On Delay Fairness for Multiple Network Coding Transmissions

Golnaz Karbaschi, Aline Carneiro Viana, Steven Martin, Khaldoun Al Agha

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On Delay Fairness for Multiple Network Coding Transmissions

Golnaz Karbaschi — Aline Carneiro Viana — Steven Martin — Khaldoun Al Agha

N° 6972

June 2009

Thème COM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6972--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

centre de recherche
SACLAY – ÎLE-DE-FRANCE

On Delay Fairness for Multiple Network Coding Transmissions

Golnaz Karbaschi , Aline Carneiro Viana , Steven Martin* , Khaldoun Al Agha†

Thème COM — Systèmes communicants
Équipes-Projets Asap

Rapport de recherche n° 6972 — June 2009 — 23 pages

Abstract:
This paper studies the unfairness issues of network coding in multi hop wireless networks. Most of the work on network coding focuses on the obtained throughput gain. They show that mixing lineally the packets at the intermediate nodes is capacity-achieving. However, network coding schemes designed *only* to maximize the throughput could be *unfairly* biased. The reason is that by mixing different flows, packets destined to one destination in order to be decoded need to wait for the reception of the whole mixed set of encoded packets that may be totally independent in terms of final destination. This may lead to highly unfair delay for small block data. To mitigate this unfairness, relay nodes may mix only packets going to the same destination. We call this strategy *FairMix*. Although FairMix may limit the maximum attainable throughput, it aims to make distinct for decoding delay of each destination corresponding to the size of the data block. In order to investigate this trade off, we compare the FairMix performance with a naive network coding which mixes packets destined to different destinations. The simulation under lossy wireless links, limited memory and bandwidth resources, and different block sizes shows that FairMix is effective in improving fairness among destinations in comparison to naive network coding.

Key-words: Data delivery, wireless networks, network coding, decoding delay

* LRI/University Paris XI.
† LRI/University Paris XI

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

L'iniquité en termes de délai dans les transmissions multiples utilisant le codage réseau

Résumé :
Ce travail porte sur l'évaluation des questions liées à l'iniquité causée par le codage dans les réseaux sans fils a multi saut. La plupart des travaux sur le codage réseau se concentre sur le gain de débit. Ils montrent que le mélange linéaire de paquets dans les nIJuds intermédiaires du réseau améliore la capacité du réseau. Toutefois, les schémas de codage réseau uniquement conçues pour maximiser le débit sont injustes en termes de délai de décodage. La raison en est que par le mélange de différents flux, les paquets destinés à une destination doivent attendre la réception de l'ensemble mixte de paquets codés afin de pouvoir être décodés, ce qui peut être totalement indépendant en matière de destination finale. Ca peut injustement retarder de décodage de petits blocs de données. Pour pallier cette injustice, les nIJuds intermédiaires doivent mixer seulement les paquets allant vers la même destination. Nous appuyant sur cette étude, nous proposons la stratégie FairMix. Bien que FairMix puisse limiter le débit maximal, il vise à différencier le décodage de chaque destination. Nous comparons la performance de FairMix avec le codage réseau naïf, c'est- à-dire, le codage mêlant les paquets destinés à des destinations différentes. En considérant des liens non fiables, des ressources limités en termes de mémoire et de bande passante, et des différentes tailles de blocs, nous montrons que FairMix est efficace pour améliorer l'équité en termes de délai de décodage entre les destinations par rapport au codage réseau naïf.

Mots-clés : Livraison de données, réseaux sans fils, codage réseau, délai de décodage

1 Introduction

Theoretical and empirical studies suggest that significant gains can be obtained by using network coding in multi-hop wireless networks. In this scheme, intermediate nodes instead of simply forwarding the packets, send out a linear combination of all the packets that they have received so far. We explain the intuition of network coding through a simple example shown in Figure 2. In the figure, node A intends to send the message m_1 to node C and node C wants to send the message m_2 to node A. Due to transmission range limitation both these transmissions are done via node B. By traditional packet forwarding, four transmissions are required to perform both message transferring. In comparison, using a simple network coding, the two messages can be transferred by only three transmissions: node A and C send out their messages to node B. Node B transmits for A and C a new encoded packet by mixing linearly m_1 and m_2. Since A and C have a copy of their packets they can simply obtain m_2 and m_1 respectively through the encoded packet.

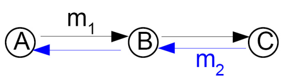


Figure 1: A simple illustration of network coding.

It is well known that network coding allows a multicast to reach the *max-flow min-cut* capacity in a lossless wireline network [1]. Nowadays there has been a recent interest in employing network coding in multi hop wireless networks since it allows to exploit natural advantages offered by shared wireless medium [3]. Indeed, since wireless links cause packets to be spread about in a probabilistic manner, there is no reason to restrict information to a path such as wireline networks. Rather, each node that is potentially a relay can encode the packets it receives and sends them out. In such an approach, the concept of routing can be broken and the challenge lies on how to mix the incoming packets. In this way, network coding can be classified into two classes: inter-session coding where coding is allowed among packets belonging to different flows and intra-session coding in which coding is limited to packets belonging to the same flows. The original works on network coding show how intra-session coding improves the throughput of both unicast and multicast session in a lossy wireless networks. However, it is also shown that in the case of existing multiple flows, intra-session coding is not necessarily optimal [23] and in general inter-session coding across the flows is needed to achieve optimal throughput. Most of the work to date focus on throughput gain obtained by network coding on multicast sessions through inter- or intra-session coding [14, 16, 18, 20, 22]. Yet, there is not much work addressing the possible trade-off in using either of these strategies. This paper represents our first efforts in this direction. We argue that the benefit of higher gained throughput in inter-session coding happens at the cost of unfairness among users which are waiting for different data flows.

In network coding, a whole block of K packets mixed through coding at the intermediate nodes in the network, requires the reception of K independent encoded packets at each destination in order to be decoded correctly. Therefore, each packet in the data block must wait the reception of the whole block before it can be decoded, even if not all the packets in the block belong to the same destination. Hence, no specific packet can be set apart from other packets. As a result, a destination that is waiting for a single urgent packet, should wait to receive large enough encoded packets to recover

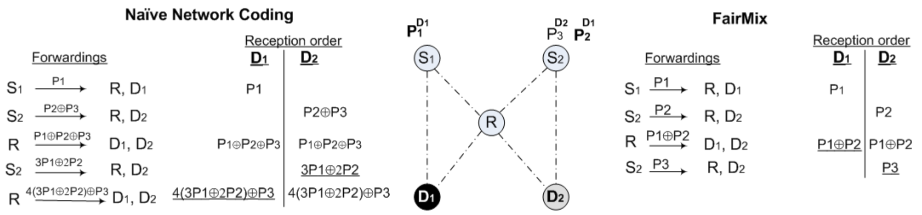


Figure 2: A simple example of how FairMix improves the fairness at the network resource usability and decreases decoding delay at destinations. Separating the coding based on the destination of the packets (i.e. D_1 and D_2) leads to less required number of packets for each destination to decode its packets.

all packets that are coded together. This may lead to a large *unfair* decoding delay for small blocks in the presence of various block sizes. Looking from a service provision's perspective, it makes nodes demanding for few resources (e.g. few bandwidth – for destinations expecting few packets) being penalized by high consuming nodes. Considering multi-hop wireless applications which require transmitting heterogeneous files (e.g. video or music files) to multiple destinations, delay performance may be critical to satisfaction of the users. In this case, it is essential and extremely challenging to deal with unfairness issues.

