Wireless Network Coding with Improved Opportunistic Listening

Antonios Argyriou, *Member, IEEE*

***Abstract*—Network coding has been shown to be a very promising technique for increasing the throughput in wireless ad hoc networks. In this paper, we set to define simple extensions for distributed medium access control (MAC) protocols that optimize the transmission of network-coded packets independently of the actual network coding algorithm. The main characteristic of the proposed protocol is that it improves the efficiency of coding decisions and allows verifying the decodability of packets before they are transmitted. The aforementioned goals are achieved first by adopting minor extensions to the channel access scheme, and second by introducing a new algorithm that manages intelligently the data packets that are stored at the MAC queue of each network node. The later algorithm improves the knowledge of a node regarding the available correct coding opportunities by the use of opportunistic acknowledgments, and by maintaining virtual buffers for the overheard data packets. For the proposed protocol, we also develop an analytical performance model that is used for evaluating its performance in conjunction with simulations. Extensive simulations are presented for several ad hoc network topologies that test all the features of the proposed protocol. Significant throughput improvement can be observed when our protocol is compared with network coding schemes that do not exploit the full potential of opportunistic listening.**

***Index Terms*—Network coding, wireless ad hoc networks, MAC protocol, opportunistic listening, carrier sensing, IEEE 802.11.**

I. Introduction

RECENTLY, network coding has emerged as an alternative concept for designing packet networks [1]. Network coding suggests that packet networks should not only route packets between hosts, but they should also intelligently process/code packets in order to approach multicast capacity [2]. This basic principle has initiated significant research efforts and the first results indicate that network coding is a versatile technique that can be applied in several domains [3]. The most important benefits that can be obtained by employing network coding include higher throughput and reliability. In the case of wireless networks, network coding can also be combined with wireless broadcast to increase the information content per transmission. Fig. 1 demonstrates the basic idea of wireless network coding. In this simple topology, nodes N_4 and N_5 employ opportunistic listening for overhearing the transmissions of packets b and a respectively. This means that if N_2 broadcasts a linear combination of packets a and b (e.g. $a \oplus b$), this single coded packet is enough for recovering the desired packets a and b at N_4 and N_5 respectively. Without network coding and broadcasting, four transmissions would be required instead of three. This idea has been applied in 802.11-based multi-hop wireless networks and throughput improvements by a factor of more than 3 have been demonstrated for elastic data transfers [4], [5]. Because of the versatility of network coding, it has been shown that it can be applied at different layers of the protocol stack [6], [7]. Therefore, it is necessary that the significant performance gains that network coding can offer are properly evaluated when future protocols for distributed wireless networks are designed.

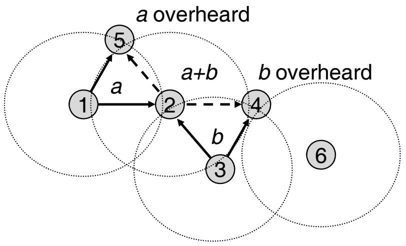


Fig. 1. Example of wireless network coding with opportunistic listening. Dashed lines indicate the transmission of a single coded packet.

One topic that has not been studied thoroughly in the literature is the interaction between opportunistic wireless network coding and the medium access control (MAC) protocol. Most of the existing studies are theoretical and make several assumptions either about the structure of the network or the channel access scheme. For example Liu *et al.* [8], as well as Sengupta *et al.* [9], have independently derived theoretical bounds for the achievable throughput of network coding in large ad-hoc networks. Furthermore, Sagduyu and Ephremides have investigated theoretically cross-layer interactions between network coding and a MAC protocol based on time division multiple access (TDMA) [10]. They proposed the use of a centralized scheduling scheme in order to optimize coding decisions. Even though of great value, these works do not shed light into practical aspects that are related with the use of a distributed MAC protocol. More recently, Chaporkar and Proutiere investigated adaptive network coding and scheduling for wireless ad-hoc networks [11]. They showed that wireless network coding can perform worse than non-coded packet transmission since wireless broadcast can reduce spatial reuse. To this aim, they suggest the use of a lightweight network coding scheme that does not exploit opportunistically overheard packets. In another related work that identified the same problem, the authors propose an algorithm that determines whether transmitting coded or uncoded packets is the most

Manuscript received March 25, 2008; revised August 7, 2008; accepted October 26, 2008. The associate editor coordinating the review of this paper and approving it for publication was V. K. N. Lau.

