Calculus of Service Guarantees for Network Coding

Ali Mahmino and Jérôme Lacan

***Abstract*— A large class of networks is able to provide some guarantees in terms of quality of service, end-to-end delays and throughput to data flows. In return, the data flows must verify constraints of burstiness and throughput. The aim of this work is to introduce and evaluate the network coding for independent flows in such networks. First, we present efficient coding nodes strategies allowing the building of output flows as a combination of a subset of all the input flows. These strategies are evaluated in terms of maximal output throughput, maximum buffer size and maximal crossing-delays of the network node. In a second part, we show that a generalization of these results to a complete network can be obtained through a transfer matrix whose entries are expressed in terms of network calculus. Thanks to the formalism used to characterize the flows, the obtained results can be considered as guarantees in terms of the burstiness, buffers size or end-to-end delays.**

***Index Terms*— Network Coding, network calculus, delays, average throughput, buffer sizing.**

I. Introduction

Several works has demonstrated the potential of the network coding to improve the throughput or the reliability of multicast, broadcast or unicast flows in practical cases [1] [2] [3] [4] [5][6].

Compared with the theoretical hypothesis on the network, which assume a fluid traffic and a synchronization of the network nodes, practical applications must cope with several problems, such the variables delays and rates on the different paths.

In this work, we consider the integration of network coding in the large class of networks providing quality of service (QoS) guarantees. In these networks, the data flows verify constraints of burstiness and maximal throughput and in return, the network provides guarantees in terms of end-to-end delays, minimal throughput to the data flows. Examples of such networks are ATM networks or IntServ and DiffServ IP networks.

To quantify the different parameters, we used the network calculus framework [7]. This theory allows to better understand some fundamental properties of integrated services networks, scheduling and buffer or delay dimensioning.

The main issues addresses here are :

1) can we improve the levels of guarantee offered to the flows (maximum end-to-end delay, throughput) with network coding ?
2) can we improve the network parameters (link use, buffer size) with network coding ?

The hypothesis made in this work are the following:

The input flows, considered as a sequence of packets of same length, are non-synchronized and they can be temporarily idle. They have one or several sources and one or several receivers. Each flow verifies constraints of burstiness and throughput. The network is represented by the graph $G = (V, E)$ where V is a set of nodes and E is a set of directed edges. The edges have a given capacity. The set of nodes is split into three categories. First one is the source nodes which generates the flows. The second subset contains the *coding nodes* which are able to perform network coding operations following a given strategy (service policy). They are composed of network elements like buffers and/or shapers, each one guarantying a service level. The other nodes are the *receiver nodes* which receive and decode the combined flows.

We assume that a linear network code is determined *a priori* for this network. Consequently, each coding node knows how to combine its input flows to produce the output flows.

The objectives are the following. From

- the constraints on the input flows
- the guaranteed services of the network elements
- the strategy of the coding nodes

we express guarantees on :

- the delay for the receivers to receive the data
- the level of utilization of a link by a flow
- the buffer sizes

The first problem we addressed is to determine the coding node strategy. Since the input flows are independent, non-synchronized, possibly idle, and have different rate and burstiness, all the packets can not be obviously combined. This implies that some packets are combined following the network code and other ones are simply multiplied by the coefficient determined by the linear

A. Mahmino and J. Lacan are with ENSICA. A. Mahmino is also with the group Ingénierie Réseaux et Télécommunications/IRIT. J. Lacan is also with the group OLC of the LAAS/CNRS

network code and forwarded without combination with other packets (or equivalently, combined with the null packet).

The node strategy must be able to take into account the interests of both the flows and the network. Indeed, it must ensure a minimum ”level of combination” in the output flow in order to decrease the total amount of data transmitted on the output link. On the other side, it must not ”too much” constraint the flows by e.g. ”too much” delaying packets in order to combine them with packets of another flow because this operation will increase the total crossing-network delay observed by the receiver.