To mitigate the unfairness in decoding delay of naive inter-session network coding, one may encode together only packets going to the same destination. This strategy that is called *FairMix* in this paper, can be interpreted as an intra-session network coding where each session is identified by its end-point destination. Lets describe FairMix through a simple example. Consider the scenarios shown in Fig. 2 where sources S_1 and S_2 send blocks of different sizes for the destinations D_1 and D_2: block size 2 is composed of packet of P_1 and P_2 destined to D_1 while block size 1 is composed of P_3 destined to D_2. The nodes' forwarding order is the same for both naive network coding and FairMix. In the naive network coding (left side in Fig. 2), no distinction between destinations or packets is performed at the nodes. Therefore, at each node all or a number of packets going to *any* destination and from *any* source are linearly mixed together and sent out. As shown in this example, this requires the reception of at least 3 encoded packets before each destination can be able to decode its own packets. Now consider the FairMix that mixes the packets from *any* source to *each* distinct destination separately (right side of Fig. 2). In the example, with the reception of 2 encoded packets, D_1 is able to decode its block, while D_2 decodes its block at the third reception. Note that D_2 could be able to decode its block at the first reception if S_2 had scheduled P_3 to be sent at first order.

It is easy to see that FairMix inherits characteristics of intra-session network coding such as reducing transmission count and obtaining higher but not necessarily optimal throughput compared to simple packet forwarding. FairMix aims to, besides of enjoying from network coding advantages through mixing all the packets going to the same destination, make a distinct for decoding delay of each destination corresponding to the size of the data blocks. We define the *decoding delay* as the difference between the time instant of sending the first packet of a block and the time instant of receiving the last packet of the block at each specific destination. This shows the completion time for transmitting one block to its destination.

We expect that in FairMix, a destination waiting for large blocks does not cause a large decoding delay for a destination waiting for small blocks. To investigate this issue, we compare naive inter-session network coding and FairMix by simulations. The goal is to better understand their performance in terms of decoding delay and fairness. We study the performance of FairMix under: (1) multi-hop and lossy wireless networks, (2) different block sizes, (3) varying number of common sources per destination. We show the benefit of FairMix in terms of fairness, specially in the case that data block sizes destined to different destinations are not the same. In addition, FairMix provides considerable delay gains compared to naive network coding for destinations waiting for small block sizes. This last point is particularly interesting in providing different quality of services with differentiated priorities since servicing the high priority traffic (such as urgent messages) separately from ordinary traffics can be easily envisioned.

The remainder of this paper is structured as follows. In Section 3, we discuss the system and network coding model. Section resumes the related work. Sections 4 describes two coding schemes which mix the packets differently. Section 5 presents our evaluation study and Section 6 concludes this work and discuss future works.

2 Related work

Ashswede *et al.* [1] using information-theoretic reasoning showed that with network coding a source node can multicast its information at the maximum rate which was attainable in a unicast communication. [16] shows that linear coding in which the coding coefficients are chosen randomly from a finite size field (such as Galois Field), is sufficient to achieve this maximum rate. The random linear network coding provides a fully distributed methodology for performing network coding, whereby each node in the network selects independently and randomly a set of coefficients and uses them to form linear combinations.

After these seminal papers, several works have discussed and proved the gain obtained by network coding in multicast, unicast or broadcast session in wired or wireless networks through inter or intra-session coding [14, 1, 16, 18, 20, 22]. [14] presents an algebraic framework for network coding and [13] addresses the practical issues facing integration of network coding in the current network stack . Less work address the delay performance of network coding. A. Eryilmaz et al. has modeled de decoding delay of network coding in a single hop transmission [6]. To the best of out knowledge, there is not much work which deal with trade offs and negative impact of network coding that one may get through different strategies for packet mixing. Our paper differs from existing works in the sense that it argues on fairness issues of network coding in multi hop wireless network. While the question of fair perceived performance has been a well investigated research field in wireless networks [2, 10], its study in the wireless network coding area is still new.

3 Network coding model

We consider a wireless multi hop network consisting of N nodes, equipped with omni-directional antennas and randomly scattered on a geographical area. Each node is able to communicate wirelessly with a subset of nodes (*neighbors*) that are in its transmission range, R_t.

We assume multiple multicast sessions such that each source node has one or several files for one or several destination nodes. Note that, if some flows are multicast, all the destinations that have requested the same unique file are assumed to be grouped together which is called a *community*. Thus, in this paper the term of *destination* refers to both community and single destinations. Each file f_i demanded by a destination d_i is then divided into blocks with K_i *native* packets. We assume a distributed system in which in general the data packets of each block of file f_i may be dispersed among different sources. In this case, the total number of packets at source s_i, i.e. K_{s_i}, contains some number of packets of data blocks destined to different destinations: $K_{s_i} = \sum_{j=1}^{|d|} K_{s_i}^{d_j}$, where $|d|$ is the total number of destinations in the network. Note that in the case that a destination d_j has not demanded anything from source s_i, its $K_{s_i}^{d_j}$ will be zero. The total number of packets in the system is then calculated as $K = \sum_{i=1}^{|s|} \sum_{j=1}^{|d|} K_{s_i}^{d_j}$, where $|s|$ is the total number of sources.

We consider a random linear network coding in which nodes act as relays and forward linear random combinations of packets they have heard from the medium or natively possess [16]. Suppose that node n_i has in its buffer a collection of packets denoted as P_1,...,P_M and receives packets of Q_1,...,Q_N from its neighbors. The encoded packet generated at time t at node n_i (denoted as $X_{n_i}(t)$) is computed as:

$$X_{n_i}(t) = \sum_{i=1}^{i=M} g_i(t)P_i + \sum_{j=1}^{j=N} h_j(t)Q_j \quad (1)$$

where $g_i(t)$ and $h_i(t)$ are coding coefficients that are generated randomly at time t and in Galois field F(q). Note that since all operations are modulo operations over a finite field, mixing packets through linear network coding does not increase the packet size. Moreover, [19,6] show that by choosing randomly the coefficients from a large enough field, probability of generating *dependent* combinations of packets is ignorable.

Each encoded packet contains both the coefficients and the encoded information. Therefore, a relay node is able to regenerate an encoded packet following Eq. 1. We assume that each node as soon as having a packet to send, is a potential transmitter which competes to take the wireless medium and when it has an opportunity to send, it broadcasts an encoded packet to its in-range neighbors. Moreover, to reduce the overhead the relay nodes forward only the $innovative$ packets (i.e. linearly independent packets to the existing packets at a node) [21, 6, 13, 3].

