Delay Analysis of Network Coding in Multicast Networks with Markovian Arrival Processes: A Practical Framework in Cache-Enabled Networks

Fatemeh Rezaei, Ahmadreza Momeni, and Babak Hossein Khalaj

***Abstract*—We develop an analytical framework for queueing and delay analysis in the case of a number of distinct flows arriving at a network node, where Asynchronous Partial Network Coding (APNC) is applied for an efficient packet transmission. In order to perfectly model and analyze the practical communication networks, the arriving flows are assumed to be general Markovian Arrival Processes (MAPs). As a key example, we apply the proposed model to the problem of coded caching in single bottleneck caching networks. In addition, we verify the accuracy of the proposed model through simulations and real trace-driven experiments.**

***Index Terms*—Markovian arrival process, queueing analysis, delay, network coding, caching.**

I. Introduction

NETWORK coding (NC) was initially introduced and shown to increase efficiency of multicast networks in [1]. Network coding can be used to improve throughput and robustness of such networks by means of algebraically combining packets that belong to different information flows passing through an intermediate network node. As a key motivating example, recently in caching networks, network coding is applied to reduce transmission load on the multicast links [2], and improve system performance [3], [4].

Delay is an important quality of service (QoS) metric in recent communication networks, such as 5G cellular networks and vehicular networks. The traffic of user requests plays an important role in delay analysis of the networks. In contrast to common independent and identically distributed (i.i.d) traffic models, such as Independent Reference Model (IRM) [5], [6], Markovian Arrival Processes (MAPs) [7] can model the requests traffic in practical networks much more accurately. In this paper, we study delay analysis of network coding in multicast networks by considering MAP traffic flows.

A. Related Works

Delay in multicast networks that apply network coding has been studied from different perspectives. For instance, the average decoding delay at receivers of a broadcast network, applying random linear network coding, is studied in [8]. The authors in [9] study a context-aware network coding and scheduling in wireless networks by considering two types of delay-sensitive and delay-tolerant network traffic. Minimizing the total transmission cost of network coded multicasting in cognitive radio networks, by considering a delay constraint, is studied in [10]. The authors in [11] propose a general framework to develop optimal and adaptive joint network coding and scheduling schemes and study the mean packet delay as a function of the throughput. Video-aware opportunistic network coding schemes that take into account the decodability of network codes by several receivers and deadlines of video packets are also studied in [12] and [13].

A key question arising in a number of problems relates to the amount of delay packets of different information flows experience in average while crossing a certain node in the network that adopts network coding approaches. Such problem has been addressed under various network scenarios and network coding schemes [14]–[18]. The work in [14] analyzes the queueing delay at a single node in a network at which inter-session network coding is performed over the flows entering the node under two different schemes referred to as synchronous and asynchronous partial NC (SNC and APNC, respectively), where the inputs are assumed to be of the renewal model. In the synchronous case, the encoding node requires each packet in every flow to be encoded with a packet of all other flows that pass through a given node. In the asynchronous case, at the beginning of an encoding process, the node picks the packets of the flows whose buffers are not empty, then combines and sends them out. Through the analysis of these two different schemes, it is shown that the synchronization requirement leads to a detrimental effect on delay. In [15], the authors analyze the queueing delay at a node at which packets arrive according to a Bernoulli process and are stored in an infinite-capacity buffer. Packets are then removed in blocks to be transmitted to multiple destinations each of which receiving the coded packets through an independent erasure channel. The transmission process for each coded packet continues until all destinations receive the packet correctly. In [16], it is shown that in a serial network with geometric input, the total transmission delay for a block of K packets is upper bounded by a constant plus K divided by the capacity of the worst-case or bottleneck link.

The aforementioned works on network coding perform their analysis under the assumption that information flows correspond to renewal arrival processes [19] in which the

This work was supported by the Iran National Science Foundation under Grant 95824827.

F. Rezaei and B. H. Khalaj are with the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran. A. Momeni is with the Department of Electrical Engineering, Stanford University, USA. e-mails:(f_rezaei@ee.sharif.edu, khalaj@sharif.edu, amomenis@stanford.edu.)

inter-arrival times of packets are i.i.d. random variables (e.g., geometric processes in the case of slotted networks or Poisson arrivals otherwise). It should be noted that such processes cannot effectively model packet flows in practical networks [5], [6]. Therefore, the effect of network coding on packet delay should be explored for the case of more practical arrival processes. The Markovian arrival process, and its subrandom processes (e.g., Markov-Modulated Poisson Process (MMPP) [20]−[23]) are good candidates for modeling such flows. In [24] and [25], a single-server queue with a Markovian arrival process is studied. Also, the authors of [17] explore the achievable rate region in a butterfly scenario when the two sources generate packets according to a two-state Markov-modulated fluid process, demonstrating that network coding outperforms traditional routing unless the network is asymmetric. Finally, in [18], the energy-delay tradeoff is investigated in a two-way relay channel where packet flows are assumed to be Markovian arrival discrete processes.

B. Contributions

In this paper, we analyze a practical multicast network where the arrival flows are independent general Markovian arrival processes and asynchronous partial network coding is applied to transmit the arrival packets. As a key motivating example, we apply this model to the case of caching networks, where some packets are cached locally in order to create coded multicast opportunities and simple packet decodablity, as will be discussed in section IV. In [2], we studied coded caching schemes in single bottleneck caching networks, by assuming IRM traffic model. Subsequently in [26], we studied heteregeneous cellular caching networks, without network coding, by considering IRM traffic. Since, the IRM traffic model is a memoryless process and ignores all temporal correlations in the stream of requests, it cannot effectively model the packet flows in practical networks [5], [6]. It should be noted that in recent communication networks, the user experienced delay is an important QoS metric and should be perfectly analyzed. However, the delay analysis results achieved under IRM traffic model cannot accurately reflect the results in an operational network. Therefore, in order to take into account a realistic traffic model, leading to reasonable analytic results, we consider the MAP model. In this paper, we analyze single bottleneck caching networks by considering MAP traffic flows and APNC, leading to a more accurate and comprehensive analysis in the case of practical scenarios. The main contributions of this paper can be summarized as follows:

- We analyze asynchronous partial network coding under Markovian arrival process flows, using matrix geometric methods [27], in order to perfectly model the practical multicast networks.
- We derive the steady-state queue lengths of the flows at arbitrary time t and present delay analysis of applying asynchronous partial network coding in a network with distinct MAP flows.
- We apply the proposed model and analysis to single bottleneck caching networks and analyze such networks by considering the practical models of MAP traffic flows and APNC.

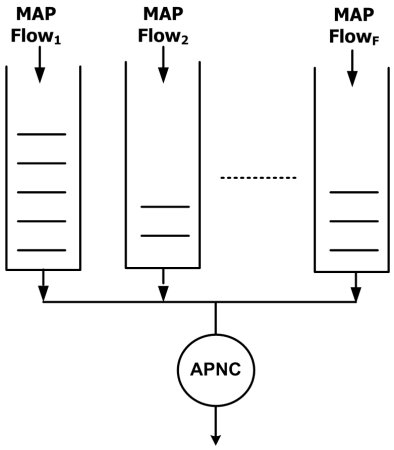

Fig. 1. A network node with F input MAP flows, applying asynchronous partial network coding.

- We show that the proposed analyis based on the MAP flows can significantly improve traffic modeling in practical networks, by running real trace-driven experiments.

The rest of the paper is organized as follows. Section II describes the system model. In section III, the main analytic approach is presented. Applying the proposed model to single bottleneck caching networks is discussed in section IV. In section V, the performance evaluation through numerical results is presented. Finally, section VI concludes the paper.

II. System Model

We consider a network node that receives packets from F distinct MAP flows. Received packets are then stored in the corresponding buffers and asynchronous partial network coding is applied at each node, as illustrated in Fig. 1. It should be noted that since synchronous network coding leads to an unstable system [2], [14], we will focus on the more practical asynchronous partial network coding case. We use the matrix geometric methods proposed in [24] and [25], where the authors studied a single-server queue with a single Markovian arrival process, in order to analyze our network of interest with F distinct MAP flows arriving at a network node where asynchronous partial network coding is applied to transmit the packets of these MAP flows.

A. Arrival Processes

The arrival process of each flow is assumed to be a general Markovian arrival process [24], described by stochastic matrices $\tilde{C}_f$ and $\tilde{D}_f$ of size $m_f \times m_f$. More specifically, for the fth flow, we consider a continuous time Markov process with m_f transient states, where the infinitesimal generator is given by $\tilde{Q}_f = \tilde{C}_f + \tilde{D}_f$. Assuming that the underlying Markov process of the fth flow is in a transient state i, at the end of the sojourn time in that state, there occurs a transition to another (or possibly the same) state, and the transition may or may not correspond to an arrival epoch. The rate at which the corresponding process transits from state i to state j with a packet arrival is given by $(\tilde{D}_f)_{ij}$. Similarly, the rate at which the corresponding process transits

TABLE I
OBTAINING THE TOTAL ARRIVAL STATE, $J_{tot}(t)$, BASED ON THE INDIVIDUAL ARRIVAL STATES, IN AN EXAMPLE WITH $F = 2$ AND $m_1 = m_2 = 2$.

$J_1(t)$	$J_2(t)$	$J_{tot}(t)$
1	1	$1 + (1-1) \times m_2 + (1-1) = 1$
1	2	$1 + (1-1) \times m_2 + (2-1) = 2$
2	1	$1 + (2-1) \times m_2 + (1-1) = 3$
2	2	$1 + (2-1) \times m_2 + (2-1) = 4$

from state i to state j with no packet arrival is given by $(\tilde{C}_f)_{ij}$, $i \neq j$. The diagonal elements of $\tilde{C}_f$ are also given by $(\tilde{C}_f)_{ii} = -(\sum_{j=1, i\neq j}^{m_f} (\tilde{C}_f)_{ij} + \sum_{j=1}^{m_f} (\tilde{D}_f)_{ij})$.

According to the model, the fth MAP flow has m_f states. The state of the fth flow at time t, which is independent of the other flows' states, is denoted by $J_f(t)$, $1 \leq J_f(t) \leq m_f$. In order to have a unique scalar state at time t, representing the states of all of the F flows at time t, we define the total arrival state at time t as $J_{tot}(t) := 1 + \sum_{f=1}^{F-1} \left((J_f(t) - 1) \prod_{k=f+1}^{F} m_k \right) + (J_F(t) - 1)$. In fact, in order to derive the scalar total arrival state at time t, we define a numerical system, in which each place is a multiplication of the number of states of the lower places. For example, the lowest place is 1, the second place is m_F, the third place is $(m_{F-1} \times m_F)$, and so on. Based on the definition, it is clear that the number of the total arrival states is $m_{tot} = \prod_{f=1}^{F} m_f$. For example, in the case that $F = 2$ and $m_1 = m_2 = 2$, $J_{tot}(t)$ is obtained according to Table I.

