Network Coding for Wireless Broadcast: Rate Selection with Dynamic Heuristics

Song Yean Cho, Cédric Adjih

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Network Coding for Wireless Broadcast: Rate Selection with Dynamic Heuristics

Song Yean Cho — Cédric Adjih

N° ????

Novembre 2007

Thème COM

Rapport de recherche

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

Network Coding for Wireless Broadcast: Rate Selection with Dynamic Heuristics

Song Yean Cho , Cédric Adjih

Thème COM — Systèmes communicants
Projets Hipercom

Rapport de recherche n° ???? — Novembre 2007 — 19 pages

Abstract: Network coding is a novel method for transmitting data, which has been recently proposed, and has been shown to have potential to improve wireless network performance.

In this article, we study using network coding for one specific case of multicast, broadcasting. Precisely, we focus on (energy-)efficient broadcasting in a multi-hop wireless networks: transmitting data from one source to all nodes with a small number of retransmissions. It is known that finding an efficient method to broadcast, is essentially summarized in selecting proper transmission rates of each node.

Our contribution, is proposing a simple and efficient method for determining a rate selection. Our method adapts dynamically and uses only local dynamic information of neighbors: Dynamic Rate Adaptation from Gap with Other Nodes (D.R.A.G.O.N.). The rationale of this rate selection method is detailed from some logical arguments. Experimental results illustrate the behavior of the method, and its excellent performance.

Key-words: wireless networks, network coding, broadcasting, multi-hop, adaptive algorithm

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Codage réseau pour la diffusion dans les réseaux sans fil: sélection de débit par des heuristiques dynamiques

Résumé : Le codage réseau est une nouvelle méthode pour transmettre des données qui a été proposée recemment, et dont le potentiel pour améliorer la performance des réseaux sans fil a été démontré.

Dans cet article, nous utilisons le codage réseau pour un cas spécifique de multi-diffusion: la diffusion à tout le réseau. Plus précisemment, nous nous intéressons à une diffusion efficace en termes d'énergie: la transmission d'une source à tous les noeuds avec un petit nombre de retransmissions. Il est connu que trouver une méthode efficace de diffusion, se ramène essentiellement à choisir les débits appropriés de chaque noeud.

Notre contribution est la proposition d'une méthode simple et efficace pour déterminer cette sélection de débits. Notre méthode s'adapte dynamiquement et utilise exclusivement l'information locale (dynamique) des voisins: adaptation dynamique du débit à partir de l'écart avec d'autres noeuds. L'idée derrière cette sélection de débit est détaillée avec des arguments logiques. Des résultats expérimentaux illustrent le comportement de la méthode et son excellent performance.

Mots-clés : réseaux sans fil, codage réseau, diffusion, multi-sauts, algorithme adaptatif

1 Introduction

Seminal work from Ahlswede, Cai, Li and Yeung [1] has shown that *network coding*, where intermediate nodes mix information from different flows, can achieve higher throughput for multicasting than classical routing. Since then, network coding has been shown to be specially useful in the context of wireless communications and in the context of multicast.

In this article, we will focus on using network coding as an *efficient* method to broadcast data to the entire wireless network, where the cost is the total number of transmissions. Such an approach can be considered to be both *energy-efficient*, considering the energy cost of each transmission on devices with limited battery life, or to be a more general heuristic for efficiency in order to alleviate the congestion in the network. Results from information theory indicate that, for single source broadcast, that optimality may be achieved by a simple coding method, *random linear coding*. [2,3] Then the key parameter for a given instance of a network is essentially the average retransmission rate of the nodes — established on the entire duration of the stream (even for packet networks, see [3] for a recent synthesis of such results): the broadcast efficiency in terms of the number of transmissions can be evaluated with the average rates.

As a result, finding an optimal solution to minimize the cost consists of finding the optimal average rate of every node [2, 3]: a *rate selection*. The problem is then reformulated:

- Problem statement: how to find a rate selection with good performance?

In fact, [5, 4, 6] have shown that optimal rate selection may be effectively obtained by solving linear programs – thus in polynomial time; and possibly in a distributed fashion ([6] for instance).

However, a different, even simpler, approach is possible: previous work [7] has shown that a simple rate selection heuristic would achieve asymptotically the optimal efficiency for homogeneous large and dense wireless networks of the plane: essentially setting the same rate on every node. For actual networks, this rate selection may be adjusted with simple heuristics using local topology: such heuristics have been explored in [8]. They performed well; however in special cases such as sparse networks, it was found that the performance decreased, and local topology information does not seem sufficient to detect such cases.

In this article, we start from the same general idea of finding not necessarily an optimal rate selection, but a simple and efficient rate selection. We also start from the logic used in methods in [8], but with the novel approach of using simple dynamic information to adapt the rates, rather than static topology information. There are several advantages in doing so; first, the problematic cases which cannot be detected from static local topology information, may be discovered from dynamic information. Second, in a real network, the network itself evolves dynamically, (for instance packet loss rate or topology may change), and hence the rate selection can no longer be a fixed constant rate. These dynamic features require dynamic adaption in any case.

Our key contributions are the proposal of a new heuristic for rate selections, its analysis based using insights from theory, and an experimental investigation of the performance with simulations.

The rest of the paper is organized as follows: section 2 details the general framework, section 3 provides some background about network coding, section 4 details the new heuristic, DRAGON, section 5 analyzes performance with experimental results and section 6 concludes.

2 Framework

As indicated in the section 1, this article focuses on an efficient network coding method to perform broadcast.

We assume that a fixed number of nodes are present in a multi-hop wireless network. Inside the network, there is one source which will transmit a finite number of packets (referred to as the *generation size*). Other nodes are retransmitting the packets with network coding (as described in section 3.1).

The objective is broadcasting: eventually every node will have obtained a copy of the packets originated from the source.

In general terms, as seen in section 1, the performance of a network coding method is derived from the average rates of the nodes, and we focus on methods which explicitly select rates of the node: rate selection (see section 3.1.4). Once the rates of the nodes are defined, there exists a limit for the rate of the source, the *maximum broadcast rate*. Below it, nodes have asymptotically great chances to decode all packets, and beyond it, they cannot.