4 Mitigating unfairness in Network Coding

4.1 FairMix Overview

One solution to alleviate the unfairness in decoding delay of inter-session network coding is limiting the encoding only on packets which go to the same destination. This mixing strategy that in this paper is called *FairMix*, can be interpreted as an intra-session network coding where each session is identified by its end-point destination. The difference is that generally in intra-session coding, packets are limited to be encoded separately from other sessions' packets while in FairMix the rational is that each destination does not need to wait for the packets of other destinations to be able to decode its own data blocks content. As a result, in this coding scheme packets are encoded separately from other destinations' packets.

Therefore, FairMix which separately mixes the packets of *any* source to *each* destination can be considered as a *fair* intra-session network coding and inherits characteristics of intra-session such as reducing transmission count and obtaining higher but not necessarily optimal throughput compared to simple packet forwarding. Building this coding scheme allows us to easily compare the fairness issues of network coding in inter-session case which for obtaining the highest capacity, all the flows are mixed. This is in contrast to FairMix that limits the coding in the favor of fairness.

Since FairMix separates data blocks based on their destinations, each node N_i maintains *virtual queues* which keeps track of each destination; Node N_i maintains a virtual queue per destination d_j, denoted as $\mathcal{V}_{d_j}$, which contains packets at N_i which are destined to d_j. Note that information of destination at each encoded packet can be retrieved easily from the packet header. Once N_i has a transmission opportunity it chooses a queue by a scheduling policy and generates an encoded packet across all the packets in that queue[1]. As a result, throughout the network the linear equations of packets of each destination are treated independently from other destinations'. Hence, in FairMix, a destination d_i can decode a whole block of $K_{d_i} = \sum_{j=1}^{|s|} K_{s_j}^{d_i}$ packets upon receiving K_{d_i} (instead of K) independent encoded packets from any relay nodes.

At the following, we address important design issues to efficiently implement FairMix.

4.2 Packet specification

Two additional information which should be inserted into the encoded packets are: (1) the destination id, denoted as d_i, for which the packet is oriented to and (2) the data block index I, which identifies the index of the block being transmitted by the source nodes. Therefore, besides the coding coefficients and the coded packet, i.e. $(g_i^j(t), X^j(t))$, each transmitted packet in FairMix also contains d_i and I.

4.3 Buffer management

In order to mix the packets based on their destination, we need to keep them separately. The most straightforward approach for buffer management is storing the packets based on their destinations in separate memories. However, this may lead to consume a large amount of memory space. In FairMix, to implement efficiently the buffer management, the following structure is maintained at each node: Node N upon receiving a packet p (coded or native) stores it in its buffer which is a FIFO memory, denoted as $Memo_N$. Since the packets for distinct destinations may arrive randomly and not necessarily in order, each node maintains a virtual queue per destination, denoted as $\mathcal{V}_{d_i}$. The virtual

[1] Impact of different scheduling policy, such as Round Robin (RR) or FIFO, on the FairMix performance has been explored in Section 5

queue $\mathcal{V}_{d_i}$ contains the pointers to the places of packets in $Memo_N$ whose destinations are d_i. If the buffer is full, node N generates a new encoded vector by mixing the received packets to a random entry in $\mathcal{V}_{d_i}$, described in details hereafter.

For buffer management and storing the encoded packets in destination based queues, the most straightforward approach, is storing the packets due to their destinations in separate memories. However this may lead to consume a lot of averaged-size memory (based on number of separate destinations) which is not an optimal approach. To implement efficiently the buffer management, the following structure which is inspired from [13] is maintained at each node: Node N upon receiving a packet p (coded or original) will store it in its buffer which is a FIFO memory, denoted as $Memo_N$. Since the packets for distinct destinations may arrive randomly and not necessarily in order, each node maintains a virtual queue per destination, denoted as $\mathcal{V}_{d_i}$. The virtual queue $\mathcal{V}_{d_i}$ contains the pointers to the places of packets in $Memo_N$ whose destinations are d_i. An example for buffer management at node N in a network with three destinations of d_1, d_2 and d_3, is shown in Figure 3.

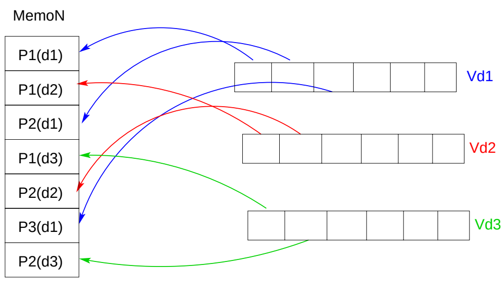


Figure 3: An example of buffer management in node N. There are three destinations of d_1, d_2 and d_3. Packets oriented towards different destinations are stored in a common buffer of $Memo_N$ and their places are pointed by pointers in the virtual queue of $\mathcal{V}_{d_i}$

In this work, we consider all packets (or destinations) have the same priority [2]. Therefore, FairMix uses Round Robin (RR) as scheduling policy in order to serve all queue with the same priority [6]. RR is then performed across all $\mathcal{V}_{d_i}$, as follows: in each round a linear combination of packets of one virtual queue is transmitted; since we consider limited bandwidth, only one linear combination is transmitted per round; for the next round, the next non-empty queue is then scheduled to be served.

4.4 End-to-end acknowledgement

When a destination decodes a block of data it is interested in, all the decoded packets are passed to the application layer. Therefore, once a block is decoded, the packets of that destination and their corresponding coefficients stored in the intermediate nodes are no longer of use to the network. To reduce the storage requirement without affecting the network delay, we delete the obsolete information from the network nodes. Let the destination d_i be waiting for the I^{th} data block sized K_i packets. Upon receiving

[2]The study of FairMix in environments requiring differentiated priorities is leaf for future works.

enough number of packets for its expected data block, it sends back an acknowledgement (cf. ack), denoted as $ack(d_i, I)$, for all its neighbors. The ack packet is transferred to the source nodes as a packet with high priority [3]. The source nodes by receiving an ack from any destination start sending the next block of the file. Each node N receiving or overhearing an $ack(d_i, I)$, extracts the destination id and the data block index I. Then following the pointers in virtual queue of $\mathcal{V}_{d_i}$ it is able to remove the related packets of the concerned index. Moreover, each node N tracks the last block index of each destination by saving the last heard index from previously acknowledged data blocks in $index\_last(d_i)$. Therefore, in this mechanism which is inspired from VACCINE [8], the nodes drop the packets that have already reached the destination and they do not accept them in their buffer anymore. Moreover, in the case of receiving a packet with higher index than $index\_last(d_i)$, they assume that they have missed an ack for the previous block. In this case, they free their buffer from the packets of the ancient block index and store the new one.