For the purpose of expressing the system model in terms of the total arrival states, we define matrices

$C_f = I_1 \otimes I_2 \otimes \cdots \otimes I_{f-1} \otimes \tilde{C}_f \otimes I_{f+1} \otimes \cdots \otimes I_F$,

$D_f = I_1 \otimes I_2 \otimes \cdots \otimes I_{f-1} \otimes \tilde{D}_f \otimes I_{f+1} \otimes \cdots \otimes I_F$,

where I_f stands for the identity matrix of size $m_f \times m_f$, and $\otimes$ is the Kronecker product [28] of the matrices. In fact, C_f and D_f denote the corresponding transition rates (as described in the definition of $\tilde{C}_f$ and $\tilde{D}_f$) of the Markov arrival process of the fth flow, in terms of the total arrival states.

For notational simplicity, we also define two other matrices $C_{tot} = \sum_{f=1}^{F} C_f$ and $D_{tot} = \sum_{f=1}^{F} D_f$. It should be noted that there is an underlying Markov process describing the transitions among the total arrival states $\{1, \cdots, m_{tot}\}$. The matrix $Q_{tot} = C_{tot} + D_{tot}$ is the infinitesimal generator of this underlying Markov process. Without loss of generality, we assume that Q_{tot} is irreducible. The steady-state vector π_{tot} of this Markov process satisfies equations

$$\pi_{tot} Q_{tot} = \mathbf{0}, \quad \pi_{tot} \boldsymbol{e} = 1, \tag{1}$$

where $\boldsymbol{e}$ is a column vector of 1's. It should be noted that by definitions, the mean arrival rate of the fth MAP flow is characterized by $\lambda_f = \pi_{tot} D_f \boldsymbol{e}$.

B. Service Process

We employ the asynchronous partial network coding scheme, which opportunistically encodes the packets from distinct flows present in the queues, by linear coding (through bitwise XOR). If only one of the flows has at least one packet in its queue, then just that uncoded packet will be forwarded. On the other hand, if more than one of the flows have at least one packet present in their queues, then the service packet will include coded packets from these flows. The CDF of the encoded packets service time is denoted by $H(\cdot)$, with mean service time h. Different network coding schemes can be simply analyzed by obtaining the service time distribution of the encoded packets. We will discuss it more and provide an example in section IV.

TABLE II
KEY NOTATIONS

Notation	Semantics
F	Number of the arrival flows at a network node
m_f	Number of the transient states of the fth MAP flow
$\tilde{C}_f$, $\tilde{D}_f$	Transition rate matrices of the fth MAP flow
$\tilde{Q}_f$	Infinitesimal generator of the fth Markov process
$J_f(t)$	Arrival state of the fth MAP flow at time t
$J_{tot}(t)$	Total arrival state of the F flows at time t
m_{tot}	Number of the total arrival states
C_f, D_f	Transition rate matrices of the fth flow in terms of the total arrival states
C_{tot}	Sum of C_f matrices of all of the F flows
D_{tot}	Sum of D_f matrices of all of the F flows
Q_{tot}	Infinitesimal generator of the total Markov process
π_{tot}	Steady-state vector of the total Markov process
λ_f	Mean arrival rate of the fth MAP flow
$H(\cdot)$	CDF of the encoded packets service time
h	Mean service time
τ_k	Epoch of the kth daparture from the system
$\vec{\xi}_k$	Queue lengths at τ_k^+
$J_{tot,k}$	Total arrival state at τ_k^+
$\tilde{A}_{\vec{n}}(x)$	Probability matrix of arrivng $\vec{n}$ packets during a service given that the system is nonempty
$\tilde{B}_{\vec{n}}(x)$	Probability matrix of arrivng $\vec{n}$ packets during a service given that the system is empty
$\tilde{P}(\vec{n}, t)$	Probability matrix of the number of arrivals in $(0, t]$
$\mathbf{x}_{\vec{n}}$	Steady-state queue lengths distribution at departures
$\mathbf{y}_{\vec{n}}$	Steady-state queue lengths distribution at an arbitrary time

III. MAIN ANALYSIS

In this section, we analyze the system model described in section II and derive the steady-state queue lengths at departure epochs and subsequently at arbitrary time t, and finally, we present the average packet delays of the F flows in the studied network. Table II shows key notations adopted for the system model and analysis presented in this paper.

We begin by defining the embeded Markov renewal process at departure epochs as follows. Define τ_k to be the epoch of the kth daparture from the system, with $\tau_0 = 0$. Let $\vec{\xi}_k = \begin{pmatrix} \xi_k^1 & \xi_k^2 & \cdots & \xi_k^F \end{pmatrix}^T$ denote the vector whose elements represent the number of packets in the corresponding buffers, i.e., queue lengths, at τ_k^+. We also define $J_{tot,k}$ to be the total arrival state of the MAP flows at τ_k^+. By considering the

queue lengths and total arrival state at departure epochs, i.e., $(\vec{\xi}_k, J_{tot,k})$, the embeded Markov renewal process at departure epochs is obtained. Consequently, $(\vec{\xi}_k, J_{tot,k}, \tau_{k+1} - \tau_k)$ is a semi-Markov process (SMP) on the state space $\{(\vec{n}, j) : \vec{n} \geq \boldsymbol{o}, 1 \leq j \leq m_{tot}\}$, where $\boldsymbol{o}$ is a column vector of 0's. (By definition, an inequality symbol between two vectors states that the inequality holds for all of the corresponding elements of these vectors.) In this paper, we assume that for each flow, we have $\rho_f = \lambda_f h < 1$, meaning that the semi-Markov process is positive recurrent.

We define two matrices $\tilde{A}_{\vec{n}}(x)$ and $\tilde{B}_{\vec{n}}(x)$ of size $m_{tot} \times m_{tot}$ as follows. (It should be noted that we define these two matrices as the vector extensions of the block elements of the transition probability matrix of the SMP in a single flow network [24].)

$(\tilde{A}_{\vec{n}}(x))_{ij} = P\{$ given a departure at time 0, where at least one packet remains in one of the buffers and the total arrival state is i, the next departure occurs no later than time x, at which the total arrival state is j, and during that service there were $\vec{n}$ arrivals$\}$,

$(\tilde{B}_{\vec{n}}(x))_{ij} = P\{$ given a departure at time 0, which left all buffers empty and the total arrival state is i, the next departure occurs no later than time x, at which the total arrival state is j, leaving $\vec{n}$ packets in the system$\}$.

In order to derive closed form equations for $\tilde{A}_{\vec{n}}(x)$ and $\tilde{B}_{\vec{n}}(x)$, we consider the following definitions. Let $N_f(t)$ be the number of arrivals of the fth queue in $(0, t]$. We consider the matrix $\tilde{P}(\vec{n}, t)$ such that its (i, j) entry is defined as

$$\tilde{P}_{ij}(\vec{n}, t) = P\{\vec{N}(t) = \vec{n}, J_{tot}(t) = j | \vec{N}(0) = \boldsymbol{o}, J_{tot}(0) = i\}, \quad (2)$$

where $\vec{N}(t) = \begin{pmatrix} N_1(t) & N_2(t) & \cdots & N_F(t) \end{pmatrix}^T$. The matrices $\tilde{P}(\vec{n}, t)$ satisfy the (forward) Chapman-Kolmogorov equations [29]

$$\frac{d}{dt}\tilde{P}(\vec{n}, t) = \tilde{P}(\vec{n}, t)C_{tot} + \sum_{f=1,\ n_f \neq 0}^{F} \tilde{P}(\vec{n} - \boldsymbol{e}_f, t)D_f, \quad \vec{n} \geq \boldsymbol{o},\ t \geq 0, \quad (3a)$$

$$\tilde{P}(\boldsymbol{o}, 0) = I, \quad (3b)$$

where I is the identity matrix of size $m_{tot} \times m_{tot}$, and $\boldsymbol{e}_f$ is a column vector with entries 0 except for the fth element which is 1. In addition, the multi-variable matrix generating function $P(\vec{z}, t) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_F=0}^{\infty} \tilde{P}(\vec{n}, t) z_1^{n_1} \cdots z_F^{n_F}$, $\vec{z} = \begin{pmatrix} z_1 & z_2 & \cdots & z_F \end{pmatrix}^T$, is explicitly given by

$$P(\vec{z}, t) = \exp\left[(C_{tot} + \sum_{f=1}^{F} D_f z_f)t\right], \quad |\vec{z}| \leq 1,\ t \geq 0. \quad (4)$$

Now, from the definition of $\tilde{P}(\vec{n}, t)$, it is clear that

$$\tilde{A}_{\vec{n}}(x) = \int_0^x \tilde{P}(\vec{n}, t)\, dH(t), \quad (5)$$

$$\tilde{B}_{\vec{n}}(x) = \int_0^x e^{C_{tot}t} D_{tot} \tilde{A}_{\vec{n}}(x - t)\, dt. \quad (6)$$

We define the transform matrices of $\tilde{A}_{\vec{n}}(x)$ as

$$A_{\vec{n}}(s) = \int_0^{\infty} e^{-sx}\, d\tilde{A}_{\vec{n}}(x), \quad (7)$$

$$A(\vec{z}, s) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_F=0}^{\infty} A_{\vec{n}}(s) z_1^{n_1} \cdots z_F^{n_F}. \quad (8)$$

Analogous definitions hold for the transforms of $\tilde{B}_{\vec{n}}(x)$. Using the properties of $\tilde{P}(\vec{n}, t)$, it can be shown that

$$A(\vec{z}, s) = \int_0^{\infty} \exp\left[(-sI + C_{tot} + \sum_{f=1}^{F} D_f z_f)x\right] dH(x), \quad (9)$$

$$B(\vec{z}, s) = (sI - C_{tot})^{-1} D_{tot} A(\vec{z}, s). \quad (10)$$

For notational simplicity, we define the matrices $A_{\vec{n}} = A_{\vec{n}}(0) = \tilde{A}_{\vec{n}}(\infty)$, $A(\vec{z}) = A(\vec{z}, 0)$, and $A = A(\boldsymbol{e}, 0)$, and also the corresponding definitions for $\tilde{B}_{\vec{n}}(x)$.

Using the framework described so far, we will derive the steady-state queue lengths of the F flows at the departure epochs in the next part.

A. Steady-State Queue Lengths at Departures

Let us define the elements of the vector $\mathbf{x}_{\vec{n}} = \begin{pmatrix} x_{\vec{n},1} & x_{\vec{n},2} & \cdots & x_{\vec{n},m_{tot}} \end{pmatrix}$, $\vec{n} \geq \boldsymbol{o}$ as follows: $x_{\vec{n},j}$, $1 \leq j \leq m_{tot}$, is the steady-state probability that after the departure epochs, there are $\vec{n}$ packets in the buffers and the total arrival state is j. According to the definitions, by applying asynchronous partial network coding in the service process, the steady-state probability vector $\mathbf{x}_{\vec{n}}$ is obtained from

$$\mathbf{x}_{\vec{n}} = \mathbf{x}_{\boldsymbol{o}} B_{\vec{n}} + \sum_{k_1=0}^{n_1+1} \cdots \sum_{k_F=0}^{n_F+1} \left(\mathbf{x}_{\sum_{f=1}^{F} k_f \boldsymbol{e}_f} A_{\sum_{f=1}^{F} \min(n_f, n_f - k_f + 1)\boldsymbol{e}_f} \right) - \mathbf{x}_{\boldsymbol{o}} A_{\vec{n}}. \quad (11)$$

The min operator in the subscript of $A_{(\cdot)}$ arises due to the fact that if the fth buffer is empty at a departure, we need n_f arrivals to reach n_f packets. However, if it is not empty, we need as many arrivals as the difference between the desired number of packets (n_f) and the number of existing packets in the buffer (k_f) plus one another arrival to replace the packet sent during the transmission process.

