QoS-Driven Network Coded Wireless Multicast

Wei Pu, Chong Luo, *Member, IEEE,* Feng Wu, *Senior Member, IEEE,* and Chang Wen Chen, *Fellow, IEEE*

***Abstract*—Emerging wireless multicast applications simultaneously impose two requirements to the underlying communication networks: to provide sufficient bandwidth and to support a variety of quality-of-service (QoS) sensitivities. For the first requirement, network coding has been proposed recently as an effective way of improving bandwidth utilization. However, almost all previous works about network coding focus on the throughput gain without considering the QoS requirements. Optimal network code construction in wireless multicast under different QoS constraints remains as a significant challenge. In this research, we study the QoS-driven network coding problem. We use large deviation principle to establish the relationship among source rate, link condition, QoS requirement, and network code. Using this relationship, under given QoS requirements, we solve the optimal network code construction problem. The proposed network code supports maximal source rate without violating the QoS requirements. These results constitute the foundations for future designing and implementing network coding based wireless multicast protocols.**

***Index Terms*—Large deviations, multicast, network coding, queueing networks, quality of service (QoS), wireless.**

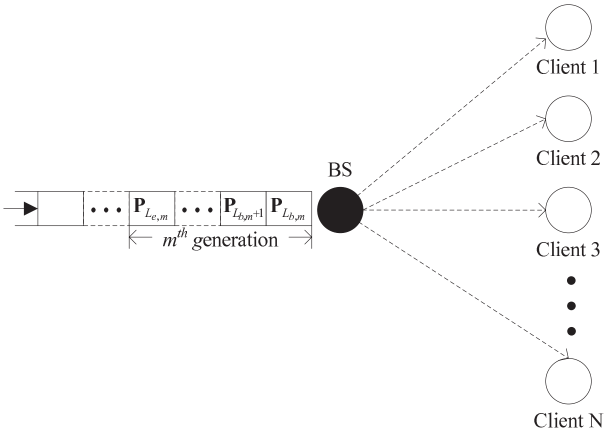


Fig. 1. System model.

I. Introduction

As broadband wireless access becomes increasingly pervasive, the demand for multicast protocol to support one-to-many wireless applications such as video conferencing and multimedia streaming intensifies rapidly. Compared with traditional voice services, these new applications are not only bandwidth consuming but also QoS sensitive. Due to the time varying wireless fading channels, the heterogeneous link conditions, and the QoS requirements, it is a challenging task to design a wireless multicast protocol that scales well to client number, adapts to channel variation and link heterogeneity, and efficiently utilizes the bandwidth resources.

Recently, *network coding* [1][2][3] is introduced to wireless multicast [4][5]. Network coding makes full use of the broadcast nature of wireless channels. By algebraically combining the packets, network coding scheme can serve different clients simultaneously, which results in significant network throughput gain. However, almost all of the existing works about wireless network coding are studied under the framework of information theory, linear programming, abstract algebra, or combinatorics. With different emphases, these frameworks focus on designing optimal network code under certain metrics such as achievable network throughput or encoding/decoding complexity. These pioneer results provide us with fundamental insights to the achievable performance of network coding. However, these theoretical results implicitly assume unbounded decoding delay and unlimited buffer size. In contrast, practical applications usually have a variety of QoS requirements and finite buffer sizes. Studying network coding's performance and designing optimal network code under these constraints are important tasks for practical protocol design.

In this paper, we present our research results on QoS-driven network coding for downlink wireless multicast with one access point (AP) broadcasting to multiple clients. We focus on two kinds of QoS requirements: the buffer overflow constraint and the delay violation constraint. The main contributions of this research include:

- We formulate QoS-driven network coding as a stochastic optimization problem. Our result indicates that as the code length increases, the maximal source rate that the network code can support increases and the QoS exponents corresponding to the maximal source rate decreases. By translating the QoS requirements to the constrains of the minimal QoS exponents, we establish the relationship among QoS exponents, network code length, multicast client number, and clients' link conditions using large deviations principle (LDP) [6].
- Given the client number and their average packet error rates (PERs), we propose a practical algorithm to calculate the optimal network code length that supports the maximal source rate while satisfying QoS constraints. In another word, the algorithm decides the effective capacity [7] of wireless broadcast channels when network coding is employed.

Manuscript received February 11, 2009; revised June 9, 2009; accepted August 10, 2009. The associate editor coordinating the review of this paper and approving it for publication was J. Zhang.

This work was carried out when W. Pu was an intern at Microsoft Research Asia.

W. Pu is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, Anhui, China, 230026 (e-mail: puipp@mail.ustc.edu.cn).

C. Luo and F. Wu are with the Internet Media Group, Microsoft Research Asia, Beijing, China, 100080 (e-mail: {fengwu, Chong.Luo}@microsoft.com).

C. W. Chen is with the Department of Computer Science and Engineering, University at Buffalo, the State University of New York. Buffalo, NY, USA, 14260-2000 (e-mail: chencw@buffalo.edu).

Digital Object Identifier 10.1109/TWC.2009.090203

The rest of this paper is organized as follows. Section II summarizes related work. In Section III, we define the system model and formulate QoS-driven network coding as a stochastic optimization problem. In Section IV, we adopt LDP to establish the relationship among link condition, network code, client number, and QoS requirements, and propose a practical algorithm to solve the optimization problem. In Section V, we present numerical results which verify our analytical results. Section VI concludes the paper.

II. Related Work

Driven by the demands of contemporary new applications, both network coding and downlink multicast scheduling are hot topics in recent wireless networking research. In the following, we summarize existing research results in these two topics that are related to our work.

In the research of downlink multicast, Lee et al [8] establish several strong laws of large number to characterize asymptotic throughput in a wireless multicast system using incremental redundant packets for forward error correction (FEC). This work provides deep insights into how channel error, client number, and network throughput interact with each other. However, their emphasis is to characterize asymptotic throughput region and the corresponding optimal FEC code rate without considering delay and buffer constraints. In this research, network coding, instead of FEC, is adopted. Network coding's rateless nature avoids the process of determining optimal code rate. We focus on constructing optimal network codes under different QoS constraints.

In [9], the authors propose a dynamic rate adaptation algorithm to optimize the average throughput subject to the statistical QoS constraints in wireless multicast using feedbacks of channel state information from the receivers. They apply the results from link layer effective capacity to establish the impact of the delay requirements on multimedia data rate. However, no packet level diversity coding scheme, such as network coding is introduced in their work to mitigate independent random packet transmission errors.

In [10], the authors formulate wireless multicast as a restless bandit problem, which is PSPACE-complete. They simplify the problem using Whittle relaxation and show that there exists an index policy to solve the relaxed problem. Their solution requires each client's state information at each transmission time. In wireless multicast, acquiring full state information costs significant overhead and makes the system ill scaled, although an estimation mechanism is proposed to reduce feedback cost. In [11][12], the authors use LDP to analyze the behavior of stochastic networks with different scheduling algorithms for downlink unicast. In [13], the authors propose greedy algorithms for real-time video multicast in WiMAX networks. They adopt layer video coding and adaptive modulation and coding (AMC). They aim at fair allocation of resources to maximize utility function which is defined to reflect video quality. They assume that link layer can provide static bandwidth resources according to different physical layer mode without considering link layer issues such as queueing delay and queueing stability. In contrast, the theme of this research is to provide guaranteed link layer bandwidth to upper layer application.