The material in this paper was presented in part at IEEE Globecom, New Orleans, USA, December 2008.

A. Argyriou is with Philips Research, High Tech Campus 34, Eindhoven, 5656 AE, The Netherlands (e-mail: anargyr@ieee.org).

Digital Object Identifier 10.1109/TWC.2009.080396

optimal decision depending on the status of the transmission buffer [12]. A simulation study by Fasolo *et al.* [13] presents results that highlight the sub-optimal performance of existing 802.11-based MAC protocols and network coding. Also recently, Le *et al.* claimed that no particular link scheduling mechanism is needed for ensuring high performance over 802.11 but instead proper buffer sizing is enough [14]. No formal proof is provided though while limited topologies and traffic patterns are considered. The most closely related work with this paper is the COPE system that is the first practical implementation of the wireless network coding principle [5]. Although several of the limitations of existing MAC protocols for network coding have been identified, none of the aforementioned works introduced any enhancements that address these problems.

In this paper, we focus on distributed MAC protocols and we propose mechanisms that allow the efficient realization of network coding for increasing the system throughput. We refer collectively to these mechanisms as the NC-MAC protocol. NC-MAC attempts to exploit to the fullest opportunistic listening by allowing nodes to decode overheard transmissions of both data packets and acknowledgments. The above process is further optimized by the use of opportunistic acknowledgments as well as virtual buffers that are used for keeping track of the overheard data packets in the neighborhood of a particular node. Also, NC-MAC introduces different data packet types depending on whether they contain network-coded data, uncoded data, or data that can be used for coding. Finally, the channel access scheme is also related to the type of the transmitted data packet. For demonstrating the features of the NC-MAC protocol, we also develop a simple network coding algorithm. As a case study, we present NC-MAC as a set of incremental extensions to the IEEE 802.11 MAC. Nevertheless, the main design principles behind the proposed protocol are not tied specifically to IEEE 802.11, but they can be applied to any distributed MAC protocol that employs a carrier sense multiple access (CSMA) scheme.

The rest of this paper is organized as follows. In Section II, we present a short overview of the main problems that arise from wireless network coding implementations that are unaware of the MAC protocol. Motivated by these problems, we present the basic features of the proposed NC-MAC protocol in Section III, while the channel access scheme is presented in detail in Section IV. Subsequently, in Section V we describe in detail how the NC algorithm can be easily implemented when the basic protocol is in place. All the protocol features are jointly characterized by an analytical throughput model in section VI. In Section VII, we present our comprehensive comparative simulation and numerical results. Finally, Section VIII presents our conclusions.

II. Problem Statement and Motivation

In this section we initially provide an analysis behind the problems of wireless network coding that we wish to address. The identification of these problems is essential for defining the appropriate generic mechanisms that can offer the transmission of coded packets with high efficiency independently of the coding algorithm.

A. Opportunistic Listening

As we mentioned in the Introduction, the broadcast property of the wireless channel is the basic characteristic of wireless ad hoc networks that is exploited by network coding. However, opportunistically overhearing packet transmissions as we demonstrated in Fig. 1, does not always guarantee improved performance of the network coding algorithm. In this figure, we assumed that N_1 transmits a unicast packet a that is intended for node N_2, but it is also overheard by N_5 who is within the carrier sensing range of both N_1 and N_2. If a node is not the intended receiver (in this figure N_5), it cannot acknowledge correct reception of overheard packets. This means that the node that employs network coding (in this figure N_2), should either not code packets for which it is not certain that they were overheard, or it can code despite the absence of this knowledge. However, it is impossible for N_2 to know if a coding decision is correct unless it has precise knowledge of the packets that are stored at each node in the neighborhood. Incorrect coding decisions will require retransmissions that lead to under-utilization of the wireless channel and eventually lower throughput. In existing wireless network coding schemes, packets that are meant for broadcast are sent with the 802.11 unicast mechanism (pseudo-broadcast). In that case the problem is addressed with the use of asynchronous cumulative out-of-band acknowledgments [5]. The disadvantages of this solution are first the stochastic estimation of the fate of already transmitted packets, the increased packet delay, and the need for separate medium contention rounds for the transmission of out-of-band batch acknowledgments. These problems are exacerbated of course in situations of high network contention. Another important problem is the existence of hidden nodes that can also significantly deteriorate the performance of network coding. The reason is that the existence of hidden nodes increases the probability of unidentified packet loss of both regular and coded packet transmissions (in this figure N_6). Therefore, it is crucial is to improve the robustness of the coding algorithm by avoiding incorrect coding decisions for each individual packet.