In this article, we introduce a heuristic for selecting rates: DRAGON. It does not assume a specific type of network topology; the only assumption is that one transmission reaches several neighbors at the same time.

The rate selection of DRAGON is constructed in a similar manner to previously published heuristics (further described in section 4); as a result, the property of efficiency of DRAGON is inherited from these heuristics (see section 4). The previous heuristics had been proven to be either asymptotically optimal or experimentally efficient, with the following models of multi-hop wireless network: where nodes are distributed uniformly (at random or in a grid); and with *unit disk graphs* models [15], where two nodes are neighbors whenever their distance is lower than a fixed radio range.

However, DRAGON is also constructed as a feedback control: from the expected dynamic behavior of network coding (namely a linear increasing with time of the information received, see section 3.2.1), it will detect inadequacies in the rate selection, and will adjust it accordingly. Owning to this afterfact adaptation ability, it is able to use a pessimistic strategy, which decreases the rates for nodes whose transmissions are not absolutely guaranteed to be beneficial.

In addition, the performance, the number of transmissions per source packet, is identical after scaling the rates. We will assume that such rates are scaled so that nodes operate below the wireless capacity.

3 Network Coding Background

In this section, several known results about network coding are presented: practical aspects in section 3.1, and theoretic performance in section 3.2.

3.1 Practical Network Coding

3.1.1 Linear Coding

Network coding consists in performing *coding* inside the network. One notable method of coding is *linear coding* [9] (see also [10]).

Starting with the assumption that all packets have identical size, with linear coding, the packets can viewed as vectors of coefficients of a fixed Galois field $\mathbb{F}_q^n$. Then this makes possible to compute linear combinations of them: this is the coding operation in linear coding. Since all packets originate from the source, at any point of time a node v will possess a set coded packets, every of which is a linear combination of the original source packets:

$$\mathrm{i}^{th}\text{coded packet at node } v : p_i^{(v)} = \sum_{j=1}^{j=k} a_{i,j} P_j$$

where the $(P_j)_{j=1,\ldots,k}$ are k packets generated from the source. The sequence of coefficients for $p_i^{(v)}$, $[a_{i,1}, a_{i,2}, \ldots, a_{i,n}]$ is the *coding vector* of coded packet $p_i^{(v)}$.

If, for instance, the field is $\mathbb{F}_2$, then the packets are viewed as sequence of bits, and the addition of vectors is an *exclusive or*. The coefficients are 0 or 1, i.e. every coded packet is the exclusive-or of a subset of source packets.

3.1.2 Random Linear Coding

With linear coding, an issue is selecting coefficients for the previous linear combinations. Whereas centralized deterministic methods exist, [2] presented a coding method, which does not require coordination of the nodes, *random linear coding*: when a node wishes to transmit a packet, computes a linear combination of all the coded packets that it possess, with randomly selected coefficients (α_i), and sends the coded packet:

$$\text{coded\_packet} \ = \ \sum_i \alpha_i p_i^{(v)}$$

This approach is made practical, with the proposition in [12], to add a special header containing coding vector of the transmitted packet. Therefore, the packets are blocks of symbols over a field $\mathbb{F}_p^n$: $(s_1, s_2, ..., s_h)$, and with the header of coefficients of coding vector $(g_i)_{i=1,\ldots,D}$, and they are transmitted as $(g_1, g_2, \ldots, g_D; s_1, s_2, \ldots, s_h)$. The new coding vector (g_i) is computed from the randomly selected coefficients α_i and the previous coding vectors $(a_{i,j})$ as seen in section 3.1.1.

3.1.3 Decoding, Vector Space, and Rank

The node will recover the source packets $\{P_j\}$ from the packets $\{p_i^{(v)}\}$, considering the matrix of coefficients $\{a_{i,j}\}$ from section 3.1.1. Decoding amounts to inverting this matrix, for instance with Gaussian elimination.

Thinking in terms of coding vectors, at any point of time, it is possible to associate with one node v, the *vector space*, Π_v spawned by the coding vectors, and which is identified with the matrix. The dimension of that vector space, denoted D_v, $D_v \triangleq \dim \Pi_v$, is also the *rank* of the matrix. In the rest of this

article, by abuse of language, we will call *rank of a node*, that rank and dimension. Ultimately a node can decode all source packets when the rank is equal to the the total number of source packets (*generation size*). See also [11].

It is a direct metric for the amount of useful received packets, and a received packet is called *innovative* when it increases the rank of the receiving node.

3.1.4 Rate Selection

In random linear coding, the remaining decision is *when* to send packets. This could be done by deterministic algorithms; for instance, [11] proposes algorithms which take a decision of sending or not another packet upon reception.

In this article, we consider "rate selections": at every point of time, an algorithm is deciding the rate of every node. We denote $\mathcal{V}$ the set of nodes, and $C_v(\tau)$ the rate of the node $v \in \mathcal{V}$ at time τ. Then, random linear coding operates as indicated on algorithm 1. With this scheduling, the parameter

Algorithm 1: Random Linear Coding with Rate Selection

1.1 **Source scheduling:** the source transmits sequentially the D vectors (packets) of a generation with rate C_s.
1.2 **Nodes' start and stop conditions:** The nodes start transmitting when they receive the first vector but they continue transmitting until themselves **and their neighbors** have enough vectors to recover the D source packets.
1.3 **Nodes' scheduling:** every node v retransmits linear combinations of the vectors it has, and waits for a delay computed from the rate distribution.

which varies, is the delay, and we choose to compute it as an approximation from the rate $C_v(t)$ as: delay $\approx 1/C_v(t)$.

Notice that all the rates $\{C_v\}$ may be scaled by the same amount, as long as the network stays below the wireless network capacity limits (considering interferences). The cost is identical. We will assume that a scaling is performed so that, arbitrarily, $C_s = M$, where M is the average number of neighbors.