4.5 Packet coding

Algorithm FairMix as shown in Figure 4 is as follows: Node N, upon receiving a packet p, extracts its destination d_i and index I. It then checks if the destination has already acknowledged the reception of packets with this I. In the case that the packet is still valuable to be processed, node N checks if the packet is linearly independent from all the packets that are pointed by pointers in $\mathcal{V}_{d_i}$. If packet p is an innovative packet, node N stores it in the first empty place in the buffer $Memo_N$ and keeps its related pointer in $\mathcal{V}_{d_i}$. Note that since packets are inserted in the buffer based on their arrival and are removed based on the receiving of their acks, the empty places in the buffer are not in order. Therefore, the first empty place should be found by a simple search through the buffer. Since we assume nodes with limited buffer sizes, the buffer may be full. In that case, the received packet p is independently encoded to a random number of entries j of $\mathcal{V}_{d_i}$, and stored in the buffer, as follows:

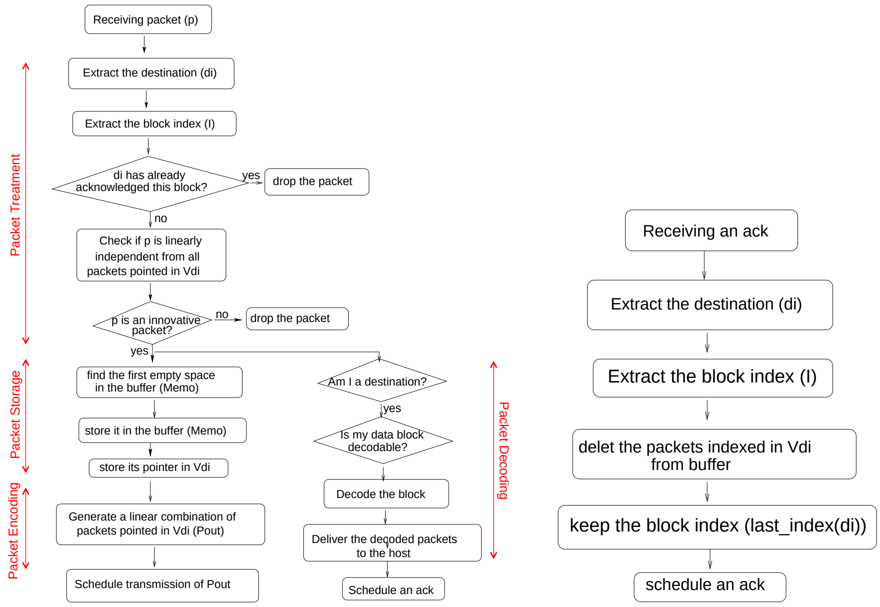
$$r = random[1, |\mathcal{V}_{d_i}|]$$
$$\forall j \in \{1, r\}_{\mathcal{V}_{d_i}}, Memo_N[j] = g_j^1(t).Memo_N[j] + g_j^2(t).p$$

in which $|\mathcal{V}_{d_i}|$ is the total number of pointers stored in $\mathcal{V}_{d_i}$ and $g_j^1(t)$ and $g_j^2(t)$ are random values chosen from the Galois field. Subsequently, a random linear combination of packets oriented to d_i is calculated and denoted as P_{out} as below :

$$P_{out} = \sum_{\forall j \in \mathcal{V}_{d_i}} Memo_N[j].g_j(t)$$

The generated packet at node N is then scheduled for transmission at the first transmission opportunity.

The decoding of data blocks is done at the destination nodes. Therefore, at each packet reception, if a receiving node N is a destination, say d_i, it verifies if it is able to decode its whole data block, i.e. if it has received the $K_{s_i}^{d_i}$ innovative packets. If so, the decoded packets are delivered to the application layer and an ack packet $ack(d_i, I)$ is scheduled for transmission.



(a) Flowchart of FairMix implementation in coding/decoding packets

(b) Flowchart of FairMix implementation at receiving an ack packet

Figure 4: Flowchart of FairMix implementation

The psudo code of FairMix algorithm in coding/decoding and managing the buffer is shown in Algorithm **??**.

5 Performance Analysis

This section describes the experiments we have conducted to assess the performance and fairness of network coding in two packet mixing strategies: FairMix and naive inter-session network coding. The goal is to show how network coding operates fairly in the presence of different block sizes destined for different destinations. Through this section by NetCode we mean naive inter-session network coding.

Our experiments reveal that, by separately coding the block sizes, FairMix considerably reduces the decoding delay for destinations waiting for small block sizes, independently of the loss rate of the network and number of destinations or sources. In addition, contrarily to NetCode, FairMix guarantees obtaining decoding delays proportional to block sizes of destinations. This shows how fair our strategy is in resource distribution among nodes, i.e. high demanding nodes wait longer time than low demanding ones.

Note that, in the presence of highly different block sizes, FairMix may increase the decoding delay of packet of large blocks, when compared to naive network coding. This, in fact, proves the fair resource distribution of FairMix. It occurs due to the fact that, by separating the coding per destination and by serving the queues with the same priority and bandwidth, packets in one queue should wait the service of the other queues before being scheduled to be transmitted [3]. Instead, in naive network coding, packets may be coded together and have equivalent chances to be transmitted. Thus, in FairMix, smaller block sizes take advantage of the fair resource distribution while larger block sizes may have a higher decoding delay. Nevertheless, the average decoding delay and fairness of system is considerably improved.

5.1 Evaluation methodology

We use a custom time-based network coding simulator, called NCWN [9]. FairMix and NetCode are implemented under this simulator. The time between sending a packet at a fixed rate until it is fully received at the next hop is defined as one time unit. Each node can only send or receive a packet at a time unit. Transmissions over the wireless channel are assumed to be broadcast and can be heard by all the in-range neighbors. Moreover, MAC layer is assumed to be an idealized version of CSMA/CA without collision. This is realized by a random scheduler that at each time unit chooses a transmitter randomly among all the *potential* transmitters (i.e the nodes that have at least one packet to send). The 1- and 2-hop neighbors of the senders are kept silent to prevent interferences. This is repeated until no more nodes are eligible to transmit at a given time step. Due to the fact that packet collision has the same impact on FairMix and NetCode, we believe that it does not play an important role in showing the difference in fairness by both protocols. Moreover, we modeled the random nature of wireless links (due to fading effects, etc.) by considering a packet loss parameter (called Loss Rate in the simulations), which represents the probability of packet drop over the links.

We consider a network with N nodes which are randomly placed in a square with area of S. The values of S, N, and the transmission range R_t of the nodes are chosen such that the density of the network is high enough to ensure a total connectivity (as shown in [17]). In the shown results, the number of nodes is set to 100. Multiple sessions are set between multiple random sources to multiple random destinations. The results are the average of repeating 100 times each experiment and confidence interval of 95%.

We use two metrics for evaluation: (i) decoding delay, which is the difference between the time instant of sending the first packet of a block and the time instant of receiving the last packet of the block at each specific destination and (ii) fairness, measured by ratio of obtained decoding delay and block size and by the "equality" of fairness. The equality is measured by Jain Fairness Index [11] and shows how far the service portion for each destination, is from equality in fairness.