In [2], the authors prove that random linear network coding, can asymptotically approach network capacity, and discuss its application in wired and wireless networks. Random network coding avoids complex scheduling algorithm [14] and centralized network code construction process by translating them into local addition and multiplication routines over Galois field $\mathbb{F}_q$. In contrast to scheduling algorithms, the coding approach scales well to client number. It does not require clients to feed back their instantaneous link conditions and state information, i. e. whether a packet has been received by a client or not, at each transmission time to achieve optimal throughput. These properties enable random network coding to be widely used in various practical systems [15][16]. In this research, we also adopt random network coding as the cornerstone of our work.

Recently, rateless digital fountain codes such as LT codes and Raptor codes [17] are introduced in wireless communication. In fact, fountain codes can be seen as a special form of network code over Galois field $\mathbb{F}_2$, which is optimized to achieve linear encoding and decoding complexity. Fountain codes are particularly suitable for bulk data distribution. In [18], the author propose a rate adaptation scheme to maximize multicast throughput with the help of fountain codes. However, to achieve good performance, fountain codes' code length are usually large, which makes them unsuitable for applications with stringent QoS requirements. In [4], the authors report that throughput and delay improvements of network coding in wireless multicast as compared with scheduling approaches. In [19], the authors propose a heuristic algorithm to calculate check nodes' degree distribution that can improve network throughput while meeting packet loss rate requirements.

These existing coding approaches propose different algorithms to maximize network throughput. Generally speaking, unlike channel aware scheduling, coding based approaches do not require each client's state at the transmitter. This feature avoids the ACK explosion problem and makes the system simple and well scaled. However, in practice, it is equal important to support a variety of QoS levels as to achieve network capacity. To the best of our knowledge, no existing work has been reported on how network coding can affect the dynamic behavior of wireless networks and how much network coding gain we can expect to achieve when QoS requirements are imposed. In this paper, we answer these two questions in the scenario of wireless downlink multicast with one access point (AP) broadcasting to a group of clients. The main results obtained in this research provide a foundation for future network coding based multicast protocol design and research on QoS-driven network coding in multihop wireless networks.

III. System Model and Problem Formulation

A. System Model

We study downlink multicast in a slotted time division multiplexed wireless LAN consisting of one wireless AP and N clients, as illustrated in Fig. 1. The AP continuously broadcasts packets to the N clients. We focus on a single multicast group, so multicasting and broadcasting mean the same, and will be used interchangeably. The set of clients is denoted by $\mathcal{N} = \{R_1, R_2, \cdots, R_N\}$. The input traffic is assumed to be deterministic with constant rate a packet per time slot (pps). We model the communication channel as Bernoulli ON-OFF channel. When the channel is ON, packets can be received without error. When the channel is OFF, packets cannot be

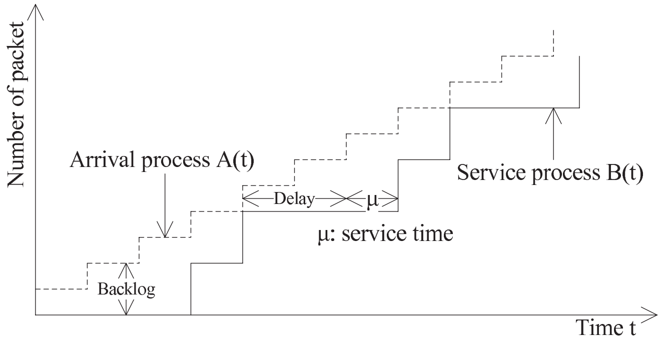


Fig. 2. Traffic characterization.

successfully received. Whether the channel is ON or OFF is independent across different clients and transmission slots. The OFF probability from the AP to R_i is denoted by c_i, $1 \le i \le N$, which is assumed to be available at the AP. This simple yet practical ON-OFF channel model has been widely used in theoretic analysis [10]. The AP only knows each client's average packet error rate c_i, but is not able to predict the channel condition of the next time slot. This assumption comes from practical consideration in multicast scenario because obtaining each receiver's channel side information before each transmission time costs significant overhead. Results obtained under this model are ready to be extended to more general finite-state Markov channel model [20][21].

Packets are denoted by $\mathbf{P}_i$, $i \in \mathbb{N}$, where $\mathbb{N} = \{0, 1, 2, \cdots\}$. We assume that all packets have the same length of L_P bits, so that they can be combined together by random network coding without padding. Certain number of successive packets starting from the head of AP's queue are treated as a *generation*. Each generation contains L packets. The AP's queue is managed with first-come-first-serve policy, in which packets within the same generation are batch served. The AP is blocked when its buffer size is smaller than L and will advance to the next generation when all clients have successfully decoded all packets in the current generation. In other words, a generation of packets will be removed together from the AP's buffer when they have been successfully decoded by all clients. Packets within the m^{th} generation are sent using random network coding. The transmitted packet at slot t can be expressed as:

$$\mathbf{P}(t) = \sum_{k=L_{b,m}}^{L_{e,m}} a_k(t)\mathbf{P}_k \tag{1}$$

where $t \in \mathbb{N}$ is time slot index, $a_k(t)$ is network coding coefficient, chosen uniformly from Galois field with size q, denoted as $\mathbb{F}_q$. $L_{b,m}$ and $L_{e,m}$, $0 \le L_{b,m} \le L_{e,m}$, are minimum and maximum packet indices of the m^{th} generation. Therefore, network code length $L = L_{e,m} - L_{b,m} + 1$. A client does not acknowledge every single packet, rather, it sends an ACK immediately after it has successfully decoded all the L packets in current generation. Note that choosing field size q will introduce $\lceil L \log_2 q \rceil$ bits overhead in each packet due to the space needed for storing the L coefficients $a_k(t)$ in the packet head.

B. Problem Formulation

Definition 1 (Backlog and Delay [22]): Let AP's input process be $A(t)$ and output process be $B(t)$, $t \in \mathbb{N}$. $A(t)$ is the total number of packets arrived at AP from time slot 0 to t, and $B(t)$ is the accumulated number of packets departed from AP's buffer from time slot 0 to $t-1$. The *backlog* seen by the i^{th} packet is defined as:

$$q(i) = A(t-1) - B(t) \tag{2}$$

where t is the i^{th} packet's arrival time. The backlog includes the packets that are currently being served but does not include the currently arrived packets themselves. The *delay* of the i^{th} packet is defined as:

$$d(i) = \inf\{\tau - \mu_i : B(t+\tau) - A(t) \ge 0\} \tag{3}$$

where $d(i)$ is the queueing time of the i^{th} packet who enters the queue at time t, and μ_i is its service time. Note that from the above definition, the delay does not include the service time. These two concepts are illustrated in Fig. 2.

