounds on the packet loss probability.

Therefore, using the union bound,

$$P(\sum_{i=1}^{d}\max_{e\in\Gamma}\left(MI_{\{X^{(n_\Gamma)}_{i,e}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e})_+)\right)>0)$$
$$\leq\sum_{(e_1,\ldots,e_d)\in\mathcal{L}}P(\sum_{i=1}^{d}\left(MI_{\{X^{(n_\Gamma)}_{i,e_i}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e_i})_+)\right)>0). \quad (37)$$

Also, since packets can only be dropped from the flow originating from source S_m in subsequent links on the network, we have that any flow $A^m_{t,e} < A^m_t$, where A^m_t is defined as in the previous section to be the total number of data packets generated by S_m in time t. This means that fewer packets are dropped in link e as the link gets farther away from the source S_m since the flow has already been 'thinned out' by dropping packets in the previous links. Also, since the arrivals are i.i.d. (from Assumption 6), we have

$$\sum_{\mathcal{L}}P(\sum_{i=1}^{d}\left(MI_{\{X^{(n_\Gamma)}_{i,e_i}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e_i})_+)\right)>0)$$
$$\leq\sum_{\mathcal{L}}P(\sum_{i=1}^{d}\Big(MI_{\{X^{(n_\Gamma)}_{i}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i})_+)\Big)>0),$$

where $X^{(n_\Gamma)}_i$ is as defined in (10). Hence, we have from (37),

$$P(\sum_{i=1}^{d}\max_{e\in\Gamma}\left(MI_{\{X^{(n_\Gamma)}_{i,e_i}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e})_+)\right)>0)$$
$$\leq|\Gamma|^dP(\sum_{i=1}^{d}\left(MI_{\{X^{(n_\Gamma)}_{i}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i})_+)\right)>0). \quad (38)$$

However, note that unlike $f(x)$ in Section IV, $g : \mathbb{R}^d \to \mathbb{R}$

$$g(x)\doteq\sum_{i=1}^{d}MI_{\{x_i>C\}}-\min(\bar{B},(C-x_i)_+)$$

is not a continuous function and hence, we cannot apply the contraction principle directly. Therefore, we upper bound $g(x)$ by the function $\bar{g}(x)$ as follows. Fix any small $0 < \beta < C - \bar{B}$. Then we define

$$\bar{g}(x) \doteq \sum_{i=1}^{d} \bar{g}_i(x_i)$$

where

$$\bar{g}_i(x_i) \doteq \begin{cases} M & \text{for} \quad x \geq C \\ \frac{M}{\beta}(x - C + \beta) & \text{for} \quad C - \beta \leq x < C \\ \frac{\bar{B}}{\bar{B}-\beta}(x - C + \beta) & \text{for} \quad C - \bar{B} \leq x < C - \beta \\ -\bar{B} & \text{for} \quad x < C - \bar{B} \end{cases}$$

as shown in Figure 5.

Thus the contraction principle applied to the vector sequence $\bar{X}^{(n_\Gamma)}$ indexed by n_Γ with rate function $I_{\bar{X}}(\bar{x})$ in (11), together with (38) implies that the sequence of random variables

$$W^{(n_\Gamma)} \doteq \sum_{i=1}^{d} \bar{g}_i(X_i^{(n_\Gamma)})$$

satisfies an LDP with rate function

$$I_W(w, d, \bar{B}) \doteq \inf\{I_{\bar{X}}(\bar{x}) : \bar{g}(\bar{x}) = w\}. \tag{39}$$

Thus, we have for n large enough,

$$P(\sum_{i \in W_k} L_{\Gamma,i}^m - B_{\Gamma,i}^m > 0) \quad \leq \quad e^{-n_\Gamma I_W(0,d,\bar{B})}. \tag{40}$$

Substituting (40) in (33), using the exponential tightness[4] of $P(\mathcal{E}_T)$ from (the path version of) Lemma 2, choosing a fixed $\bar{T}$ large enough such that the first term in (33) dominates (the argument is identical to that in (27) in Theorem 1), and noting that from Assumption 5, $|\Gamma|^d$ is polynomial in n, we have the probability that a packet dropped on path Γ between source S^m and the destination at time-slot 0 is lost (cannot be recovered) is asymptotically bounded as follows

$$\lim_{n_\Gamma \to \infty} \frac{1}{n_\Gamma} \log P(D_\Gamma^{(n_\Gamma)} > d) \leq -I_W(0, d, \bar{B}) \tag{41}$$

proving the following result.

