



**HAL**  
open science

## When Network Coding improves the Performances of Clustered Wireless Networks

Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier, Gilles Roussel

► **To cite this version:**

Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier, Gilles Roussel. When Network Coding improves the Performances of Clustered Wireless Networks. WD'11, 2011, Canada. pp.1-6, 10.1109/WD.2011.6098153 . hal-00794865

**HAL Id: hal-00794865**

**<https://hal.science/hal-00794865>**

Submitted on 7 Mar 2013

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

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

# When Network Coding improves the Performances of Clustered Wireless Networks

Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier, Gilles Roussel  
Université Paris-Est, Laboratoire d'Informatique Gaspard Monge – LIGM  
Marne-La-Vallée, France  
{Isalhi, Ghamri, Lohier, Roussel}@univ-mlv.fr

**Abstract**—This paper introduces a network coding scheme that significantly increases the performances of clustering algorithms in wireless multi-hop networks. Knowing the promising capabilities of network coding over broadcast mediums, we propose a simple and non-invasive packet scheduling mechanism based on proactive code selection and overheard messages to optimize intra and inter-cluster communications. Our solution uses bitwise XOR operations at the packet level to combine several messages in one transmission. Simulations show an important gain in the network reliability, and a major enhancement of the throughput without any substantial computation or network overhead.

**Keywords**—XOR coding; wireless multi-hop networks; clustering

## I. INTRODUCTION

Partitioning nodes into a set of disjoint groups (clustering) is a well-established approach to attain scalability, connectivity and to improve communications of large-scale wireless multi-hop networks. Indeed, several algorithms have been proposed to organize wireless networks into a connected hierarchy of clusters, whether for mesh networks [1]-[2]; wireless sensor networks [3]-[4] or even vehicular ad hoc networks [5]-[6]. In such solutions, every cluster contains a super node called *cluster-head* that is in charge of coordinating communications with its cluster members. The communication between clusters is assured by a backbone formed exclusively by cluster-heads. Clustering is very convenient for load balancing and data aggregation; it reduces channel contention and collisions and prolongs the network lifetime in energy-constrained networks.

As clustered solutions offer more robustness, scalability and stability to wireless networks, they also systematically imply a sub-optimality of the achievable throughput, since partitioning a network necessarily suggests a reduction of its bandwidth. In the literature, clusters are abstractly separated according to many criteria such as mobility, topology or energy consumption. But in fact, they are often physically overlapped, and nodes may frequently receive messages from nodes that are not in the same cluster, via what we designate as *overhearing links* (cf. Fig.1). Hence, we propose to exploit such redundancy to enhance clustered wireless networks. Indeed, our objective is to keep taking full advantage of the benefits of clustering, but avoiding the loss of bandwidth due to the logical separation of clusters. Our idea is to use network coding to mix forwarded packets with overheard ones from other clusters. By doing so, we strongly believe that the performances of clustered wireless networks will be significantly enhanced in terms of throughput, latency and reliability.

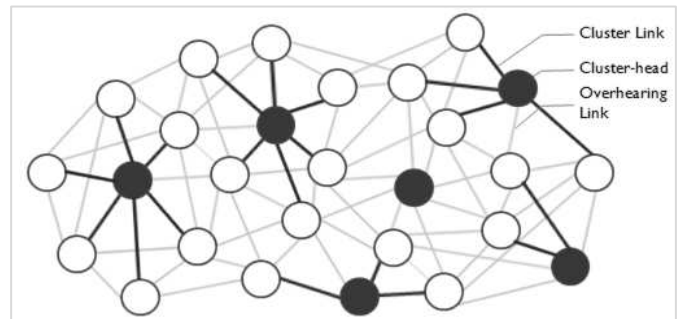


Figure 1 - Overhearing in clustered wireless networks

This paper describes more precisely CNC (Clustered Network Coding) a new coding scheme based on topology inferring that allows the utilization of one-hop coding to encode different flows of packets from adjacent clusters via the exploitation of overhearing links. We evaluate CNC (Clustered Network-Coding), our new coding scheme for clustered wireless multi-hop networks through a detailed performance analysis considering different data traffic patterns and network densities. Our empirical study shows an important improvement of the global network throughput under high load and a significant gain when the medium becomes unreliable.

The paper is organized as follows. Section II details the wireless network coding background, while Section III discusses related work of clustering algorithms for wireless multi-hop networks. Section IV describes the system model. In section V we define CNC our coding scheme for clustered wireless networks, and Section VI presents our system performances. Section VII concludes the paper.