3.2 Performance of Wireless Network Coding

3.2.1 Network Coding and Maximum Broadcast Rate

One of the notable existing results is that *random linear coding* [2], performs asymptotically as efficiently as any other network coding method (it is *capacity achieving*), for the case of single source multicast [14].

Another notable result, relates to the performance of random linear coding, is one of a series of information-theoretic results about (the *capacity region* with) network coding, starting with [1]. It is the following [3]: with random linear coding, the source may transmit at a rate arbitrarily close to some fixed rate, the *maximum broadcast rate* – which is the min-cut of an hypergraph (defined later) – and at the end of the broadcast, all the destinations can decode with a probability p_e. The error probability p_e may be made arbitrarily small, by increasing the generation size. This asymptotic result is independent of field size [3].

As indicated, the maximum broadcast rate of the source is the rate limit, above which the nodes will not be able to decode, is given by the min-cut from the source to each individual destination of the network, viewed as a hypergraph for wireless networks [13, 6].

The idea behind the min-cut, is to cut the network into two parts, and check the total rate transmitted from nodes in the part including the source, to nodes of the other part. It turns out the minimum such total rate between partitions, for all possible partitions, is exactly the maximum broadcast rate.

Formally: let us consider the source s, and one of the broadcast destinations $t \in \mathcal{V}$. The definition of an *s-t cut* is: a partition of the set of nodes V in two sets S, T such as $s \in S$ and $t \in T$. Let us denote ΔS, the set of nodes of S that are neighbors of at least one node of T; the *capacity of the cut* $C(S)$ is defined as the maximum rate between the nodes in S and the nodes in T:

$$\Delta S \triangleq \{v \in S : \mathcal{N}(v) \cap T \neq \emptyset\} \quad \text{and} \quad C(S) \triangleq \sum_{v \in \Delta S} C_v \tag{1}$$

Let $Q(s,t)$ be the set of s-t cuts. The *min-cut* between s and t is the cut in $Q(s,t)$ with the minimum capacity. Let us denote $C_{\min}(s,t)$ as its capacity. From [13, 6], the maximum broadcast rate is given by the minimum of capacity of the min-cut of every destination, $C^{\text{all}}_{\min}(s)$. Formally:

$$C_{\min}(s,t) \triangleq \min_{(S,T) \in Q(s,t)} C(S) \; ; \; C^{\text{all}}_{\min}(s) \triangleq \min_{t \in \mathcal{V} \setminus \{s\}} C_{\min}(s,t) \tag{2}$$

From [14, 3], when using random linear coding, with *any scheduling*, the maximum broadcast rate is the the capacity of the min-cut computed using average rates in the previous equations.

From the previous results, the dynamic behavior of the rank of one node is also known: assume that the source is transmitting with a rate C_s lower or equal to the maximum broadcast rate $C^{\text{all}}_{\min}(s)$. Then the rank of one node vt, will grow linearly with time τ as: $D_t(\tau) \approx C_s \times \tau$.

If the source is transmitting with a rate too large, then it will be bounded by the min-cut: $D_t(\tau) \approx C_{\min}(s,t) \times \tau$.

4 Heuristics for Rate Selection

In this section, we describe related work, prior heuristics for rate selection in section 4.3, and the related proposed dynamic heuristic, `DRAGON` in section 4.4. Before, we introduce the general concept of *cut at destination* and *local received rate*, in section 4.1, from which all these heuristics are actually derived; and they are connected to the maximum broadcast rate obtained as in section 3.2.1.

4.1 Local Received Rate: Cut At Destination

The cut at destination is a special case of a cut as defined section 3.2.1. Formally, it is the partition S/T of the nodes, where T includes only the destination node $T = \{t\}$ (hence $S = \mathcal{V} \setminus \{t\}$). An example is represented on Fig. 1: the node t has neighbor nodes v_1, v_2, v_3 and v_4 (this is the set ΔS of the cut in section 3.2.1); the capacity of the cut is the sum of the rate of these nodes, $C(S) = C_{v_1} + C_{v_2} + C_{v_3} + C_{v_4}$ (as given by (1)).

The capacity of the cut $C(S)$ may be pictured as the total rate that flows through the boundary between S and T, and the capacity of cut at destination

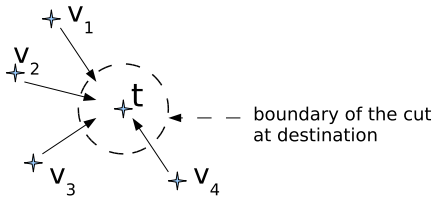

Figure 1: Cut S/T at destination: $\Delta S = \{v_1, v_2, v_3, v_4\}; T = \{t\}$

is simply the total rate at received one node from its neighbors: the *local received rate*.

4.2 Cut At Destination, Min-Cut and Efficiency

From section 3.2.1, maximum broadcast rate of the source is the capacity of the min-cut. Therefore it is lower than the capacity of any cut at destination (= any local received rate). Notice that in general, the min-cut need not to be the cut

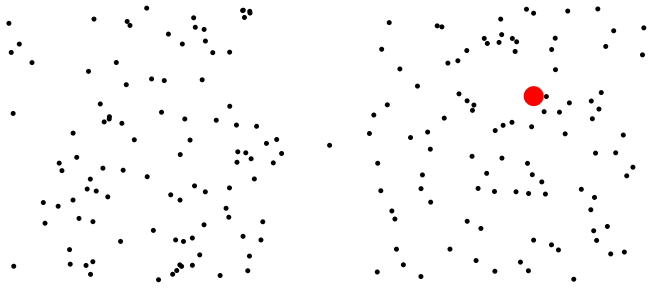

Figure 2: Example topology (source is the larger node)

at one destination: for instance in one example topology of Fig. 2, one unique node is connecting the left and right parts of the network. If, for instance, all the nodes had an identical average rate lower than the source, then that center node would be the bottleneck, and the min-cut would be the partition S/T with: on one side, all nodes in right part which containing the source plus that center (set S); and on the other side, all the nodes of the left part (set T).