Theorem 2: Consider a path Γ from source $S^m = N_0$ to destination $N_{|\Gamma|}$ in a network satisfying the topological requirements in Assumption 5. Also, assume that all sources S^j in the network have i.i.d. packet arrival process, $\{A_i^j\}$ satisfying Assumption 6 with mean $E(A_0^j) < C$. Also, if the source generates auxiliary packets with rate $\bar{B}$, then the probability that a packet dropped from source S^m in time-slot 0 cannot be recovered with delay $D_{\Gamma,m}^{(N)} < d$ is asymptotically bounded as

$$\lim_{N \to \infty} \frac{1}{n_\Gamma} \log P(D_{\Gamma,m}^{(n_\Gamma)} > d) \leq -I_W(0, d, \bar{B}) \tag{42}$$

□

[4]Note that Lemma 2 generalizes to a path because of the facts that on each edge the probability of loss decays exponentially, and that the number of edges in a path can grow at most polynomially.

In the following section, we perform numerical simulations to show that $I_W(0, d, \bar{B})$ is strictly positive and scales linearly in d for i.i.d. arrival processes.

Remark 3: In a queueing network with buffering at intermediate nodes, each node needs to have a buffer of size $b = \Theta(d)$ allocated for *every* flow passing through it. This follows from many-sources large deviations for a single server queue [18]. Botvich and Duffield show that at a single link, a buffer of $\Theta(n_\Gamma b)$ is necessary to achieve a loss probability that decays as $e^{-n_\Gamma I(b)}$, and $I(b) \approx \delta b + \nu$ (see (1)). Consequently, the buffer size at each intermediate node scales similarly (since loss can occur at any of the links in the path of the flow).

Since, we have assumed that $n_\Gamma = \Omega(N^\alpha)$ (recall from (32) that n_Γ is a lower bound on the number of flows through any intermediate router), the above argument implies that the total buffering required in the network (with N nodes) scales as $\Omega(N^{1+\alpha})$.

On the other-hand, for comparable QoS with network coding, Theorem 2 requires $\Theta(d)$ buffers *per source-destination flow*. This implies that the total buffer in the network scales as $\Theta(Nd)$ (as there are $\Theta(N)$ source-destination pairs). This gives the *spatial buffer multiplexing* a per-node buffering gain of $\Omega(N^\alpha)$ over traditional queueing at intermediate nodes.

VI. Numerical Results

A. Single Link

Under the i.i.d. Assumption 6, we can show that the rate function for the single link packet loss probability using network coding $I_Y(0, d, \bar{B})$ derived in (13) scales linearly in d if the mean arrival rate $E[A] \in (C - \bar{B}, C)$. In this paper (due to space constraints), we demonstrate this for the simple case where $\{A_i^m\} \sim Bernoulli(p)$ with $p = 0.6$ (hence $E[A_i^m] = 0.6$) over a link of capacity 0.9. The rate function for each A can be derived from the convex dual of the Log Moment generating function (MGF) of the Bernoulli random variable to be

$$I_A(x) = x \log(x/p) + (1 - x) \log((1 - x)/(1 - p)).$$

Using standard rate function computations (for vectors with i.i.d. elements) [21], we can write the rate function for the sequence $\bar{X}^{(n)}$ as

$$I_(\bar{X})(\bar{x}) \quad = \quad \sum_{i=1}^{d} I_A(x_i). \tag{43}$$

Substituting in (13), for $y = 0$, we have

$$\begin{aligned} & I_{Y_k}(y, d, \bar{B}) = \inf\{I_{\bar{X}}(\bar{x}) : f(\bar{x}) = y\} \\ = \quad & \min_{x_i \in [0,1]} \sum_{i=1}^{d} x_i \log(x_i/p) + (1 - x_i) \log((1 - x_i)/(1 - p)) \\ & \text{such that} \sum_{i=1}^{d} f_i(x_i) = 0\}, \end{aligned} \tag{44}$$