## II. BACKGROUND

Recent advances in the applications of the Network Coding theory introduced by Ahlswede *et al.* in [14] are refining some of the conventional techniques in communication networks. For example, authors of [15] propose a new multicast model based on network coding that overtakes the classical *store-and-forward* transportation pattern by encoding several incoming packet-flows using linear combinations at intermediate nodes. Many protocols such as, Avalanche [17] or CodeTorrent [15] have used random linear coding operations to improve forwarding and dissemination schemes over communication networks, by dividing streams of data and encoding them before transmissions. Joint work of both the wireless network and the network coding communities shows in [16] the prospective benefits of using simple coding operations, such as XOR, at

packet level to increase the performances of wireless networks. In all these cases, the network features a significant gain in terms of reliability, delay and throughput.

The key challenge in contemporary network coding resides no more in the way nodes encode packets. Indeed, since COPE, simple XOR operations have been preferred to complex linear random combination of packets that involves several constraints on the data traffic and the nodes' capabilities. Yet, important issues still characterize network coded architectures. The first concerns the "topology inference" or *how to perform optimal coding decisions which imply a successful decoding process at all the destined nodes?* And the second is rather about the reliability of communications when the network endures high loss rates. In this paper, we answer these question via an adaptation of one-hop coding, a mechanism introduced in [7]. Our new coding scheme is based on a lightweight local topology inference that implies less overhead and a simple yet effective packet coding/reception acknowledgement that ensures reliable network coding under lossy networks.

### III. RELATED WORK

Clustering aims at building an overlay backbone over a wireless network that splits nodes into a set of clusters. This architecture has been widely used for large-scale wireless networks to build a virtual hierarchy to support many data-transportation schemes such as routing, data dissemination, distributed storage and data aggregation. Because wireless networks are naturally subject to intermittent connectivity due to channel disruptions and nodes mobility, there is a large body of work that focuses on clusters stability and convergence time within dynamic networks. Authors of [8] used network clustering as a routing scheme for mobile ad hoc networks by proposing C-OLSR a clustered protocol that provides a higher achievable throughput with less overhead generation than classic ad hoc protocols. In [2], Chatterjee *et al.* propose WCA, a weight based distributed clustering algorithm that helps to achieve a better node "reachability" and route stability. Another example is ZigBee [9], a standardized hierarchy-based routing protocol proposed by the ZigBee Alliance which is based on IEEE 802.15.4 [10], and is widely employed for WSN (Wireless sensor Networks) to provide robustness and to reduce energy consumption. In wireless sensor networks, clustering mechanisms are also extensively studied whether to support data-aggregation, to guarantee energy efficiency or to enhance communications. Abbasi *et al.* provided a comprehensive survey [11] on existing clustering algorithms for WSNs. In LEACH [12] for instance, clustering is combined with sleep/awake mechanisms to extend the network lifetime, while in [13] the authors use both residual energy and communication costs to build clusters. Of course, further work in this field could have been mentioned, but to the best of our knowledge, none of the existing solutions has used network coding to improve the performances of clustered networks.

Throughout this work, we describe a new coding scheme capable of improving multi-hop networks by exploiting redundant links inherent to clustered architectures. We focus on the fact that none of the existing clustering solutions take full advantage of such "dead" links between overlapping clusters. Thanks to broadcast nature of the wireless medium, we propose to exploit overheard messages, with a single-hop coding mech-

anism to reduce the channel overhead and to improve the global throughput. To do so, we assume nodes perform XOR operations at packet level, and maintain an *overhearing queue* where all *out*-cluster messages are stored. Before packet transmission, nodes perform a coding operation by XORing the outgoing packet with overheard packets. Unlike in [7], instead of XORing together arbitrarily head-packets of the transmission queue, optimal coding operations are applied to the next packet to be transmitted with packets coming from other clusters thanks to our local topology inference mechanism.

### IV. SYSTEM OVERVIEW

The major aim of an efficient clustering scheme is to attain stability and maximize the clusters lifetime with low overhead and minimized convergence time. In that aim, algorithms often include synchronous or asynchronous self-organization components to cope with the inherent dynamicity of wireless multi-hop networks. During data transmission, the overlay backbone may behave differently depending on the characteristics of the network. For example, in the case of mobile nodes the maintenance of the hierarchy requires more control messages and a shorter cluster lifetime than in a static network. An important hypothesis we rely on, is the fact that the clustering scheme uses message-based topology maintenance<sup>1</sup> and passes through at least three phases: cluster head election, cluster formation and the data transmission; which is the case in every clustering algorithms we have found in the literature.