The proposed coding strategy is described through an node architecture based on leaky bucket shapers and synchronized buffers. This strategy ensures that the maximal rate of the output flows is the maximum of the maximal rates of the input flows. The extension of these results to a network with several nodes is also presented. It allows in particular to obtain maximum bounds on delays at the receivers side. This generalization is done through the use of a transfer matrix, like in [8], defined over the min-plus algebra of network calculus (similarly than for finite fields for the network coding).

This result could be very useful for practical bandwidth allocation in networks with guaranteed services. Indeed, all the obtained results are expressed in terms of strong (non probabilistic) guaranteed in terms of average rate or buffer sizing.

The next Section presents related work in the domains of network coding and guaranteed services with network calculus for aggregated flows. The proposed coding node strategy and the associated results are presented in Section III. The generalization to the network is proposed in Section IV. A discussion about decoding issues is presented in Section **??** and the last Section concludes.

II. Related Work

A. Network coding : practical approaches

To cope with the problems of non-synchronized packets or variable networks, several work proposed strategies for coding nodes policies. In different contexts, [9], [10] or [2] present solutions ensuring the encoding of all the packets in coding nodes. In [9], the problems of implementation are solved by considering that the network coding is done at application level and that the nodes have more capabilities than classical router or switches. In the two latter ones, the implementations are based on particular network nodes strategies to cope e.g. with the asynchronous data arrivals which involves buffering information at coding nodes in order to code them with other incoming information from the same batch. An alternative proposed for cyclic networks is to take a continuous coding approach [8] [11] where information from different time periods is combined.

To the best of our knowledge, the problem of extracting guaranteed service of networks implementing network coding was not addressed.

B. Network Calculus : A System Theory for Computer Networks

Network Calculus is a set of recent developments that provide deep insights into flow problems encountered in networking. This theory allows to better understand some fundamental properties of integrated services networks, window flow control, scheduling and buffer or delay dimensioning.

A detailed presentation of these concepts can be found in [7]. Other pioneering work on this subject are [12], [13], [14], [15], [16].

The following definitions and results are directly extracted from [7].

1) A data stream F transmitted on a link can be described by the *cumulative function* $R(t)$, such that for any $y > x$, $R(y-x)$ is the quantity of the data transmitted on this link in time interval $[x, y]$.
2) Let F be un data stream with cumulative function $R(t)$. Let α be a wide-sense increasing function. We say that α is an arrival curve of F (or equivalently R) if for any $0 \leq t_1 \leq t_2$, $R(t_2) - R(t_1) \leq \alpha(t_2 - t_1)$. A common class of arrival curves are the affine functions $\gamma_{r,b}(t) = rt + b$ for $t > 0$ and 0 otherwise.
3) The min-plus convolution of two functions X and Y is defined as $X(t) \otimes Y(t) = inf_{0 \leq s \leq t}(X(s) + Y(t-s))$. It can be shown that α is an arrival curve of R if and only if $R \leq R \otimes \alpha$.
4) A leaky bucket controller is a device that analyzes the data on a flow R(t) as follows. There is a pool (bucket) of fluid (data) of size b. The bucket is initially empty. The bucket has a hole and leaks at a rate of r units of fluid (data) per second when it is not empty. Data that would cause the bucket to overflow is declared non-conformant, otherwise the data is declared conformant. A leaky bucket controller with leak rate r and bucket size b forces a flow to be constrained by the arrival curve $\gamma_{r,b}$.
5) Let R^{out} be the output flow of a node with one input flow R. We say that the node offers a service curve $\beta(t)$ to R if for any $t > 0$, $R^{out}(t) \geq R(t) \otimes \beta(t)$.
6) Assume a flow $R(t)$, constrained by arrival curve $\alpha(t)$ traverses a system that offers a service curve

of β. The output flow R^{out} is constrained by the arrival curve $\alpha \oslash \beta$, where $\alpha(t) \oslash \beta(t) = sup_{i=1...n}\{R(t+v) - R(v)\}$.