Note that, as briefly discussed in Section 1, the benefits of mixing the maximum number of flows to better exploit the bandwidth and to obtain higher throughput, has been widely explored in the literature (ex. [22, 5, 15]). Thus, in this paper, for the lack of brevity, we omitted the results which confirm this idea.

5.2 Simulated Results

5.2.1 The case of different block sizes

We investigate the performance of both strategies (called FM and NC for resp. FairMix and NetCode) in different scenarios of test. In the first scenario, we evaluate the impact of block size and channel loss rate on decoding delay obtained by NC and FM. In this test, there are 8 distinct destinations which are waiting for blocks of data that are distributed among 4 randomly placed sources in the network. To cover a diversity in data block sizes, the corresponding block sizes of the destinations vary from 2 to 110 packets. Each set of experiment is done with a different loss rate in {15% , 30%, 50%}. As shown in Fig. 5(a), the decoding delay for different block sizes by network coding is the same for all the destinations, while FM has differentiated them very well and independently from the loss rate. For instance, for the loss rate of 15%, the decoding delay of d_8 with data block of 2 packets is reduced about 90% by FM.

Note that FM with providing more fair service portion has increased the delay for large blocks in comparison to NC. This in fact is normal in a fair system which distributes more fairly the network resources. Therefore, as the next step we need to evaluate the overall fairness for *all* the flows.

We expect that decoding delay changes proportional to block size of concern. Therefore, we define a *Fairness* parameter as the ratio of block size to its decoding delay. Enlarging the block size should increase the decoding delay and vice versa. Hence, in a fair system we expect that the variations of fairness parameter versus block size should be small. We compare this parameter for FM and NC in Fig. 5(b) for the same experiment as above. As shown in this figure, changes of fairness versus block size for NC is much higher than FM. It confirms the fact that in NC reducing the block size could not result in decreasing the decoding delay. This leads to highly variable service portion that the system provides for different users compared to the average. Apart from NC, by differentiating the delay proportional to block size, FM can greatly improves its fairness variations. In the literature, there are several mathematical and conceptual definitions for accurately measuring the overall fairness by system. To provide a mathematical evaluation of the fairness, we have calculated the Jain Fairness Index [11] for the both strategies: $JainIndex = \frac{(\sum x_i)^2}{n.\sum x_i^2}$ where x_i is the normalized service for the user i in terms of our defined fairness, and n is the number of users. The result shown in Fig. 5(c) shows a great difference between NC and FM. This difference is not reduced even by changing the channel loss rate. Moreover, it shows how close the allocation provided by FM is to the "equality": it is approximately 90% fair, i.e. 30% more than the fairness provided by NC.

To evaluate how the number of sources impacts the fairness provided by FairMix, in the next experiment shown in Fig. 5(d), we have set 4 randomly placed destinations waiting for 4 different block sizes. The data blocks are distributed on variable number of sources that changes from 1 to 6. Note that $1s/d$ in the figure means that the data blocks of 4 destinations with different block sizes are placed in one source, while $2s/d$ means that one more source is added to generate the data blocks for the 4 destinations, and so on. As it can be observed in Fig. 5(d), by increasing the number of sources the decoding delay is decreased for both NC and FM. This shows an obvious result that having multiple sources in the network leads to higher diversity of information which

[3]This wait time depends on the scheduling policy managing the process of queue service. Evaluating the impact of different scheduling is one of our future direction.

is beneficial for both protocols. The comparison of NC and FM shows again that FM is much more successful to make a distinction between destinations with different block sizes.

For the scenario of Fig. 5(a) we have measured the average transmission count per node for both NC and FM. The result is shown in Fig. 7. As it can be seen, the FM scheme by reducing the innovative packets sends more un-useful packets which increases the overhead.

5.2.2 Users's equal share case: equal block sizes

To see the impact of FairMix on the case of equal share for the users, the same experiment of changing number of sources from 1 to 6 for 4 randomly placed destinations is repeated. In this experiment the block size for all the destinations is set to 45 packet. We expect that all the destinations receive the same performance from the network. The results shown in Fig. 8 confirms it and shows that NetCode and FairMix give the same decoding delay for all the destinations.

In Fig. 9 we have compared the average transmission count and received innovative packet per node for both NetCode and FairMix. This results show that the overhead and the amount of innovative packet count for the both approaches are nearly the same for all the scenarios of different source number. This shows that FairMix by providing the same performance as NetCode, does not impose higher overhead and lower innovation to the network.

5.2.3 Impact of Scheduling Policy

In this experiment we have evaluated the impact of scheduling policy on the performance of network. Therefore, at each node we replaced the Round Robin (RR) scheduler with a FIFO, i.e. at each node, packets of each flow are serviced in the order of their arrivals. The experiment is done for three different loss rate in $\{15\%, 30\% ,50\%\}$. The results shown in Fig. 6 suggest that FIFO scheduling for all the loss rates, leads to less fair service level for large block flows. The reason is that RR, services the large block flows more frequently and more regularly than FIFO scheduling. This leads to less average decoding delay for them. Moreover, it can be seen in the figure that by FIFO, the service level for the flows with median block sizes is not changed a lot, while the fairness for small block flows is reduced. It shows that since FIFO services the packet of each block based on their arrival times, small blocks may wait longer than the case of RR scheduling. It can be concluded from this experiment that RR gives a more fair service to all the flows.

5.2.4 Impact of Galois Field Size

In network coding, one of the parameters that increases the probability of sending innovative encoded packet (by choosing random coefficients) is the Galois Field size in which the random coefficients are chosen randomly. Therefore, in order to prevent transmitting useless packet that do not increase the decoding matrix rank at the receivers, the size of Galois Field can be increased. Note that increasing the field size needs to piggyback more number of bits at each encoded packet. However, if increasing the field size can decrease the transmission of useless packet, the slightly increase in the size of packets is ignorable. Fig. 11 shows the summary of the experiment to show the impact of field size on performance. The field size is increased from 8 bit till 12 bit and there are 4 destinations waiting data blocks of 80, 40, 10 and 2 packets . As it can be seen, by increasing the field size, the decoding delay of FairMix and NetCode is not changed very much. It shows that the field size of 8 bit already gives an appropriate performance without a large amount of useless packets and so larger field size do not change the results too much.

Furthermore, comparing the average transmission count per node for both approaches for different field sizes (Fig. 12) confirms the previous results: the 8 bit field size is enough to guarantee a very low useless packet. This figure also shows a slightly higher overhead by FairMix at different field sizes.

5.2.5 Comparing with Simple Flooding

Till now, in both FairMix and Netcode whenever a node has a packet to send, it broadcasts for all its in-range neighbors. In Fig. 13 we have compared the decoding delay of NetCode, FairMix (with both scheduling policy of RR and FIFO) with the case that each node does a simple broadcasting without combining the packets (network coding). This experiment is done at loss rate = 15% and for 8 destination with different block sizes. The results confirms the great advantage of combining the messages through network coding instead of transmitting them separately. Moreover, the higher decoding delay by simple flooding than FairMix shows that although FairMix encodes the packets less than naive network coding by encoding the packets separately based on their destinations, but still it mixes the packets of different sources which are oriented to the same destination. Therefore, it still benefits from advantages of network coding.

