An Analytical Study of Network Coding in the Presence of Real-Time Messages

Marc Aoun
Philips Research
Eindhoven, The Netherlands
marc.aoun@philips.com

Paul Beekhuizen
Eurandom, TU/e
Eindhoven, The Netherlands
beekhuizen@eurandom.tue.nl

Antonios Argyriou
Electrical & Computer Eng. Dpt.
University of Thessaly, Greece
anargyr@uth.gr

***Abstract*—With network coding, two data packets are transformed into one by a simple XOR-operation. The transformed packet is transmitted and each original packet can be retrieved at its destination node through a similar XOR-operation. Network coding is an important research topic as it radically challenges existing networking paradigms.**

In this paper we provide an analytical study of the impact network coding has on the delivery of real-time packets (i.e., packets with deadlines). We model a router as a queueing system where packets arrive from two independent Poisson flows. We obtain an exact expression for the goodput of the system and study the goodput gain that can be achieved by performing network coding. We verify the validity of the model through simulations.

I. Introduction

The idea of coding packets in data networks prior to routing has spurred a plethora of research activities in the networking area. Network coding gained a lot of interest in both wireline and wireless networks because of its potential to radically affect the way networks operate.

The increased interest in network coding comes at a time when the Internet becomes a carrier of more time-sensitive information and new generation of systems such as wireless sensor networks offer unprecedent monitoring capability of time-critical physical environments. High data rate applications like video-on-demand and high quality interactive video communications demand timeliness guarantees as well.

When packets with timeliness restrictions undergo network coding, key QoS parameters might be improved but also compromised, so additional insight into the performance of systems with network coding and timeliness requirements is needed. In this paper we therefore develop an analytical model of network coding in the presence of real-time packets, i.e. packets with delivery deadlines.

We model a router node to which two types of packets arrive and must be forwarded, and we provide a detailed derivation of its real-time goodput. The model we propose is an M/M/1/2 queueing system, adjusted in such a way that it can perform network coding. We provide exact expressions for the stationary real-time goodput at the router node, and verify them with simulations. Our work provides analytical tools for further exploration of the network coding concept. To the best of our knowledge, our work is the first to model network coding while looking at it from a real-time perspective.

II. Related Work

This paper advances network coding in the direction of performance modeling in the presence of real-time flows associated with specific packet deadlines. We review the most noteworthy of the research works related to this area next.

In the real-time queueing theory domain, Lehoczky [7] analysed an $M/M/1$ queue with deadlines to service endings and the edf-policy in heavy traffic. He argued that, since the deadlines of all stored packets have to be taken into account, this queue gives rise to a Markov process on a statespace of infinite dimension. Lehoczky shows that the Markov process collapses to a tractable one-dimensional process in heavy traffic. Lehozcky later used these results to analyse control policies in [8], and extended his analysis to Jackson networks in [9]. Doytchinov et al. [3] extended this analysis to a $GI/G/1$ queue with deadlines, and Kruk et al. [6] to networks of such queues.

Delay sensitive traffic in the presence of network coding was studied in [10]. The authors adopted a statistical QoS measure that expresses the decay rate of the buffer at the middle node in a butterfly network. This router node is the bottleneck in the butterfly case, which explains the reasons why the authors focus on its buffer behavior. Although this metric is enough for approximating the delivered QoS per flow, it does not express it directly in terms of the achieved packet error rate and goodput. Shah et al. [11] start with the goal of minimizing the backlog of coded packets at receiving nodes. They design an online algorithm so that the linear packet combinations that are generated, are chosen in such a way that their actual span excludes any linear combination that is already known to the receivers.

Eryilmaz et al. [4] study the delay benefits of network coding in wireless multicast and multiple unicast scenarios. They present a model that considers only a single-hop transmission and the random coding across packets from the same flow (intra-session network coding). Online network coding and delay minimization was more recently presented in [2]. A precise model is not presented in that work, although a simple delay analysis for the wireless channel with Bernoulli erasures is performed. Another interesting modeling work can be found in [13] where the authors used stochastic network calculus for calculating the throughput in a coded butterfly network. Never-

978-1-4244-7188-1/10/$26.00

theless, the model does not cover delay and timeliness aspects. A recent work from Goseling et al. [5] aims at modeling the performance of coded queueing systems. Instead of focusing on providing exact expressions, the authors provide bounds on the performance of a coded tandem queuing network with two independent flows. Nevertheless, the extension of this work to packets with deadline requirements is not straightforward.