B. Metadata for Coded Packets

The problems we described in the previous paragraph can significantly affect the performance of the actual network coding algorithm. The unreliability of wireless transmissions, and the possibility of unordered delivery of network-coded packets, forces existing protocols to use an additional packet header. This header contains all the necessary metadata for decoding the initial packet. In particular, when a node receives a packet with a MAC address different from its own, it checks this header to see if it should decode the packet to extract the new information. Otherwise it stores the packet in a buffer and marks it as an opportunistically received packet[1]. However, this header constitutes an important overhead and requires new functionality between the Network and MAC layers. But if we look more carefully, we can see that this overhead is necessitated because packets are not synchronously acknowledged, and they may also need to be forwarded over multiple hops.

[1]With COPE [5], the node that transmits a coded packet does not know whether it was actually received.

Therefore, the challenge here is to minimize the additional header overhead, but at the same time ensure correct decoding decisions at the receiving nodes.

III. Basic NC-MAC Protocol

Our discussion in Section II provided an overview of the problems that we desire to address in this paper. In this section we present the proposed NC-MAC protocol that is implemented with minor modifications as part of IEEE 802.11. With NC-MAC a node can transmit three types of data packets: 1) a regular unicast uncoded data packet, 2) a unicast uncoded data packet targeted to multiple opportunistic receivers, and 3) a coded packet targeted to multiple receivers. Furthermore, a new type of acknowledgment packets is introduced but its purpose is explained in the next section. For these three types of data packets the MAC headers are created as follows: 1) For regular packets the MAC header includes a single destination address that indicates unicast transmission. 2) For an opportunistic packet the MAC header destination fields include the addresses of the intended opportunistic receivers. 3) For a coded packet the destination address fields in the MAC header include all the addresses of the nodes that must decode the network-coded packet. The header indicates which packet type is contained in a two-byte field. While neither the proposed protocol nor the coding algorithm dependent on this precise header structure, the requirements that we analyzed should be satisfied.

A. Managing Opportunistically Overheard Data Packets and Acknowledgments

The main mechanism for improving coding decisions is an additional form of acknowledgments for opportunistically overheard packets. This is a mechanism of receiving in-band notifications that ensures correct coding decisions while it renders unnecessary the explicit cross-layer communication between higher protocol layers and the MAC protocol. We call this type of acknowledgments, *opportunistic acknowledgments* (OACKs) to distinguish from the ACKs that are required for the reliable data delivery[2]. OACKs do not ensure full reliability since no retransmission timer is associated with opportunistically overheard packets. Nevertheless, OACKs offer a fairly simple and powerful technique that allows a node to have a better indication regarding the status of the overheard packets at each neighboring/receiving node. In conjunction with OACKs, multiple virtual *opportunistic packet buffers* (OPBs) are maintained for each transmitting neighbor of a particular node. The OPBs are mainly responsible for tracking "who sent what" and "who acknowledged what". Therefore, to minimize storage requirements to the absolute minimum we define certain rules under which data packets are stored. To demonstrate how the previous two ideas can work together, we consider again our simple example in Fig. 1. In this example N_1 should require from N_5 to OACK packet a since it has detected transmissions from N_2 to N_5. This OACK will indirectly notify N_2 regarding the opportunity to use packet a

[2]The PHY frame structure is the same. Since OACKs are control packets, they are transmitted at lowest possible PHY rate to ensure high probability of successful delivery.