One link between the cut at destination and min-cut comes from [7]: a specific rate selection was studied (*"IREN/IRON": Increased Rate for Exceptional Nodes, Identical Rate for Other Nodes*), where all nodes have the same rate, except from the source and the nodes near the edge of the network. One result was that the min-cut is essentially equal to the number of neighbors in some specific cases. This implies that in those cases, the min-cut is essentially one cut at destination.

In addition, under the same asymptotic conditions, it was proved on average the proportion of non-innovative transmissions would converge towards 0. This proved asymptotic optimal efficiency - for the metric of the number of transmissions per broadcast packet, and an advantage over non-coding.

4.3 Prior Static Heuristics using Cut At Destination

From section 4.2, we saw that the min-cut is related to the local received rate in some cases of homogeneous networks. However, when the nodes have different numbers of neighbors, a first step is to adjust it, in order that the local received rate would be at least the broadcast rate of the source, M, for every destination: indeed, every node in the network should receive at least a total rate M from its neighbors.

A heuristic was explored in [8], inspired from [11], using the following logic. A simple way to ensure that the local received rate of every node is at least equal to M, is the following: when a node has h neighbors, every of its neighbors would have a rate at least equal to $\frac{M}{h}$. Then the local received rate is always at least the sum of h such rates; indeed greater or equal to M. This yields the following rate selection:

- IRMS (Increased Rate for the Most Starving node) [8]: the rate of a node v is set to C_v, with:

$$C_v = k \max_{u \in H_v} \frac{M}{|H_u|}$$

  where H_w is the set of neighbors of w, $|H_w|$ is its size, and $k = 1$ is a global adjustment factor.

In [11], and in a slightly different context (not explicitly a rate selection, broadcast all-to-all), theoretical arguments were given for ability to decode in the case of general networks when $k \geq 3$; but again, [7] is showing that $k = 1$ is sufficient, asymptotically.

In [8], IRMS ($k = 1$) was explored experimentally, and although overall good performance was observed, in the case of sparser networks, phenomena occurred where only a few nodes would connect one part of the network to another, in a similar fashion to the center node in Fig. 2. In networks similar to Fig. 2, the rate of the nodes linking two parts of the network, would be dependent on how many of them are present: such information is not available from local topology information.

4.4 New Dynamic Heuristic, DRAGON

The previous heuristics for rate selection were static, using simple local topology information, and the rates would be constant as long as the topology would remain identical.

In this article, a different approach is chosen. The starting point is the observation that with fixed rates, the rank of a node v, will grow linearly with time, as $D_v(\tau) \approx C_{\min}(s, v) \times \tau$, i.e. proportionally to the capacity of the min-cut from the source s to the node (see section 3.2.1). With a correct rate selection, one would expect this min-cut to be close to the source rate $C_{\min}(s, v) \approx M$ in every node, and hence would expect that all the ranks of all nodes grow at the same pace. Failure to do so is a symptom that the rate selection requires adjustment.

From $D_v(\tau)$, the rank of a node v at time τ, let us define $g_v(\tau)$, the maximum gap of rank with its neighbors, normalized by the number of neighbors, that is:

$$g_v(\tau) \triangleq \max_{v \in H_u} \frac{D_v(\tau) - D_u(\tau)}{|H_u|}$$

We propose the following rate selection, DRAGON *Dynamic Rate Adaptation from Gap with Other Nodes*, which adjusts the rates dynamically, based on that gap of rank between one node and its neighbors as follows:

- DRAGON: the rate of node v is set to $C_v(\tau)$ at time τ as:
  - if $g_v(\tau) > 0$ then: $C_v(\tau) = \alpha g_v(\tau)$
    where α is some constant
  - Otherwise, the node stops sending encoded packets until $g_v(\tau)$ becomes larger than 0

This heuristic has some strong similarities, with the reasonings presented in section 4.3, and with IRMS. Consider the cut at one destination v: then DRAGON ensures that every node will receive a total rate at least equal to the average gap of one node and its neighbors scaled by α, that is the local received rate at time τ verifies:

$$C(\mathcal{V} \setminus \{v\}) \geq \alpha \left(\frac{1}{|H_v|} \sum_{u \in H_v} D_u(\tau) - D_v(\tau) \right)$$

which would ensure that the gap would be closed in time $\approx\leq \frac{1}{\alpha}$, if the neighbors did not receive new innovative packets.

4.5 Theoretical Analysis of DRAGON

4.5.1 Overview

In effect, DRAGON is performing a feedback control, which intends to equalize the rank of one node to the rank of its neighbors, by adapting the rate of the nodes.

However, notice that the precise control-theoretic analysis of DRAGON is complicated by the fact the rank gap does not behave like a simple "physical" output, and that we have the following properties, for two neighbor nodes u, v:

- If $D_u > D_v$, then every transmission of u received by v will increase the rank of v (is innovative).
- If $D_u \leq D_v$, then a transmission of u received by v, may or may not, increase the rank of v, both cases may occur.

The last property may be illustrated in the network on Fig. 2. Consider the center node: it is the only node connecting the two parts of the network, hence all nodes on the left part of the network received vectors that went through it, and any transmission received from the left, will not be innovative for the center node. Now, imagine that another node is added, out of range of center node, such as it also connects the two parts of the network. In that case, some transmissions from the left are likely to become innovative for the center node (depending on the overall rate selection).

As a result, there is some uncertainty in the control about the effect of increasing the rate of a neighbor that has greater rank than another node. A refined dynamic approach would be gather detailed statistics about the innovation rate, and use this information in the control; however the approach used in DRAGON is simpler and more direct: if a node has lower rank than all its neighbors, it will stop sending packets - this amounts to pessimistically estimate that transmissions from nodes with higher rank are non-innovative.

Although this would tend to make the rates less stable with time, intuitively this property might allow DRAGON to be more efficient. Note also from sec-

tion 3.2.1, that there is no direct penalty for unstable rates: only the average rates matter for the maximum broadcast rate (and also for the cost).

4.5.2 Insights from Tandem Networks

Although the exact modelling of the control is complex, some insight may be gained from approximation.