As we focus on clustering algorithms in general, common properties are to be defined in order to meet any configuration of the network. Table 1 summarizes different attributes that can characterize most of clustering schemes. We assume each node has eavesdropping capabilities<sup>2</sup> i.e., is able to receive from its neighborhood unicast packets that are not addressed to him, which is a realistic assumption since the wireless radio channel is by definition a shared medium. We also assume that nodes are capable of performing bit-wise XOR operations at the packet level to use one-hop coding.

TABLE I – SYSTEM TERMINOLOGY

Attribute	Description
Cluster Head - CH	Node in charge of inter-cluster communications
CH election phase	CH are identified at the end of this phase using a specific clustering algorithm
Cluster formation phase	At the end of this phase each node belongs to one cluster and is either CH or registered with at least one CH
Communication phase	The backbone is built and nodes transmit their data during this phase,
Convergence time	Duration in which the backbone is functional
Signalization packets	Control packets transmitted during the cluster formation and maintenance phases
Overhearing links	Links that are not part of the backbone

<sup>1</sup> Topology maintenance in our context consists on exchanging short hello messages containing the sender's ID (among possible other information) in order to build and maintain the clusters.

<sup>2</sup> Eavesdropping can be implemented in a real wireless network by activating promiscuous mode at intermediate nodes. [7] details this procedure for the 802.11 MAC. Similar feature is available in 802.15.4 networks, the dominant standard for WSNs.

### A. System Description

We consider a wireless multi-hop network of  $N$  nodes organized into  $M$  distinct clusters within a square area  $A$ . Each cluster contains  $m$  special nodes called cluster-heads that are elected at the beginning of the construction phase. After this election, nodes that are not CHs have to set up their membership (choose which cluster they belong to) in order to start data transmission. We define  $T_{cvg}, T_{elc}$  and  $T_{frm}$  such as ( $T_{cvg} = T_{elc} + T_{frm}$ ).  $T_{cvg}$  is the convergence time of the algorithm; it includes the cluster head election time ( $T_{elc}$ ) and the cluster formation duration ( $T_{frm}$ ). For design simplicity, we assume nodes have the same transmission range  $r$ . We suppose the localization of the active nodes to be uniform. We denote by  $C_j = \{n_j\}$  the set of nodes  $n_j$  that are members of the same cluster  $C_i$  with  $i \in \{1, 2, \dots, M\}$  and  $j \in \{1, 2, \dots, N\}$ . Note that a node can be member of only one cluster (cf. equation 1).

$$\forall j, k \in M, \quad j \neq k \Rightarrow C_j \cap C_k = \emptyset \quad (1)$$

#### 1) Cluster head election phase

In this step of duration  $T_{elc}$ , every node has to determine whether it is a cluster head or not. The cluster head election phase can be *ID-based*, *Probabilistic* or *Weight-based* as each system can be suitable for a certain type of network. For example, ID-based techniques are not proper for energy constrained networks (e.g., WSNs) as they penalize nodes regardless of their energy consumption, but can be very efficient in case of high mobility where CH must be elected rapidly. In this work, we consider a system where the three classes are represented. We do so by using a combination of the parameters that can describe each one of the above-mentioned classes.

#### 2) Cluster formation phase

Once CH's are elected, the fastest way to inform nodes in the neighborhood is to broadcast a short control message (ADV) announcing that  $n_i (\forall i \in M)$  is a CH. Each non-CH node then sends an acknowledgment message using any medium access control protocol to indicate which cluster it belongs to. Note that some extra control messages might be sent between the CH and the cluster members in order to organize the data communication phase (TDMA scheduling, radio channel occupation, encryption parameters, etc.). Our system assumes that after  $T_{cvg}$  the  $M$  clusters are completely constructed.

#### 3) Data transmission phase

This phase allows effective data packets to be transmitted either in an end-to-end, dissemination or aggregated way. The information is sent from a node to another depending on its position in the hierarchy. For instance, communication between two nodes of the same cluster takes at most two hops (if source and destination are not in the same transmission range), otherwise, one-hop transmission is performed. In Inter-Cluster Communications, the backbone formed by the CHs allows nodes from different clusters to communicate via inter-cluster routes. In this case, the source transmits its packets to its CH; the latter will then choose a multi-hop path above the backbone to relay it to the destination. Note that, as shown in Fig. 2, a Hierarchy Maintenance phase might be necessary, where each CH broadcasts control packets to inform its members that it is still available and to maintain the overlay backbone of CHs.