Delay and backlog are two important metrics for queueing system. They are widely used to evaluate communication quality. In networking protocol design, besides the average delay and average backlog, we are also interested in system's deviation behavior, which is related to QoS constraints. We use (d_{max}, p_d) and (q_{max}, p_q) to parameterize QoS requirements. d_{max} and q_{max} are given positive constants, representing the maximum tolerable delay and backlog. $p_d, p_q \in [0, 1]$ are given parameters. The QoS-driven network coding problem is to select optimal code length L that maximize the supported source rate a under backlog and delay constraints. Using notations defined above, the problem is formulated as follows:

$$\begin{array}{ll} \text{Find} & L^* = \arg\max_{L \in \mathbb{N}^+}\{\, a \,\} \\ \text{Subject to} & \mathsf{P}_\mathsf{r}(q(\infty) > q_{max}) \le p_q \text{ and/or} \\ & \mathsf{P}_\mathsf{r}(d(\infty) > d_{max}) \le p_d \end{array} \tag{4}$$

where $\mathbb{N}^+ = \mathbb{N}\backslash\{0\}$, $q(\infty)$ and $d(\infty)$ are random variables denoting the steady-state backlog and delay, incurred by a typical customer. AP's buffer overflow constraint $\mathsf{P}_\mathsf{r}(q(\infty) > q_{max}) \le p_q$ means that the probability of the steady-state backlog exceeding q_{max} is no larger than p_q. The delay violation constraint $\mathsf{P}_\mathsf{r}(d(\infty) > d_{max}) \le p_d$ guarantees that the probability of the steady-state waiting time longer than d_{max} is upper bounded by p_d. Note that even though it is not explicitly expressed, the maximal supported source rate a is related to the packet size L_P, network code length L, the client number N, and their link conditions c_i.

IV. Throughput, QoS, and Network Code

A. QoS and LDP

Strictly speaking, it is impossible to achieve QoS guarantee under time varying wireless channels. There is always a nonzero probability that queue overflow and delay violation happen. To overcome this difficulty, the concept of statistical QoS is proposed for wireless networks [7][23][24][25][26][9]. The idea behind statistical QoS is to bound the delay violation and queue overflow probability within a given threshold. However, the complexity of wireless channels and communication protocols often prevent the approach of tracking delay violation or queue overflow probabilities. LDP is proposed as an effective technology to obtain an approximated solution. In [26], the author proves under mild assumptions that both

TABLE I
NETWORK CODE OVERHEAD

$P_r(\Phi_L > L)$	$q = 2^4$	$q = 2^6$	$q = 2^8$	$q = 2^{12}$
$L = 10$	0.0676	0.0164	0.0039	0.0003
$L = 20$	0.0676	0.0177	0.0041	0.0002
$L = 40$	0.0662	0.0166	0.0028	0.0001
$L = 80$	0.0679	0.0159	0.0038	0.0003
q^{-1}	0.0625	0.0156	0.0039	0.0002

backlog and delay satisfies LDP. They can be approximated as:

$$\mathsf{P_r}(q(\infty) > x) \approx e^{-\theta_Q x}, \text{ for sufficiently large } x \quad (5)$$

and

$$\mathsf{P_r}(d(\infty) > y) \approx e^{-\theta_D y}, \text{ for sufficiently large } y \quad (6)$$

where θ_Q and θ_D can be determined by the input process and the service process using LDP. These two parameters characterize the asymptotic deviation behavior of $q(\infty)$ and $d(\infty)$. When θ_Q is large, the decaying rate of buffer overflow probability is high, which means that a good QoS is guaranteed. For extreme case, when $\theta_Q \to \infty$, the statistical QoS approaches to the traditional QoS with deterministic sharp backlog bound. When $\theta_Q \to 0$, no QoS requirement is imposed. We can draw similar conclusion for delay violation. Following the term in [7], θ_Q and θ_D are called QoS exponents.

B. QoS Driven Network Coding

In this section, we establish the relationship among client number, link conditions, network code and QoS exponents. First, we formally define QoS exponents as a function of code length L:

$$\theta_Q(L) \triangleq -\lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(q(\infty) > x) \quad (7)$$

$$\theta_D(L) \triangleq -\lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(d(\infty) > x) \quad (8)$$

Before calculating the QoS exponents, let us first characterize the overhead of random network coding coming from the inequality between L and Φ_L, where Φ_L is the number of coded packets required to decode the L source packets. Due to the randomized encoding process, Φ_L is a random variable. However, the following lemma reveals that for practical field size, this number is highly concentrated on L. This result suggests that it is safe to assume that a client can decode the source packets with L network coded packets.

Lemma 1: Let Φ_L denote the packet number required to successfully decode L source packets. Assume that each transmitted packet's network coding coefficients are never all zero. Φ_L converges to L in probability:

$$\Phi_L \xrightarrow{\mathsf{P_r}} L, \quad as\ q \to \infty \quad (9)$$

The rate of the convergence can be determined as follows:

$$\mathsf{P_r}(\Phi_L > L) = O(q^{-1}) \quad (10)$$

The proof is given in Appendix A. Table I gives simulation results for typical field size and network code length. They verifies Lemma 1 in that the random network coding overhead due to singularity is approximately independent with the code length L and the probability that decoder requires more than L packets approximately equals to q^{-1}. In practical network coding based multicast protocols, q is usually chosen to be 2^8 or 2^{12}. For these two typical field size settings, the overhead is less than 0.1%, which can be safely neglected. Based on this result, we simplify the problem (4) by assuming that $\Phi_L \equiv L$. Note the portion of the packet head storing the networking coding coefficient is $\frac{\lceil L \log_2 q \rceil}{L_P}$. If the packet size L_P is sufficiently large, this overhead can be small. However, for some applications such as VoIP, whose typical packet size is around 40 – 60 bytes, such overhead can be significant. Therefore, we have to take into account these overhead bits in network code optimization.

Equipped with these assumptions, the following theorem determines QoS exponents.

Theorem 1: The code length L, client number N, link condition c_i, $1 \le i \le N$, and QoS exponents θ_Q, θ_D satisfy:

$$\theta_Q(a, L) = \theta^* \quad (11)$$

$$\theta_D(a, L) = a\theta^* \quad (12)$$

where a is the source's arrival rate and $\theta^* > 0$ is the largest solution to the following equation:

$$\sum_{k=0}^{\infty} e^{ka\theta}(\prod_{i=1}^{N} I_{1-c_i}(L, k+1) - \prod_{i=1}^{N} I_{1-c_i}(L, k)) = e^{\theta L(1-a)} \quad (13)$$

$I_p(p, q)$ is regularized incomplete beta function (see Appendix B). Note that (13) having no positive solution means the system cannot support source rate a.