III. Model

A. System and Queueing Model

We focus our analysis on a router element R to which two flows of packets arrive and need to be routed. The router node is capable of opportunistically coding two plain packets belonging to different flows, and transmitting the resulting coded packet instead of two plain transmissions.

We model the router R as a single server queueing system. Packets from each of the two flows arrive to node R according to a Poisson process with parameters λ_1 and λ_2, respectively. Such packets will be referred to as *plain*, i.e. not coded. Packets of the first flow are type-1 packets, those of the second flow are type-2 packets, and a coded packet is referred to as type-3. The service time of a packet at the router, i.e. the time it takes node R to retransmit a received packet, follows an exponential distribution with parameter μ. The sum of the two arrival rates λ_1 and λ_2 is denoted by λ.

The queueing system has a single-space waiting buffer, which can hold one packet while the server is busy. The buffer is an overwrite one; a plain packet is overwritten by a newer arriving one if they are of the same type, otherwise they are coded together and the resulting coded packet occupies the waiting buffer. A coded packet is overwritten by any newly arriving packet. The server has an additional buffer that holds the packet currently being served. The result is thus a modified version of an $M/M/1/2$ queueing system (note that, in an $M/M/1/K$ system, the K refers to the number of buffer positions, including the one in service).

We shall restrict our real-time requirements on the packets to the same constant relative deadline *d* that is set once the packet becomes available at node R. The absolute deadline of a packet corresponds to the sum of its arrival moment at node R and its relative deadline. For the packet to meet its real-time requirement, it should be retransmitted by node R before its absolute deadline expires. The restriction to one relative deadline value *d* neither impacts the quality nor the representativeness of the analysis. It is merely used to simplify the underlying presentation. The extension of our analysis to different deadline distributions is straightforward.

Any plain packet whose deadline has expired is removed from the system. A coded packet is removed if the deadlines of both of its plain packets have expired.

B. System State

The state of the queueing system at the router is fully conveyed through the characterization of its waiting buffer's status and the status of its server. Two variables will be used to keep track of the system state: variable L, that defines the current number of packets in the system (included any packet being served), and variable N, that defines the type of the packet residing in the buffer, if any. Knowledge of the type of a packet when it is in service is not required for the analysis.

Variable L can take one of three values: 0, 1, or 2. Note that a coded packet is counted as a single packet in the system. Variable N can also take one of three values: 1 (packet in buffer is of type-1), 2 (packet in buffer is of type-2), or 3 (packet in buffer is a coded packet, type-3).

IV. Goodput Analysis

We define the *goodput* of a flow as the number of packets of that flow that are served by the router within their deadline requirement, per unit of time. Given the symmetry of the problem, we will concentrate our analysis on the goodput of type-1 packets. The analysis for packets of type-2 is identical.

A packet of type-1 arriving to the system can find it in one of five states:

1) *Empty (L = 0)*: The server is available, and there is no packet occupying the waiting buffer.
2) *Busy Server (L = 1)*: The server is servicing a packet, but the waiting buffer is free.
3) *Full, type-1 packet in buffer (L = 2, N = 1)*: the server is busy. The waiting buffer is occupied by a type-1 packet.
4) *Full, type-2 packet in buffer (L = 2, N = 2)*: the server is busy. The waiting buffer is occupied by a type-2 packet.
5) *Full, type-3 packet in buffer (L = 2, N = 3)*: the server is busy. The waiting buffer is occupied by a coded packet.

Theorem 1. *The goodput of type-1 packets is given by the following expression:*

$$\gamma_1 = \lambda_1 \sum_{a,b} \mathbb{P}(L = a, N = b)\mathbb{P}(success|L = a, N = b), \tag{1}$$

where $\mathbb{P}(L = a, N = b)$ is the probability that the system is in state $(L = a, N = b)$ and $\mathbb{P}(success|L = a, N = b)$ is the probability that an arbitrary type-1 packet arriving to the system in state (L = a, N = b) meets its deadline.