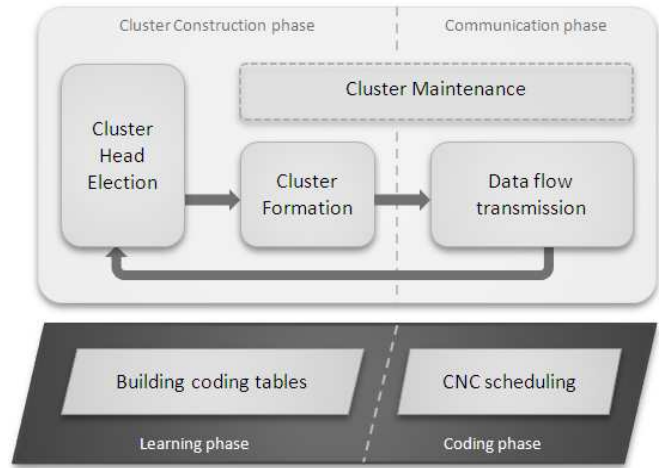


Figure 2 – CNC overview

## V. CLUSTERED NETWORK CODING

### A. CNC Scheduling System

As shown in Fig. 2, our scheme is twofold. First, it takes advantage of the exchanged control messages during the cluster construction phase in order to infer the local topology of the node and building its coding table. Then, it uses this coding table in order to perform the most optimal coding decisions. To do so, intermediate nodes perform XOR coding operations before data transmissions by combining eavesdropped packets with the head of the transmission queue, taking into account that the next-hop(s) node has to be able to decode the outgoing coded packet.

Fig. 3 depicts this scheduling system, and shows that it requires an overhearing queue ( $Q_{ovh}$ ) in addition of the input queue ( $Q_{in}$ ). While  $Q_{ovh}$  contains only eavesdropped packets,  $Q_{in}$  can include packets from other nodes or native-packets generated by the node itself. Packets in  $Q_{ovh}$  are then stored in a memory ( $Mem$ ) during a period of time where they are used to encode/decode packets. The coding strategy module ( $CSM$ ) decides from the set of available packets in  $Q_{ovh}, Q_{in}$  and  $Mem$  and depending on the nature of data delivery (dissemination or routing) which packets to include into the outgoing coded packet. Note that when no coding can be performed, the packet scheduling scheme does not hold the packet and no noteworthy delay is applied.

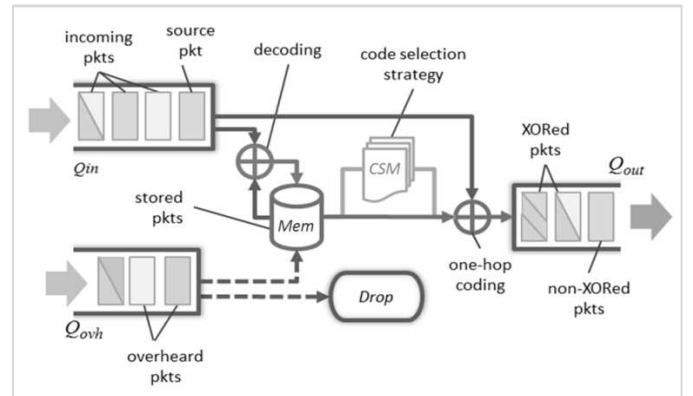


Figure 3 – Queuing system overview

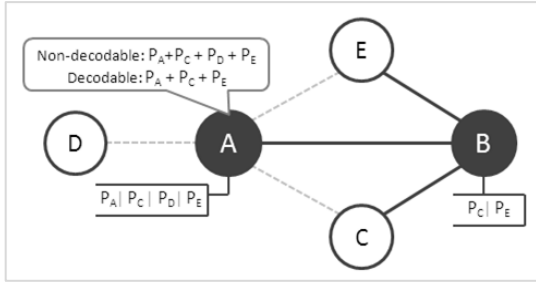


Figure 4 – One-hop coding illustration

## B. CNC Reliable Network Coding

### 1) Topology Inference & Packet Encoding

In [7], the problem of topology inference is partially solved (not without a tradeoff) thanks to the exchange of reception reports at regular intervals of time. Indeed, in order to perform a successful coded transmission, one has to be aware of what the next hop(s) has already received and overheard in order to include or not a packet in the code (cf. Fig. 4). The problem is partially solved, because in case of non-negligible loss rate, reception reports are lost and coding failures increase. The tradeoff of such a solution is obviously the large overhead induced by control messages used in reception reports.