Proof: We denote by $T(i), i \in \mathbb{N}$ the interarrival time of the i^{th} packet, i. e. the time interval between the arrival epochs of the i^{th} packet and the $(i-1)^{th}$ packet. We denote by $D'(i), i \in \mathbb{N}$ the service time of the i^{th} generation. Define

$$T'(i) = \sum_{j=iL}^{(i+1)L-1} T(i) \quad (14)$$

then $T'(i)$ can be seen as the interarrival time of a batch arrival process in which every L packets are batched and arrived together. As we assume that the BS is blocked when it has less then L packets, the batch arrival process and the original arrival process sends the same packets from the BS at any time slot. If we treat a generation of packets as a single element, and denote by $q'(i)$ and $d'(i)$ the backlog and delay of the i^{th} element in the batched arrival process. Recall that $q(i)$ and $d(i)$ are the backlog and delay of the i^{th} packet in the original process. The interval between the arrival time of packet i and packet $j = L\lfloor \frac{i}{L} \rfloor + L - 1$ is upper bounded $\lceil \frac{L}{a} \rceil$. From the definition, the arrival time of packet j is the same as the arrival time of the $\lfloor \frac{i}{L} \rfloor^{th}$ element in the batched arrival process. We have

$$d'\left(\left\lfloor \frac{i}{L} \right\rfloor\right) \le d(i) \le d'\left(\left\lfloor \frac{i}{L} \right\rfloor\right) + \left\lceil \frac{L}{a} \right\rceil \quad (15)$$

During the interval $\lceil \frac{L}{a} \rceil$, at most $\left\lceil \frac{\lceil \frac{L}{a} \rceil}{L} \right\rceil$ generations can be served. We have

$$\begin{aligned} Lq'\left(\left\lfloor \frac{i}{L} \right\rfloor\right) \le q(i) &\le Lq'\left(\left\lfloor \frac{i}{L} \right\rfloor\right) + L\left\lceil \frac{\lceil \frac{L}{a} \rceil}{L} \right\rceil + L \\ &\le Lq'\left(\left\lfloor \frac{i}{L} \right\rfloor\right) + \left\lceil \frac{L}{a} \right\rceil + 2L \end{aligned} \quad (16)$$

Assume that batched arrival process's steady-state backlog and delay, incurred by a typical element is denoted by $q'(\infty)$ and $d'(\infty)$. Then from (16)(15), the stationary distribution of the original process's backlog $q(\infty)$ and delay $d(\infty)$ satisfies:

$$Lq'(\infty) \le q(\infty) \le Lq'(\infty) + \left\lceil \frac{L}{a} \right\rceil + 2L \tag{17}$$

$$d'(\infty) \le d(\infty) \le d'(\infty) + \left\lceil \frac{L}{a} \right\rceil \tag{18}$$

From the left inequality of (17),

$$\begin{aligned}\lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(q(\infty) > x) &\le \lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(Lq'(\infty) > x)\\ &= \frac{1}{L} \lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(q'(\infty) > x)\end{aligned} \tag{19}$$

From the right inequality of (17),

$$\begin{aligned}&\lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(q(\infty) > x)\\ &\ge \lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(Lq'(\infty) + \left\lceil \frac{L}{a} \right\rceil + 2L > x)\\ &\ge \frac{1}{L} \lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(q'(\infty) > x - \frac{1}{a} - 2)\\ &= \frac{1}{L} \lim_{x\to\infty} \frac{x - \frac{1}{a} - 2}{x} \frac{1}{x - \frac{1}{a} - 2} \log \mathsf{P_r}(q'(\infty) > x - \frac{1}{a} - 2)\\ &= \frac{1}{L} \lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(q'(\infty) > x)\end{aligned} \tag{20}$$

From (19)(20),

$$\lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(q(\infty) > x) = \frac{1}{L} \lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(q'(\infty) > x) \tag{21}$$

Starting from (18), using the similar derivation as (19)(20), we have,

$$\lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(d(\infty) > x) = \lim_{x\to\infty} \frac{1}{x} \log \mathsf{P_r}(d'(\infty) > x) \tag{22}$$

From [27, Theorem 4.1], $\theta_D(a, L)$ is the largest solution that satisfies the following equation:

$$\Lambda_{T'}(-\theta) + \Lambda_{D'}(\theta) = 0 \tag{23}$$

where the asymptotic logarithmic moment generating function $\Lambda_{T'}(\theta)$ is

$$\begin{aligned}\Lambda_{T'}(\theta) &= \lim_{n\to\infty} \frac{1}{n} \log \mathsf{E} e^{\theta \sum_{i=0}^{n-1} T'(i)}\\ &= \log \mathsf{E} e^{\theta T'(0)} = \log e^{\theta \frac{L}{a}} = \theta \frac{L}{a}\end{aligned} \tag{24}$$

and $\Lambda_{D'}(\theta)$ is:

$$\begin{aligned}\Lambda_{D'}(\theta) &= \lim_{n\to\infty} \frac{1}{n} \log \mathsf{E} e^{\theta \sum_{i=0}^{n-1} D'(i)}\\ &= \log \mathsf{E} e^{\theta D'(0)} = \log \mathsf{E} e^{\theta K_{L,N}}\end{aligned} \tag{25}$$

where $K_{L,N}$ is defined in Appendix B. From (23)(24)(25)(33), we have:

$$\begin{aligned}e^{\theta \frac{L}{a}} &= \mathsf{E} e^{\theta K_{L,N}} = \sum_{k=L}^{\infty} \mathsf{P_r}(K_{L,N} = k) e^{\theta k}\\ &= \sum_{k=0}^{\infty} e^{(k+L)a\theta} (\prod_{i=1}^{N} I_{1-c_i}(L, k+1) - \prod_{i=1}^{N} I_{1-c_i}(L, k))\end{aligned} \tag{26}$$

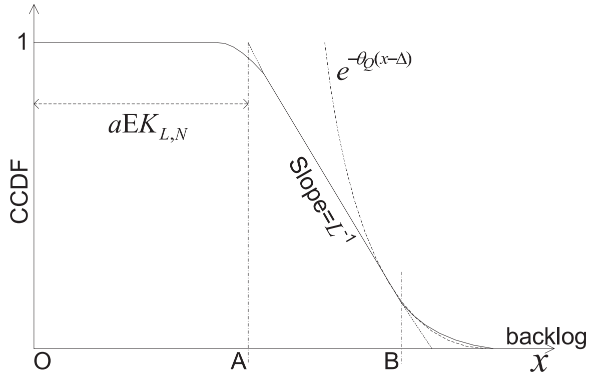


Fig. 3. Approximation of the backlog distribution.

From [27, Theorem 4.3] and (21), we have

$$\theta_Q(a, L) = \frac{1}{L} \Lambda_{T'}(\theta_D(a, L)) = \frac{1}{L} \theta_D(a, L) \frac{L}{a} = \frac{\theta_D(a, L)}{a} \tag{27}$$

From (27)(26), we prove that (12)(11)(13) hold. Note that the symbol definition in [27] is different from ours, so a translation is needed when applying their results. ■

Theorem 1 characterizes the large deviation behavior of the QoS-driven network coding scheme. The QoS exponent θ^* can be numerically calculated using standard nonlinear optimization algorithms.