Proof: The goodput γ_1 of type-1 packets is by definition equal to the probability that the service of an arbitrary packet of flow 1 is successful, i.e. meets its deadline. By the PASTA property (Poisson Arrivals See Time Averages, see [12]), the probability that an arbitrary type-1 packet arrives to the system in state $(L = a, N = b)$ is equal to the steady-state probability that the system is in state $(L = a, N = b)$. The overall service success probability is therefore given by summing, over the entire state space, the probability that the system is in state $(L = a, N = b)$ times the success probability conditioned on the occurrence of this state. ■

Finding γ_1 amounts to finding the success probabilities per system state, $\mathbb{P}(\text{success}|L = a, N = b)$, and the different state probabilities $\mathbb{P}(L = a, N = b)$. We derive these probabilities in the following two subsections.

A. Success Probability per System State

We start by finding the probabilities $\mathbb{P}(\text{success}|L = a, N = b)$ that a packet of type-1 arriving to the system when it is in state $(L = a, N = b)$ is served by the router server and completes its service before the expiration of its deadline. The arrival of the packet is considered as time origin. Therefore, the absolute deadline of the packet is equal to its relative deadline.

1) Case 1: Arrival to an Empty System (L = 0): A packet arriving to an empty system directly enters service. Its success probability is equal to:

$$\mathbb{P}(success|L = 0) = (1 - e^{-\mu d}), \tag{2}$$

which is equal to the probability that the service time it experiences is smaller than its deadline d.

2) case 2: Arrival to a Busy Server (L = 1) or Full System (L = 2) with N = 1 or 3: Due to the overwrite property of the waiting buffer, an arrival of a type-1 packet to a Busy Server state is identical to an arrival when the system is full and the buffered packet is of type-1 or type-3. Indeed, in all three cases, the new arriving packet will occupy (or overwrite) the buffer and wait to be serviced. Therefore $\mathbb{P}(\text{success}|L = 2, N = 1)$ and $\mathbb{P}(\text{success}|L = 2, N = 3)$ are both equal to $\mathbb{P}(\text{success}|L = 1)$, given by:

$$\begin{aligned}\mathbb{P}(success|L = 1) = \int_0^d &\mu e^{-\mu t} e^{-\lambda_1 t}(1 + \lambda_2 t)e^{-\lambda_2 t} \\ &\times (1 - e^{-\mu(d-t)})dt.\end{aligned} \tag{3}$$

To derive Equation (3), we first condition on the length of the residual service time *t* of the current packet in the server. Being exponentially distributed, this residual service time has a density equal to $\mu e^{-\mu t}$. Given that the residual service time is t, a buffered packet is successfully serviced if all following conditions are met:

1) The server becomes free before the deadline expiration of the buffered packet, i.e., $t \leq d$.
2) The packet is not overwritten while waiting in the buffer.
3) The service time of the packet is smaller than the remaining time the packet has until deadline expiration.

Condition 1 is accounted for in the integration region (0 to d). Condition 2 is met if and only if, starting from the arrival moment 0 of the buffered packet until the residual service time *t* is completed, no new type-1 packet arrives and at most one arrival of type-2 occurs. Indeed, the first arrival of a type-2 packet will not result in overwriting the buffered packet, since it will be coded with it (resulting in a type-3 packet). Any arrival of type-1 is not tolerated since it overwrites both a type-1 and a type-3 buffered packet. The term $e^{-\lambda_1 t}$ gives the probability that no type-1 arrivals occur during t. The term $(1 + \lambda_2 t)e^{-\lambda_2 t}$ gives the probability that at most one type-2 arrival occurs.

Finally, condition 3 is met if the service time experienced by the packet is *at most* equal to the remaining time $(d - t)$ until deadline expiration. The term $(1 - e^{-\mu(d-t)})$ is equal to the probability that the service time is smaller than $d - t$.

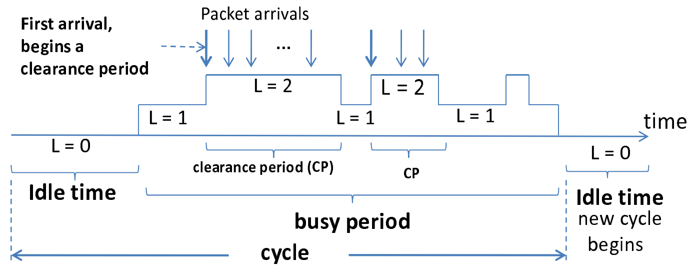

Fig. 1: A cycle consisting of idle time and busy period.