Consider one path $(s = v_0, v_1, \dots v_n)$ from the source s to a given node v_n, such as on Fig. 3.

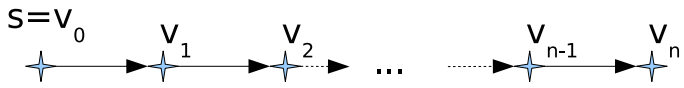

Figure 3: Line network

Denote $D_k \triangleq D_{v_k}$, $C_k \triangleq C_{v_k}$. Assume now that the ranks in the nodes verify $D_{k+1}(\tau) < D_k(\tau)$, for $k = 0, \dots, n-1$, and then a fluid approximation yields the following equations:

$$\frac{dD_{k+1}(\tau)}{d\tau} = C_k(\tau) + I_k(\tau) \text{ with } \alpha_k = \frac{\alpha}{|H_{v_{k+1}}|} \tag{3}$$

$$C_k(\tau) = \alpha_k(D_k(\tau) - D_{k+1}(\tau - \delta_k(\tau))) \tag{4}$$

where $I_k(\tau)$ is the extra innovative packet rate at v_k from other neighbors than v_{k-1} and $\delta_k(\tau)$ is the delay for node $k+1$ to get information about the rank of its neighbor k. If rank is piggybacked on each transmitted packet, then $\delta_k(\tau) \approx \frac{1}{C_k(\tau)}$.

If we make the approximation that $\delta_k(\tau) = 0$, and if we consider a linear network composed exclusively of the path, then $\alpha_k = \frac{1}{2}\alpha$ and $I_k(\tau) = 0$. Let $\beta \triangleq \frac{1}{2}\alpha$. In that case, (3) and (4) yields a sequence of first order equations for D_k, solvable with standard resolution methods:

$$D_k(\tau) = M\tau - (1 - e^{-\beta\tau})\frac{kM}{\beta} - e^{-\beta\tau}P_k(\tau) \tag{5}$$

where $P_k(\tau)$ is a polynomial of τ with $P_k(0) = 0$. This result shows that in that case the ranks are $D_k(t) \approx M\tau$ plus an additional term: when $\tau \ll \frac{1}{\beta}$, this term is $\approx P_k(\tau)$ and is 0 for $\tau = 0$; whereas when $\tau \gg \frac{1}{\beta}$ this term is $\approx \frac{kM}{\beta}$.

The conclusion is that, for a line network, when $\tau \to \infty$, the rank of the nodes $v_0, v_1, \dots, v_k$ are such as the dimension gap between two neighbors is $\frac{M}{\beta}$, and this occurs after a time on the order of magnitude of $\frac{1}{\beta} = \frac{2}{\alpha}$: the rank of the nodes decreases linearly from the source to the edge of the network.

5 Experimental Results

5.1 Model, Metrics, Environment and Scenarios

In order to evaluate the performance of `DRAGON`, we performed extensive simulations, which are detailed in this section. The focus of the `DRAGON` algorithm is

on wireless ad hoc networks, and simulations were performed either for random uniform graphs (inside a square) or with the reference network of Fig. 2, in which one node connecting two parts. This last network is of special interest because it exemplifies features found in sparse networks, where static heuristics fail.

Parameter	Value(s)
Number of nodes	200
Range	defined by M, expectation of number of nodes in one neighborhood. $M = 8$
Position of the nodes	random uniform i.i.d -or- net. on Fig. 2
Generation size	500
Field $\mathbb{F}_p$	$p = 1078071557$
α (in `DRAGON`)	$\alpha = 1$

Table 1: Default simulation parameters.

The default simulation parameters are given in table 1. The simulator used was self-developed, with an ideal wireless model with no contention, no collision and instant transmission/reception. For comparison purposes, NS-2 (version 2.31) was also used its default parameters.

The metric for (energy-)efficiency that is used in simulations is: the total number of transmitted packets per source packet, and is denoted as E_{cost}. $E_{\text{cost}} \triangleq \frac{\text{Total number of transmitted packets}}{\text{Generation size}}$. As mentioned in section 1, the optimal rate selection may be computed from a linear program [6]: it is the rate selection which minimizes E_{cost}. In some scenarios, we computed numerically this minimum E_{cost} (by a linear program solver), to obtain a reference point.

5.2 General Behavior of Wireless Network Coding

A first general view of the behavior of wireless broadcast with network coding would separate the broadcast in three durations: the starting phase, the general phase, and the termination phase. We simulated with IRMS, with default parameters, on the network on Fig. 2. Results are illustrated on Fig. 5.2: the average innovative packet rate received by nodes in the network, depending on the time. In the starting phase, the nodes have not enough vectors, and then the efficiency of network coding is limited. Hence the innovative rate is limited. In the general phase, network coding acts as predicted. In the termination phase, some nodes have all the packets (and are able to decode), but some other do not. When the generation size is sufficiently large, the performance is dominated by the behavior in the central part.

5.3 Efficiency of `DRAGON`

In this section, we start the analysis of the performance of various heuristics, by considering their efficiency from E_{cost}. Simulations were performed on several graphs with default parameters but $M = 6$ – relatively sparse networks – and with three rate selections: optimal rate selection, IRMS, and `DRAGON`. The Fig. 5.3 represents the results (for an average of 6 random graphs).