Given $\mathbf{x}_{\boldsymbol{o}}$, the vectors $\mathbf{x}_{\vec{n}}, \vec{n} \neq \boldsymbol{o}$ are obtained according to (11). In order to derive $\mathbf{x}_{\boldsymbol{o}}$, we define the marginal probabilities $\boldsymbol{x}_0^{[q]} = \begin{pmatrix} x_{0,1}^{[q]} & x_{0,2}^{[q]} & \cdots & x_{0,m_{tot}}^{[q]} \end{pmatrix}$, $1 \leq q \leq F$, where the element $x_{0,j}^{[q]}$ is the probability that the qth queue at a departure epoch is empty, and the total arrival state is j. (Note that the qth queue may have had no packets to participate in the encoding process of the transmitted packet at the given departure epoch.)

In order to compute $\boldsymbol{x}_0^{[q]}$, we will introduce matrix $G^{[q]}$ as follows. First, we define the *level l* for the qth queue to be the set of states $\left\{\left(\sum_{f=1,f\neq q}^{F} n_f \boldsymbol{e}_f + l\boldsymbol{e}_q,\ j\right) | n_f \geq 0,\ 1 \leq j \leq m_{tot}\right\}$. Let s_1 represent a state belonging to level $n_q + r$ of the qth queue with total arrival state i, i.e., $s_1 = \left((n_1, \cdots, n_{q-1},\ n_q + r,\ n_{q+1}, \cdots, n_F)^T, i\right)$. Similarly, s_2 represents a state belonging to level n_q of the qth queue with total arrival state j, i.e., $s_2 = \left((n_1', \cdots, n_{q-1}',\ n_q,\ n_{q+1}', \cdots, n_F')^T, j\right)$, where $n_q \geq 0, r \geq 1,\ 1 \leq i, j \leq m_{tot}$.

Define matrix $\tilde{G}^{[r,q]}(k,\ x)$ with elements $\tilde{G}_{ij}^{[r,q]}(k,\ x)$ to denote the probability that the first passage from state s_1 to state s_2 occurs in exactly k transitions no later than time x, and s_2 is the first state visited at level n_q of the qth queue.

Let us consider the matrix transform $G^{[q]}(z_q,\ s) = \sum_{k=1}^{\infty} \int_0^{\infty} e^{-sx}\, d\tilde{G}^{[1,q]}(k,\ x) z_q^k, |z_q| \leq 1, Re(s) \geq 0$. In the rest of the paper, we will use the positive stochastic matrices $G^{[q]} := G^{[q]}(1,\ 0)$, with invariant probability vectors $\boldsymbol{g}^{[q]}$ satisfying

$$\boldsymbol{g}^{[q]} G^{[q]} = \boldsymbol{g}^{[q]},\quad \boldsymbol{g}^{[q]}\boldsymbol{e} = 1. \tag{12}$$

Now, we define $E^{[q]}$ as a diagonal matrix with elements $E_j{}^{[q]}$ denoting the conditional probability that if the qth queue is empty and the total arrival state is j, then all other queues are empty. In other words

$$E_j{}^{[q]} = \frac{x_{\boldsymbol{o},j}}{x_{0,j}^{[q]}}. \tag{13}$$

Proposition 1: The vectors $\boldsymbol{x}_0^{[q]}$ are given by

$$\boldsymbol{x}_0^{[q]} = \frac{\boldsymbol{g}^{[q]}\left(-E^{[q]} {C_{tot}}^{-1} D_{tot} + (I - E^{[q]})\right)^{-1}}{\boldsymbol{g}^{[q]}\left(I - A + (\boldsymbol{e} - \beta^{[q]})\boldsymbol{g}^{[q]}\right)^{-1}\boldsymbol{e}}, \tag{14}$$

where $\beta^{[q]} = \frac{d}{dz_q} A(\vec{z})|_{\vec{z}=\boldsymbol{e}}\boldsymbol{e}$.

Proof: The proof is provided in the Appendix.

The only unknown part of (14) is $E^{[q]}$. In what follows, we will introduce an approximate decoupling approach for calculating $E^{[q]}$.

1) Decoupling the Queues: Let $S = \{1, 2, \cdots, F\}$. We assume that for any queue q, $1 \leq q \leq F$, and for each choice of the set $Z^*, Z^* \subseteq S \setminus \{q\}$, the event that after a departure, the qth queue becomes empty and the total arrival state is j, is independent from the event that after a departure, each queue belonging to the set Z^* becomes empty and the total arrival state is j. This assumption implies that

$$x_{\boldsymbol{o},j} = \prod_{q=1}^{F} x_{0,j}^{[q]}. \tag{15}$$

We call the above assumption the *decoupling assumption* and will exploit it in what follows. It should be noted that we are not assuming that the flows are fully decoupled. In fact, the *decoupling assumption* in this paper is much less strict than fully decoupling of the flows, and it is only about the time steps that a queue is empty. This assumption simplifies the computations and will be verified through numerical results in section V. We note that (15) and (13) lead to the following simplification:

$$E^{[q]} = \prod_{f=1,f\neq q}^{F} diag(\boldsymbol{x}_0^{[f]}). \tag{16}$$

Considering (14) and (16), we have a system of nonlinear matrix equations, which is solved by fixed point iteration method [30]. Based on our simulations, the iterations converge very fast starting from $\boldsymbol{x}_0^{[q]} = \pi_{tot}, 1 \leq q \leq F$.

So far, we have obtained $\boldsymbol{x}_0^{[q]}$, representing the vector probability that after a departure epoch, the qth queue is empty. Subsequently, $\mathbf{x}_{\boldsymbol{o}}$ is given by (15).

Finally in this part, we will obtain the steady-state queue length of the qth queue at departure epochs. We begin by defining $\boldsymbol{x}_{n_q}^{[q]} = \left(x_{n_q,1}^{[q]} \quad x_{n_q,2}^{[q]} \quad \cdots \quad x_{n_q,m_{tot}}^{[q]}\right)$, where $x_{n_q,j}^{[q]}, 1 \leq j \leq m_{tot}$, is the steady-state probability that after a departure, there will be n_q packets in the qth queue and the total arrival state will be j. It should be noted that according to the definitions, the relationship between $\boldsymbol{x}_{n_q}^{[q]}$ and $\mathbf{x}_{\vec{n}}$ is given by

$$\boldsymbol{x}_{n_q}^{[q]} = \sum_{n_1=0}^{\infty} \cdots \sum_{n_{q-1}=0}^{\infty} \sum_{n_{q+1}=0}^{\infty} \cdots \sum_{n_F=0}^{\infty} \mathbf{x}_{\left(\sum_{f=1,f\neq q}^{F}(n_f\boldsymbol{e}_f) + n_q\boldsymbol{e}_q\right)}. \tag{17}$$

Moreover, the vector generating function of $\boldsymbol{x}_{n_q}^{[q]}$, that is $\boldsymbol{X}^{[q]}(z_q) = \sum_{n_q=0}^{\infty} \boldsymbol{x}_{n_q}^{[q]} z_q^{n_q}$, is related to the vector generating function of $\mathbf{x}_{\vec{n}}$, i.e., $\mathbf{X}_{\vec{z}} = \sum_{n_1=0}^{\infty} \cdots \sum_{n_F=0}^{\infty} \mathbf{x}_{\vec{n}} z_1^{n_1} \cdots z_F^{n_F}$, according to the following equation:

$$\boldsymbol{X}^{[q]}(z_q) = \boldsymbol{X}_{(\boldsymbol{e}+(z_q-1)\boldsymbol{e}_q)}. \tag{18}$$

In the following lemma, we derive the vector generating function of $\boldsymbol{x}_{n_q}^{[q]}$.

Lemma 1: The vector generating function $\boldsymbol{X}^{[q]}(z_q)$ satifies the following equation:

$$\boldsymbol{X}^{[q]}(z_q)(z_q I - A^{[q]}(z_q)) = z_q \mathbf{x}_{\boldsymbol{o}}(B^{[q]}(z_q) - A^{[q]}(z_q)) + \boldsymbol{x}_0{}^{[q]}(z_q - 1)A^{[q]}(z_q), \tag{19}$$

where $A^{[q]}(z_q)$ and $B^{[q]}(z_q)$ are defined similar to (18).

Proof: The proof is provided in the Appendix.

Having equations describing the vector generating functions $\boldsymbol{X}^{[q]}(z_q)$ by Lemma 1, the vectors $\boldsymbol{x}_{n_q}^{[q]}$ are readily obtained.

Finally, the mean queue length of the qth queue at departures is presented in Lemma 2.

Lemma 2: The mean queue length of the qth queue at departures is obtained from

$$\boldsymbol{X}'^{[q]}(1)\boldsymbol{e} = \frac{1}{2(1 - \pi_{tot}\beta^{[q]})}\Bigg[\boldsymbol{X}^{[q]}(1)A''^{[q]}(1)\boldsymbol{e} + u''^{[q]}(1)\boldsymbol{e} + 2\left(u'^{[q]}(1) - \boldsymbol{X}^{[q]}(1)(I - A'^{[q]}(1))\right)(I - A + \boldsymbol{e}\pi_{tot})^{-1}\beta^{[q]}\Bigg], \tag{20}$$

where for the simplicity of notations,
$u^{[q]}(z_q) = z_q \mathbf{x_o}(B^{[q]}(z_q) - A^{[q]}(z_q)) + \boldsymbol{x}_0^{[q]}(z_q - 1)A^{[q]}(z_q)$
denotes the RHS of (19).

Proof: The proof is provided in the Appendix.

B. Steady-State Queue Lengths at an Arbitrary Time

In this section, we explore the relationship between the steady-state queue lengths at an arbitrary time and the steady-state queue lengths at departures.