and f_i is defined as

$$f_i(x) = \left[\frac{(x - C)_+}{x} M - \min(\bar{B}, (C - x)_+) \right].$$

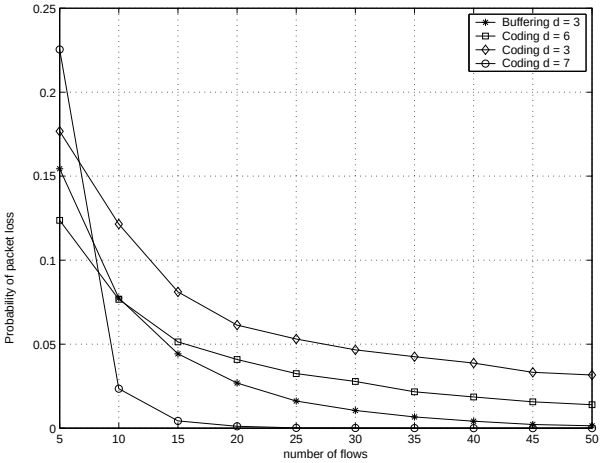

Fig. 6. Comparison of coding with buffering. Coding with $d = 3, 6$ performs marginally poorer than queueing with $b = 3$. However, coding with $d = 7$ performs better than queueing with $b = 3$. Thus, the performance of coding matches buffering for $d = O(b)$.

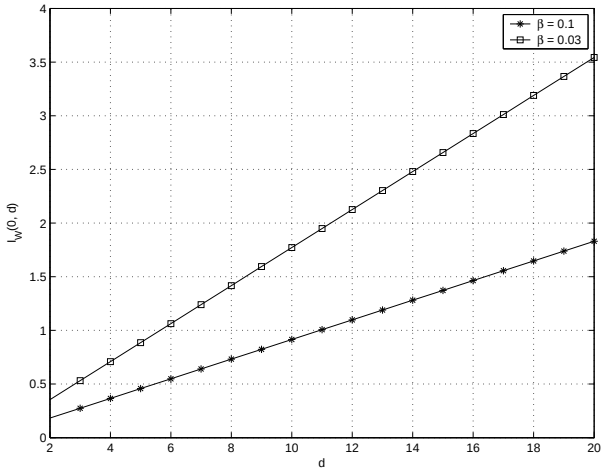

Fig. 7. Rate function for the multiple link case as a function of d

Note that $f_i(x) = 0$ at C and is strictly increasing and locally concave at C (see Figure 5). Also, since the rate function $I_A(x)$ is convex and greater than zero everywhere (except at $x = E[A]$ where $I_A(E[A]) = 0$), if $E[A] \in [C - \bar{B}, C]$ the rate function is a strictly increasing convex function in a small neighbourhood around C. Therefore (44) can be written as the convex optimization problem with convex increasing positive cost function $I_A(f^{-1}(z_i))$ under the constraint $\sum_{i=1}^{d} z_i = 0$. From standard optimization theory it follows that, the objective obtains it's minimum when each $z_i = 0$, corresponding to each $x_i = C$. Hence

$$I_{Y_k}(y, d, \bar{B}) = dI_A(C).$$

For our particular example, $I_A(C) = 0.2263$. Hence the probability of packet loss with network coding for this case, scales as $\Theta(\exp(-nd \times (0.2263)))$ showing that coding over larger blocks provides exponential gain in the probability of packet loss. This is analogous to Botvich and Duffield's [18] result for queueing, repeated in (1) where $I(b)$ scales linearly as buffer-size b in the large b regime.

We also perform a simulation for the single link case with i.i.d. packet arrivals to each source with a Poisson distribution with mean $E[A_i^m] = 58$, $m = 1, 2, \ldots, n$, $i = 0, 1, \ldots$ and capacity per-flow $C = 60$. We compute the probability of packet loss with queueing in intermediate nodes and *spatial buffer multiplexing* via network coding at the source alone and plot the results in Figure 6. We observe that similar performance in terms of packet loss probabilities can be achieved if the number of time-slots over which network coding needs to be performed d is orderwise the same as the buffer b required for queueing.