C. Approximation Algorithm

Although (13) gives the condition that θ^* should satisfy, the exponential bound (5)(6) is only accurate when deviations approach to infinite. In practical wireless communication system, we are particular interested in moderate deviation values. Depending on different applications, different modification schemes are proposed[9][7] to to compensate for the gap between LDP bound and the real queueing tail. These approaches can be generally expressed as multiplying the exponential expression with a modification term. For example, the backlog deviation (5) can be approximated as:

$$\mathsf{P_r}(q(\infty) > x) \approx \beta(\theta_Q) e^{-\theta_Q x}$$

However, the batch service character of network coding differentiates itself from existing approaches. From queueing theory's perspective, determining the exact expression of $\beta(\cdot)$ is mathematically intractable. To approximate the deviation behavior of QoS-driven network coding, we propose to use the following empirical equations:

$$\begin{aligned}&\mathsf{P_r}(q(\infty) > x)\\ &\approx \begin{cases} 1, & x < a\mathsf{E}K_{L,N};\\ 1 - \frac{x - a\mathsf{E}K_{L,N}}{L}, & a\mathsf{E}K_{L,N} \le x \le a\mathsf{E}K_{L,N} + L - \theta_Q^{-1};\\ e^{-\theta_Q(x-\Delta_Q)}, & x > a\mathsf{E}K_{L,N} + L - \theta_Q^{-1}. \end{cases}\end{aligned} \tag{28}$$

and

$$\mathsf{P_r}(d(\infty) > y) \approx \begin{cases} 1 - xaL^{-1}, & 0 \le x \le a^{-1}L - \theta_D^{-1};\\ e^{-\theta_D(x-\Delta_D)}, & x > a^{-1}L - \theta_D^{-1}. \end{cases} \tag{29}$$

where $\mathsf{E}K_{L,N}$ is the expectation of the random variable $K_{L,N}$ defined in Appendix B. First, we show how (28) is developed and explain the meaning of the parameters in it. As illustrated in Fig. 3, the curve of the complementary cumulative distribution (CCDF) of the backlog can be approximately separated into three regions. In the first region, i. e. $x \in [O, A]$, the backlog is smaller than its average, the curve can be approximated as a horizontal line. In the second region, i. e. $x \in [A, B]$, the curve descends approximately linearly due to the batch processing character of network coding. In the third region, i. e. $x \in [B, \infty)$, the curve conforms LDP bounds. Our task is to determine the turning points A and B. Point A can be heuristically calculated as

$$A = a\mathsf{E}K_{L,N} \tag{30}$$

which is approximately equal to the number of packets arrived at the BS when it is serving the L current generation packets. When $x \in [A, B]$, the CCDF curve can be approximated with a linear function whose slope equals to L^{-1}. For the tail region, we use the LDP exponential formula as an approximation. To determine the turning point B, we impose first order derivative continuity constraint, i. e. the linear and exponential curve have both the same CCDF value and the same first order derivative at point B. We have

$$B = a\mathsf{E}K_{L,N} + L - \theta_Q^{-1} \tag{31}$$

Then the parameter Δ can be determined according to the position of point B as

$$\Delta_Q = B - \theta_Q^{-1}\log(\theta_Q L) \tag{32}$$

Expression (29) is derived similarly, except that the curve of the CCDF of delay only contains the linear drop region and the LDP region. By forcing one order derivative continuity at the turning point, the turning point's position can be calculated as $a^{-1}L - \theta_D^{-1}$. And the parameter $\Delta_D = a^{-1}L - \theta_D^{-1} - \theta_D^{-1}\log(\theta_D L a^{-1})$.

Now, we are ready to solve the code optimization problem (4). Due to the overhead of the network coding coefficients, the effective length of one transmitted packet is $L_P - \lceil L\log_2 q\rceil$ bits. From (33) in Appendix B, the average service rate $L(1 - \frac{\lceil L\log_2 q\rceil}{L_P})/\mathsf{E}K_{L,N}$ at first increases due to the benefits of network coding and then decrease due to the overhead with respect to L. Further, if code length L can support source rate a, it must be able to support any source rate smaller than a. These facts suggest that the optimal L that maximizes a in (4) can be derived using two steps. The first step is to determine the maximal supported source rate a_u without considering the QoS requirements. The second step is to find the maximal source rate $a \in [a_l, a_u]$ and the corresponding code length L under QoS requirements. We summarize QoS-driven network coding in Algorithm 1.

Algorithm 1 QoS-driven network coding

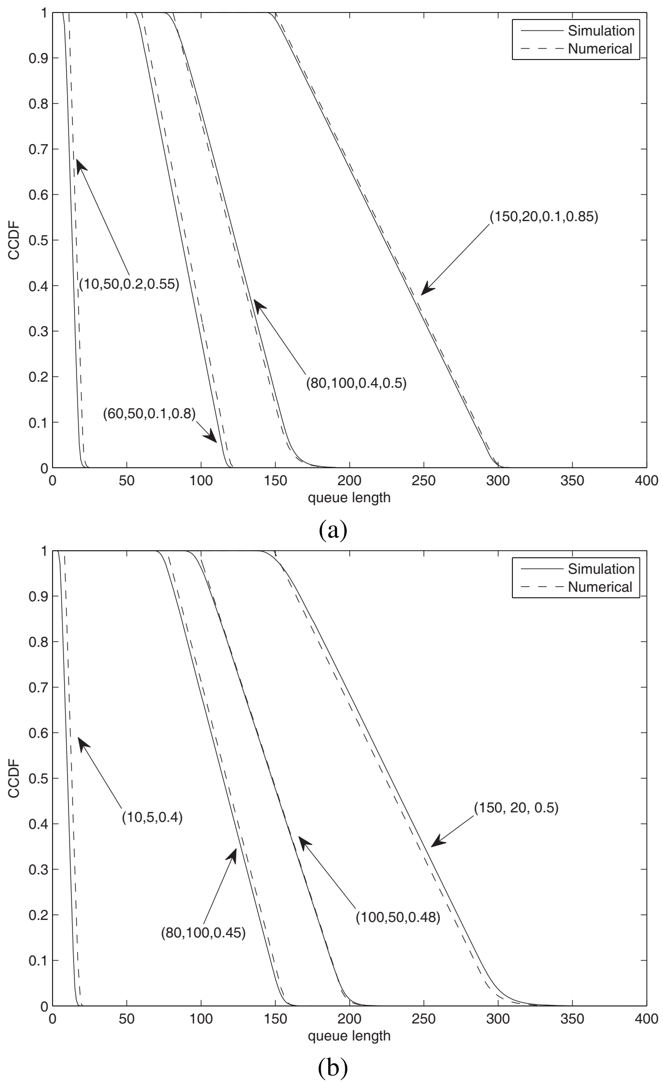
```
 1: procedure QoS_NC($(c_1, c_2, \cdots, c_N), (d_{max}, p_d), (q_{max}, p_q)$)
 2:     $L_{max} \leftarrow q_{max},\ a_l \leftarrow 1/\mathsf{E}K_{1,N}$
 3:     $a_u \leftarrow L_{max}(1 - \frac{\lceil L_{max}\log_2 q\rceil}{L_P})/\mathsf{E}K_{L_{max},N}$      ▹ Initialization
 4:     $L'_{max} = L_{max} - \Delta$      ▹ Constant $\Delta$ is the step size
 5:     $a'_u \leftarrow L'_{max}(1 - \frac{\lceil L'_{max}\log_2 q\rceil}{L_P})/\mathsf{E}K_{L'_{max},N}$
 6:     while $a'_u > a_u$ do      ▹ Within descending region
 7:         $a_u \leftarrow a'_u,\ L_{max} \leftarrow L'_{max}$
 8:         $L'_{max} = L_{max} - \Delta,\ a'_u \leftarrow L'_{max}(1 - \frac{\lceil L'_{max}\log_2 q\rceil}{L_P})/\mathsf{E}K_{L'_{max},N}$
 9:         if $L_{max} \le 1$ then
10:             $L \leftarrow 1,\ a \leftarrow a_l$, goto line 30
11:         end if
12:     end while
13:     $L_0 \leftarrow 1,\ L \leftarrow L_{max}$
14:     while $a_u - a_l > \epsilon$ do      ▹ $\epsilon$ is a small positive constant
15:         $a \leftarrow 0.5(a_l + a_u),\ L \leftarrow L_0$
16:         Calculate $\theta_D(a/(1 - \frac{\lceil L\log_2 q\rceil}{L_P}), L)$      ▹ Using (12)
17:         Calculate $\theta_Q(a/(1 - \frac{\lceil L\log_2 q\rceil}{L_P}), L)$      ▹ Using (11)
18:         while $\begin{pmatrix} \mathsf{P_r}(q(\infty) > q_{max}) > p_q \,\| \\ \mathsf{P_r}(d(\infty) > d_{max}) > p_d \end{pmatrix}$ && $L \le q_{max}$ do
19:                                                  ▹ Using (28)(29)
20:             $L \leftarrow L + \Delta$
21:             Calculate $\theta_D(a/(1 - \frac{\lceil L\log_2 q\rceil}{L_P}), L)$
22:             Calculate $\theta_Q(a/(1 - \frac{\lceil L\log_2 q\rceil}{L_P}), L)$
23:         end while
24:         if $L > q_{max}$ then      ▹ Update $a_u$, $a_l$, $L_0$
25:             $a_u \leftarrow a$
26:         else
27:             $a_l \leftarrow a,\ L_0 \leftarrow L$
28:         end if
29:     end while
30:     return $L$, $a$      ▹ Code length and maximal rate
31: end procedure
```