Let $\vec{\xi}(t) = \begin{pmatrix}\xi^1(t) & \xi^2(t) & \cdots & \xi^F(t)\end{pmatrix}^T$ denote the queue lengths at time t. We consider the continuous-parameter process $\{(\vec{\xi}(t), J_{tot}(t)), t \geq 0\}$. The time-dependent joint distribution of the queue lengths and the total arrival state is given by the conditional probabilities

$$Y(\vec{n}, j; t) = P\{\vec{\xi}(t) = \vec{n}, J_{tot}(t) = j | \vec{\xi}_0 = \vec{n}_0, J_{tot,0} = j_0\}, \quad (21)$$

for $\vec{n} \geq \boldsymbol{o}$, $1 \leq j \leq m_{tot}, t \geq 0$. We define

$$\boldsymbol{y}_{\vec{n}} = \{y_{\vec{n},1}, y_{\vec{n},2}, \cdots, y_{\vec{n},m_{tot}}\}, \quad (22)$$

where

$$y_{\vec{n},j} = \lim_{t\to\infty} Y(\vec{n}, j; t), \quad \vec{n} \geq \mathbf{0},\ 1 \leq j \leq m_{tot}. \quad (23)$$

In what follows, we will derive $\boldsymbol{y}_{\vec{n}}$. For this purpose, we first introduce and derive the *fundamental mean*, denoted by E^*, of the Markov renewal process. E^* is the sum of the products of the invariant probabilities $x_{\vec{n},j}$ of each state and the sum of its mean transition times to all other possible states. In the steady-state version of the system, E^* is the average time between an arbitrary transition (e.g., departure from the system) and the next transition. Lemma 3 presents the closed form expression of E^*.

Lemma 3: The fundamental mean E^* is given by

$$E^* = h - \mathbf{x_o} C_{tot}^{-1} \boldsymbol{e}. \quad (24)$$

Proof: The proof is provided in the Appendix.

Finally, in the following lemma and proposition, we obtain the steady-state queue lengths distribution at an arbitrary time, i.e., $\boldsymbol{y}_{\vec{n}}$.

Lemma 4: The vector $\boldsymbol{y_o}$ is given by $\boldsymbol{y_o} = \frac{-1}{E^*}\mathbf{x_o} C_{tot}^{-1}$.

Proof: The proof is provided in the Appendix.

Proposition 2: The steady-state queue lengths distribution at an arbitrary time, $\boldsymbol{y}_{\vec{n}}, \vec{n} \neq \boldsymbol{o}$, is obtained from

$$\begin{aligned} E^*\boldsymbol{y}_{\vec{n}} = \\ \mathbf{x_o} \int_0^\infty \int_0^\infty e^{C_{tot} v} dv \sum_{f=1, n_f \neq 0}^{F} D_f \tilde{P}(\vec{n} - \boldsymbol{e}_f, t)(1 - H(t))dt \\ + \sum_{k_1=0}^{n_1} \cdots \sum_{k_F=0}^{n_F} \boldsymbol{x}_{\sum_{f=1}^F k_f \boldsymbol{e}_f} \int_0^\infty \tilde{P}(\vec{n} - \sum_{f=1}^{F} k_f \boldsymbol{e}_f, t)(1 - H(t))dt \\ - \mathbf{x_o} \int_0^\infty \tilde{P}(\vec{n}, t)(1 - H(t))dt, \end{aligned} \quad (25)$$

and the vector generating function $\boldsymbol{Y}_{\vec{z}}$ of $\boldsymbol{y}_{\vec{n}}$ satisfies the following equation:

$$\begin{aligned} E^*\boldsymbol{Y}_{\vec{z}}\left(C_{tot} + \sum_{f=1}^{F} z_f D_f\right) = \\ \boldsymbol{X}_{\vec{z}}(A(\vec{z}) - I) - \mathbf{x_o} {C_{tot}}^{-1}\left(C_{tot} + \sum_{f=1}^{F} z_f D_f\right) A(\vec{z}). \end{aligned} \quad (26)$$

Proof: The proof is provided in the Appendix.

The steady-state vector probability that there will be n_q packets in the qth queue at an arbitrary time, denoted by $\boldsymbol{y}_{n_q}^{[q]}$, and its vector generating function, $\boldsymbol{Y}^{[q]}(z_q)$, are defined similar to (17) and (18). Therefore, putting the vector $\vec{z} = \boldsymbol{e} + (z_q - 1)\boldsymbol{e}_q$ in (26), simply leads to

$$\begin{aligned} E^*\boldsymbol{Y}^{[q]}(z_q)(Q_{tot} + D_q(z_q - 1)) = \\ \boldsymbol{X}^{[q]}(1)(A^{[q]}(z_q) - I) - \mathbf{x_o} {C_{tot}}^{-1}(Q_{tot} + D_q(z_q - 1))A^{[q]}(z_q), \end{aligned} \quad (27)$$

and subsequently, the vectors $\boldsymbol{y}_{n_q}^{[q]}$ for $0 \leq n_q < +\infty$, are computed.

Finally, the mean queue length of the qth queue at an arbitrary time is given by $\boldsymbol{Y}'^{[q]}(1)\boldsymbol{e}$, which is obtained in the proof of Proposition 3.

C. Average Packet Delay

So far, we have presented the steady-state queue lengths at departure epochs and subsequently, at an arbitrary time. We conclude this section by presenting delay analysis as an important QoS metric. The average packet delay of the qth flow, denoted by w_q, is defined as the average time that a packet belonging to the qth flow stays in the network node. In the following proposition, we derive the average packet delay in each of the F MAP flows in the desired network, where asynchronous partial network coding is applied.

Proposition 3: The average packet delay of the qth flow, w_q, is obtained from

$$\begin{aligned} w_q = \frac{1}{{\lambda_q}^2 E^*} \\ \Bigg[\boldsymbol{X}'^{[q]}(1)\boldsymbol{e} - \frac{1}{2}u''^{[q]}(1)\boldsymbol{e} - \mathbf{x_o} C_{tot}^{-1}(D_q A'^{[q]}(1) + \frac{1}{2}Q_{tot}A''^{[q]}(1))\boldsymbol{e} \\ + \Bigg(\boldsymbol{X}'^{[q]}(1)(A - I) + \boldsymbol{X}^{[q]}(1)\boldsymbol{A}'^{[q]}(1) - \mathbf{x_o} C_{tot}^{-1}(D_q A + Q_{tot}\boldsymbol{A}'^{[q]}(1)) \\ - E^*\pi_{tot} D_q\Bigg)(\boldsymbol{e}\pi_{tot} - Q_{tot})^{-1} D_q \boldsymbol{e}\Bigg]. \end{aligned} \quad (28)$$

Proof: The proof is provided in the Appendix.

Therefore, by applying the derivd equations in this section, the average packet delay of each of the flows in the network is obtained using the network parameters.

D. Discussion

In this section, we have presented new fundamental results for queuing analysis in networks where network coding is applied on a number of MAP flows. These results pave the

way to obtain many performance measures of interest. For example, expressions for the moments of the queue lengths at departures and at an arbitrary time can be simply obtained by differentiating (19) and (27), respectively. Moreover, it should be noted that MAP is a wide class of arrival processes and contains as special cases many processes studied in the literature. For example, we can simply obtain the results for MMPP arrival flows, by setting $\tilde{D}_f = \tilde{\Lambda}_f$ and $\tilde{C}_f = \tilde{Q}_f - \tilde{\Lambda}_f$, where $\tilde{\Lambda}_f$ is the diagonal rate matrix of the fth MMPP flow. As another example, if we consider $\tilde{D}_f = \lambda_f$ and $\tilde{C}_f = -\lambda_f$, then the MAP reduces to a Poisson process of rate λ_f. Therefore, the presented analysis in this paper provides a general framework for applying network coding in communication networks.

The fact that packet flows arriving at internet routers (both edge and backbone) cannot be accurately modeled by Poisson processes is widely accepted, and has been discussed in the literature. It is shown that traffic traces captured on both LANs and WANs exhibit Long Range Dependence (LRD) properties, and self-similar characteristics at different time scales [5],[6]. Since the long-term correlation of data traffic beyond a certain threshold does not influence the performance of a system, Markovian arrival models can be successfully employed in packet networks [31]-[33]. The MMPP models are shown to provide good matches of LRD properties and fitting procedures are proposed to match the covariance function of the Markovian model to that of second order self-similar processes over several time scales [20], [21]. Moreover, it is shown that the queuing behavior of the traffic generated by the MMPP model is consistent with the one produced by real traces collected at edge routers under several different traffic loads.Therefore, the Markovian arrival models match very well the characteristic of data traffic [22]. It will also be validated through numerical results in section V.

Subsequently, we discuss the case of limited buffer lengths. If the fth buffer can store at most a total of L_f packets, any further arriving packets will be refused entry to the system and will depart immediately without service and will be lost. Considering the limited buffer lengths constraints, i.e., $\boldsymbol{L} = (L_1, \cdots, L_F)$, the steady-state queue lengths distributions at departures, $\mathbf{x}_{\vec{n}}$, and at an arbitrary time, $\mathbf{y}_{\vec{n}}, \boldsymbol{o} \leq \vec{n} \leq \boldsymbol{L}$, are given by (11) and (25), respectively. Moreover, we have

$$\boldsymbol{y}_{n_q}^{[q]} = \sum_{n_1=0}^{L_1} \cdots \sum_{n_{q-1}=0}^{L_{q-1}} \sum_{n_{q+1}=0}^{L_{q+1}} \cdots \sum_{n_F=0}^{L_F} \mathbf{y}_{\left(\sum_{f=1, f\neq q}^{F} (n_f \boldsymbol{e}_f) + n_q \boldsymbol{e}_q\right)}, \tag{29}$$

where $0 \leq n_q \leq L_q$. By Little's Law and similar to the approach in [34], the average packet delay of the qth queue is obtained from

$$w_q = \frac{N_q}{\lambda_q (1 - P_{loss\,q})}, \tag{30}$$

where the mean queue length of the qth queue is obtained from $N_q = \sum_{n_q=0}^{L_q} n_q \mathbf{y}_{n_q}^{[q]} \boldsymbol{e}$, and the loss probability of the qth queue is

$$P_{loss\,q} = \frac{\mathbf{y}_{L_q}^{[q]} D_q \boldsymbol{e}}{\sum_{n_q=0}^{L_q} \mathbf{y}_{n_q}^{[q]} D_q \boldsymbol{e}}. \tag{31}$$

It should be also noted that according to [35], in a MAP/G/1/L queue, queue length distribution tends to its limiting value for $L \to \infty$ at a geometric rate. As a consequence, the empty buffer probability and the loss probability (and many other performance parameters) exhibit a geometric decay towards the corresponding limiting values.

IV. Applying the Proposed Model in Single Bottleneck Caching Networks

In this section, we apply the proposed queuing analysis of network coding with Markovian arrival processes to caching networks. We consider a single bottleneck caching network with one content server, F stations equipped with caches (i.e., caching nodes), and one-hop multicast transmission from the server to the stations, with an average transmission rate denoted by r_0 [bps], as illustrated in Fig. 2. It should be noted that the content server can be a macro base station in a 5G cellular network, where the stations can be caching helpers, such as micro base stations. We assume that the users requests, arriving at the stations, are drawn from a specific same-size file library $\boldsymbol{M} = \{m_i, i = 1, \cdots, M\}$ of size B bits, for notational convenience, similar to [2], [36]. This assumption can be easily removed by dividing the content items into multiple smaller files of the same length. The caching nodes are capable of storing C whole files (i.e., CB bits).

