



HAL
open science

Calculus of service guarantees for network coding

Ali Mahmino, Jérôme Lacan, Christian Fraboul

► **To cite this version:**

Ali Mahmino, Jérôme Lacan, Christian Fraboul. Calculus of service guarantees for network coding. International Symposium on Information Theory and its Applications (ISITA 2006), Nov 2006, Seoul, South Korea. pp.1-6. hal-04034856

HAL Id: hal-04034856

<https://hal.science/hal-04034856>

Submitted on 17 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Calculus of Service Guarantees for Network Coding

Ali Mahmino[†], Jérôme Lacan[‡] and Christian Fraboul^{††}

[†] ENSICA and IRIT-INPT-ENSEEIH
1, place E. Blouin, 31056, Toulouse Cedex 05, France
E-mail: ali.mahmino@ensica.fr

[‡] ENSICA and LAAS/CNRS
1, place E. Blouin, 31056, Toulouse Cedex 05, France
E-mail: jerome.lacan@ensica.fr

^{††} IRIT/INPT/ENSEEIH
2, rue Camichel - BP 7122 - 31071 Toulouse Cedex, France
E-mail: christian.fraboul@enseeiht.fr

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.

1. Introduction

Several works have 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].

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 variable 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 through-

put to the data flows. Examples of such networks are ATM networks or IntServ and DiffServ IP networks.

In these networks, a useful framework to the different constraints, bounds and service guarantees is the network calculus framework [5]. This theory can help to analyze flows, networks and its elements by algebraic methods. It allows to calculate the maximum and the minimum bounds of delay and backlog.

The main issues addressed 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 a 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. The first one is the *source nodes* generating the flows. The second subset is composed of 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 address 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 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 should 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 should avoid to strongly constrain the flows by e.g. delaying packets in order to combine them with packets of another flow. Indeed, this operation increases 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 at the network level is also presented. It allows in particular to obtain maximum bounds on delays at the receiver side. This generalization is done through the use of a transfer matrix defined over the min-plus algebra of network calculus.

The next Section presents related work in the domains of network coding and network calculus for aggregated flows. The proposed coding node strategy and the associated results are presented in Section 3. The generalization to the network is proposed in Section 4. A discussion about decoding issues is presented in Section 5 and the last Section concludes.

2. Related Work

2.1. Network coding

To implement network coding in practical networks, several work proposed strategies for coding nodes policies. In different contexts, [6], [2] present solutions ensuring the encoding of all the packets in coding nodes. In [6], 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 [7] [8] 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.

2.2. Network Calculus

Network Calculus is a framework providing deterministic bounds on end-to-end delays and backlogs. A detailed presentation of this theory can be found in [5]. Other pioneering work on this subject are e.g. [9, 10].

The following definitions and results are extracted from [5].

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 a data stream with cumulative function $R(t)$. We say that an increasing function 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. The curve $\gamma_{r,b}(t) = rt + b$ represents the arrival curve of the leaky bucket controller with leak rate r and bucket size b .
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. 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 $t > 0$, $R^{out}(t) \geq R(t) \otimes \beta(t)$.
5. Assume a flow $R(t)$, constrained by an 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 $\alpha \circ \beta$, where $(\alpha \circ \beta)(t) = \sup_{v \geq 0} \{\alpha(t + v) - \beta(v)\}$.

6. The backlog, defined as $R(t) - R^{out}(t)$ 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 $\delta(s) = \inf\{\tau \geq 0 : \alpha(s) \leq \beta(s + \tau)\}$.
7. 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 :

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

where $T > 0$ and $0 \leq \tau \leq T$.

3. Network Coding Node Strategy

The approach we develop in this paper focuses 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).

3.1. Definitions and assumptions

For sake of simplicity, we consider in this section a coding node with 2 input flows and one output flow R^{out} . Note that the results presented in this Section 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 flows R_i , for $i = 1, 2$, are constrained by the 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-a). For sake of simplicity, we consider that the values σ_1 and σ_2 are multiple of L . Let us define $\rho = \max(\rho_1, \rho_2)$ and $T = L/\rho$.

Let us defined R_i^{out} as 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.

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.