5.2.6 Probabilistic Forwarding

In this part, we have evaluated the performance of FairMix and NetCode in the case of probabilistic forwarding, instead of broadcast forwarding. The probability of transmission at each node is defined based on the number of neighbors of that node. This approach that is proposed in [4] is called *Rapid* and claims that a node located in a high density area with a large number of neighbors should send less packets[4]. The reason is due to the probable congestion in that area and redundant messages that already are heard by the neighbors. Therefore, at each node the probability of transmission is a function of reciprocal of node's degree (number of neighbors).

In Fig. 14 we compare the decoding delay of FairMix and NetCode by a simple *Rapid* transmission. In order to do a fair comparison, transmission in FairMix and NetCode is probabilistic too. The results show again the advantage of network coding based approaches in comparison to simple probabilistic forwarding. Moreover, comparing the FairMix and NetCode performance suggest that with probabilistic forwarding, the difference between decoding delay of the large blocks by FairMix and NetCode is much lower than the case of broadcast forwarding. This shows that since probabilistic forwarding can greatly reduce the amount of useless transmission (because of less transmissions), FairMix with Rapid is more fair to larger blocks in comparison to FairMix with broadcast forwarding. Therefore, in the case that transmission is conditionally and not only due to having a packet, the performance of FairMix can

[4]We implemented the simple *Rapid* without *corrective measures*. The authors in [4] propose also an *Enhanced Rapid* in which some redundancy and neighbors' interference are reduced.

be improved in terms of fairness and applied overhead to the network. Note that this may happen in mobile scenarios in which not always nodes have opportunity to send their packets. In this case the sent packet is probably a mixture of multiple packets and so the probability that it is innovative is higher.

Another way of doing probabilistic forwarding is defining the probability of transmission based on the information status of the neighbors, i.e. if a node knows the probability that its packet is not innovative for its neighbors, it can decide to send it or not.

6 Summary and Future Directions

Network coding may lead to unfair decoding delay for multiple unicast flows with different data block sizes. This is due to the fact that in network coding, nodes mix together all packets in a data block, requiring destinations to receive a large enough number of encoded packets in order to be able to decode a whole block. By this scheme, the service level given to different destinations is not fairly distributed. To counter this issue, we proposed a practical and fair network coding called FairMix, which besides of benefiting from network coding advantages, improves considerably the fairness in decoding delay. Through simulation analysis, we compared the fairness and decoding delay of FairMix and naive network coding in different scenarios of test: under lossy wireless links, limited memory resources, different block sizes, and increasing number of sources. Under all the scenarios, FairMix proved to be effective in guaranteeing fairness among destinations (approximately 90% fair, i.e. 30% more than the fairness provided by naive network coding), while assured obtaining decoding delays proportional to block sizes of destinations. In addition, for destinations waiting for small block sizes, FairMix has provided considerable delay gains compared to naive network coding: about 90% of delay reduction in 15%-loss rate scenarios.

An important extension of this work is to consider other scheduling policies while analyzing FairMix. As showed by authors in [6, 7], scheduling policies can affect the delay perceived by destinations. In this way, the analysis of how different scheduling policies improve the FairMix performance, constitutes an interesting point of investigation. Another direction is the study of FairMix in environments requiring the association of differentiated priorities per users. In particular, we think in specifying classes of services with different delay constraints as well as scheduling policies to serve queues according these constraints, and then, verify if FairMix is able to guarantee the delay required by the application. Considering the particular case of delay tolerant networks, we also expect to evaluate FairMix in the presence of mobile scenarios.

References

[1] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, and Raymond W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[2] P. Bahl, R. Chandra, P. P. C. Lee, V. Misra, J. Padhye, D. Rubenstein, and Yan Yu. Opportunistic use of client repeaters to improve performance of wlans. In *ACM Conext*, dec 2008.

[3] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proc. of Conference on Information Sciences and Systems (CISS)*, pages 169–180, August 2007.

[4] Vadim Drabkin, Roy Friedman, Gabriel Kliot, and Marc Segal. Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks. In *26th IEEE International Symposium on Reliable Distributed Systems (SRDS 2007)*, pages 13–22, 2007.

[5] A. Eryilmaz and D. S. Lun. Control for inter-session network coding. In *Workshop on Network Coding, Theory and Applications (NetCod)*, Jan 2007.

[6] A. Eryilmaz, A. Ozdaglar, and M. Medard. On delay performance gains from network coding. In *Proc. of Conference on Information Sciences and Systems (CISS)*, pages 864–870, March 2006.

[7] A. Eryilmaz, A. Ozdaglar, M. Medard, and E. Ahmed. On the delay and throughput gains of coding in unreliable networks. In *Submitted to IEEE Transactions on Information Theory*, 2007.

[8] Zygmunt J. Haas and Tara Small. A new networking model for biological applications of ad hoc sensor network. *IEEE/ACM Transaction on Networking*, 14(1):27–40, Feb. 2006.

[9] Anwar Al Hamra, Chadi Barakat, and Thierry Turletti. Network coding for wireless mesh networks: A case study. In *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 284–291, 2006.

[10] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *Proc. of IEEE Infocom*, pages 836–843, April 2003.

[11] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. In *DEC Research Report TR-301*, September 1984.

[12] Golnaz Karbaschi and et. al. On the network coding fairness (extended version). *INRIA Technical Report*, April 2006.

[13] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Medard, and Jon Crowcroft. Xors in the air: Practical wireless network coding. *IEEE/ACM Transactions on Networking*, 16(3):497–510, June 2008.

[14] Ralf Koetter and Muriel Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5):782–795, Oct. 2003.

[15] Jilin Le, John C.S. Lui, and Dah Ming Chiu. How many packets can we encode? Ů an analysis of practical wireless network coding. In *IEEE INFOCOM*, 2008.

[16] Shuo-Yen Robert Li, Raymond W. Yeung, and Ning Cai. Linear network coding. *IEEE Transaction on Information Theory*, 49(2):371–381, Feb. 2003.

[17] T.K. Philips, S.S. Panwar, and A.N. Tantawi. Connectivity properties of a packet radio network model. In *IEEE Transaction on Information Theory*, Sept. 1989.

[18] Aditya Ramamoorthy, Jun Shi, and Richard Wesel. On the capacity of network coding for random networks. *IEEE Transaction on Information Theory*, 51(8):2878–2885, Aug. 2005.

[19] R. Koetter T. Ho, M. Medard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *International Symposium on Information Theory (ISIT)*, page 442, July 2003.