In [2], we studied such networks by considering the IRM traffic model generating an i.i.d sequence of requests. The IRM traffic model ignores all temporal correlations in the stream of requests and similar to other memoryless processes cannot effectively model packet flows in practical networks [5], [6]. The Markovian arrival process is an appropriate candidate for traffic modeling in communication networks. Therefore, in order to take into account a realistic traffic model, we consider the MAP model.

We consider F queues at the content server for the requests of F stations sent to the server. Based on such a framework, at the server, the requested files of different caching nodes enter their corresponding queues and are merged together to construct the coded packets to be transmitted. Therefore, the described model in section II is applicable to the server functionality in single bottleneck caching networks. It should be noted that by means of the cache content of the stations, the coded packets are appropriately decoded as will be disscussed in what follows.

As described in section II, we employ asynchronous partial network coding for the service process. As an example of asynchronous partial network coding in caching networks, we present Asynchronous Partition Coded Caching (APCC) scheme, which is an asynchronous extension of Partition Coded Caching Scheme (PCCS) studied in [2] and [37]. These schemes consist of two phases, namely, cache placement phase, which is done during the off-peak hours of the networks, and delivery phase in which the users requests are served. The cache placement phase of APCC is similar to PCCS, in which variable $\gamma = F\frac{C}{M}$ is defined based on the network parameters. (It should be noted that this scheme is proposed for cache sizes C such that γ is an integer less

than F.) Subsequently, each file is partitioned into $\mu_0 = \binom{F}{\gamma}$ subfiles of equal size. In the cache placement phase, each station caches an equal number of subfiles of all of the M files. The cache placement is designed in order to create coded multicasting opportunities for any $\gamma + 1$ stations even with different requests. The main idea of this scheme is to enable subfiles exhchanges between the stations, given their cache contents and the recieved coded packets, such that the stations can appropriatley decode their required subfiles. It should be noted that in PCCS, stochastic arrival time of user requests is not considered, and it is assumed that the requests of all caching nodes are simultaneusly present at the server. In contrast in APCC, we consider the stochastic request arrivals. Therefore, at each service round in the delivery phase of APCC, the server creates coded small packets by linear coding (through XOR) of the required subfiles of those stations that their corresponding queues are not empty, and sends the service packet. Subsequently, at the stations, the corresponding requested subfiles are obtained by decoding the recieved coded small packets given the cache content.

As an example, we consider a simple network with three stations and a library of three files, namely A, B, C, and caches of size one, as shown in Fig. 2. Therefore, the network parameters are $F = M = 3$, $C = 1$, and consequently, we have $\gamma = 1$ and $\mu_0 = 3$. According to APCC, each file is split into μ_0 equal size subfiles. i.e., $A = (A_1, A_2, A_3)$, $B = (B_1, B_2, B_3)$ and $C = (C_1, C_2, C_3)$. The cache placement is done such that the cache content of station f is $Z_f = (A_f, B_f, C_f)$. In the delivery phase, if for example station one requests file A, station two requests file B, and station three requests file C, the missing subfiles are A_2 and A_3 for station one, B_1 and B_3 for station two, and C_1 and C_2 for station three, which enter the corresponding queues as shown in Fig. 2. Given the cache contents, stations one and two aim to exchange the missing subfiles A_2 and B_1, stations one and three aim to exchange A_3 and C_1, and stations two and three aim to exchange B_3 and C_2. By sending the coded small packets $(A_2 \oplus B_1, A_3 \oplus C_1, B_3 \oplus C_2)$, the server enables all of these exchanges between the stations. Therefore, the server creates and sends a service packet consists of three coded small packets of size one third of a whole file. The length of the service packet which serves the requests of the three stations is then one whole file. It should be noted that the mentioned example in the delivery phase is the extreme case, where all of the stations have requested files or in other words, all of the queues have packets to send. If for example only station one requests file A and station two requests file B and the corresponding queue of station three is empty, the transmitted service packet will be $(A_2 \oplus B_1, A_3, B_3)$. Another example is also provided in Fig. 2. For other values of the network parameters, the cache placement and delivery phases are obtained according to the basic model explained in [2] and [37].

In APCC, for any requested file from a given station, its cache does not contain the whole file and only contains a subset of the required subfiles. Consequently, each station needs to submit each request to the server. Therefore, the arrival processes of the queues at the server are the same as the

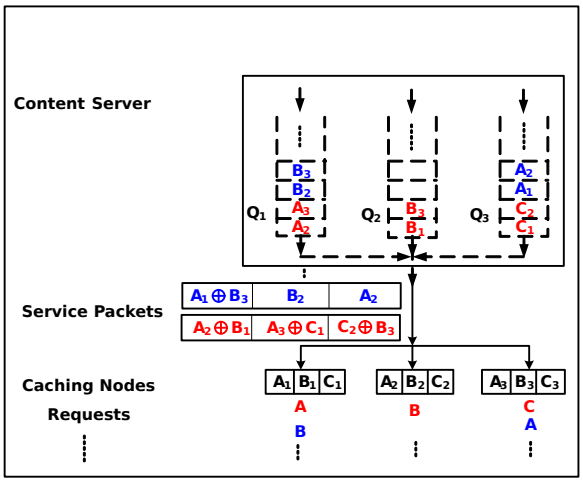

Fig. 2. An example of network coding in a single bottleneck caching network with 3 caching nodes.

requests arrival processes at the stations, which are modeled by MAP. It should be noted that a station may need to serve different users requesting the same file in different moments. In order to improve the efficiency considering this situation, we can apply a control unit at the entrance of each queue which ensures that multiple requests for each file enter at most once in the corresponding queue. It should be noted that the class of MAPs is closed under random thinning. A random thinning of a MAP $(C; D)$, that is a MAP where an event is kept with probability p, is itself a MAP with parameter matrices $(C + (1 - p)D; pD)$ [38]. Therefore, by applying the control units, the arrival of the queues are still MAPs and the analysis provided in section III is applied. On the other hand, the service time of APCC is obtained as follows. The maximum number of stations that can be served by each coded small packet is $\gamma+1$. Since at most the subfiles of $\gamma+1$ queues are combined by linear coding before transmission, the number of transmittable coded small packets at each service round is $\binom{F}{\gamma+1}$. Morever, the size of each coded smal packet is $\frac{1}{\mu_0} = \frac{1}{\binom{F}{\gamma}}$ of a whole file, according to the explained model of APCC. Therefore, the length of the service packets is designed to be $\frac{\binom{F}{\gamma+1}}{\binom{F}{\gamma}} B$ bits. Since the transmission rate of the multicast link for serving the requested contents is varying over time, due to the random channel conditions, the service time is random and is modeled by an arbitrary distribution, with mean $h = \frac{B\binom{F}{\gamma+1}}{r_0\binom{F}{\gamma}}$.

Therefore as explained in this section, by applying the service process of the desired coded caching scheme to the derived equations in section III, the performance metrics in caching networks, such as the average packet delay, can be derived.

V. Numerical Results

In this section, the analytic expressions derived in this paper are validated through simulation results and real trace-driven experiments. First, we have run a trace-driven experiment, using a real trace of video clips requests from a campus

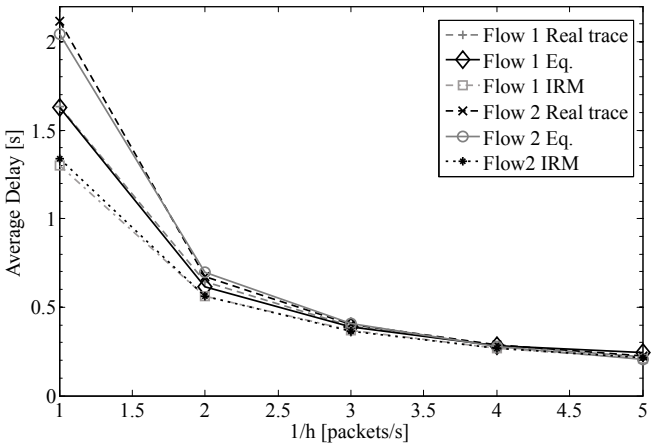

Fig. 3. Comparison of the mean packet delays of the real trace driven experiment, proposed analytic expression and traditional IRM traffic model. $F = 2$, two-state MAPs, i.e., $m_f = 2, f \in \{1, 2\}$.

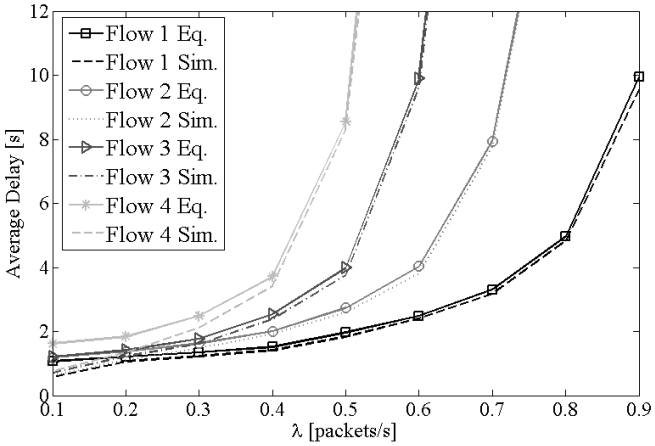

Fig. 4. Mean packet delays of APNC with $F = 4$, two-state MAPs with rates $\lambda_1 = \lambda$, $\lambda_2 = 1.25\lambda$, $\lambda_3 = 1.5\lambda$, $\lambda_4 = 1.75\lambda$ and exponential service time distribution with $h = 1$ [s].

network measurement on YouTube traffic in 2009 [39], with a total 123.3k requests for 78.9k videos.

Fig. 3 compares the average delay in the real trace-driven experiment with the derived equations for the MAP traffic model and the results of the IRM traffic model. In this figure, we study a network with $F = 2$ arrival flows, and for MAP traffic modeling, we consider a two-state MAP for each flow, i.e., $m_f = 2, f \in \{1, 2\}$. We estimate the matrices

$$\tilde{C}_1 = \begin{bmatrix} -0.8 & 0.2 \\ 0.01 & -0.14 \end{bmatrix}, \tilde{D}_1 = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.13 \end{bmatrix},$$

$$\tilde{C}_2 = \begin{bmatrix} -0.43 & 0.21 \\ 0.15 & -0.61 \end{bmatrix}, \tilde{D}_2 = \begin{bmatrix} 0.22 & 0 \\ 0 & 0.46 \end{bmatrix},$$

for modeling the real traffic. We observe that the proposed model based on the MAP traffic can properly model the real traffic of an operational network, while the IRM model cannot. In addition, it is illustrated that by increasing the average service rate, i.e., $\frac{1}{h}$, the average delay decreases.