To provide a less message-greedy answer to the question of *who got what?* CNC exploits in its learning phase (cf. Fig. 2) the control messages already used to build and maintain the clustered topology of the network in order to locally build similar neighboring tables to COPE. Using these messages nodes can list a set of overhearing links of the neighboring nodes and thus know which packet they are disposed to decode or not. For instance, consider the device *A* in Figure 4. At the end of the cluster formation phase, *A* knows that he can overhear packets transmitted from the routers *B, C, D* and *E* thanks to the control messages that have been exchanged since the initialization of the network. Node *A* will then broadcast to its neighborhood a control packet containing the identifiers of nodes that it overhears. Note that this operation takes place only after the cluster formation phase and at the end of the maintenance phase, and thus does not imply as much overhead on the network traffic as in COPE. The atomic operations of coding and decoding packets remain the same as in one-hop coding, i.e., when a set of  $N$  packets is coded, the node XORs them together and broadcasts the resulting packet. Decoding consists on XORing  $(N - 1)$  received/overheard packets with the incoming coded packets in order to retrieve the original packet.

### 2) Network Coding Reliability

The reliability in CNC is also improved compared to [7] where acknowledgement mechanisms are not adapted to provide reliable coded communications. Consider the case where the network endures a non-negligible packet-loss rate. Coded packets may be received and following to classic one-hop coding algorithm, they are acknowledged. Though, these packets may not be decodable because of a bad topology inference that led to a non-decodable packets combination. This type of decoding failures may cause dead-end paths (routes that do not reach their destinations) for end-to-end communications when the medium is unreliable or endures high packet loss rate. Moreover, COPE disables coding operation whenever the loss

rate exceeds 20%, which is often the case in multi-hop wireless networks as their density increases. This implies a sub-optimality in the bandwidth utilization. In CNC, we cope with this issue by adding a feedback field in coded packets to inform neighboring nodes about the next expected packet using the EPI (Expected Packet Id) field. For instance, if the next expected packet is  $p_i$  EPI will contain the sequence number of  $p_i$  so that the previous hop will be aware that  $p_i$  is not received yet at its next hop. This helps to perform a better topology inference, enhancing the efficiency of the selected codes. Algorithm 1 details more precisely the reception process. Lines 4, 18 and 21 show that our ACK system ensures reactivity to packet losses thanks to the EPI field when an intermediate node decodes a packet (if it is received or overheard). As a result, the coding node can know when a packet has been lost by checking the EPI field. This latter will also let nodes detect which packets could not be decoded by which node, allowing retransmissions or recoding solving *dead-end* paths issue.

---

### Algorithm 1 Packet Reception at Node *A*

---

```

1: for each incoming packet  $p$  in  $Q_{in}(A)$  do
2:   if  $p$  is non-coded then
3:     if  $A$  is dest. of  $p$  and is not the final hop then
4:        $A$  adds  $p$  to its next EPI field
5:     else
6:       if  $A$  is dest. of  $p$  and is the final hop then
7:          $A$  sends an ACK to the sender
8:       else /*  $A$  overheard  $p$  */
9:          $p$  is stored in  $Q_{ovh}(A)$ 
10:      end if
11:    else /*  $p$  is coded */
12:      if  $p$  is decodable then
13:         $p_{dec} = Decode(p)$ 
14:        if  $A$  is dest. of  $p_{dec}$  and is the final hop then
15:           $A$  sends an ACK to the sender
16:        else
17:          if  $A$  is dest. of  $p_{dec}$  and is not the final hop then
18:             $A$  adds  $p_{dec}$  to its next EPI field
19:          else /*  $A$  overheard  $p$  */
20:             $p_{dec}$  is stored in  $Q_{ovh}(A)$ 
21:             $A$  adds  $p_{dec}$  to its next EPI field
22:          end if
23:        else /*  $p$  is not decodable */
24:           $p$  is dropped from  $Q_{in}(A)$ 
25:        end if
26:      end if
27:    end for