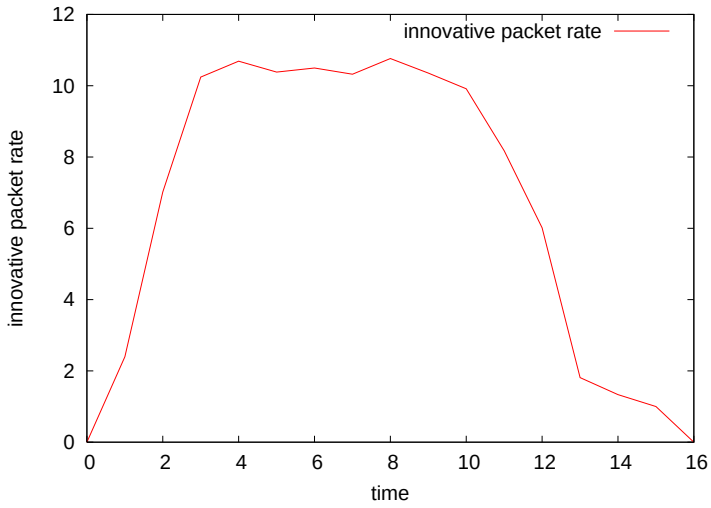

Figure 4: Average innovative packet rate vs time

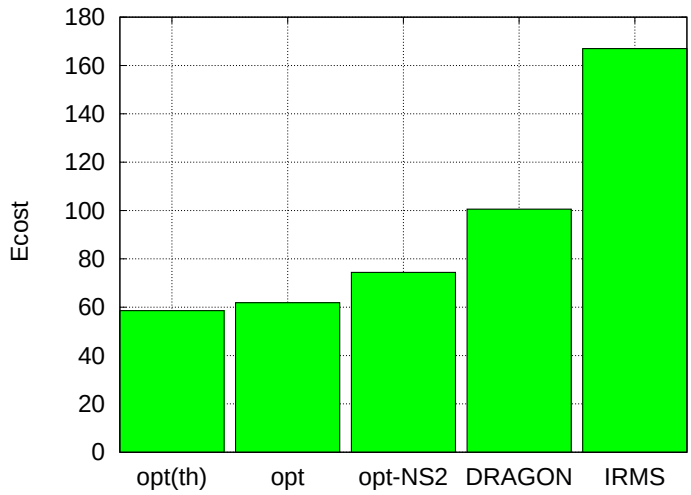

Figure 5: Performance of different heuristics

5.3.1 Theory and Practice

The first 3 bars, are unrelated with `DRAGON`, and in essence, attempt to capture the gap between theory and practice.

The first bar (label: $opt(th)$), is the optimal E_{cost} as obtained directly from the linear program solution, without simulation. The second bar (label: opt), is the actual measured E_{cost} in simulations in the ideal model wireless model, with optimal rate selection. The third bar (label: $opt - NS2$) is the actual measured E_{cost} in simulations with the simulator $NS-2$ (packet size 512, coding vector headers included).

As one might see, with the default parameters, the measured efficiency when performing actual coding with optimal rates (and generation size = 500), gives a result rather close to numerical value of the optimal: within a few percents. Another result is that the impact of the physical and mac layer, as simulated by NS-2 (with 802.11, two ray ground propagation, omni-antenna), is limited as one can see: $\approx 20\%$ (and the channel occupation rate was approximately of 1/3).

The results are close. Therefore, because the purpose of our algorithm is not to perform congestion control (and because the parameters chosen would made some simulations operate above the channel capacity), in the rest, we will present results with the ideal wireless model.

5.3.2 Efficiency of different heuristics

The two last bars of Fig. 5.3 represent the efficiency of `DRAGON` and IRMS respectively. As one may see in this scenario, the ratio between the optimal rate selection, and `DRAGON` is around 1.6, but without reaching this absolute optimum, `DRAGON` still offers significantly superior performance to IRMS.

The gain in performance comes from the fact that the rate selection IRMS, has lower maximum broadcast rate (in some parts of the network), than the actual targeted one, and hence than the actual source rate. As a result, in the parts with lower min-cut, the rate of the nodes is too high compared to the innovation rate. Whereas with `DRAGON` such phenomena should not occur for prolonged durations: this is one reason for its greater performance.

5.3.3 Impact of Field Size

Field $\mathbb{F}_p$	p=2	p=23	p=17333	p=1078071557
Result	8302/200	8150/200	8127/200	8127/200

Table 2: Impact of field size

As indicated in section 3.2.1, the asymptotic performance is not related with field size. The table 2 shows the cost-per-broadcast E_{cost} for the reference graph in Fig. 2, generation size = 200, and `DRAGON`. It appears that indeed there is little variation (2%). To be complete, notice that because `DRAGON` attempts to equalize the rank gaps, it is countering the effect that made field size secondary in the queueing model of [3]; hence performance of `DRAGON` with smaller field size is expectedly different and slightly less efficient

5.3.4 Impact of Loss Rate

By nature, random linear network coding as the property of being resilient to losses, in the sense that one loss may be compensated by just one additional transmission. The table table 5.3.4 indicates the performance of `DRAGON`, in a

Loss Rate	0 %	10 %
Result	31.99	35.596

lossless network and in a network with loss equal to 10%. The result is again

the cost-per-broadcast E_{cost} (avg. number of transmissions to broadcast one source packet). As one can compute, in the case of the lossy network, the E_{cost} multiplied by the transmission success rate of the network (0.9) gives a value of 32.0364: this is normalization of the results in order to factor out the loss rate. Then the normalized result with loss is only 0.15% higher than without loss: indeed network coding and DRAGON are highly resilient to loss in this example.

5.3.5 Impact of Density

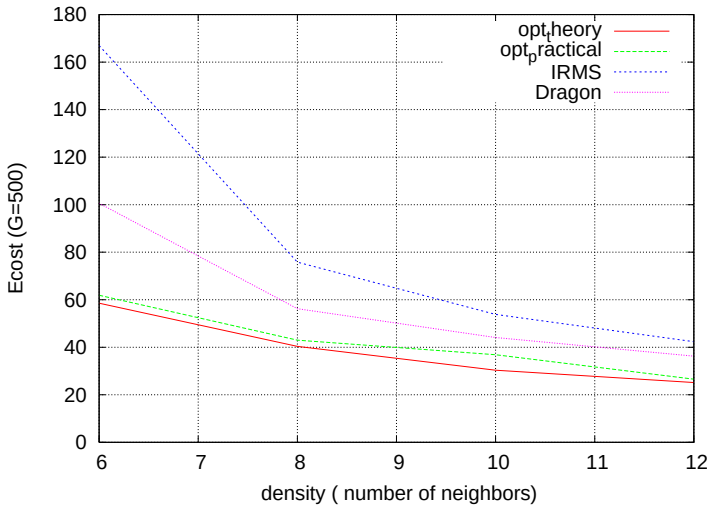

Figure 6: Increasing density

On Fig. 6, simulations were performed on random graphs (avg. of 6), with default parameters and with increasing radio range. The modified parameter is M, the average number of nodes in one disk representing radio range. As one may see, DRAGON performs well, comparatively to IRMS in sparser networks which are the most problematic cases, for reasons explained previously. For denser networks, the gap between IRMS, DRAGON and the optimal rate selection closes.