Fig. 4 shows the average delay of each flow in a network with $F = 4$ arrival flows. In this figure, the arrival flows are assumed two-state MAPs, i.e., $m_f = 2, f \in \{1, 2, 3, 4\}$, with mean arrival rates $\lambda_1 = \lambda$, $\lambda_2 = 1.25\lambda$, $\lambda_3 = 1.5\lambda$, $\lambda_4 = 1.75\lambda$. The service time distribution is exponential with the mean $h = 1$ [s]. It is illustrated that the proposed MAP equations yield very accurate results, specially by increasing λ. This result might be due to the fact that the decoupling assumption holds when the network node is relatively loaded by each of the flows.

The effect of APNC and caching is illustrated in Fig. 5, in a network with $F = 2$ flows, when the arrivals are two-state MAPs, $m_f = 2, f \in \{1, 2\}$. Fig. 5 shows that by increasing the ratio of $\frac{r_0}{B}$, which increases the average service rate, the average delay decreases. We observe that by applying APNC, the average delay decreases in comparison to the case that no network coding is applied. Moreover, it is illustrated that applying APNC and caching in the network reduces the

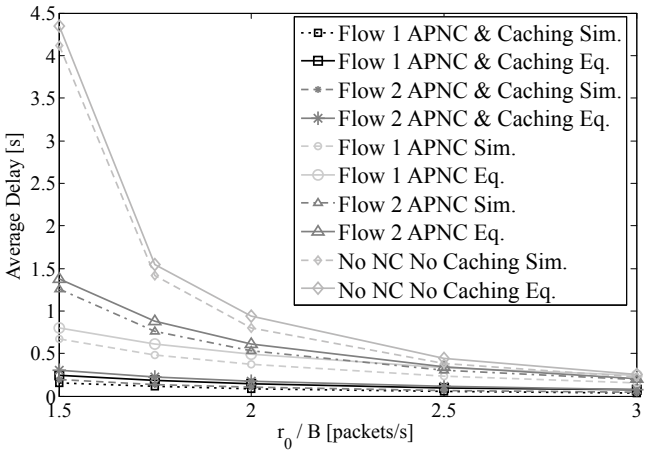

Fig. 5. Mean packet delays of APNC with $F = 2$, two-state MAPs and exponential service time distribution, compared to the systems with caching and without network coding.

average delay significantly.

Furthermore, we have provided the numerical results based on the MAP models of the real trace of internet traffic [20]-[22], in Figs. 6 and 7. The average delay of each flow performing network coding has been plotted in comparison with the performance without network coding in these figures. We have provided the results in a network with 4 MAP flows and the results in a network with 8 MAP flows in Figs. 6 and 7, respectively. Moreover, the average delay of each flow performing network coding in a network with 10 MAP flows has been plotted in comparison with the performance without network coding in Fig. 8. As shown in the figures, the results hold for different traffic types, different requests arrival rates and different number of flows. It is also illustrated that by increasing the number of data flows, the network coding gain increases. This fact shows the importance of using asynchronous network coding specially in large networks.

In Fig. 9, we have compared the average delay as a function of the number of flows in the network, for different values of the service rate. As shown in this figure, for small values of

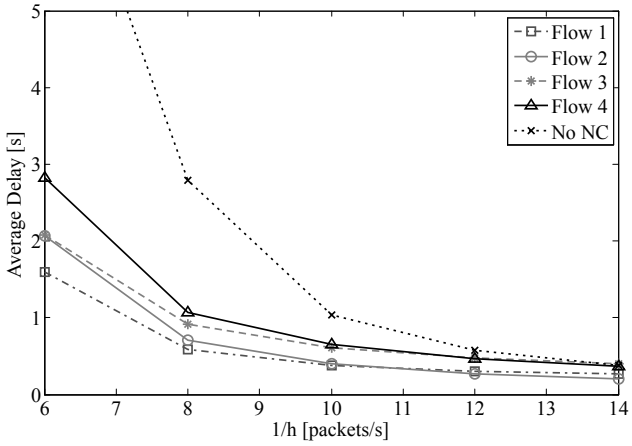

Fig. 6. Mean packet delays of APNC with $F = 4$, compared to the systems without network coding. Two-state MAPs with rates $\lambda_1 = 1.74$, $\lambda_2 = 2.09$, $\lambda_3 = 0.71$, $\lambda_4 = 1.13$, $\lambda_{tot} = 5.68$.

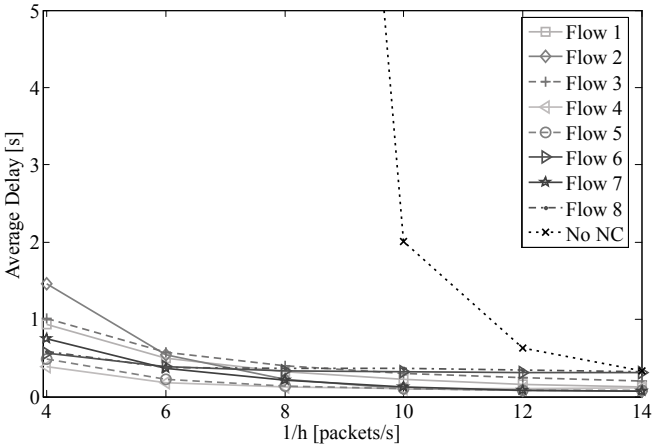

Fig. 7. Mean packet delays of APNC with $F = 8$, compared to the systems without network coding. Two-state MAPs with rates $\lambda_1 = 1.74$, $\lambda_2 = 1.71$, $\lambda_3 = 2.09$, $\lambda_4 = 0.63$, $\lambda_5 = 0.71$, $\lambda_6 = 0.47$, $\lambda_7 = 1.13$, $\lambda_8 = 0.37$, $\lambda_{tot} = 8.86$.

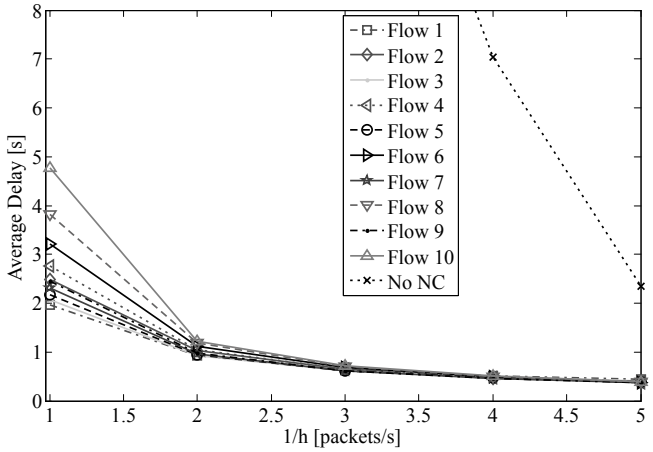

Fig. 8. Mean packet delays of APNC with $F = 10$, compared to the systems without network coding. Two-state MAPs with rates $\lambda_1 = 0.15$, $\lambda_2 = 0.36$, $\lambda_3 = 0.19$, $\lambda_4 = 0.45$, $\lambda_5 = 0.23$, $\lambda_6 = 0.54$, $\lambda_7 = 0.27$, $\lambda_8 = 0.63$, $\lambda_9 = 0.30$, $\lambda_{10} = 0.72$, $\lambda_{tot} = 3.84$.

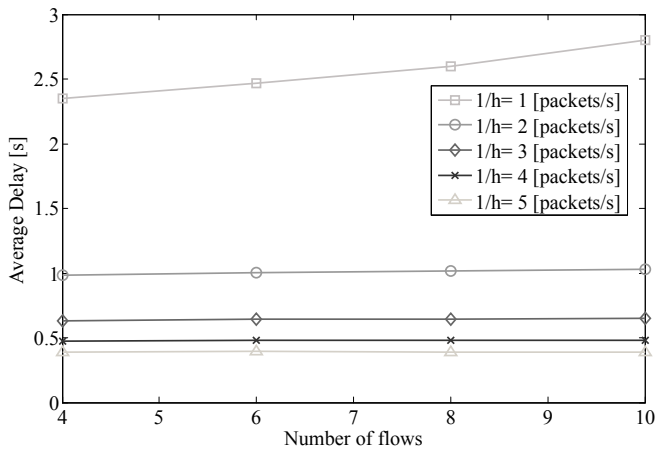

Fig. 9. Mean packet delays of APNC as a function of the number of flows.

the service rate, such as $1/h = 1[packets/s]$, by increasing the number of flows, which increases the total arrival rate, the average delay increases. However, for larger values of the service rate, the average delay slightly changes by increasing the number of flows. It should be noted that since packets of different flows are served together using asynchronous network coding, when the service rate is sufficient in comparison to the arrival rate, the average delay in the network does not significantly change by increasing the number of arrival flows. This fact illustrates the significant gain of using asynchronous network coding in communication networks.

VI. Conclusion

In this paper, we presented new fundamental results for queuing analysis in networks where asynchronous network coding is applied on a number of MAP flows. These results pave the way to obtain many performance measures of interest. Through the analysis carried out using matrix geometric methods, we provided queue lengths at departures and at an arbitrary time and utilized the analysis to characterize the mean packet delay. Since MAP is a wide class of arrival processes and matches very well the characteristics of data traffic, the presented analysis in this paper provides a general framework for applying network coding in communication networks. Furthermore, we applied the proposed analysis in caching networks as a key motivating example. Finally, the analytic results were validated through simulations and real trace-driven experiments.

Appendix

Proof of Proposition 1: First, we define $\tilde{K}_{ij}^{[q]}(k, x), 1 \leq i, j \leq m_{tot}, k \geq 1, x \geq 0$, as the conditional probability that the Markov renewal process starting in the state $\left((n_1, \cdots, n_{q-1}, 0, n_{q+1}, \cdots, n_F)^T, i\right)$, returns to a state $\left((n'_1, \cdots, n'_{q-1}, 0, n'_{q+1}, \cdots, n'_F)^T, j\right)$, for the first time in exactly k transitions, and no later than time x. $\tilde{K}^{[q]}(k, x)$ denotes the matrix with elements $\tilde{K}_{ij}^{[q]}(k, x)$, and its transform matrix

is denoted by $K^{[q]}(z_q,\ s) = \sum_{k=1}^{\infty}\int_0^{\infty} e^{-sx}\, d\tilde{K}^{[q]}(k,\ x)z_q^k, |z_q| \leq 1, Re(s) \geq 0$. In the interest of brevity, a first-passage argument similar to the proofs given in [40] is used to obtain the following equations:

$$K^{[q]}(z_q,\ s) = z_q \sum_{k_1=0}^{\infty}\cdots\sum_{k_F=0}^{\infty} \left[\left(E^{[q]}B_{\sum_{f=1}^{F} k_f \boldsymbol{e}_f}(s)+(I-E^{[q]})A_{\sum_{f=1}^{F} k_f \boldsymbol{e}_f}(s)\right)\left(G^{[q]}(z_q,\ s)\right)^{k_q}\right], \tag{A.1}$$

where

$$G^{[q]}(z_q,\ s) = z_q \sum_{k_1=0}^{\infty}\cdots\sum_{k_F=0}^{\infty}\left[A_{\sum_{f=1}^{F} k_f \boldsymbol{e}_f}(s)\left(G^{[q]}(z_q,\ s)\right)^{k_q}\right], \tag{A.2}$$

$$G^{[q]}(z_q,\ s) = z_q \int_0^{\infty} e^{-sx}\exp\Big[(C_{tot} + D_{tot} + D_q(G^{[q]}(z_q,\ s) - I))x\Big]\, dH(x). \tag{A.3}$$