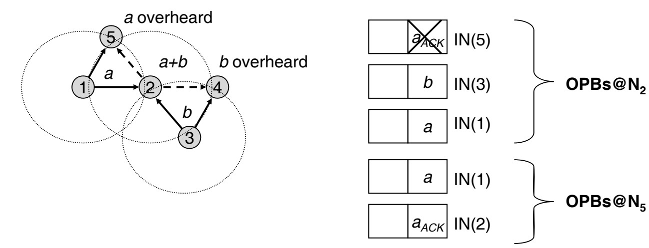


Fig. 2. Nodes use the last successfully received packet as an indication of each receiver's OPB state. The loss of opportunistic ACK is also demonstrated in this figure. N_2 estimates that N_5 did not receive packet a. However, N_5 actually has it while it also overheard the ACK sent by N_2.

for coding. Note however that while N_5 will OACK packet a, N_2 will send a regular ACK since it is the target destination of a.

We can formulate now a simple algorithm for determining the group of nodes that should OACK the transmission of a data packet originated from a specific node. The algorithm also prevents OACK implosion since we do not want every node to acknowledge unnecessarily the transmission of every regular data packet. The algorithm consists basically of two conditions that should be checked before a node populates the list of opportunistic receivers in the packet header:

Rule 1: A node N_x should request from another node N_y to OACK a packet l currently considered for transmission to $l{\rightarrow}nexthop$, iff within the last K transmissions detected from $l{\rightarrow}nexthop$ at least one was destined to node N_y. The sender makes a transition to $OACK\_requested$ state.

If within the last K transmitted packets from $l{\rightarrow}nexthop$ one was transmitted to N_y then N_x uses this information as a *soft indication* that there is potentially a packet flow from node $l{\rightarrow}nexthop$ to node N_y that could be coded. However, we want to avoid that a node receives and stores opportunistically overheard packets that are not actually used for coding. Since rule 1 does not actually ensure that there is a coding opportunity and the request of an OACK was the appropriate choice, we define a second rule that can actually satisfy this objective:

Rule 2: If a node is in the $OACK\_requested$ state and within K consecutive transmissions detected from $l{\rightarrow}nexthop$ one is a coded packet destined to node N_y, then subsequent OACKs can be sent by following rule 1. The sender makes a transition to $OACK\_validated$ state.

These rules are important to ensure dynamic adaptation to the changing channel or packet flow conditions that may arise due to packet errors or mobility. Furthermore, rules 1 and 2 ensure that at most one opportunistically overheard packet is unnecessarily stored. For example the result of these rules in Fig. 2 is that N_1 will request from N_5 to store a packet only when it identifies a transmission from N_2 to N_5. Also in the same figure, we present next to the simple toy topology the OPBs of nodes N_2 and N_5. Nodes can "flush" their OPBs by simply not acknowledging opportunistically overheard packet transmissions. These nodes are considered out of the "coding loop". Therefore, if a new innovative packet has to be transmitted to them, it is transmitted uncoded. Note that this may be either a conscious decision of the

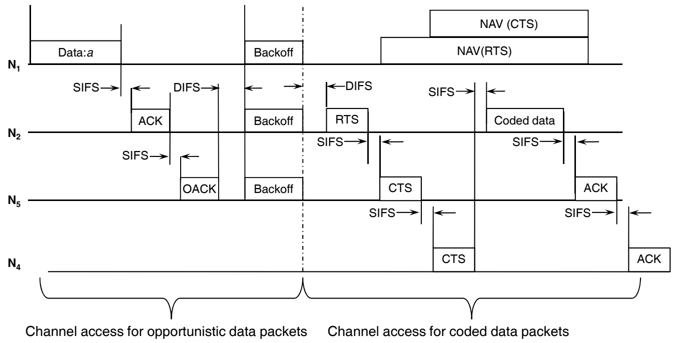


Fig. 3. Proposed channel access scheme with transmission of OACKs and ACKs. The SIFS interval between the successive acknowledgments is used for protecting from other nodes' contention. Our system supports both physical/virtual carrier sensing.

receiver (e.g. due to buffer limitations) or it may arise from a lost OACK as the example in Fig. 2 demonstrates. Either way, the protocol ensures that it has an up-to-date view of the OPBs in its neighborhood before any coding decisions are made. We believe that this level of reliability is enough instead of adopting a complex reliable multicast protocol. One more observation that can be made from Fig. 2, is that use of OPBs renders unnecessary the need for complex packet headers for distinguishing coded packets, since a virtual queue is maintained for each neighbor.