7) The backlog, defined as $R(t) - R^{out}$ for all t, satisfies $R(t) - R^{out}(t) \leq sup_{s>0}\{\alpha(s) - \beta(s)\}$. The virtual delay $d(t)$, for all t, satisfies: $d(t) \leq h(\alpha, \beta)$, where $h(\alpha, \beta) = \sup_{s>0}\{\delta(s)\}$ where $\delta(s) = \inf\{\tau \geq 0 : \ \alpha(s) \leq \beta(s+\tau)\}$.
8) The Staircase Functions $v_{T,\tau}$ used for T-periodic stream of packets of same size L which suffer a variable delay τ is defined as :

$$v_{T,\tau}(t) = \begin{cases} \frac{t+\tau}{T} & if t > T \\ 0 & otherwise \end{cases}$$

where the interval $T > 0$ and the tolerance $0 \leq \tau \leq T$.

III. Network Coding Node Strategy

For sake of simplicity, we consider in this section a coding node with 2 input flows and one output flow R^{out}. The results can be easily extended to nodes with more input and output flows.

The two input flows and the output flow are respectively represented by their cumulative functions R_1, R_2 and R^{out}. The input flows are composed of packets of length L. They are supposed to be independent, non-synchronized and they can be temporarily idle. We consider that the links have an infinite capacity (this constraint will be discussed later).

We consider that the input flow R_i, for $i = 1, 2$, are constrained by the affine arrival curve α_i, where $\alpha_i(t) = \sigma_i + L * v_{L/\rho_i, -L/\rho_i}(t)$ for $t > 0$ and 0 otherwise. This corresponds to a stair function with backlog σ_i and average rate of ρ_i (see Figure 1). For sake of simplicity, we consider that the values σ_1 and $sigma_2$ are multiple of L. Let us define $\rho = max(\rho_1, \rho_2)$ and $T = L/\rho$.

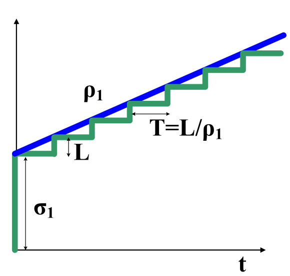

Fig. 1. arrival curve $\alpha_i(t) = \sigma_1 + L * v_{L/\rho_1, -L/\rho_1}(t)$

Our approach is network-oriented. We consider strategies focusing on minimizing the backlogs and the crossing-network delays. Such strategy leads to combine a subset of the data and then to forward the other part (multiplied by the scalar coefficient affected by the linear network code).

Let us defined R_1^{out} and R_1^{out} the cumulative functions of the subset of the data of R^{out} obtained from data of R_i (either combined with packets of other flows or simply multiplied by a coefficient).

We define the delay experienced by a data as the difference between the time when it arrives at the node and the time when it leaves it (combined or not with other data).

The backlog of a flow R_1 (resp. R_2) in the node at the time t is the amount of data "in transit" in the node.

The network calculus theory can use both discrete and continuous time model. We consider here the continuous model.

Theorem 1: There exists a service policy for a coding node ensuring that, for $i = 1, 2$:

1) The service curve β_i provided by the node to R_i is equal to $L * v_{T,-T}(t)$.
2) the maximum delay experienced by a data of R_i is $T(1 + \sigma_i/L)$.
3) the maximum backlog is equal to ρ_i.
4) R_i^{out} is constrained by $\alpha_i \oslash L * v_{T,0}$.
5) R^{out} is constrained by $L * v_{T,0}(t)$

Proof: To prove this theorem, we present an node architecte which verifies these assertions. This node performs a linear coding operation, i.e. for each flow, it multiplies the data by a scalar and add the data of the two flows.

Consider a coding node composed of two Leaky Bucket Shapers $LBS_{i_{i=1,2}}$ and two FIFO buffers $B_i, \ i = 1, 2$ as indicated in figure 2.