Using (10) and (A.3) reduces (A.1) to

$$K^{[q]}(z_q,\ s) = \left(E^{[q]}(sI - C_{tot})^{-1}D_{tot} + (I - E^{[q]})\right)G^{[q]}(z_q,\ s). \tag{A.4}$$

We define the invariant probability vector $\boldsymbol{k}^{[q]}$ of $K^{[q]} := K^{[q]}(1, 0)$, which satisfies $\boldsymbol{k}^{[q]}K^{[q]} = \boldsymbol{k}^{[q]}, \boldsymbol{k}^{[q]}\boldsymbol{e} = 1$, and we also define the vector $\boldsymbol{k}^{*[q]} = \frac{d}{dz_q}K^{[q]}(z_q)|_{z_q=1}\boldsymbol{e}$.
According to the theory of Markov renewal processes [41], $\boldsymbol{x}_0^{[q]}$ is expressed as

$$\boldsymbol{x}_0^{[q]} = \frac{\boldsymbol{k}^{[q]}}{\boldsymbol{k}^{[q]}\boldsymbol{k}^{*[q]}}. \tag{A.5}$$

From (A.4), it can be verified that

$$\boldsymbol{k}^{[q]} = \boldsymbol{g}^{[q]}\left(-E^{[q]}{C_{tot}}^{-1}D_{tot} + (I - E^{[q]})\right)^{-1}. \tag{A.6}$$

Subsequently, according to (A.4) we have

$$\boldsymbol{k}^{*[q]} = \left(-E^{[q]}{C_{tot}}^{-1}D_{tot} + (I - E^{[q]})\right)\frac{d}{dz_q}G^{[q]}(z_q)|_{z_q=1}\boldsymbol{e}. \tag{A.7}$$

Now, differentiating (A.2) yields

$$\frac{d}{dz_q}G^{[q]}(z_q)|_{z_q=1}\boldsymbol{e} = \left(I - G^{[q]} + \boldsymbol{e}\boldsymbol{g}^{[q]}\right)\left(I - A + (\boldsymbol{e} - \beta^{[q]})\boldsymbol{g}^{[q]}\right)^{-1}\boldsymbol{e}. \tag{A.8}$$

According to (A.5)-(A.8), the result follows.

Proof of Lemma 1: Writing the transform vector of each side of (11), the vector generating function $\mathbf{X}_{\vec{z}}$ satifies the following equation:

$$\mathbf{X}_{\vec{z}} = \mathbf{x_o}(B(\vec{z})-A(\vec{z}))+z_1^{-1}\cdots z_F^{-1}\left[\sum_{Z_1\subseteq S}\left(\prod_{f\in Z_1} z_f\right)\left(\mathbf{X}_{\vec{z}}|_{z_f=0\,\text{if}\,f\in Z_1} + \sum_{Z_2\subseteq S\setminus Z_1,\, Z_2\neq\varnothing}(-1)^{|Z_2|}\mathbf{X}_{\vec{z}}|_{z_f=0\,\text{if}\,f\in Z_1\cup Z_2}\right)\right]A(\vec{z}), \tag{A.9}$$

where $\mathbf{X}_{\vec{z}}|_{z_f=0\,\text{if}\,f\in Z}$ is equal to $\mathbf{X}_{\vec{z}'}$ in which $\vec{z}'$ is the same as $\vec{z}$ except that for any f in Z, the fth element of $\vec{z}'$ is zero.

In order to compute $\boldsymbol{X}^{[q]}(z_q)$, we put the vector $\vec{z} = \boldsymbol{e} + (z_q - 1)\boldsymbol{e}_q$ in (A.9) and note that for any choice of $Z^*, Z^* \subset S, Z^* \neq \{q\}$, the coefficient of $\boldsymbol{X}_{\vec{z}|_{z_f=0\,\text{if}\,f\ \in Z^*}}$ will be $\sum_{k=0}^{\lfloor\frac{|Z^*|}{2}\rfloor}\binom{|Z^*|}{2k} - \sum_{k=0}^{\lfloor\frac{|Z^*|-1}{2}\rfloor}\binom{|Z^*|}{2k+1} = 0$, which leads to (19).

Proof of Lemma 2: By setting $z_q = 1$ in (19), adding $X^{[q]}(1)\boldsymbol{e}\pi_{tot}$ to both sides and observing that $(I - A + \boldsymbol{e}\pi_{tot})$ is non-singular, we obtain

$$\boldsymbol{X}^{[q]}(1) = \pi_{tot} + u^{[q]}(1)(I - A + \boldsymbol{e}\pi_{tot})^{-1}. \tag{A.10}$$

Now, differentiating (19) leads to

$$\boldsymbol{X}'^{[q]}(z_q)(z_q I - A^{[q]}(z_q)) + \boldsymbol{X}^{[q]}(z_q)(I - A'^{[q]}(z_q)) = u'^{[q]}(z_q). \tag{A.11}$$

By setting $z_q = 1$ in (A.11), adding $X'^{[q]}(1)\boldsymbol{e}\pi_{tot}$ to both sides and noting that $(I - A + \boldsymbol{e}\pi_{tot})$ is non-singular, we obtain

$$\boldsymbol{X}'^{[q]}(1) = (\boldsymbol{X}'^{[q]}(1)\boldsymbol{e})\pi_{tot} + \left(u'^{[q]}(1) - \boldsymbol{X}^{[q]}(1)(I - A'^{[q]}(1))\right)(I - A + \boldsymbol{e}\pi_{tot})^{-1}. \tag{A.12}$$

By differentiating (A.11), setting $z_q = 1$, multiplying by $\boldsymbol{e}$ and noting that $A'^{[q]}(1)\boldsymbol{e} = \beta^{[q]}$, we have

$$\boldsymbol{X}'^{[q]}(1)\beta^{[q]} = \boldsymbol{X}'^{[q]}(1)\boldsymbol{e} - \frac{1}{2}\left(\boldsymbol{X}^{[q]}(1)A''^{[q]}(1) + u''^{[q]}(1)\right)\boldsymbol{e}. \tag{A.13}$$

Multiplying (A.12) by $\beta^{[q]}$ and substituting (A.13), the mean queue length of the qth queue at departures, i.e., $\boldsymbol{X}'^{[q]}(1)\boldsymbol{e}$, is obtained from (20).

Proof of Lemma 3: From the definition of E^* it follows that

$$E^* = -\mathbf{x_o}\left[\frac{\partial}{\partial s}B(\vec{z}, s)\right]_{\vec{z}=\boldsymbol{e},s=0}\boldsymbol{e} - [\boldsymbol{X_e} - \mathbf{x_o}]\left[\frac{\partial}{\partial s}A(\vec{z}, s)\right]_{\vec{z}=\boldsymbol{e},s=0}\boldsymbol{e}. \tag{A.14}$$

From (9) we have

$$\left[\frac{\partial}{\partial s}A(\vec{z}, s)\right]_{\vec{z}=\boldsymbol{e},s=0}\boldsymbol{e} = -h\boldsymbol{e}, \tag{A.15}$$

which in combination with (10) leads to

$$\left[\frac{\partial}{\partial s}B(\vec{z}, s)\right]_{\vec{z}=\boldsymbol{e},s=0}\boldsymbol{e} = hC_{tot}^{-1}D_{tot}\boldsymbol{e} - \left(C_{tot}^{-1}\right)^2 D_{tot}\boldsymbol{e}. \tag{A.16}$$

Next, we note that

$$C_{tot}^{-1}D_{tot}\boldsymbol{e} = C_{tot}^{-1}\left(Q_{tot} - C_{tot}\right)\boldsymbol{e} = -\boldsymbol{e}, \tag{A.17}$$

where the last equality follows from the fact that $Q_{tot}\boldsymbol{e} = \boldsymbol{o}$, since Q_{tot} is an infinitesimal generator matrix. Inserting (A.15), (A.16), and (A.17) in (A.14), and noting that $\boldsymbol{X_e e} = 1$, the result is obtained.

Proof of Lemma 4: First, we define

$$\mathbf{Y}(\vec{n}, t) = \{Y(\vec{n}, 1; t), Y(\vec{n}, 2; t), \cdots, Y(\vec{n}, m_{tot}; t)\}. \tag{A.18}$$

We also define the *vector renewal function* $\mathbf{M}_{\vec{n}}(.) = \left(M_{\vec{n},1}^{\vec{n}_0,j_0}(.) \quad M_{\vec{n},2}^{\vec{n}_0,j_0}(.) \quad \cdots \quad M_{\vec{n},m_{tot}}^{\vec{n}_0,j_0}(.)\right)$, $\vec{n} \geq \boldsymbol{o}$, of the Markov renewal process, where its components $M_{\vec{n},j}^{\vec{n}_0,j_0}(t)$ are the conditional expected number of visits to the state $(\vec{n}, j)$ in $[0, t]$, given the initial conditions $\vec{\xi}_0 = \vec{n}_0$, $J_{tot,0} = j_0$. The quantity $dM_{\vec{n},j}^{\vec{n}_0,j_0}(u)$ is then the conditional probability that the Markov renewal process enters the state $(\vec{n}, j)$, in the interval $(u, u + du)$.

Considering the state of the Markov renewal process at the epoch of the last departure before time t, we have

$$\boldsymbol{Y}(\boldsymbol{o}; t) = \int_0^t dM_{\boldsymbol{o}}(u)e^{C_{tot}(t-u)}. \tag{A.19}$$

From the key renewal theorem [31], it follows that

$$\boldsymbol{y_o} = \frac{1}{E^*}\mathbf{x_o}\int_0^\infty e^{C_{tot}t}dt = \frac{-1}{E^*}\mathbf{x_o}C_{tot}^{-1}. \tag{A.20}$$

Proof of Proposition 2: Writing the total probability considering two cases:
(a) t falls during the first service of a busy period, and
(b) t falls during the second or later services of a busy period, we obtain

$$\begin{aligned}\boldsymbol{Y}(\vec{n}; t) = &\int_0^t \int_0^{t-u} dM_{\boldsymbol{o}}(u)e^{C_{tot}v}dv \\ &\sum_{f=1, n_f\neq 0}^{F} D_f\tilde{P}(\vec{n} - \boldsymbol{e}_f, t-u-v)(1 - H(t-u-v)) \\ +\sum_{k_1=0}^{n_1}\cdots\sum_{k_F=0}^{n_F}&\int_0^t dM_{\sum_{f=1}^F k_f\boldsymbol{e}_f}(u)\tilde{P}(\vec{n}-\sum_{f=1}^{F}k_f\boldsymbol{e}_f, t-u)(1-H(t-u)) \\ &- \int_0^t dM_{\boldsymbol{o}}(u)\tilde{P}(\vec{n}, t-u)(1-H(t-u)).\end{aligned} \tag{A.21}$$

By computing the limit of (A.21) as $t \to \infty$ and applying the key renewal theorem, we obtain (25). Finally, computing the transform vector of (25) results in (26).