IV. Channel Access with NC-MAC

At this point we have introduced the concept of OACKs and we explained their role in manipulating the multiple virtual OPBs that are mapped to a single receiving MAC layer queue. Even though our protocol only requires a CSMA channel access scheme, we adopt the same distributed multiple access scheme with IEEE 802.11 that is based on binary exponential backoff (BEB) for collision avoidance. The interested reader is referred to [15] and [16] for details regarding the operation of CSMA/CA and the IEEE 802.11 MAC respectively. With the proposed protocol, the channel access mechanism differs only in the way coded and opportunistic packets are transmitted.

1) Channel Access for the Transmission of Opportunistic Packets: To implement this feature, the basic requirement is the transmission of OACKs according to rules 1 and 2. Therefore, the channel access mechanism should be engineered accordingly. Fig. 3 demonstrates the proposed channel access scheme when nodes transmit opportunistic and coded packets. N_2 acknowledges a packet since it is the next hop of a transmission from N_1. According to our network topology, this also means that N_5 can overhear transmissions both from N_1 and N_2. According to rule 1, N_1 must have requested an OACK from N_5 which sends it after a short inter-frame space (SIFS). After N_1 receives an OACK from N_5 it infers that N_5 can hear also N_2's transmissions. N_1 waits for a distributed inter-frame space (DIFS) so that it is certain that no other nodes have sent an ACK, and then it proceeds with resetting the backoff counter normally. Regarding the duration of a successful or not transmission of a opportunistic packet, we must note that it is not constant since the number of opportunistic receivers depends on the specific packet flows. If the number of opportunistic receivers is denoted as N, from Fig. 3 we can calculate the maximum duration for the successful transmission of an opportunistic packet as $T_{RO}(N) = T_{HDR}(N) + T_D(r) + \sum_N (T_{SIFS} + T_{OACK})$, where T_{HDR} is the duration of the extended MAC header and $T_D(r)$ is the duration of the data packet that depends on the selected PHY rate r. Note that the duration of successful transmission for a regular unicast data packet is constant and equal to $T_R = T_{HDR} + T_D(r) + T_{SIFS} + T_{ACK}$.

2) Channel Access for the Transmission of Coded Packets: The proposed channel access scheme makes use of the request-to-send/clear-to-send (RTS/CTS) mechanism only for transmitting coded packets. A transmitting node sends a single RTS with multiple nexthop addresses in the RTS header (i.e., to the nodes that can decode an innovative packet). CTS packets are sent after SIFS with the order being determined by the order of the receivers addresses as they appear in the RTS packet (see Fig. 3). After a CTS is received from the nodes that can actually decode an innovative packet, the transmitter sends with a single broadcast transmission the coded packet. The received packet is decoded and the appropriate OPBs are updated. Finally an ACK is sent after SIFS similar to the same principle that applies for the transmission of CTS control packets. Note that whether RTS or data packets are re-transmitted to those neighbors from which CTS or ACK was not received, depends on whether reliable multicast is employed or not. For our protocol we decided to avoid this option since we do not want to affect fundamentally the basic CSMA/CA. For a coded data packet the total maximum transmission time is similarly with before $T_C(N) = T_{HDR}(N) + T_D(r) + \sum_N (T_{SIFS} + T_{ACK})$, where N is the number of receivers that the specific coded packet is destined to.

3) Pseudo-Algorithm for Channel Access: We summarize the above analysis for the proposed channel access scheme by presenting a pseudo-algorithm in the upper part of Fig. 4. When there is packet in the MAC queue, regardless of whether the packet is coded or not, the node will sense the channel and transmit according to the BEB algorithm. This is accomplished by a call to the $xmit\_pkt\_mac()$ procedure. When the backoff timer expires, the packet is passed to the PHY with $xmit\_pkt\_phy()$ and it is transmitted. The radio at the transmitter is then set into listening mode for listening ACKs and OACKs after the end of the transmission. Now the new functionality that NC-MAC introduces is that the transmitter populates the hash table that indicates the receivers that actually received the particular packet l. As we said earlier, a coded/uncoded packet is retransmitted and the backoff algorithm is initiated, only when the transmission to one of the primary receivers fails (i.e., that belongs in the group $Innovative[]$). Depending on the group of receivers that did not ACK a coded packet, the coding algorithm at the sender will check for new coding opportunities as we explain in the next section.