B. Path with multiple links

Unlike the single link case, the mapping function $\bar{g}$ for the multiple hop case (see Figure 5) is not concave in the neighbourhood of C. However, local properties of the function $\bar{g}(x)$ around $x = C$, allow $I_W(0, d, \bar{B})$ to scale linearly as well with d. However,the analysis is considerably lengthier. Instead, we numerically compute the values of $I_W(0, d, \bar{B})$ for the Bernoulli arrival process in the previous subsection and graphically observe that the rate function does indeed scale linearly with d.

VII. Conclusion

In this paper we have studied the comparison of buffering at each intermediate link along a path versus network coding at the source and decoding at the destination. Using large deviations based analysis, we have derived upper and lower bounds on the probability of packet loss over a single link using a sliding-window based network coding scheme. By computing the rate function, we have shown that if the buffer required for coding is orderwise the same as the buffer for queueing, the same QoS (packet loss probability) can be obtained.

Next, we generalize the rate function to the case of a path with multiple links and for coding buffer of $d = \Theta(1)$. We derive an upper bound on the probability of packet drop that decays exponentially in n_Γ, the minimum number of flows through any edge along the path. We use this result to show that the network coding based scheme can provide order-wise buffer gains in large networks.

Acknowledgments

The research of S. Bhadra and S. Shakkottai was supported by NSF Grants ACI-0305644, CNS-0325788, CNS-0347400 and CNS-0519401.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204–1216, 2000.
[2] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, October 2003.
[3] T. Ho, R. Koetter, M. M. abd D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. 2003 International Symposium on Information Theory*. IEEE, 2003.
[4] T. Ho, M. Mdard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *41st Allerton Annual Conference on Communication, Control, and Computing*, Monticello, IL, October 2003.

[5] D. S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," in *Proc. 2004 International Symposium on Information Theory and its Applications (ISITA 2004)*, October 2004.
[6] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, "Achieving minimum cost multicast: A decentralized approach based on network coding," in *Proc. IEEE Infocom 2005*, March 2005.
[7] S. Deb and M. Médard, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," in *Proc. Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2004.
[8] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. INFOCOM 2005*, 2005.
[9] D. Mosk-Aoyama and D. Shah, "Information dissemination via gossip: Applications to averaging and coding," http://www.arxiv.org/, April 2005.
[10] B. Cohen, "Incentives build robustness in bittorrent," in *P2P Economics Workshop*, Berkeley, CA, 2003.
[11] D. S. Lun, M. Médard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, September 2004.
[12] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "Further results on coding for reliable communication over packet networks," in *2005 IEEE International Symposium on Information Theory*, Sydney, 2005.
[13] M. Luby, "Lt codes," in *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, November 2002, pp. 271–280.
[14] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.
[15] S. Subramanian and S. Shakkottai, "Geographic routing with limited information in sensor networks," in *The Fourth International Conference on Information Processing in Sensor Networks(IPSN)*, Los Angeles, CA, April 2005.
[16] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput delay trade-off in wireless networks," in *Proc. INFOCOM 2004*, 2004.
[17] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *IEEE INFOCOM-2001*, Anchorage, Alaska, 2001, pp. 1360–1369.
[18] D. D. Botvich and N. G. Duffield, "Large deviations, the shape of the loss curve, and economies of scale in large multiplexers," *Queueing Systems*, vol. 20, pp. 293–320, 1995.
[19] S. Shakkottai and S. Srikant, "Many-sources delay asymptotics with applications to priority queues," *Queueing Systems Theory and Applications (QUESTA),*, vol. 39, pp. 183–200, October 2001.
[20] C. Courcoubetis and R. Weber, "Buffer overflow asymptotics for a buffer handling many traffic sources," *Journal of Applied Probability*, vol. 33, 1996.
[21] A. Dembo and O. Zeitouni, *Large deviations techniques and applications*, 2nd ed. Springer-Verlag, 1998.
[22] A. Schwartz and A. Weiss, *Large deviatoins for performance analysis*. Chapman and Hall, 1995.
[23] G. de Veciana and J. Walrand, "Effective bandwidths: Call admission, traffic policing and filtering for atm networks," *Queueing Systems*, vol. 20, pp. 37–39, 1995.
[24] P. Billingsley, *Probability and Measure*, 3rd ed. Wiley, 1995.