3) case 3: Arrival to a Full System with Buffered Type-2 Packet (L = 2, N = 2): If the type-1 packet arrives to a system where the waiting buffer is occupied by a type-2 packet, both packets will be coded together, resulting in a type-3 buffered packet. The success probability of the arriving packet is therefore equal to the probability that the resulting type-3 packet is served before the deadline d of the arriving packet is expired. It is given by:

$$\mathbb{P}(success|L = 2, N = 2) = \int_0^d \mu e^{-\mu t} e^{-\lambda_1 t} e^{-\lambda_2 t}(1-e^{-\mu(d-t)})dt. \tag{4}$$

Equation (4) differs from Equation (3) in that no arrival of type-2 is allowed at all, throughout the whole sojourn time of the coded packet in the buffer. Indeed, any type-2 arrival will overwrite the coded packet. This arrival restriction on type-2 packets is reflected in Equation (4) through the term $e^{-\lambda_2 t}$.

B. System State Probabilities

In order to determine the probabilities $\mathbb{P}(L = a, N = b)$ of finding the system in a particular state $(L = a, N = b)$, we introduce the notion of cycle time, which we define as the time between two consecutive moments at which the system becomes empty (i.e. L becomes equal to 0). The cycle time *C* consists of two parts: A first part, the 'idle time' *I*, during which the system is empty, and a second part, the 'busy period' *BP*, during which the system serves packets until it becomes empty again. Fig. 1 illustrates an example of a cycle.

The busy period spreads throughout the time that the system is not in a state $L = 0$. It consists of a collection of consecutive states characterized by $L = 1$ or $L = 2$. We shall refer to the case where $L = 2$ during a busy period as clearance period (CP), which is the amount of time the buffer is continuously occupied. A clearance period ends when the buffer is cleared, i.e. when either a dispatch to the server occurs, or when the stored packet is deleted upon its deadline expiration.

The arrival that causes a state change from $L = 1$ to $L = 2$ (i.e. that starts a clearance period) is referred to as *first arrival*. As a busy period might comprise multiple clearance periods, there can be multiple first arrivals during a busy period.

The idle time lasts from the moment the system becomes empty to the arrival of a packet of either type-1 or type-2. Therefore, the mean idle time is equal to $1/(\lambda_1 + \lambda_2)$. The mean cycle time is given by the sum of the mean idle time and the mean busy period. We then have:

$$\mathbb{E}[C] = \frac{1}{\lambda_1 + \lambda_2} + \mathbb{E}[BP] \tag{5}$$

The mean busy period is given by:

$$\mathbb{E}[BP] = \frac{1}{\lambda_1 + \lambda_2 + \mu} + \frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2 + \mu} \left(\mathbb{E}[T] + \mathbb{E}[BP] \right), \tag{6}$$

where $\mathbb{E}[T]$ is the mean clearance period. We will derive $\mathbb{E}[T]$ shortly hereafter.

Equation (6) follows from a probabilistic argument: The busy period starts when a packet arrives to an empty system. After this, the expected time until the first event (an arrival or service completion) is $1/(\lambda_1 + \lambda_2 + \mu)$. Furthermore, with probability $\mu/(\lambda_1 + \lambda_2 + \mu)$, the first event is a service completion, and the busy period ends. With probability $(\lambda_1 + \lambda_2)/(\lambda_1 + \lambda_2 + \mu)$, the first event is an arrival. In this case, the arriving packet is stored in the buffer and the buffer has to be cleared, which takes in expectation $\mathbb{E}[T]$ time units. Once the buffer has been cleared, the expected time before the system becomes empty is again equal to $\mathbb{E}[BP]$ due to the memorylessness of service times. Evaluating (6) yields

$$\mathbb{E}[BP] = \frac{1}{\mu} + \frac{\lambda_1 + \lambda_2}{\mu} \mathbb{E}[T]. \tag{7}$$