V. Performance Evaluation

In this section, we numerically evaluate QoS-driven network coding scheme under different client number, link conditions, and QoS requirements. All of the experiments use finite field size 2^8.

First, we investigate the effectiveness of using LDP to approximate the backlog and delay distribution. We study both homogeneous and heterogeneous link conditions. In the homogeneous case, the curves are indexed by 4-tuple (L, N, c, a), where c is the common PER of the clients. In the heterogeneous case, links' PER are generated according to uniform distribution with range $[0.1, 0.4]$ and therefore the curves are indexed by triple (L, N, a). Fig. 4 illustrates the CCDF of the backlog and Fig. 5 illustrates the CCDF of the delay. The curves with legend *numerical* are obtained by LDP (28)(29). The curves with legend *simulation* are obtained by Monte Carlo simulations. These experiment results show that the LDP approach can approximate backlog and delay distribution with high accuracy. Therefore, it is reasonable to use LDP approximation to determine the optimal network code length.

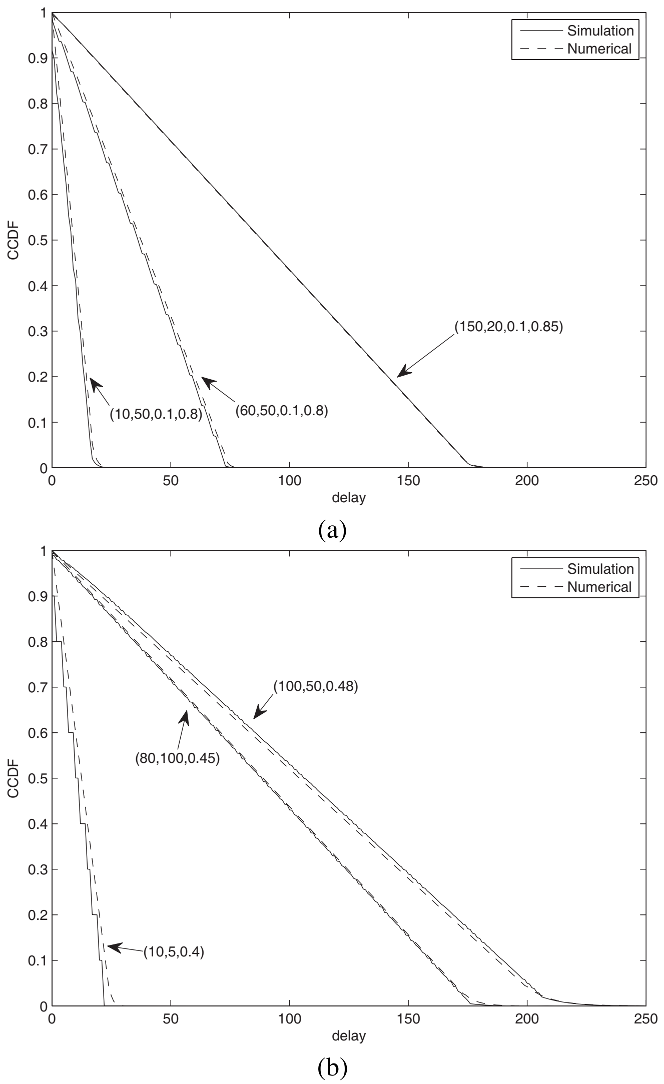
The second group of experiments present the results of network code optimization, obtained by the QoS-driven network coding algorithm (Algorithm 1). The network code length is updated with step size 5. The results are summarized in



(a)

(b)

Fig. 4. CCDF of the backlog in homogeneous networks with common link PER c (a) and in heterogeneous networks with link PER uniformly distributed with range $[0.1, 0.4]$ (b).



(a)

(b)

Fig. 5. CCDF of the delay in homogeneous networks with common link PER c (a) and in heterogeneous networks with link PER uniformly distributed with range $[0.1, 0.4]$ (b).

Table II. As the packet size can affect the optimal network code length, we study two typical packet length settings, i. e. $L_P = 2 \times 10^5$ bits and $L_P = 400$ bits. The first packet length setting represents the scenarios that can support long packet while the second one presents the scenarios with restricted packet length, such as VoIP. In Table II, each line corresponds to a typical topology and link configuration, including receiver number N, link error rate c if the links are homogeneous. If the links are heterogenous, link error rates are generated according to uniform distribution with range $[0.1, 0.4]$. Each column corresponds to a typical set of QoS constraints (q_{max}, p_q) and (d_{max}, p_d). The results are presented as (a, L), where a is the maximal supportable source rate, L is the optimal network code length.