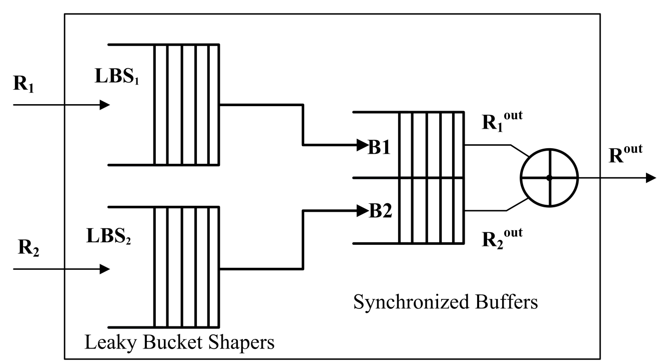

Fig. 2. coding node Architecture

The flow R_1 (resp. R_2) traverses the leaky bucket shaper LBS_1 (resp. LBS_2) and the buffer B_1 (resp. B_2) in sequence. LBS_1 (resp. LBS_2) has a buffer size σ_1 (resp. σ_2) and offers a service $L * v_{T,0}(t)$ to the flow. This operation consists in shaping the flow such that there is at least a time interval T between two packets. It can be shown the flow R_i (resp. R_2), which is α_1-smooth (resp. α_2-smooth) is conformant with LSB_1 (resp LBS_2).

We consider that the multiplication by the scalar is performed at the output of the leaky bucket and that this operation does not add additional delay.

The buffers B_1 and B_2 are *synchronized* and offer the same service curve of $L * v_{T,-T}(t)$ to the flows. In other words, after each time interval T, each buffer authorizes a packet to leave (if there is a packet) simultaneously. The output flows of the buffers, called R_1^{out} (resp. R_2^{out}), are combined (by an addition) to obtain $R^{out}(t)$. When there is only one packet available (the other buffers does not contain a packet), it is simply forwarded (i.e. it is combined with a null packet). Note that this operation does not modify or delay the flow.

Let us now determine the service curve offered by the node to the flows. The flow which leaves LBS_1 (resp. LSB_2) is equal to $[R_1(t) \otimes L * v_{T,0}(t)]$ (resp. $[R_2(t) \otimes L * v_{T,0}(t)]$. Then it is served by the buffer B_1 (resp. B_2) with a service curve $L * v_{T,-T}(t)]$. It follows that :

$$\begin{aligned} R_1^{out}(t) \quad & \leq (R_1(t) \otimes L * v_{T,0}(t) \otimes L * v_{T,-T}(t) \\ & \leq R_1(t) \otimes (L * v_{T,0}(t) \otimes L * v_{T,-T}(t) \\ & \leq R_1(t) \otimes L * v_{T,-T})(t) \end{aligned}$$

We can then deduce that $L * v_{T,-T}(t)$ is a service curve offered by the node to R_1. The same result holds for R_2.

From the service and the arrival curves of R_1 (resp. α_2 of R_2), we can obtain the upper bounds of the backlog and the delay (see Section II-B- point 7) indicated in the theorem.

The arrival curve of R_1^{out} (resp. R_2^{out}) can also be directly obtained from the arrival curve of R_1^{out} (resp. R_2^{out}) and the service curve offered by the node (see Section II-B- point 7).

Finally, the flow R^{out} is built by taking, at each time interval T, one packet of size L if there is one packet in the flows R_1^{out} or R_2^{out} and no packet if there is no packet in R_1^{out} and R_2^{out}. Let i be the integer such that $\rho = \rho_i$. The arrival curve of R_i^{out} is then $L * v_{T,-T})(t)$. The other arrival curve has a maximum slope of ρ_i (with stair shape). Then, the arrival curve of R^{out} is the maximum of the two arrival curves, i.e. $\alpha_1 \oslash L * v_{T,0} \vee \alpha_2 \oslash L * v_{T,0} = L * v_{T,0}(t)$.