```

---

## C. CNC Code Selection

### 1) CNC for Unicast

As shown in Fig. 5, in unicast scenarios, where data packets are sent from one source CH to another via an inter-cluster route, the coding selection scheme remains the same as in one-hop coding, i.e., intermediate node in the route performs XOR coding by selecting the best side-packet(s) to combine with the head of the transmission queue. This allows the next hop of the route to retrieve its destined packet while allowing neighboring nodes to retrieve side-packets. In the example depicted in Fig. 4, *A* would have sent  $p' = p_a \oplus p_b$  allowing the next-hop

$B$  and  $D$  to decode  $p_a$  which obviously outperforms store & forward schemes. Note that while the code selection remains the same as in one-hop coding, reliability and topology inference mechanisms are applied following Section V.B. This changes the behavior of the algorithm when the radio channel becomes unreliable. For instance, when the loss rate is high and coded packets cannot be retrieved because of missing side packets, the intermediate node add in the EPI field of its forwarded packet the identifier of the missing packet which will be recoded using a new set of side packets and retransmitted.

## 2) CNC for Dissemination

Clustering is a convenient solution for dissemination-based networks. It ensures scalability and reduces the broadcast domain of relaying nodes, avoids *broadcast storms* and provides satisfying performances, while keeping a low traffic load. Still, in such networks, the bandwidth utilization can be strongly underexploited. In fact, while propagating data among the hierarchy and since clusters are often overlapped, an important fraction of data-traffic overheard by neighboring nodes is dropped at lower levels because unicast communications imply so. With CNC this non-exploited packets are combined and transmitted by intermediate nodes to their neighbors allowing to a maximum of them to extract a maximum number of packets. Consider the following canonical example of an inter-cluster communication between four neighboring cluster-heads  $A, B, C$  and  $D$  with the following queues configuration:

$$Q_{in}(A) = \{p_a, p_b, p_c, p_d\}; Q_{ovh}(B) = \{p_b, p_d\}$$

$$Q_{ovh}(C) = \{p_a, p_b\}; Q_{ovh}(D) = \{p_a, p_c\}$$

Where  $p_a, p_b, p_c, p_d$  are packets that have to be propagated to nodes  $A, B, C$  and  $D$ . Disseminating packets means sending in the fastest possible way a maximum number of packets to a maximum number of nodes. As detailed in Fig. 5, whenever a node  $i$  receives a new packet  $p$ , if  $p$  is not unicast, one-hop coding is not performed. Instead, the node checks in its coding table for packets that can be decoded by a maximum of its neighbors, then encode them with side-packets stored in  $Mem$ . In Fig. 4 for instance, it is far more efficient for  $A$  to transmit a coded packet  $p$ , with  $p = p_c \oplus p_b$ , rather than a classic store and forward solution where  $A$  just sends  $p_a$ , the head of its transmission queue. Thanks to CNC all destined neighbors can retrieve a packet, whether it is  $B$  and  $C$  that decode  $p_c$  or  $D$  that decodes  $p_b$ . While with sending  $p_a$  only  $B$  receives an innovative packet, i.e., a packet that they haven't already received or overheard. The careful reader may notice that COPE coding scheme does not perform better than CNC because it does not take into account the nature of the data traffic. As a matter of fact, if we apply COPE code selection algorithm to the previous example,  $A$  would have sent  $p'$  (with  $p' = p_a \oplus p_b$ ) wherein only  $B$  and  $D$  can retrieve innovative data from  $p'$ .

## VI. PERFORMANCE EVALUATION

In this section, we evaluate of CNC using different traffic patterns and with different network densities. We process the throughput of the network, the packet overhead and we show that our coding scheme does not affect the convergence time of clustering algorithms. We compare the results with a no-coding scenario and with a COPE-based clustered network. We use the Qualnet 4.5 simulation environment [18] to assess our solution.

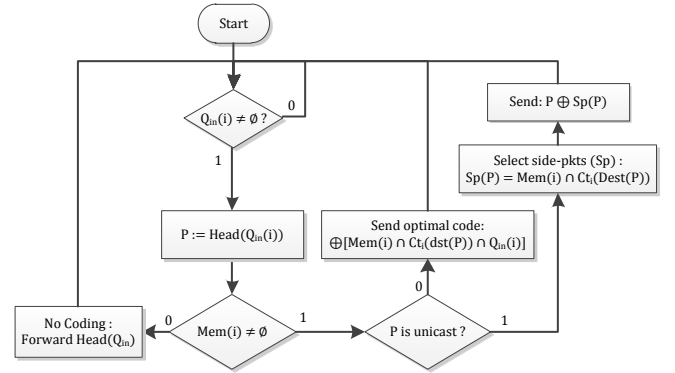


Figure 5 - Code Selection Scheme

We chose the IEEE 802.11b protocol for wireless communications. We consider two scenarios: CBR /UDP for dissemination and FTP/TCP for unicast. In this latter case, the system triggers randomly a number of unicast communications between two arbitrary nodes within the 10 first seconds of simulation. Table 2 show the simulation attributes:

TABLE II – SIMULATION SETTINGS

Simulation time	1000 s
Simulation Area	1000 x 1000   $10^3 \times 10^3$
Number of seeds	15
Number of Nodes	50
Routing protocol	Static
FTP Generation Size	15 packets
CBR data rate	10 000 packet/s
Reflection model	Two-ray ground
Propagation model	Lognormal shadowing
Channel Capacity	1 Mbps
Buffer Size	100 packets
Packet Size	500B

### A. Simulation Results

*FTP/TCP traffic.* Fig. 6, represents simulation results for FTP traffic in terms of total transport-level throughput, control packet overhead (assuming the same loss rate at all links) and data delivery ratio. Fig. 6(a) shows the results for end-to-end communications. At low traffic (2-4 TCP flows) there is only a moderate gain from both **COPE** and **CNC** compared to the **no-coding** scenario. The reason resides in the fact that network coding opportunities are less frequent compared to a more loaded network which is confirmed as the number of TCP flows increases. The results also show a slight improvement of the throughput, where CNC is higher compared to COPE. This is because COPE spends a considerable part of the available bandwidth exchanging reception reports in order to perform efficient coding decision. Fig. 6(b) plots the overhead growth of control packets to maintain the clustered hierarchy of the network, in addition to the overhead induced by network coding control messages. It represents the fraction of the bandwidth used up by control packets and extra control fields in the 802.11 frame. As predicted, the results clearly indicates the low impact in terms of overhead when using CNC compared to COPE, since our scheme is less invasive and thanks to our local topology inference system that takes advantage of the control messages used in clusters formation and maintenance. Fig.

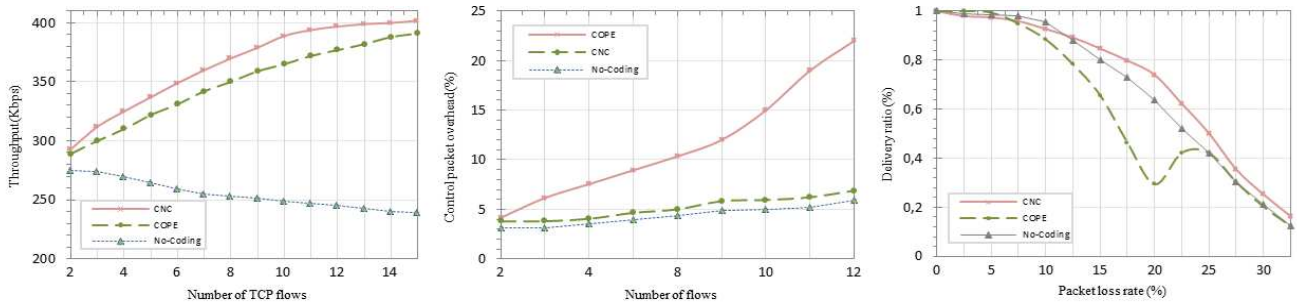


Figure 6 – CNC vs. COPE vs. No-coding: Throughput, Overhead and Delivery Ratio Analysis

6(c), meanwhile, shows that COPE's performances decrease significantly as the network endures a high loss rate. Once the loss rate exceeds 20%, COPE disables all network coding operations which lead to the same ratio as with no coding at all. CNC, however, keeps the advantage of using network coding at low-medium loss rate, then provides the same performances as with no coding at all, thanks to the EPI field, intermediate nodes cannot find any optimal coding opportunity when the loss rate is too important, which leads to performances close to a classic store and forward scenario. Fig. 7 indicates the average delivery ratio for CBR dissemination-based communications (assuming 10% loss rate at all links). It shows that, with its specific code selection strategy and its reliability mechanisms, CNC outperforms COPE in terms of data propagation speed, this is due to the fact that CNC scheduling scheme performs more efficient packet combinations compared to COPE. Obviously, CNC and COPE perform faster data dissemination compared to no coding, since they both enhance the bandwidth utilization of the network.