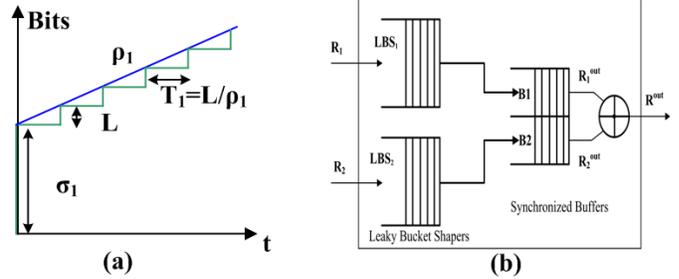


Figure 1: (a) staircase arrival curve. (b) coding node Architecture.

3.2. Coding node architecture

The architecture we propose is represented in 1(b). It is composed of two greedy Leaky Bucket Shapers, LBS_1 and LBS_2 , and two FIFO buffers B_1 and B_2 . 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 that the flow R_1 (resp. R_2), which is α_1 -smooth (resp. α_2 -smooth) is conformant with LSB_1 (resp. LSB_2).

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 , the buffers authorize a packet to leave (if there is one) simultaneously. The two output packets of the buffers are then multiplied by a scalar and added. We consider that these operations do not add additional delays. If one of the buffer does not contain any packet, the packet leaving the other buffer is multiplied by the scalar and forwarded (i.e. it is combined with a null packet).

The output flows of the buffers are represented by their cumulative functions R_1^{out} and R_2^{out} . Actually, these functions also represent the amount of data of R^{out} obtained respectively from data of R_1 and R_2 .

Under these assumptions, we can obtain the following properties for the node :

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 σ_i .*

4. R_i^{out} is constrained by $\alpha_i \otimes L * v_{T,-T}$.
5. R^{out} is constrained by $(\alpha_1 \otimes L * v_{T,-T}) \vee (\alpha_2 \otimes L * v_{T,-T})$.

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

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

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

The backlog and the delay for R_1 and R_2 can be then deduced. The maximum backlog of the flow R_i is given according to (point 6 - section 2.2) by:

$$\begin{aligned} b_i(t) &\leq \sup_{s \geq 0} \{\alpha_i(s) - L * v_{T,-T}(s)\} \\ &\leq \sup_{s \geq 0} \{\sigma_i + L * v_{T_i,-T_i}(s) - L * v_{T,-T}(s)\} \\ &\leq \sigma_i. \end{aligned}$$

Following (point 6 -section 2.2), the maximum delay d_i experienced by a packet from a flow R_i is upper-bounded by :

$$\begin{aligned} d_i(t) &\leq \sup_{t \geq 0} \{\inf\{\tau \geq 0 : \alpha_i(t) \leq L * v_{T,0}(t + \tau)\}\} \\ &\leq \sup_{t \geq 0} \{\inf\{\tau \geq 0 : \sigma_i + L * v_{T_i,-T_i}(t) \leq L * v_{T,0}(t + \tau)\}\} \\ &\leq \frac{\sigma_i + L}{\rho} = T(1 + \sigma_i/L) \end{aligned}$$

(see Section 2.2- point 7) indicated in the theorem.

This flow is then served by the buffer with service curve $L * v_{T,-T}$. Then an arrival curve of R_i^{out} is equal to $\alpha_i \otimes \{L * v_{T,0} \otimes L * v_{T,-T}\} = \alpha_i \otimes L * v_{T,-T}$.

Since 1) the buffers B_1 and B_2 are synchronized, 2) the packets leaving the two buffers are added (XOR-ed) and 3) an arrival curve of R_i^{out} is equal to $\alpha_i \otimes L * v_{T,-T}$, then the flow R^{out} is necessarily constrained by $(\alpha_1 \otimes L * v_{T,-T}) \vee (\alpha_2 \otimes L * v_{T,-T})$. ■

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 $Max_{i=1,2}(\rho_i)$.

4. Network-level Bounds

The aim of this section is to analyse and to determine the properties of a network composed of nodes built as described in the previous section.

Let us consider a delay-free communication network represented by an acyclic directed graph $G = (V, E)$ with a vertex set $V = \{v_1, \dots, v_m\}$ and an edge set $E = \{e_1, \dots, e_p\}$. 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 , $i = 1, \dots, k$, 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, \dots, R_r are respectively constrained by the arrival curves $\alpha_{1,in}, \dots, \alpha_{r,in}$. It offers to each of its input flows a service curve towards each of the output flows. Let $\beta_{i,j}$, $i = 1, \dots, r$ and $j = 1, \dots, s$ 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 :

$$\alpha_{j,out} = \alpha_{1,in} \otimes \beta_{1,j} \vee \dots \vee \alpha_{r,in} \otimes \beta_{r,j}$$

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

We consider that the receivers decode the flows from the input flows. The delays due to the decoding process are not studied since the main goal of this paper is to evaluate the impact of the network coding on the network.