This result can be retrieved by considering that R^{out} is by construction a flow of packets of size L, with a time interval which is a multiple of T between two packets. Then it is necessarily constrained by $L * v_{T,0}$. ■

Notes: The hypotheses taken on the capacity of the links was that all the links have an infinite capacity. From the proposed node architecture, we can reduce it without damage to $Max_{i=1,2}(\rho_i)$.

IV. Network-level Bounds

In the last Section, an analysis of network coding strategy at node level was proposed. In this section, we extend the analysis at the network-level.

Let us consider a delay-free communication network represented by an acyclic directed graph G = (V,E) with a vertex set $V = \{v_1, \ldots, v_m\}$ and an edge set $E = \{e_1, \ldots, e_p\}$. We allocate to each edge e_i a capacity C_{e_i}. We consider that s vertex among the m are source nodes and r are receiver nodes. We assume that the source nodes generate some flows R_i, $i = 1, \ldots, k$, respectively constrained by an arrival curve α_i. Each source node offers a given service curve to its flows towards its different output links.

A coding node with input links and output links combines the input flows to produce the output flows following a linear network code determined ***a priori***. We consider that its input flows $R_1, \ldots, R_r$ are respectively constrained by the arrival curves $\alpha_{1,in} \ldots, \alpha_{r,in}$. It offers to each of its input flows a service curve towards each of the output flows. Note that this service curve can be the constant function equal to 0. Let $\beta_{i,j},\ i = 1, \ldots, r$ and $j = 1, \ldots, s$ be the service curve offered by the coding node to the input flow R_i towards the j^{th} output flow $R^{j,out}$.

It $R^{j,out}$ is constrained by the arrival curve $\alpha_{j,out}$, then we assume that the coding node policy satisfies the following property :

$$\alpha_{j,out} = \alpha_{1,in} \oslash \beta_{1,j} \vee \ldots \vee \oslash \alpha_{rR,in} \oslash \beta_{r,j}$$

This property is verified by the coding strategy proposed in the last section. Note that any coding strategy verifying this property can use the results presented in this section.

We consider that the receivers produce n output flows $R_1^*, \ldots, R_n^*$ respectively constrained by the arrival curves $\alpha_1^*, \ldots, \alpha_n^*$. Each receiver offers a service curve to its input flows towards its output flows.

An example of such network is given in Figure 3.

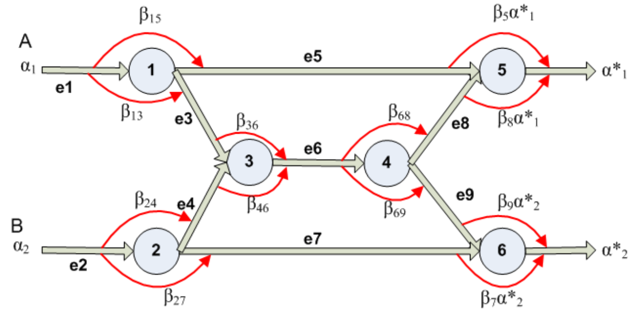

Fig. 3. network output flows

This section aims at determining the service offered by the network to the input flows towards the output flows. In others words, we aim at defining a transfer matrix M whose entries are service curves such that :

$$[\alpha_1^*, \ldots, \alpha_n^*] = [\alpha_1, \ldots, \alpha_k] \oslash M \tag{1}$$

The proposed construction follows step-by-step the construction of the transfer matrix for a linear network code presented in [8]. The main difference is that, in [8], the coding nodes perform linear combinations over a finite field such e.g.

$$Y = X_1 * \beta_1 + X_2 * \beta_2$$

where $X_1,\ X_2$ and Y are random processes representing the flows and β_1 and β_2 belong to a finite field.

In our context, the node perform operations on service and arrival curves, such e.g.

$$Y = X_1 \oslash \beta_1 \vee X_2 \oslash \beta_2$$

where $X_1,\ X_2$ and Y are arrival curves and β_1 and β_2 are service curves.

However, the operations on the curves must be manipulated carefully. Indeed, the construction of the transfer matrix is based on products of matrices over a finite field. Here, we have to define the operations that must be used for service curves and matrices of service curves.