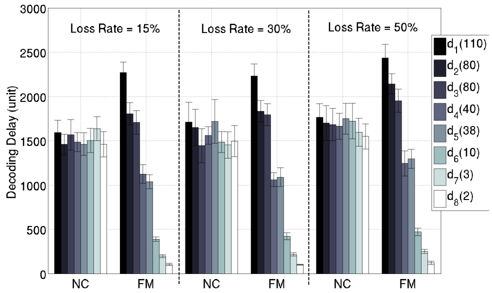
[20] Tracey Ho Harish Viswanathan. Dynamic algorithms for multicast with intra-session network coding. *IEEE Transaction on Information Theory*, 55(2):797–815, Feb. 2009.

[21] J. Widmer and J.Y. Le Boudec. Network coding for efficient communication in extreme networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 284–291, 2005.

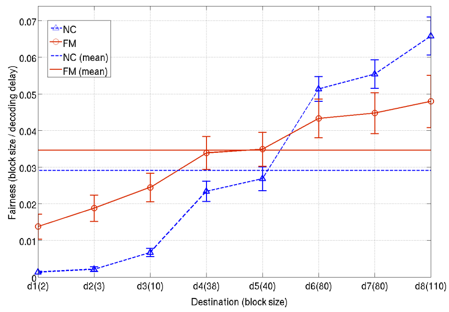
[22] Min Yang and Yuanyuan Yang. A linear inter-session network coding scheme for multicast. In *IEEE International Symposium on Network Computing and Applications*, pages 177–184, May 2008.

[23] Raymon W. Yeung. Multilevel diversity coding with distortion. *IEEE Transaction on Information Theory*, 41(2):412Ű422, Mar. 1995.

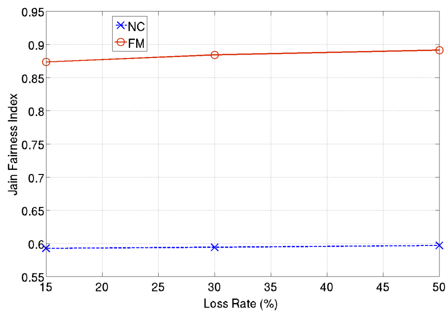
Contents



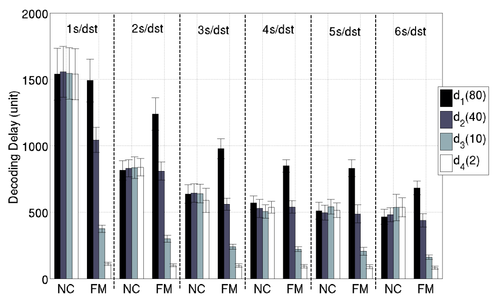
(a) Loss rate's impact in decoding delay



(b) Fairness measurement



(c) Jain Fairness Index



(d) Number of sources' impact in decoding delay

Figure 5: FairMix evaluation in terms of decoding delay and fairness, in a 100-node network with different loss rates, block sizes, and number of sources.

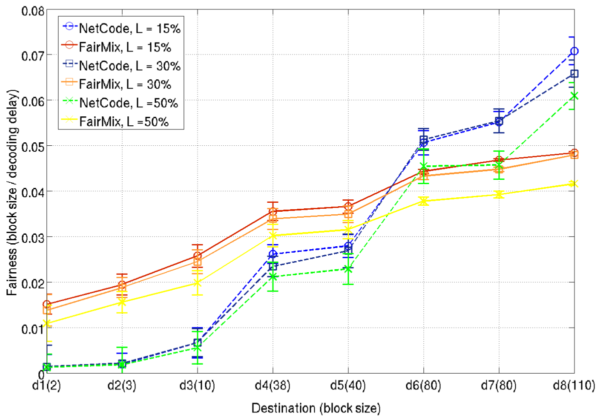


Figure 6: Fairness changes for NetCode and FairMix at different channel loss rates

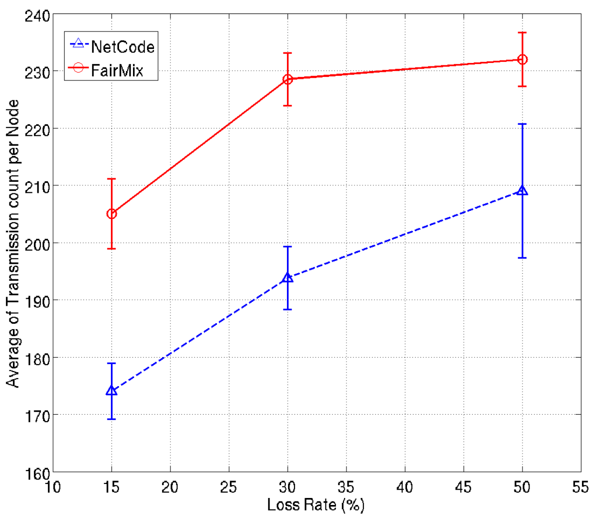


Figure 7: Average Transmission Count per node vs. Channel Loss Rate

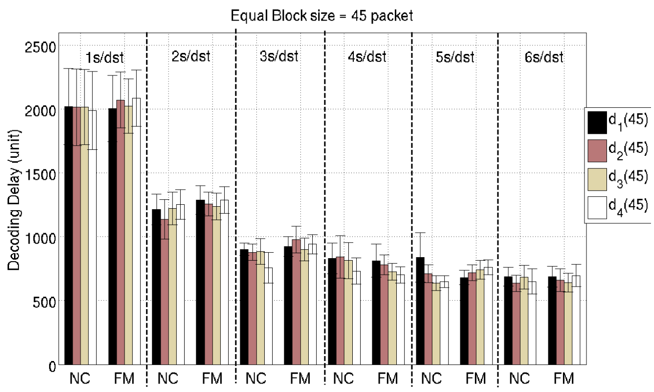
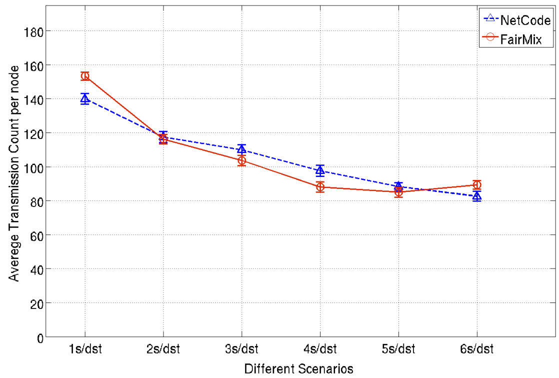
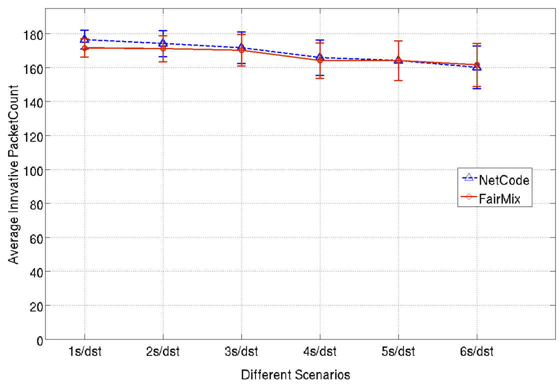