Fig. 7 represents the impact of generation size: the efficiency of the optimal rate selection (theoretical and with simulations), IRMS, and DRAGON is represented. As one might see, for the range of generation size selected, little variation is observed.

5.4 Closer Analysis of the Behavior of DRAGON

5.5 Impact of α

In DRAGON, one parameter of the adaptation is α, and is connected to the speed at which the rates adapt. The table I indicates the total number of transmissions made, for the reference graph Fig. 2, and for DRAGON with different values of α; and also for IRMS.

As one might see, first, the efficiency of IRMS on this network is rather low (1/4 of DRAGON): indeed, the topology exemplifies properties found in the cases

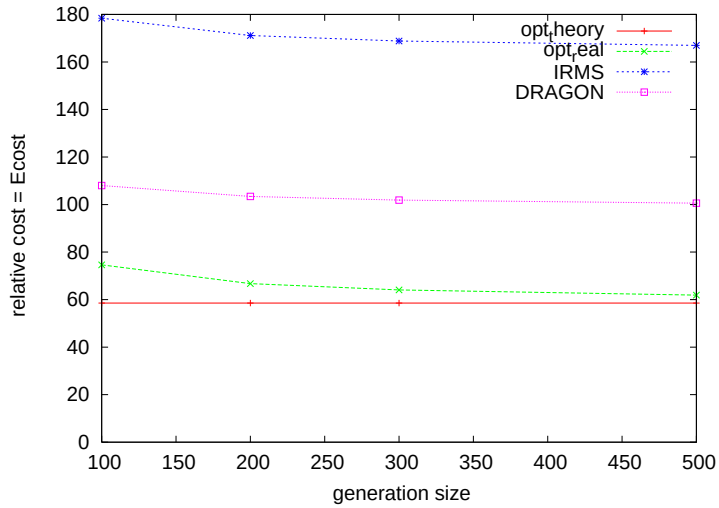

Figure 7: Impact of generation size

Value of α	$\alpha = 1$	$\alpha = 5$	$\alpha = 10$	IRMS
Total cost	16083	16272	17734	64411

Table 3:

of networks where IRMS was found to be less efficient: two parts connected by one unique node. For various choices of α, it appears that the performance of `DRAGON` decreases when α increases. This evidences the usual tradeoff between speed of adaptation and performance. However the decrease in performance is rather limited: we ascribe this fact to two factors. The first one, again, that for identical average rates, different rate selections will have similar performance, whether the rates are oscillating dramatically or not. The second one, is that in `DRAGON`, the nodes stop sending encoded packets, whenever they are unsure if its transmissions are beneficial to their neighbors.

5.5.1 Comparison with Model

On Fig. 8, some results are represented, when running `DRAGON`, with $\alpha = 10$ on the reference network in Fig. 2. Consider one node. At every time, it has a rank, and this rank can be compared with the number of packets already sent by the source: the difference between the two should be ideally 0, and a larger value is an indication of the delay in in propagation of the source packets. Hence that difference can be taken as a metric. We make the following statistics: for each node, we identify the maximum value of that difference over the entire simulation. We then plot one point on the graph: the x coordinate is the distance of the node to the source, whereas the y is this difference.

Therefore the graph indicates how the "gap of rank with the source" evolves with distance. First, we see that there is a large step near the middle of the graph: this is the effect of the center node, which is the bottleneck and which obviously induces further delay. Second, two linear parts are present on each side of the step: this confirms the intuitions given by the models in section 4.5.2,

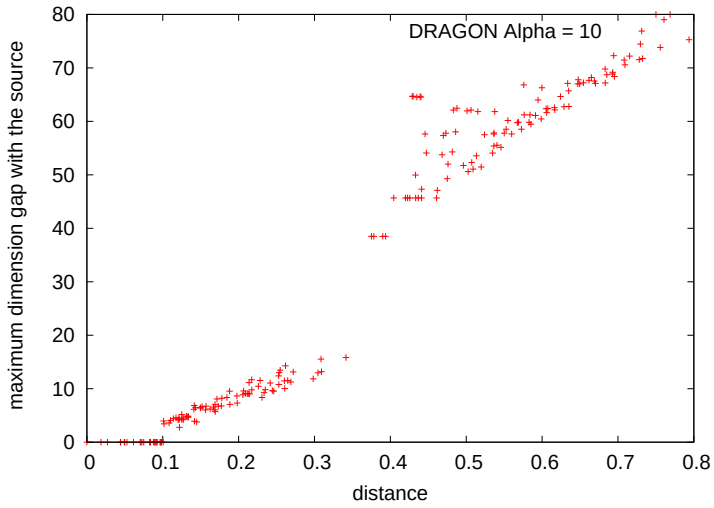

Figure 8: Maximum rank gap with the source

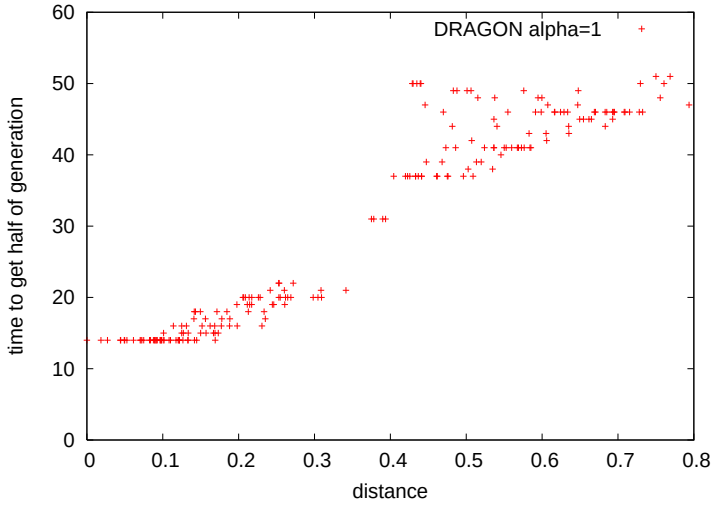

Figure 9: Propagation with distance $\alpha = 1$

about a linear decrease of the rank of the node from the source to the edge of the network.