V. Network Coding Algorithm

We explained in the previous sections the concept of OACKs and OPBs, as well as the proposed channel access scheme. The aforementioned techniques all synergetically

$xmit\_pkt\_mac(l)$

$xmit\_pkt\_phy(l)$ //when the backoff timer expires
$listen\_ACKs(l)$
$hash(l) = \{...\}$
if if $Innovative[l]$ have received l **then**
  $CW = \min(CW_{min}, CW_{max})$
else
  $CW = \min(2CW_{min}, CW_{max})$ //double the CW
end if
$start\_backoff\_timer(CW)$

$code\_pkt\_mpdu(l_x)$

if $l_x \rightarrow nxthop == hash(l_y)$ **then**
  if $l_y \rightarrow nxthop == hash(l_x)$ **then**
    $l_x = l_x \oplus l_y$
    $l_x \rightarrow nxthop+ = \{l_y \rightarrow nxthop, ...\}$
  end if
end if
l_y++
$xmit\_pkt\_mac(l_x)$

Fig. 4. Pseudo-algorithms for the channel access scheme and the coding algorithm.

contribute towards improving the ability of the coding algorithm to make correct coding decisions. Regarding the coding algorithm itself, our system supports the transmission of MAC service data units (MSDUs) coded with simple bitwise XOR operation while it requires decoding at the next hop. This last requirement is a mixed blessing since packets may be decodable at multiple nexthops. This option would demand a cross-layer design that requires either additional packet headers as we explained earlier [5], or explicit support from the routing layer [7]. On the other hand, by asserting this constraint we simplify the design of the coding algorithm that can be designed so as to support this single-hop XOR-based network coding algorithm.

Now the MSDUs that are passed from the upper layers are encoded by using the algorithm shown in the lower part of Fig. 4. The main procedure $code\_pkt\_mpdu()$ is executed for a packet l_x that resides at the head of the outgoing MAC queue. The decodability of different options is examined next. The first packet l_x in the outgoing MAC queue is coded only if it has been overheard or transmitted by a neighboring node. To implement the above lookup procedure, we use in our system an efficient hash table that contains as keys the packets in the outgoing MAC queue. The hash function returns the nodes that received that particular packet. For example in Fig. 2, the result of the hash function $hash(\alpha)$ executed at N_2, will return N_5 according to the contents of the OPBs. Hash tables provide constant-time $O(1)$ lookup on average. When a valid coding combination is found, the packet l_x is coded while the set of destination nodes that must receive it are updated accordingly (i.e., with the assignment $l_x \rightarrow nxthop+ =$). This procedure is repeated until the optimal number of packets that can be coded is identified. A coded packet is created this way. It is also possible that the algorithm may not code a packet, and transmit it as opportunistic packet if the rules 1 and 2 allow it. When these two rules do not apply, the packet is transmitted as a regular uncoded packet.

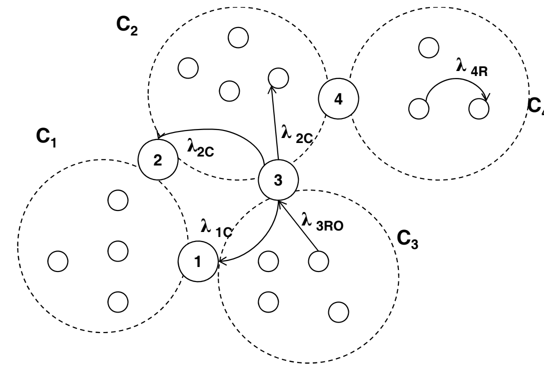


Fig. 5. The analytical model considers clusters of nodes and inter-cluster connections through relays.