An example of such network is given in Figure 2.

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

$$[\alpha_1^*, \dots, \alpha_n^*] = [\alpha_1, \dots, \alpha_k] \otimes M \quad (1)$$

where the α_n^* are the input flows of the receivers.

The proposed construction follows the construction of the transfer matrix for a linear network code presented in [7] excepted for the decoding part. The main difference is that, in [7], 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

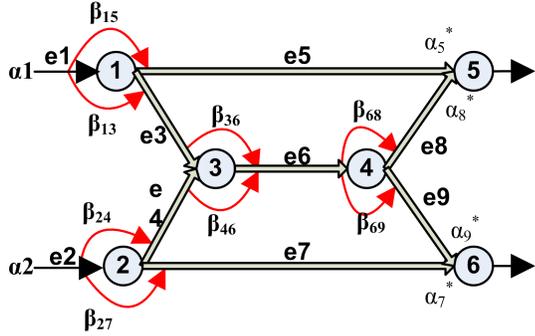


Figure 2: Simple delay-free, acyclic, directed network with two sources A and B generating the input flows R_1, R_2 with respective arrival curves α_1 and α_2 . The two sinks (or receiver nodes) are the nodes 5) and 6). The nodes 1), 2), 3) and 4) are the *Coding Nodes*. Each node offers a given service curve to their input flows toward their output flows.

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

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

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

For 1) and 3), see [5]. The equation 2) can be directly obtained for the definition of the \otimes 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 $A = (a_{i,j})_{i=1,\dots,p;j=1,\dots,p}$ 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 2, 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 $F = (f_{i,j})_{i=1,\dots,p;j=1,\dots,p}$ 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 2, 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 a strict upper-triangular matrix. It is then nilpotent and we can compute the matrix $I + F + F^2 + F^3 + \dots$ which indicates the states of the input flows in the network.

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

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

The transfer matrix can be obtained :

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

For the network of the Figure 2, M is equal to :

$$\begin{bmatrix} 0 & 0 & \beta_{13} & 0 & \beta_{15} & \beta_{13} \otimes \beta_{36} \\ 0 & 0 & 0 & \beta_{24} & 0 & \beta_{24} \otimes \beta_{46} \\ 0 & \beta_{13} \otimes \beta_{36} \otimes \beta_{68} & \beta_{13} \otimes \beta_{36} \otimes \beta_{69} \\ \beta_{27} & \beta_{24} \otimes \beta_{46} \otimes \beta_{68} & \beta_{24} \otimes \beta_{46} \otimes \beta_{69} \end{bmatrix}$$

Now we can find each flow from the receivers input flows by calculation of the max value of all the deconvolution between the network input flows and the own column of this flow in the preceding matrix.

$$[\alpha_5^* \quad \alpha_8^* \quad \alpha_9^* \quad \alpha_7^*] = [\alpha_1 \quad \alpha_2] \otimes$$

$$\begin{bmatrix} \beta_{15} & \beta_{13} \otimes \beta_{36} \otimes \beta_{68} & \beta_{24} \otimes \beta_{46} \otimes \beta_{68} & 0 \\ 0 & \beta_{13} \otimes \beta_{36} \otimes \beta_{69} & \beta_{24} \otimes \beta_{46} \otimes \beta_{69} & \beta_{27} \end{bmatrix}$$

For example:

$$\alpha_8^* = \{\alpha_1 \otimes (\beta_{13} \otimes \beta_{36} \otimes \beta_{68})\} \vee \{\alpha_2 \otimes (\beta_{24} \otimes \beta_{46} \otimes \beta_{68})\}$$

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.

5. 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 [8]. 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.

6. 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, Jan. 2004.
- [2] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *41st Allerton Conference*, Oct. 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 Tech. Rep. MSR-TR-2004-78*, Aug. 2004.
- [4] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," *42st Allerton Conf.*, Oct. 2004.
- [5] 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.
- [6] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
- [7] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.
- [8] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "on randomized network coding," *41st Annual Allerton Conference*, Oct. 2003.
- [9] R. L. Cruz, "A calculus for network delay, part i : Network elements in isolation," *IEEE Trans. on Information Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.
- [10] —, "A calculus for network delay, part ii : Network analysis," *IEEE Trans. on Information Theory*, vol. 37, no. 1, pp. 132 – 141, January 1991.