It remains to determine $\mathbb{E}[T]$, the expected length of the buffer clearance period. Without loss of generality, we assume that the clearance period starts at time 0. We condition on t, the time at which the next event (arrival or service completion) occurs. Because the time until the next event is exponentially distributed with parameter $\lambda_1 + \lambda_2 + \mu$, we obtain:

$$\begin{aligned}\mathbb{E}[T] = &\int_d^\infty (\lambda_1 + \lambda_2 + \mu) e^{-(\lambda_1+\lambda_2+\mu)t} d \, dt \\ &+ \int_0^d \frac{\mu}{\lambda_1 + \lambda_2 + \mu} (\lambda_1 + \lambda_2 + \mu) e^{-(\lambda_1+\lambda_2+\mu)t} t \, dt \\ &+ \int_0^d \frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2 + \mu} (\lambda_1 + \lambda_2 + \mu) e^{-(\lambda_1+\lambda_2+\mu)t} (t + \mathbb{E}[T]) dt.\end{aligned} \tag{8}$$

The three integrals (from top to bottom) cover the following three possibilities: First, if there are no events before time d, the buffer is cleared at time d because the deadline of the buffered packet becomes expired. Second, if the first event occurs at time $t < d$, and the first event is a service completion, the buffer is cleared at time t because the buffered packet enters service. Third, if the first event occurs at time $t < d$ and is a packet arrival, the arriving packet overwrites the buffered packet, and a new buffer clearance period begins.

After rewriting and evaluating the integrals (the first two using partial integration), we obtain:

$$\mathbb{E}[T] = \frac{1 - e^{-(\lambda_1+\lambda_2+\mu)d}}{\mu + (\lambda_1 + \lambda_2) \, e^{-(\lambda_1+\lambda_2+\mu)d}}. \tag{9}$$

A standard argument from renewal theory implies that the probability that the system is in a certain state is given by the mean time the system spends in this state during a cycle, divided by the mean cycle time. Let C_i be the total time the system has a packet of type *i* in the buffer during a cycle time. In other words, C_i is the total amount of time the system is in state $(L = 2, N = i)$ during a cycle. The system state probabilities are then obtained using:

$$\mathbb{P}(L = 0) = \frac{\mathbb{E}[I]}{\mathbb{E}[C]}, \tag{10}$$

$$\mathbb{P}(L = 2, N = i) = \frac{\mathbb{E}[C_i]}{\mathbb{E}[C]}, and \tag{11}$$

$$\mathbb{P}(L = 1) = 1 - P(L = 0) - \sum_{i=1}^{3} P(L = 2, N = i). \tag{12}$$

To find $\mathbb{E}[C_i]$, we define $T_{i,j}$ as the cumulative amount of time that the buffer is occupied by type-i packets during a clearance period, given that the *first arrival* of that clearance period was a type-j packet. The expected value of C_i is given by:

$$\begin{aligned}\mathbb{E}[C_i] &= \sum_{j=1}^{2} \frac{\lambda_j}{\lambda + \mu} (\mathbb{E}[T_{i,j}] + \mathbb{E}[C_i]) &(13)\\ &= \sum_{j=1}^{2} \frac{\lambda_j}{\mu} (\mathbb{E}[T_{i,j}]). &(14)\end{aligned}$$

It now finally remains to find the expected values of the different $T_{i,j}$. The expected value of $T_{1,1}$ is given by:

$$\begin{aligned}\mathbb{E}[T_{1,1}] = &\int_0^d (\lambda + \mu) e^{-(\lambda+\mu)t} \left[\frac{\mu}{\lambda + \mu} t + \frac{\lambda_1}{\lambda + \mu} \left(\mathbb{E}[T_{1,1}] + t \right) \right. \\ &\left. + \frac{\lambda_2}{\lambda + \mu} (\mathbb{E}[T_{1,2}] + t) \right] dt + d e^{-(\lambda+\mu)d}.\end{aligned} \tag{15}$$