VI. Analytical Model

Developing an accurate performance model for our system is not trivial since there is a need to characterize jointly the MAC protocol and network coding in a multi-hop setting [11], [17]. Therefore, we narrow our focus in a static mesh network for which only certain nodes are configured to act as relays. The network we are interested consists of clusters of nodes. Each cluster i is composed of R_i regular nodes that transmit packets within the cluster, and RO_i regular nodes that transmit packets to other clusters. Furthermore, each cluster has access to C_i relay nodes. Fig. 5 presents an example of our system model with 4 clusters and 4 relays. Relay nodes are the nodes that can code packets since they can transmit and receive packets in more than one cluster. All nodes within a cluster can hear each other. Now each relay node interconnects M_R clusters and so it is subject to the contention caused by this exposure. More specifically a relay has to contend with $R \cdot M_R$ other regular nodes plus $M_R \cdot (C-1)$ other relays.

We adopt a queueing model for modeling our system, and we define different traffic classes to distinguish different packet types. The network is assumed to be in the saturation condition, that is every node is backlogged and always has a packet to transmit. The mean data traffic load that is destined from each node inside another node in the same cluster i, is denoted as λ_{R_i} (packets/slot), the rate of opportunistic packets that are destined to any of the relays is λ_{RO_i}, while the rate of coded packets transmitted from the relay j is λ_{C_i}. We assume that the total outgoing traffic from a cluster i is split equally between the C_i available relays. If we look it from the perspective of the relay, the total traffic that is concentrated to a specific relay j is $\lambda_{IN}(j) = \sum_{i=1}^{M_R} \lambda_{RO_i}(j)$. Now the service rate that is seen by each packet type is denoted as μ_C, μ_{RO}, and μ_R respectively. Also, the queue utilization at the relay is given by $\rho(j) = \frac{\lambda_{IN}(j)}{\mu_C(j)}$, and it has to be $\rho(j) > 1$ for stability.

A. Packet Transmission Probabilities

NC-MAC is essentially the same q-persistent CSMA/CA protocol that is used in the DCF mode of IEEE 802.11. In a slotted q-persistent CSMA each node senses the channel at the beginning of each time slot and if the channel is free it transmits with probability q [15]. If the conditional collision probability is given as p, a collision will occur if either a node or a relay transmits a packet inside the same contention zone. Let τ denote the probability that a node transmits in a random slot. Then if both the C relays and the $R+RO$ regular nodes are backlogged and if the network is homogeneous, the total collision probability would be $p = 1-(1-\tau)^{R+RO+C-1}$ [18]. However, for NC-MAC the average transmission probability τ is different depending on the type of the transmitted packet, and also for each relay it depends on the number of clusters M_R it interconnects. We calculate the collision probabilities next. The conditional collision probability for RTS packets preceding a coded packet transmission only from relay j can be calculated as

$$p_C(j) = 1 - \prod_{i=1}^{M_R(j)} \Big\{(1-\frac{\lambda_{R_i}}{\mu_{C_i}}\tau_{R_i})^{R_i}(1-\frac{\lambda_{RO_i}}{\mu_{C_i}}\tau_{RO_i})^{RO_i} \times (1-\frac{\sum_{k=1}^{i}\lambda_{C_k}}{\mu_{C_i}}\tau_{C_i})^{C_i-1}\Big\}. \quad (1)$$

This equation means that the transmission of an RTS packet, that precedes the transmission of the coded packet itself, will succeed only when none of the regular nodes in the adjacent clusters transmit a regular data packet, and similarly no node transmits an opportunistic or coded packet. To ameliorate the effect of high contention, we adopted the RTS scheme for this type of packets which eventually results into reducing the p_C. Note that in this equation the effect of hidden nodes is not included since we observed that the short duration of the RTS minimizes the probability of such collisions.