Figure 8: Decoding delay in the case of having the same block size for destinations

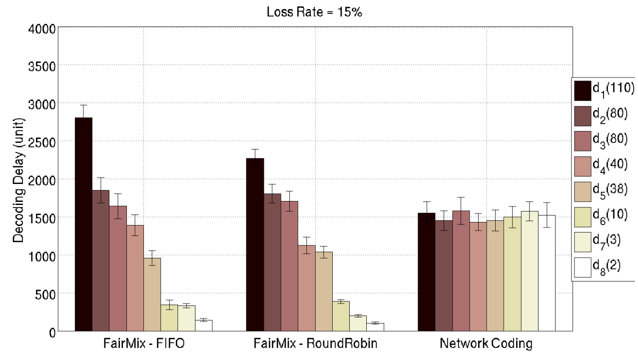


(a) Average Transmission Count per node

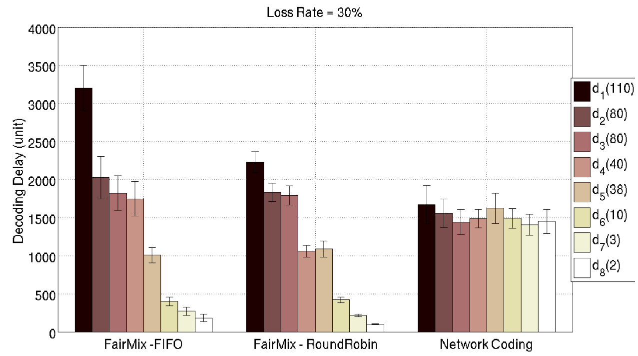


(b) Average Innovative packet Count per node

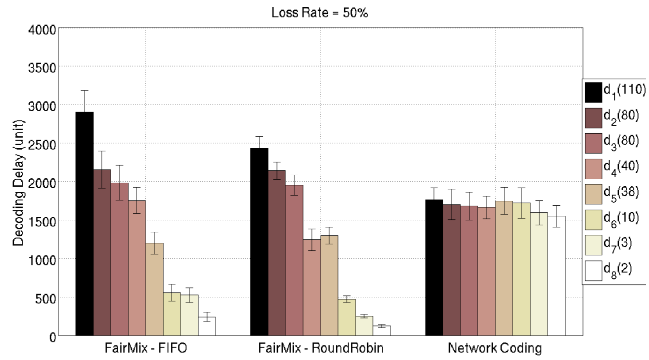
Figure 9: The resource utilization in the case of equal block sizes for all the destinations



(a) Loss Rate = 15%



(b) Loss Rate = 30%



(c) Loss Rate = 50%

Figure 10: The resource utilization in the case of equal block sizes for all the destinations

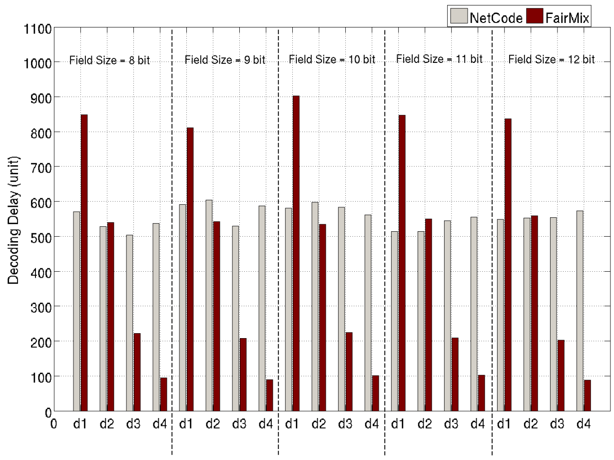


Figure 11: Impact of Galois Field size on the decoding delay by FairMix and NetCode.

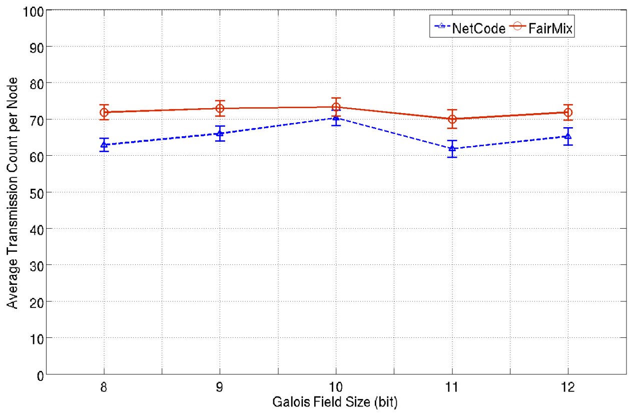


Figure 12: Average transmission count per node by changing the Galois Field size.

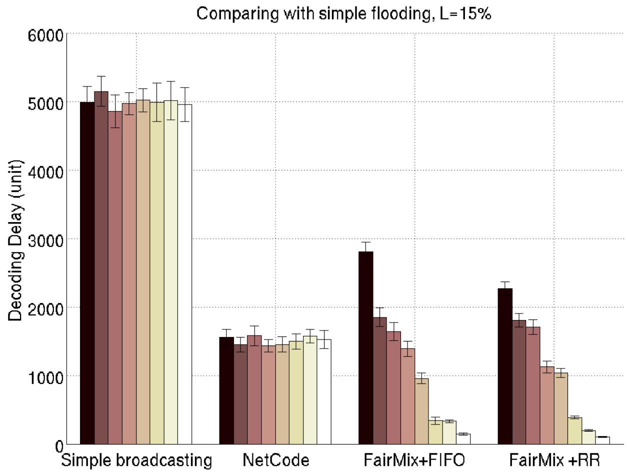


Figure 13: Comparing the decoding delay of FairMix (with RR and FIFO scheduling), NetCode and the simple flooding.

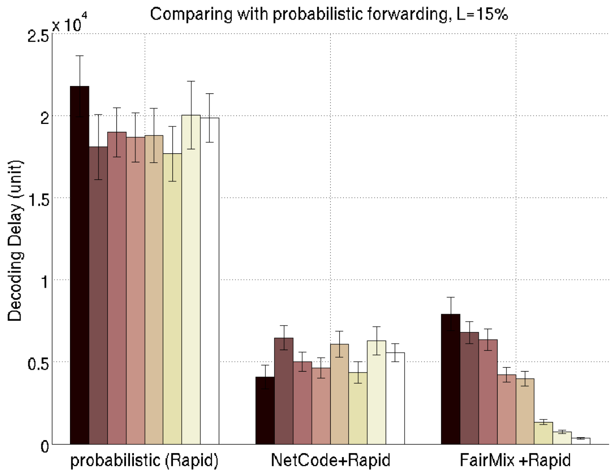


Figure 14: Comparing the decoding delay of FairMix and NetCode with probabilistic forwarding (Rapid) with the simple probabilistic forwarding.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399