Equation (15) conveys the following: After the *first arrival* an event is bound to occur: either a service completion, or a new arrival. We condition on t, the time at which the next event occurs. Because of the memorylessness of the system, t has density $(\lambda + \mu) e^{-(\lambda+\mu)t}$. With probability $\frac{\mu}{\lambda+\mu}$, this event is a service completion, and the clearance period is equal to t. With probability $\frac{\lambda_2}{\lambda+\mu}$, the first event is a type-2 arrival. Due to memorylessness, this can be seen as the beginning of a new clearance period with a type-2 arrival as *first arrival*, i.e., with expectation $\mathbb{E}[T_{1,2}]$. However, up to time t, there was a type-1 packet in the buffer, so the expected *cumulative* amount of time type-1 packets spend in the buffer is given by $t + \mathbb{E}[T_{1,2}]$. With probability $\frac{\lambda_1}{\lambda+\mu}$, the first event is a type-1 arrival. Likewise, this can be seen as a new clearance period with a type-1 arrival as a *first arrival*, i.e., with expectation $\mathbb{E}[T_{1,1}]$ and cumulative amount of time equal to $t + \mathbb{E}[T_{1,1}]$. Finally, with probability $e^{-(\lambda+\mu)d}$, no event happens prior to the deadline expiration and subsequent removal of the type-1 first arrival. In that case, the buffer would have been occupied by this type-1 packet for an amount of time equal to d.

Likewise, the expected values of $T_{1,2}$ and that of $T_{1,3}$ are respectively given by:

$$\mathbb{E}[T_{1,2}] = \int_0^d (\lambda+\mu)e^{-(\lambda+\mu)t}\left[\frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{1,3}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{1,2}]\right]dt, \tag{16}$$

$$\mathbb{E}[T_{1,3}] = \int_0^d (\lambda+\mu)e^{-(\lambda+\mu)t}\left[\frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{1,1}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{1,2}]\right]dt. \tag{17}$$

Let $\alpha = 1 - e^{-(\lambda+\mu)d}$. The upper set of integrals results in the following set of three equations with three unknowns, which can be solved to find $\mathbb{E}[T_{1,1}]$, $\mathbb{E}[T_{1,2}]$ and $\mathbb{E}[T_{1,3}]$:

$$\mathbb{E}[T_{1,1}] = \alpha\left[\frac{1}{\lambda+\mu} + \frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{1,1}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{1,3}]\right] \tag{18}$$

$$\mathbb{E}[T_{1,2}] = \alpha\left[\frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{1,3}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{1,2}]\right] \tag{19}$$

$$\mathbb{E}[T_{1,3}] = \alpha\left[\frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{1,1}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{1,2}]\right] \tag{20}$$

Following a similar reasoning,

$$\mathbb{E}[T_{2,1}] = \alpha\left[\frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{2,1}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{2,3}]\right] \tag{21}$$

$$\mathbb{E}[T_{2,2}] = \alpha\left[\frac{1}{\lambda+\mu} + \frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{2,3}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{2,2}]\right] \tag{22}$$

$$\mathbb{E}[T_{2,3}] = \alpha\left[\frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{2,1}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{2,2}]\right] \tag{23}$$

and

$$\mathbb{E}[T_{3,1}] = \alpha\left[\frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{3,1}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{3,3}]\right] \tag{24}$$

$$\mathbb{E}[T_{3,2}] = \alpha\left[\frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{3,3}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{3,2}]\right] \tag{25}$$

$$\mathbb{E}[T_{3,3}] = \alpha\left[\frac{1}{\lambda+\mu} + \frac{\lambda_1}{\lambda+\mu}\,\mathbb{E}[T_{3,1}] + \frac{\lambda_2}{\lambda+\mu}\,\mathbb{E}[T_{3,2}]\right] \tag{26}$$

Once the different $\mathbb{E}[T_{i,j}]$ are found, they are used in Equation (11) to find the state probabilities $\mathbb{P}(L = 2, N = i)$. Replacing the result of Equations (10), (11), and (12) along with the different conditional success probabilities (Equations (2)-(4)) into Equation (1) yields the type-1 goodput of the system, which is a function of λ_1, λ_2, μ and d.