Proof of Proposition 3: Let N_q denote the average number of packets in queue q at an arbitrary time. It is readily seen that

$$N_q = \frac{d\boldsymbol{Y}^{[q]}(z_q)}{dz_q}|_{z_q=1}\boldsymbol{e}. \tag{A.22}$$

Using Little's Law, the average packet delay in the qth queue is given by

$$w_q = \frac{N_q}{\lambda_q}. \tag{A.23}$$

The final step is to find the mean queue lengths at an arbitrary time, in order to obtain N_q.
By setting $z_q = 1$ in (27) and (19), and using (10), we obtain $\boldsymbol{Y}^{[q]}(1) = \pi_{tot}$. By differentiating (27), setting $z_q = 1$, adding $\boldsymbol{Y}'^{[q]}(1)\boldsymbol{e}\pi_{tot}$ to both sides, and multiplying by $D_q\boldsymbol{e}$, we have

$$\begin{aligned}\boldsymbol{Y}'^{[q]}(1)\boldsymbol{e} = &\frac{1}{\lambda_q}\boldsymbol{Y}'^{[q]}(1)D_q\boldsymbol{e} + \frac{1}{\lambda_q E^*}\Big(\boldsymbol{X}'^{[q]}(1)(A-I) \\ &+ \boldsymbol{X}^{[q]}(1)\boldsymbol{A}'^{[q]}(1) - \mathbf{x_o}C_{tot}^{-1}(D_qA + Q_{tot}\boldsymbol{A}'^{[q]}(1)) - E^*\pi_{tot}D_q\Big) \\ &(\boldsymbol{e}\pi_{tot} - Q_{tot})^{-1}D_q\boldsymbol{e}.\end{aligned} \tag{A.24}$$

By double differentiating (27) and setting $z_q = 1$, we obtain

$$\begin{aligned}\boldsymbol{Y}'^{[q]}(1)D_q\boldsymbol{e} = &\frac{1}{E^*}\Big(\boldsymbol{X}'^{[q]}(1)\boldsymbol{e} - \frac{1}{2}u''^{[q]}(1)\boldsymbol{e} \\ &- \mathbf{x_o}C_{tot}^{-1}(D_qA'^{[q]}(1) + \frac{1}{2}Q_{tot}A''^{[q]}(1))\boldsymbol{e}\Big).\end{aligned} \tag{A.25}$$

Finally, by substituting (A.25) in (A.24), and using (A.22) and (A.23), the result is obtained.

References

[1] R. Ahlswede, N. Cai, S. Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204-1216, July 2000.
[2] F. Rezaei and B. H. Khalaj, "Stability, rate and delay analysis of single bottleneck caching networks," *IEEE Trans. Commun.*, vol. 64, no.1, pp. 300-313, Jan. 2016.
[3] J. Chen, V. C. S. Lee, K. Liu, and J. Li, "Efficient cache management for network-coding-assisted data broadcast," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3361-3375, Apr. 2017.
[4] M. K. Kiskani and H. R. Sadjadpour, "Multihop caching-aided coded multicasting for the next generation of cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2576-2585, Mar. 2017.
[5] V. Paxson and S. Floyd, "Wide-area traffic: the failure of poisson modeling," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 226-244, 1995.
[6] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causess," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835-846, 1997.
[7] M. F. Neuts, "Models based on the Markovian arrival process," *IEICE Trans. Commun.*, E75-B, pp. 1255-1265, 1992.
[8] I. Chatzigeorgiou and A. Tassi, "Decoding delay performance of random linear network coding for broadcast," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7050-7060, Aug. 2017.
[9] T. Tran and T. Nguyen, "Context-aware interflow network coding and scheduling in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9299-9318, Nov. 2016.
[10] Y. Qu, C. Dong, S. Guo, S. Tang, H. Wang, and C. Tian, "Spectrum-aware network coded multicast in mobile cognitive radio Ad Hoc networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5340-5350, June 2017.
[11] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," *in Proc. ACM Int. Conf. Mobi. Comput. Netw.*, Sept. 2007.
[12] H. Seferoglu and A. Markopoulou, "Opportunistic network coding for video streaming over wireless," *in Proc. IEEE Int. Conf. Pack. Video*, Nov. 2007.
[13] H. Seferoglu and A. Markopoulou, "Video-aware opportunistic network coding over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 713-728, June 2009.
[14] O. H. Abdelrahman and E. Gelenbe, "Queueing performance under network coding," *in Proc. IEEE Inf. Theory Workshop Netw. Inf. Theory*, pp. 135-139, 2009.

[15] B. Shrader and A. Ephremides, "Queueing delay analysis for multicast with random linear coding," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 421-429, 2012.
[16] T. Dikaliotis, A. Dimakis, T. Ho, and M. Effros, "On the delay of network coding over line networks," *in Proc. IEEE Int. Symp. Inf. Theory*, 2009.
[17] P. Parag and J. Chamberland, "Queueing analysis of a butterfly network for comaring network coding to classical routing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1890-1908, 2010.
[18] J. Charith Gunasekara, A. Alfa, and P. Yehampath, "A queueing theoritic model for opportunistic network coding," *in Proc. Int. Conf. Comput., Netw. Commun.*, pp. 999-1004, 2013.
[19] D. R. Cox, *Renewal Theory*, London, U. K.:Methuen, 1962.
[20] A. T. Andersen and B. F. Nielsen, "A Markovian Approach for Modeling Packet Traffic with Long-Range Dependence," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 5, pp. 719-732, June 1998.
[21] P. Salvador, "Multiscale fitting procedure using Markov Modulated Poisson Processes," *Telecommun. Systems*, vol. 23, no. 12, pp. 123-148, 2003.
[22] L. Muscariello, M. Mellia, M. Meo, M. A. Marsan, and R. L. Cigno, "Markov models of internet traffic and a new hierarchial MMPP model," *J. Comput. Commun.*, vol. 28, no. 16, pp. 1835-1851, 2005.
[23] F. Hamidi-Sepehr, H. D. Pfister, and J. Chamberland, "Delay-sensitive communication over fading channels: queueing behavior and code parameter selection," *IEEE Trans. Veh. Technol.*, vol. 64, no. 9, pp. 3957-3970, Sept. 2015.
[24] D. Lucantoni, K. Meier-Hellstern, and M. Neuts, "A single-server queue with server vacations and a class of non-renewal arrival processes," *Advances in Applied Probability*, vol. 22, no. 3, pp. 676-705, 1990.
[25] D. Lucantoni, "New results on the single-server queue with a batch Markovian arrival process," *Stochastic Models*, vol. 7, no. 1, pp. 1-46 1991.
[26] F. Rezaei, B. H. Khalaj, M. Xiao, and M. Skoglund, "Delay and Stability Analysis of Caching in Heterogeneous Cellular Networks," *in Proc. IEEE Int. Conf. Telecommun.*, May 2016.
[27] G. Latouche and V. Ramaswami, "Introduction to matrix analytic methods in stochastic modeling," *SIAM J. Applied Math.*, vol. 5, 1999.
[28] H. Neudecker, "A note on kronecker matrix products and matrix equation systems," *SIAM J. Applied Math.*, vol. 17, no. 3, pp. 603-606, 1969.
[29] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*, ser. McGraw-Hill series in electrical engineering: Communications and signal processing. Tata McGraw-Hill, 2002.
[30] C. T. Kelly, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, 1995.
[31] M. Grosslauser and J. Bolot, "On the relevance of long-range dependencies in network traffic," *IEEE/ACM Trans. Netw.*, vol. 7 no. 5, pp. 629-640, 1999.
[32] S. Ben Fredj, T. Bonald, A. Proutiere, G. Regnie, and J. Roberts, "Statistical Bandwidth Sharing: a Study of Congestion at Flow Level," *in Proc. ACM SIGCOMM*, pp. 111-122, Aug. 2001.
[33] T. Bonald, A. Proutiere, G. Regnie, and J. Roberts, "Insensitivity Results in Statistical Bandwidth Sharing," *in Proc. Int. Teletraffic Conf.*, Nov. 2001.
[34] D. I. Choi, "MAP/G/1/K queue with multiple thresholds on buffer," *Comm. Korean Math. Soc.*, vol. 14, no. 3, pp. 611-625, 1999.
[35] A. Baiocchi, "Analysis of the loss probability of the map/g/1/k queue," *Commun. Statist.-Stochastic Models*, vol. 10, no. 4, pp. 867-893, 1994.
[36] M. Ji, G. Caire, and A. F. Molisch, "The throughput-outage tradeoff of wireless one-hop caching networks," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6833-6859, 2015.
[37] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856-2867, 2014.
[38] B. F. Nielsen, "Note on the Markovian Arrival Process," *Stochastic Processes*, 1998.
[39] M. Zink, K. Suh, Y.Gu, and J.Kurose, "Characteristics of YouTube network traffic at a campus network - Measurements, models, and implications," *Int. J. Comput. Telecommun. Netw.*, vol. 53 no. 4, pp. 501-514, 2009.
[40] M. F. Neuts and D. M. Lucantoni, "Simpler proofs of some properties of the fundamental period of the map/g/1 queue," *J. Applied Probability*, vol. 31, no. 1, pp. 235-243, 1994.
[41] E. Cinlar, "Markov renewal theory," *Advances in Applied Probability*, vol. 1, no. 2, pp. 123-187, 1969.

Fatemeh Rezaei received the B.Sc., M.Sc. and PhD degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2010, 2012 and 2017, respectively. She spent a sabbatical stay at KTH University, Stockholm, Sweden, in 2015. Her current research interests include caching analysis in communication networks, heterogeneous cellular networks, wireless and mobile communications, 5G networks, and distributed systems.

Ahmadreza Momeni received the B.S. degree in electrical engineering from Sharif University, Tehran, Iran in 2015, and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Stanford University, Stanford, CA. His current research interests include performance optimization, and decision making under uncertainty with a focus on online advertising, and social networks.

Babak Hossein Khalaj received the B.Sc. degree in Electrical Engineering from the Sharif University of Technology, Tehran, Iran, in 1989, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1993 and 1996, respectively. He has been with the pioneering team at Stanford University, where he was involved in adoption of multi-antenna arrays in mobile networks. He joined KLA-Tencor in 1995, as a Senior Algorithm Designer, working on advanced processing techniques for signal estimation. From 1996 to 1999, he was with Advanced Fiber Communications and Ikanos Communications. Since then, he has been a Senior Consultant in the area of data communications, and a Visiting Professor at CEIT, San Sebastian, Spain, from 2006 to 2007. He has co-authored many papers in signal processing and digital communications. He holds two U.S. patents and the recipient of Alexander von Humboldt Fellowship from 2007 to 2008. He was the Co-Editor of the Special Compatibility Standard Draft for ANSIT1E1 Group from 1998 to 1999.