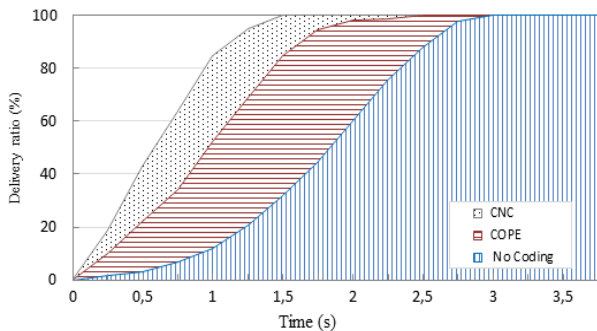


Figure 7 - Delivery ratio versus time (convergence time)

## VII. CONCLUSION

In this work, our objective is to keep taking full advantage of the benefits of clustering in wireless multi-hop networks, but avoiding the loss of bandwidth due to the logical separation between nodes of different clusters. Our idea is to use network coding to mix packets with eavesdropped packets from other clusters to enhance the performances of clustered networks. Our solution, CNC relies on exchanged packets used during the cluster formation phases in order to infer the local topology of intermediate nodes. They can then perform more efficient XOR coding decisions for unicast and dissemination-based communications and ensure a better bandwidth utilization of the network. Furthermore, we extend the concept of one-hop coding using reliability mechanisms to allow more efficient coding decisions in case of erratic environments. Performance evalua-

tion results show that CNC outperforms one-hop coding (COPE) in terms of throughput, control overhead and convergence time especially as the number of flows increases. Besides, when the network is unreliable, CNC offers the best data delivery rate compared to coded and non-coded schemes. Our future work focuses on the deployment of CNC on a real IEEE 802.11 network testbed.

## REFERENCES

- [1] S. Banerjee, S. Khuller, A Clustering Scheme for Hierarchical Control in Multi-Hop Wireless Networks, Proc. of IEEE INFOCOM, Apr. 2001.
- [2] M. Chatterjee, S. Das, D. Turgut, WCA: Weighted Clustering Algorithm for mobile Ad Hoc networks, Cluster Computing (2002) 193–204.
- [3] K. Sohrabi et al., Protocols for self-organization of a wireless sensor network, IEEE Personal Communications 7 (5) (2000) 16–27.
- [4] G. Gupta, M. Younis, Load-balanced clustering in wireless sensor networks, in: Proceedings of the International Conference on Communication (ICC 2003), Anchorage, Alaska, May 2003
- [5] H.-J. Reumerman, M. Roggero, M. Ruffini, The Application-based Clustering Concept and Requirements for Intervehicle Networks, IEEEcommunication Magazine, April 2005.
- [6] H.Su, X. Zhang, Clustering-Based Multichannel MAC Protocols for QoS Provisionings Over Vehicular Ad Hoc Networks, IEEE Transactions on Vehicular Technology, Vol. 56, No. 6, November 2007.
- [7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, J. Crowcroft, XORs in the air: Practical network coding, in ACM SIGCOMM, Sept. 2006.
- [8] Francisco J. Ros, Pedro M. Ruiz Cluster-based OLSR extensions to reduce control overhead in mobile ad hoc networks, IWCMC '07.
- [9] ZigBee Alliance, "ZigBee Specifications". Dec. 2006.
- [10] IEEE Standard for Information Technology, "Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)". 2003.
- [11] Abbasi A A, Younis M. A survey on clustering algorithms for wireless sensor networks. Computer Communications, 2007, 30: 2826-2841.
- [12] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-Efficient Communication Protocol for Wireless Sensor Networks," Proceedings of the 33th Hawaii International Conference on System Sciences, 2000.
- [13] O. Younis, S. Fahmy, HEED: A Hybrid Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks, IEEE Transactions on Mobile Computing, vol. 3, no. 4, Oct-Dec 2004.
- [14] R. Ahlswede, L. Ning Cai, R. W. Yeung, Network Information Flow, in IEEE Information Theory. July 2000.
- [15] J. Park, D. Lun, Y. Yi, M. Gerla, M. Medard. A network coding-based adhoc multicast protocol, in IEEE Wireless Communications, Feb. 2006.
- [16] T. Ho, R. Koetter, M. Medard, D. R. Karger, M. Effros, The benefits of coding over routing in a randomized setting, in IEEE ISIT, June 2003.
- [17] C. Gkantsidis P.R. Rodriguez, Network coding for large scale content distribution, in Proc. of IEEE INFOCOM. Aug. 2005.
- [18] <http://www.scalable-networks.com>.