V. Numerical Results

In this section, we study numerically the goodput gain of network coding relative to the no-coding base case. We are particularly interested in finding how much the goodput can be increased by applying network coding. We define γ_{base} as the total goodput of the router without coding, and γ_{coding} as the total goodput of the router with coding. An important remark to be made here is that γ_{base} can be found using Equation (1), by considering the existence of a single arriving flow instead of two, with parameter $\lambda = \lambda_1 + \lambda_2$. In other words,

$$\gamma_{base}\left\{\lambda_1, \lambda_2, \mu, d\right\} = \gamma_1\left\{\lambda_1 + \lambda_2, 0, \mu, d\right\}. \tag{27}$$

We define the goodput gain as the relative increase in goodput, i.e., as $(\gamma_{coding} - \gamma_{base})/\gamma_{base}$.

In the sequel, we fix $\mu = 1$. We can make this assumption without loss of generality; all parameters are relative to each other, so for any set of parameters, we can scale time in such a way that $\mu = 1$ without changing the goodput of the system.

Fig. 2 provides the goodput gain as a function of the arrival rates λ_1 and λ_2, for d equal to 1. A major conclusion to be drawn here is that the gain provided by network coding is always maximized when λ_1 is equal to λ_2. This result is logical, since equal arrival rates provide the most opportunities for coding.

Fig. 3 provides the maximal goodput gain for d between 0 and 5, and λ between 0 and 5. Similarly, Fig. 4 provides the maximal goodput gain for the same range of deadlines and larger values of λ, between 5 and 15.

For a fixed arrival rate λ, the gain increases for an increasing deadline value. A larger deadline provides a bigger probability of success for coded packets, hence a positive return on applying network coding. Similarly, for a fixed deadline d, the gain increases for an increasing arrival rate; This can be explained by the more efficient buffer usage that network coding provides; In the base case, every arrival overwrites the buffer. The higher the arrival rate, the more frequent such overwritings occur. On the other hand, with network coding, an arrival of type-1 when the buffer is occupied by a type-2 packet, or vice-versa, does not result in an overwriting, but in a coding operation which maintains both packets.

As shown in Fig. 3, the gain provided by network coding remains limited in the operation region ($\lambda < 2$, $d < 1$), where low arrival rates limit the opportunities of coding, and the short deadlines causes most of the coded packets to miss their timeliness requirement anyway.

Finally, as conveyed in Fig. 4, the gain of network coding for high arrival rates and large deadlines can reach up to 30%, a substantial improvement compared to the base case scenario.

Numerical Results Validation

To assess the accuracy of our model and the obtained numerical results, the queueing system was simulated in Java. The

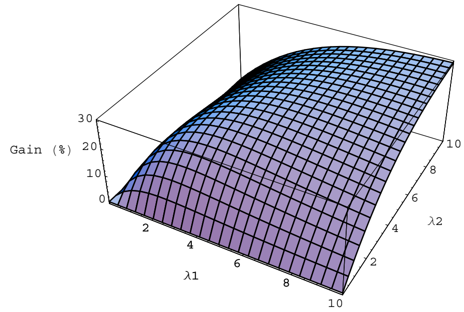

Fig. 2: Goodput gain (%), function of λ_1 and λ_2 when d = 1.

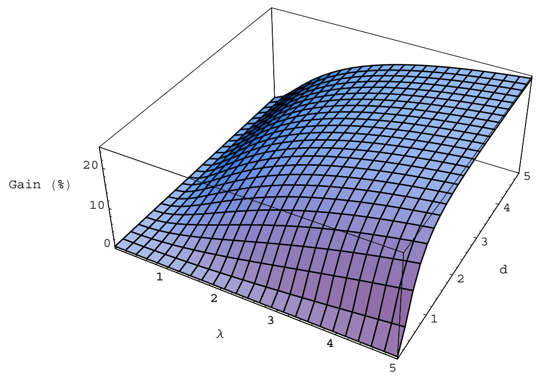

Fig. 3: Goodput gain (%) for values of λ between 0 and 5 and values of d between 0 and 5.

barchart in Fig. 5 provides a comparison between numerical and simulated maximum gain results, for integer values of λ between 1 and 8, and a deadline equal to 1. Every simulation outcome corresponds to the average of 50 simulation runs. The corresponding standard deviation values have been omitted from the barchart, due to their small values (the maximum encountered standard deviation value was equal to 0.1%).

As conveyed in Fig. 5, the simulation output accurately matches the numerical results.