We recall three rules of network calculus :

1) $(\alpha \oslash \beta_1) \oslash \beta_2 = \alpha \oslash (\beta_1 \otimes \beta_2)$
2) $(\alpha \oslash \beta_1) \vee (\alpha \oslash \beta_2) = \alpha \oslash (\beta_1 \wedge \beta_2)$
3) $(\alpha_1 \vee \alpha_2) \oslash \beta = (\alpha_1 \oslash \beta) \vee (\alpha_2 \oslash \beta)$

For 1) and 3), see [7]. The equation 2) can be directly obtained for the definition of the $\oslash$ operation.

The two first points concern some operations including only one arrival curve. These operations are then typically the one that must be used to compute the multiplication of matrices of service curves. The last one includes two arrival curves. Thus, it will be used in the vector-matrix multiplication.

Recall that p, k and n represent respectively the number of edges, input flows, and output flows. We will say that an edge i (or a flow) is connected to an edge j (or a flow) if the head of the edge i is the tail of the edge j. Let us now define the following matrices:

Let $A = (a_{i,j})_{i=1,\ldots,k;j=1,\ldots,p}$ be defined as follows. If the flow R_i is connected to the flow on the edge j, then $a_{i,j} = \beta_{i,j}$ is the service curve offered to R_i towards the flow on the edge j; else $a_{i,j} = 0$.

For the network of the Figure 3, we have :

$$A = \begin{bmatrix} 0 & 0 & \beta_{13} & 0 & \beta_{15} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_{24} & 0 & 0 & \beta_{27} & 0 & 0 \end{bmatrix}$$

Let $F = (f_{i,j})_{i=1,\ldots,p;j=1,\ldots,p}$ be the adjacency matrix defined as follows. If the edge i is connected to the edge j, then $f_{i,j} = \beta_{i,j}$ is the service curve offered to the flow on the edge i towards the flow on the edge j; else $f_{i,j} = 0$.

For the network of the Figure 3, we have :

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_{36} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_{46} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_{68} & \beta_{69} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Intuitively, the multiplication of a vector of inputs by M indicates the state of the inputs after one hop. Similarly, the multiplication of a vector of inputs by M^i indicates the state of the inputs after i hops. Since the graph G is acyclic, the adjacency matrix can be represented as an strict upper-triangular matrix. It is then nilpotent and we can compute the matrix $I + F + F^2 + F^3 + \ldots$ which indicates the states of the input flows in the network.

For the network of the Figure 3, $I + F + F^2 + F^3 + \ldots$ is equal to :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \beta_{36} & 0 & \beta_{36} \otimes \beta_{68} & \beta_{36} \otimes \beta_{69} \\ 0 & 0 & 0 & 1 & 0 & \beta_{46} & 0 & \beta_{46} \otimes \beta_{68} & \beta_{46} \otimes \beta_{69} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \beta_{68} & \beta_{69} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Let $B = (b_{i,j})_{i=1,\ldots,p;j=1,\ldots,n}$ be defined as follows. If the edge i is connected to the output flow R_j^*, then $b_{i,j} = \beta_{i,j}$ is the service curve offered to the flow on the edge i towards the output flow R_j^*; else $b_{i,j} = 0$. For the network of the Figure 3, we have :

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & \beta_{5\alpha_1^*} & 0 & 0 & \beta_{8\alpha_1^*} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta_{7\alpha_2^*} & 0 & \beta_{9\alpha_2^*} \end{bmatrix}$$

Following the similar construction of the transfer matrix presented in [8], we can obtain the transfer matrix :

$$M = A \times (I + F + F^2 + \ldots) \times B$$

For the network of the Figure 3, M is equal to :

$$\begin{bmatrix} \beta_{15} \otimes \beta_{5\alpha_1^*} \wedge \beta_{13} \otimes \beta_{36} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*} & \beta_{13} \otimes \beta_{36} \otimes \beta_{69} \otimes \beta_{9\alpha_2^*} \\ \beta_{24} \otimes \beta_{46} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*} & \beta_{27} \otimes \beta_{7\alpha_2^*} \wedge \beta_{24} \otimes \beta_{46} \otimes \beta_{69} \otimes \beta_{9\alpha_2^*} \end{bmatrix}$$

We can then obtain the values of α_1^* and α_2^* by applying Equation 1.