From Table II, comparing the results under the same topology and different source packet length, we can see that short packet length can significantly reduce the maximal supportable source rate. When $L_P = 2 \times 10^5$ bits, there is a trade off between QoS constraints and maximal supportable throughput. When we require more stringent QoS guarantee, the average throughput decreases. However, when $L_P = 400$ bits, in the first three network configurations, the maximal throughput does not change when QoS becomes more stringent. The reason is that when the packet length is small, as illustrated before, due to the overhead of storing the network coding coefficients, there is a peak of the throughput when L becomes large. If the throughput corresponding to the peak can support the QoS requirements, both the maximal supportable source rate and the optimal network code length will stay on that point. When the packet is long enough, such impact is not significant. Note that we adjust the code length with step size 5, the optimal code length is not changed if the optimal code length has only very small change. Further, due to the nonlinear relationship among client number, link conditions, maximal throughput, and QoS requirements, there is no directly sequential relationship of the code length under different network configurations (different rows in Table II).

VI. Conclusion

For modern network protocol design, providing QoS guarantee is as important as maximizing throughput. Network coding has been proved to be an efficient technology to improve

TABLE II
PERFORMANCE OF QOS DRIVEN NETWORK CODING.

a, L	$L_P = 2\times 10^5$ bits		$L_P = 400$ bits	
	$(q_{max}, p_q) = (200, 10^{-4})$ $(d_{max}, p_d) = (180, 10^{-4})$	$(q_{max}, p_q) = (200, 10^{-6})$ $(d_{max}, p_d) = (180, 10^{-6})$	$(q_{max}, p_q) = (200, 10^{-4})$ $(d_{max}, p_d) = (180, 10^{-4})$	$(q_{max}, p_q) = (200, 10^{-6})$ $(d_{max}, p_d) = (180, 10^{-6})$
$N = 100, c = 0.4$, homogeneous	0.49, 71	0.48, 66	0.29, 11	0.29, 11
$N = 40, c = 0.2$, homogeneous	0.71, 76	0.70, 66	0.46, 6	0.46, 6
$N = 60$, heterogeneous	0.46, 46	0.45, 41	0.31, 6	0.31, 6
$N = 120$, heterogenous	0.45, 51	0.44, 41	0.29, 6	0.28, 6

throughput. To put network coding theory into practice, in this paper, we addressed the network coding problem in downlink wireless multicast under stochastic QoS constraints. We identified the relationship among network throughput, network code length, client number, and link conditions. Using this relationship, we proposed a practical algorithm to calculate the optimal network code length, with which maximal source rate can be supported. Although we have assumed that the source rate is constant, our results are ready to be extended to more general stochastic source traffic models. QoS-driven network coding does not require frequent measurement of physical layer information. Only links' average PERs are required for determining the optimal code length. This feature makes it easy to be integrated into existing protocol stack.

In this research, we focus on intra session network coding in single session multicast. When there are multiple multicast sessions, the optimal policy to perform QoS-driven inter session network coding is still an open problem. Furthermore, if AP can explore certain degree of channel side information, designing optimal QoS-driven network code under advanced physical layer technologies such as AMC to enhance communication quality is left to the future work.

APPENDIX A
PROOF OF LEMMA 1

In this appendix, we prove Lemma 1. As we focus on one particular generation, (1) can be simplified as:

$$\mathbf{P}(t) = \sum_{k=0}^{L-1} a_k(t)\mathbf{P}_k$$

Let random variable $\Phi_L(q)$ be the number of coded packets required to successfully decode the L source packets. We use the term *knowledge space* at time t to denote the linear space spanned by $\{\mathbf{P}(i),\ 0 \le i \le t\}$. Let $N_{k,L}$ be the number of packets required to make the knowledge space's dimension grow from $k-1$ to k. From the channel model, $N_{k,L}$ is geometric distributed. As we forbid all zero code word, i. e., $\prod_{k=0}^{L} a_k(t) \neq 0$, $N_{k,L}$'s distribution is:

$$\mathsf{P_r}(N_{k,L} = i) = \begin{cases} 1_{[i=1]}, & k = 1; \\ \frac{q^L - q^{k-1}}{q^L - 1}\left(\frac{q^{k-1}-1}{q^L-1}\right)^{i-1}, & 1 < k \le L. \end{cases} \quad \forall i \in \mathbb{N}^+.$$

where 1_X is the indicator function that takes value 1 when condition X is satisfied and 0 otherwise. Because $\Phi_L(q) = \sum_{k=1}^{L} N_{k,L}$ and $N_{k,L} \ge 1$, we have the following relation:

$$\begin{aligned}\mathsf{P_r}(\Phi_L(q) > L) &\le \sum_{k=1}^{L} \mathsf{P_r}(N_{k,L} > 1) = \sum_{k=1}^{L}(1 - \mathsf{P_r}(N_{k,L} = 1)) \\ &= \sum_{k=0}^{L-1} \frac{q^k - 1}{q^L - 1} \le \sum_{k=0}^{L-1} \frac{q^k}{q^L} = \sum_{k=1}^{L} q^{-k}\end{aligned}$$

And:

$$\begin{aligned}\mathsf{P_r}(\Phi_L(q) > L) > \mathsf{P_r}(N_{L,L} > 1) &= 1 - \mathsf{P_r}(N_{L,L} = 1) \\ &= \frac{q^{L-1} - 1}{q^L - 1} = \frac{1}{q} - \frac{1}{q + q^2 + \cdots + q^L} \\ &\ge q^{-1} - q^{-L}\end{aligned}$$

From the above two bounds, we get:

$$\lim_{q\to\infty} \mathsf{P_r}(\Phi_L(q) \neq L) = 0$$

i. e. $\Phi_L(q) \xrightarrow{\mathsf{P_r}} L$ and the deviation satisfies:

$$\mathsf{P_r}(\Phi_L(q) > L) = O(q^{-1})$$

APPENDIX B

In this appendix, we establish some auxiliary results that will be used in the main content. Assume that there are L packets $\mathbf{P}_k, 0 \le k \le L-1$ to be transmitted to N clients. Let random variable K_L^i, $1 \le i \le N$ denote the number of coded packet transmitted from the AP for client i to decode the L source packets. Let $K_{L,N} \triangleq \max\{K_L^i, 1 \le i \le N\}$ denote the number of coded packet the AP needs to send for all clients to decode the source packets. From Lemma 1, we assume that each client needs to successfully receive L packets. According to our ON-OFF channel model, K_L^i is the sum of L identical independent geometric distributed random variables. Using X_i to denote any of these random variables, we have

$$\mathsf{P_r}(X_i = j) = c_i^{j-1}(1 - c_i), \quad j \ge 1$$

As the sum of geometric distributed variables conforms negative binomial distribution. The probability mass function of K_L^i is:

$$\mathsf{P_r}(K_L^i = k) = \binom{k-1}{L-1} c_i^{k-L}(1 - c_i)^L, \quad k \ge L$$

The probability mass function of $K_{L,N}$ can be expressed as:

$$\begin{aligned}
&\mathsf{P_r}(K_{L,N} = k) = \mathsf{P_r}(\max_{1\le i\le N} K_L^i = k)\\
=&\mathsf{P_r}(K_L^i \le k, 1 \le i \le N) - \mathsf{P_r}(K_L^i \le k-1, 1 \le i \le N)\\
=&\prod_{i=1}^{N} \mathsf{P_r}(K_L^i \le k) - \prod_{i=1}^{N} \mathsf{P_r}(K_L^i \le k-1)\\
=&\prod_{i=1}^{N} I_{1-c_i}(L, k-L+1) - \prod_{i=1}^{N} I_{1-c_i}(L, k-L)), \quad k \ge L \quad (33)
\end{aligned}$$

where $I_p(a,b) \triangleq \frac{\int_0^p t^{a-1}(1-t)^{b-1}dt}{\int_0^1 t^{a-1}(1-t)^{b-1}dt}$ is regularized incomplete beta function. In the above induction, we use the fact that K_L^i are independent.