VI. Conclusion

We developed in this paper an analytical model that captures the impact of network coding on the delivery of real-time packets. The proposed model is an M/M/1/2 router queueing system with network coding capability and per-packet deadlines. We provided exact expressions for the stationary goodput of the system and numerical results of the achievable goodput gain. We further verified the validity of our model through simulations. We show that depending on the operation region, the gain of network coding can reach up to 30%.

In addition to the novel approach that looks at network coding from a real-time perspective, our work paves the way for future research; among the topics we foresee as interesting is a comparative study between network coding and lead-time based service reneging, where denial-of-service decisions can be made based on remaining time until deadline expiration. We further foresee the importance of qualitative and quantitative studies of network coding in real-time wireless sensor networks, along with the different requirements, conditions and new challenges pertaining to this emerging domain.

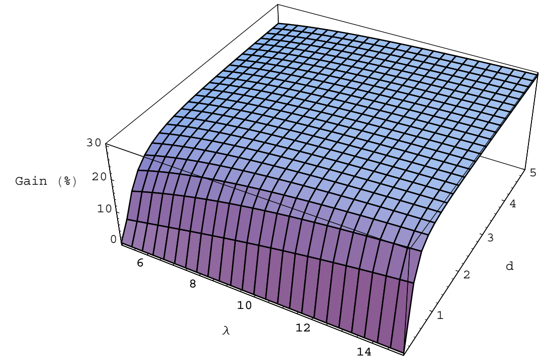

Fig. 4: Goodput gain (%) for values of λ between 5 and 15.

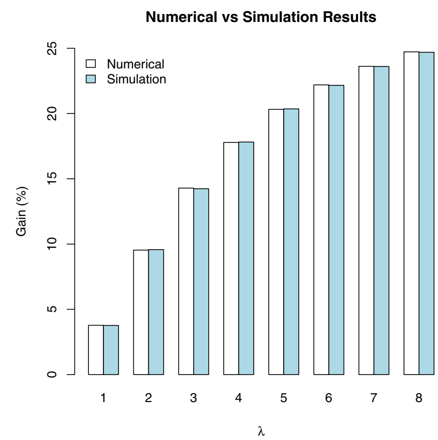

Fig. 5: Numerical vs simulation results,when d = 1.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[2] J. Barros, R. A. Costa, and J. Widmer. Effective delay control for online network coding. In *INFOCOM*, April 2009.

[3] B. Doytchinov, J. Lehoczky, and S. Shreve. Real-time queues in heavy traffic with earliest-deadline-first queue discipline. In *The Annals of Applied Probability* 11(2):332–378, 2001.

[4] A. Eryilmaz, A. Ozdaglar, and M. Medard. On delay performance gains from network coding. In *Conference on Information Sciences and Systems*, 2006.

[5] J. Goseling, R. J. Boucherie, and J.-K. Ommeren. Energy consumption in coded queues for wireless information exchange. In *Workshop on Network Coding, Theory, and Applications (NetCod)*, June 2009.

[6] L. Kruk, J. Lehoczky, S. Shreve, and S-N. Yeung. Earliest-deadline-first service in heavy-traffi acyclic networks. In *The Annals of Applied Probability* 14(3):1306–1352, 2004.

[7] J. Lehoczky. Real-time queueing theory. In *Proc. of IEEE Real-Time Systems Symposium* 1996, pp. 186–195.

[8] J. Lehoczky. Using real-time queueing theory to control lateness in real-time systems. In *Performance Evaluation Review* 25:158–168, 1997.

[9] J. Lehoczky. Real-time queueing network theory. In *Proc. of IEEE Real-time Systems Symposium 1997*, pp. 58–67.

[10] P. Parag and J.-F. Chamberland. Queuing analysis of a butterfly network. In *IEEE International Symposium on Information Theory (ISIT)*, 2008.

[11] D. Shah, M. Medard, and J. K. Sundararajan. On queueing in coded networks – queue size follows degrees of freedom. In *Information Theory Workshop*, 2007.

[12] R.W. Wolff. Poisson arrivals see time averages. In *Operations Research*, 30(2):223–231, 1982.

[13] K. Wu, W. Jia, Y. Yuan, and Y. Jiang. Performance modeling of stochastic networks with network coding. In *Workshop on Network Coding, Theory, and Applications (NetCod)*, June 2009.