This simple example demonstrated the interest of the network coding in this context. Indeed, compared to a traditional approach (with two multicast sources and two receivers) multiplexing the flows, the network coding allows to improve the guaranteed throughput if we consider that the edge 6 has a finite capacity. In this case, the deterministic bound on the end-to-end delays is also improved.

V. Decoding issues

The approach used in the last sections was entirely focused on optimizing the network parameters such the delay, the backlog and the throughput. Even if we assume that a network code was designed and that the coding nodes always perform the same linear operations on the input flows, the variable throughput, the different lengths of the multiple paths, the jitter and the losses occurring in real networks could lead to a near-random code.

The analysis of the decoding performance of proposed approach is out the scope of this paper. However, many recent works have shown that randomized coding could provide a interesting statistical level of reliability [11]. Moreover, in networks with guarantees of service, the "random" parameters are minimized. It follows that a detailed analysis of the network parameters should lead to the construction of codes ensuring a high level of reliability.

VI. Conclusion

This paper has provided a solution to introduce the network coding in networks with service guarantees. A coding strategy was proposed to obtain minimal upper bounds on the rate of the output flow without excessive buffering and delays. The second part of this paper has presented a method to obtain global service curves of the network. This method is based on a transfer matrix whose the entries are service curves.

This work can be extended by several ways. First, the produced code is partially-random and the decoding performance will be precisely evaluated. Another point concerns the strategy of the coding nodes and more particularly the analyse of the trade-offs between the network parameters and the end-users parameters.

References

[1] Y. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE J. Selected Areas in Comm.*, vol. 22, no. 1, January 2004.

[2] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *41st Allerton Conference on Communication, Control, and Computing*, October 2003.

[3] Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," *Microsoft Research Technical Report MSR-TR-2004-78*, August 2004.

[4] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," *42st Allerton Conf. Communication, Control and Computing*, October 2004. [Online]. Available: citeseer.ifi.unizh.ch/705799.html

[5] M. Effros, M. Mdard, T. Ho, S. Ray, D. Karger, R. Koetter, and B. Hassibi, "Linear network codes: A unified framework for source, channel, and network coding," *DIMACS workshop on network information theory*, 2003.

[6] W. H. S. Katti, D. Katabi and R. Hariharan, "The importance of being opportunistic: Practical network coding for wireless environments," *In Proc. 43rd Allerton Conference on Communication, Control, and Computing*, September 2005.

[7] J.-Y. L. Boudec and P. Thiram, *Network Calculus A Theory of Deterministic Queuing Systems for the Internet*, ser. Series: Lecture Notes in Computer Science, Vol. 2050. Springer Verlag, 2001.

[8] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, October 2003.

[9] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[10] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, 2005.

[11] J. S. M. E. T. Ho, M. Médard and D. R. Karger, "on randomized network coding," *in Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing*, october 2003.

[12] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proceedings of the IEEE*, vol. 83, no. 10, pp. 1374–1296, october 1995.

[13] R. L. Cruz, "A calculus for network delay, part i : Network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–131, January 1991.

[14] ——, "A calculus for network delay, part ii : Network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 132 – 141, January 1991.

[15] C. S. Chang, *Performance Guarantees in Communication Networks*, ser. New York. Springer Verlag, 2000.

[16] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan, "Performance bounds for flow control protocols," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 310–323, June 1999.

[17] R. K. M. M. T. Ho, B. Leong, "Distributed asynchronous algorithms for multicast network coding," *NetCod,First Workshop on Network Coding, Theory, and Applications, WiOpt*, April 2005.