Acknowledgment

The authors would like to thank the reviewers for pointing out that in certain applications, the bits in the packet head used for storing network coding coefficients cannot be neglected.

References

[1] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[2] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Weung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[4] A. Eryilmaz, A. Ozdaglar, and M. Médard, "On delay performance gains from network coding," in *Proc. 40th Annu. IEEE Conf. Inform. Sci. Syst. (CISS'06)*, Princeton, NJ, USA, Mar. 2006, pp. 864–870 (invited).

[5] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1906–1918, Nov. 2005.

[6] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, 2nd ed. New York: Springer-Verlag, 1998.

[7] D. Wu and R. Negi, "Effective capacity: a wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, July 2003.

[8] I.-C. Lee, C.-S. Chang, and C.-M. Lien, "On the throughput of multicasting with incremental forward error correction," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 900–918, Mar. 2005.

[9] X. Zhang and Q. Du, "Cross-layer modeling for QoS-driven multimedia multicast/broadcast over fading channels in mobile wireless networks," *IEEE Commun. Mag.*, vol. 45, no. 8, pp. 62–70, Aug. 2007.

[10] V. Raghunathan, V. Borkar, M. Cao, and P. R. Kumar, "Index policies for real-time multicast scheduling for wireless broadcast systems," in *Proc. 27th Annu. IEEE Conf. Comput. Commun. (INFOCOM'08)*, Phoenix, AZ, USA, Apr. 2008, pp. 1570–1578.

[11] S. Shakkottai, "Effective capacity and QoS for wireless scheduling," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 749–761, Apr. 2008.

[12] L. Ying, R. Srikant, A. Eryilmaz, and G. E. Dullerud, "A large deviations analysis of scheduling in wireless networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 5088–5098, Nov. 2006.

[13] S. Deb, S. Jaiswal, and K. Nagaraj, "Real-time video multicast in wimax networks," in *Proc. 27th Annu. IEEE Conf. Comput. Commun. (INFOCOM'08)*, Phoenix, AZ, USA, Apr. 2008, pp. 1579–1587.

[14] S. Gaurav, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. ACM MobiCom'06*, Los Angeles, CA, Sept. 2006, pp. 227–238.

[15] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Allerton Conf. Commun., Control, and Comput.*, Monticello, IL, USA, Oct. 2003 (invited).

[16] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. 24th Annu. IEEE Conf. Comput. Commun. (INFOCOM'05)*, Miami, FL, USA, Mar. 2005, pp. 2235–2245.

[17] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[18] U. C. Kozat, "On the throughput capacity of opportunistic multicasting with erasure codes," in *Proc. 27th Annu. IEEE Conf. Comput. Commun. (INFOCOM'08)*, Phoenix, AZ, USA, Apr. 2008, pp. 520–528.

[19] X. Zhang and Q. Du, "Adaptive low-complexity erasure-correcting code-based protocols for QoS-driven mobile multicast services over wireless networks," *IEEE Trans. Veh. Technol.*, vol. 55, no. 5, pp. 1633–1647, Sep. 2006.

[20] D. Dalalah, L. Cheng, and G. Tonkay, "Modeling end-to-end wireless lossy channels: a finite-state Markov approach," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1236–1243, Apr. 2008.

[21] M. Hassan, M. M. Krunz, and I. Matta, "Markov-based channel characterization for tractable performance analysis in wireless packet networks," *IEEE Trans. Wireless Commun.*, vol. 3, no. 3, pp. 821–831, May 2004.

[22] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. New York: Springer-Verlag, 2001.

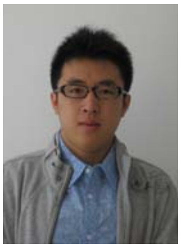
[23] C.-S. Chang, "Stability, queue length and delay of deterministic and stochastic queueing networks," *IEEE Trans. Autom. Control*, vol. 39, no. 5, pp. 913–931, May 1994.

[24] ——, "Effective bandwidth in high-speed digital networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1091–1100, Aug. 1995.

[25] C.-S. Chang and T. Zajic, "Effective bandwidths of departure processes from queues with time varying capacities," in *Proc. 14th Annu. IEEE Conf. Comput. Commun. (INFOCOM'95)*, Boston, MA, USA, Apr. 1995, pp. 1001–1009.

[26] C.-S. Chang, *Performance Guarantees in Communication Networks*. London: Springer-Verlag, 2000.

[27] D. Bertsimas, I. C. Paschalidis, and J. N. Tsitsiklis, "On the large deviations behavior of acyclic networks of G/G/1 queues," *Ann. Appl. Prob.*, vol. 8, no. 4, pp. 1027–1069, Nov. 1998.



Wei Pu (SM'09) received his BS from the University of Science and Technology of China (USTC) in 2004. He is currently pursuing a Ph.D in electrical engineering at USTC.

His research interests include multimedia streaming system (e.g. QoS assurance, user behavior analysis, network monitoring and management), coding (e.g. Error control coding, space time coding, and network coding), queueing (e.g. scheduling, congestion control, and routing).

Chong Luo (M'05) received the B.S. degree in computer science from Fudan University, Shanghai, China, in 2000. She received the M.Sc. degree in computer science from National University of Singapore, Singapore, in 2002. She is currently pursuing the Ph.D. degree in Shanghai Jiao Tong University, Shanghai, China. She is an Associate Researcher in Microsoft Research Asia. Her research interests include multimedia communications, peer-to-peer networks, and wireless sensor networks.

Feng Wu (M'99-SM'06) received his B.S. degree in Electrical Engineering from XIDIAN University in 1992. He received his M.S. and Ph.D. degrees in Computer Science from Harbin Institute of Technology in 1996 and 1999, respectively. He is now a Lead Researcher/Research Manager. His research interests include image and video representation, media compression and communication. He has authored or co-authored over 100 papers published in journals and International Conferences and Forums.

Chang Wen Chen (S'86-M'90-SM'97-F'04) received his BS from University of Science and Technology of China in 1983, MSEE from University of Southern California in 1986, and Ph.D. from University of Illinois at Urbana-Champaign in 1992. He was elected an IEEE Fellow for his contributions in digital image and video processing, analysis, and communications and an SPIE Fellow for his contributions in electronic imaging and visual communications.