Finally, one may find simulations results for $\alpha = 1$ on Fig. 9 and for $\alpha = 5, \alpha = 10$ on Fig. 10. As previously, one dot represents a node in the network, and the x coordinate is the distance of the node to the source ; but this time the y-coordinate is the time at which the node has received exactly half of the generation size. This yields further indication on the propagation of the coded packets from the source. Indeed if we compare the difference of time between nearest node from the source, and furthest node, we get a propagation time. It is around 35 for $\alpha = 1$, 6 for $\alpha = 5$ and 4 for $\alpha = 10$: roughly, it is inversely proportional to α, as expected from section 4.5.2. In addition, by comparing

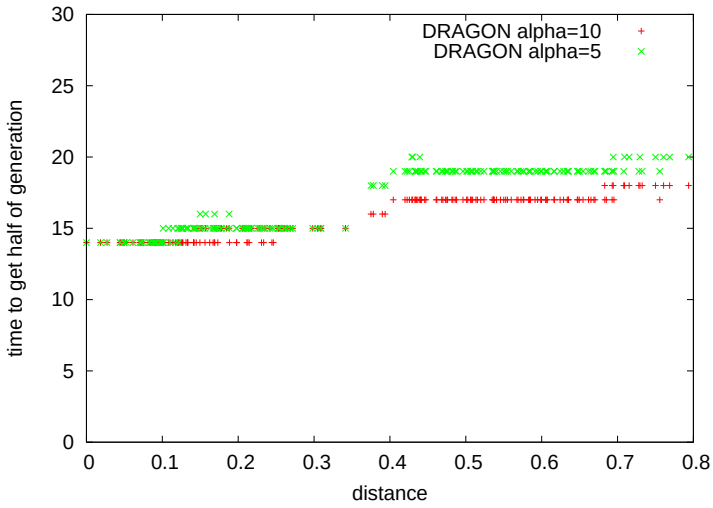

Figure 10: Propagation with distance $\alpha = 5, \alpha = 10$

Fig. 9 to Fig. 10: one sees that with higher values of α (greater reaction to gaps), the impact of the bottleneck in center node, is dramatically reduced. Note also that a smaller rank gap makes real-time decoding more likely.

6 Conclusion

We have introduced a simple heuristic for performing network coding in wireless multi-hop networks: `DRAGON`. It is based on the idea of selecting rates of each node, and this selection is dynamic. It operates as a feedback control, whose target is to equalize the amount of information in neighbor nodes, and hence indirectly in the network. The properties of efficiency of `DRAGON` are inherited from static algorithms, which are constructed with a similar logic. Experimental results have shown the excellent performance of the heuristics. Further work includes addition of congestion control methods.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, *"Network Information Flow"*, IEEE Trans. on Information Theory, vol. 46, no.4, Jul. 2000

[2] T. Ho, R. Koetter, M. Médard, D. Karger and M. Effros, *"The Benefits of Coding over Routing in a Randomized Setting"*, International Symposium on Information Theory (ISIT 2003), Jun. 2003

[3] D. S. Lun, M. Médard, R. Koetter, and M. Effros, *"On coding for reliable communication over packet networks"*, Technical Report #2741, MIT LIDS, Jan. 2007

[4] Z. Li, B. Li, D. Jiang, L. C. Lau, *"On Achieving Optimal Throughput with Network Coding"* Proc. INFOCOM 2005.

[5] Y. Wu, P. A. Chou, and S.-Y. Kung, *"Minimum-Energy Multicast in Mobile Ad Hoc Networks using Network Coding"*, IEEE Trans. Commun., vol. 53, no. 11, pp. 1906-1918, Nov. 2005

[6] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, *"Minimum-Cost Multicast over Coded Packet Networks"*, IEEE/ACM Trans. Netw., vol. 52, no. 6, Jun. 2006

[7] C. Adjih, S. Y. Cho and P. Jacquet, *"Near Optimal Broadcast with Network Coding in Large Sensor Networks"*, 1^{st} Workshop Information Theory for Sensor Networks, Sante Fe, Jun. 2007 (WITS'07).

[8] S. Y. Cho, C. Adjih and P. Jacquet , *"Heuristics for Network Coding in Wireless Networks"*, Proc. International Wireless Internet Conference (WICON 2007), Texas, USA, October, 2007, Accepted.

[9] S.-Y. R. Li, R. W. Yeung, and N. Cai. "Linear network coding". IEEE Transactions on Information Theory , Februray, 2003

[10] R. Koetter, M. Medard, *"An algebraic approach to network coding"*, IEEE/ACM Transactions on Networking, Volume 11, Issue 5, Oct. 2003

[11] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, *"A Network Coding Approach to Energy Efficient Broadcasting"*, INFOCOM 2006

[12] P. A. Chou, Y. W, and K. Jain, *"Practical Network Coding"*, Forty-third Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, October 2003

[13] A. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, *"Capacity of Wireless Erasure Networks"*, IEEE Trans. on Information Theory, vol. 52, no.3, pp. 789-804, Mar. 2006

[14] D. S. Lun, M. Médard, R. Koetter, and M. Effros, *"Further Results on Coding for Reliable Communication over Packet Networks"* International Symposium on Information Theory (ISIT 2005), Sept. 2005

[15] B. Clark, C. Colbourn, and D. Johnson, *"Unit disk graphs"*, Discrete Mathematics, Vol. 86, Issues 1-3, Dec. 1990

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

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