For regular packets we can derive a simpler formula since these transmissions are exposed only to the contention from the nodes inside their own cluster and the RTS packets from the C_i relays. So we have that

$$p_R(i) = 1-(1-\frac{\lambda_{R_i}}{\mu_{R_i}}\tau_{R_i})^{R_i-1} \times (1-\frac{\lambda_{RO_i}}{\mu_{R_i}}\tau_{RO_i})^{RO_i}(1-\frac{\sum_{k=1}^{i}\lambda_{C_k}}{\mu_{R_i}}\tau_{C_i})^{C_i} \quad (2)$$

Regarding the opportunistic packets, we have to recall that according to our protocol they are destined to relays[3]. At this point in our analysis we need to account for the effect of hidden nodes. The transmission of opportunistic packets to a relay node will be vulnerable to the transmissions from all the nodes that are within the transmission range of that particular relay. Thus, the resulting expression will be more complex. The probability of collisions *only due to hidden nodes* for all the transmissions of RO packets from cluster i will be

$$p_H(i) = 1-\frac{\lambda_{RO_i}}{\mu_{RO_i}}\tau_{RO_i}\prod_{m=1}^{M_R(j)-1}\Big\{(1-\frac{\lambda_{R_m}}{\mu_{RO_i}}\tau_{R_m})^{R_m} \times (1-\frac{\lambda_{RO_m}}{\mu_{RO_i}}\tau_{RO_m})^{RO_m}(1-\frac{\sum_{k=1}^{m}\lambda_{C_k}}{\mu_{RO_i}}\tau_{C_m})^{C_m}\Big\}. \quad (3)$$

Essentially this equation means that the transmission of an opportunistic packet to each relay will succeed only when none of the nodes (either regular or relays) in the adjacent clusters of that particular relay transmits. The first term $\frac{\lambda_{RO_i}}{\mu_{RO_i}}\tau_{RO_i}$ is the transmission probability of opportunistic packets to any of the attached relays. It is possible to express this quantity in this way since we assume for simplicity that the traffic load is distributed symmetrically. Furthermore, the probabilities of this type of collisions depend on the targeted relay and of course its neighborhood, which makes them independent for each relay. That is why the form of this equation is similar to (1). But despite hidden nodes, we also have to account for the collisions that are caused inside the same cluster during the service cycle of an opportunistic packet. Therefore, we have

$$p_{RI}(i) = 1-(1-\frac{\lambda_{R_i}}{\mu_{RO_i}}\tau_{R_i})^{R_i}(1-\frac{\lambda_{RO_i}}{\mu_{RO_i}}\tau_{RO_i}))^{RO_i-1} \times (1-\frac{\sum_{k=1}^{i}\lambda_{C_k}}{\mu_{RO_i}}\tau_{C_i})^{C_i}. \quad (4)$$

Note that the total collision probability p_{RO} is equal to $p_{RI} + p_H$ since these two probabilities are independent.

At this moment we have derived the conditional transmission probabilities, and so we can easily calculate the average number of transmissions for each packet type. The average number of transmissions has to account for the fact that L-truncated ARQ is employed. So we have for regular packets that

$$E[N_R] = \frac{1-p_R^{L-1}}{1-p_R}. \quad (5)$$

Similar equations can be written for $E[N_C]$ and $E[N_{RO}]$.

The next step is to account for the binary exponential backoff algorithm that is used in conjunction with ARQ. The backoff algorithm controls the evolution of the contention window as we described earlier. It has already been shown that its evolution can be expressed as a geometric distribution which means that its average value will be [19]

$$\overline{W} = \sum_{l=0}^{L} p^l(1-p)2^l\frac{W}{2} + p^{L+1}2^L\frac{W}{2}. \quad (6)$$

To obtain another equation for solving numerically the problem, we can easily calculate the transmission probability for an arbitrary slot as follows

$$\tau_R = \frac{E[N_R]}{E[N_R]+\bar{W}_R}. \quad (7)$$

Accordingly we can derive the transmission probabilities for the other packet types.

B. Collision Duration

As a consequence of the previous analysis, we can see that the average collision time for different packet types will also be different and therefore has to be calculated accordingly. By using the detailed values for the conditional transmission probabilities as we derived them in (1), (2), (3), and (4), we can estimate accurately the average total collision time. For example for coded packets it will be

$$\overline{T}_{coll,C} = \frac{p_C(1-p_C)^L}{1-p_C}\bar{t}_{coll,C}. \quad (8)$$

[3]These packets are also received by nodes inside their cluster but these transmissions do not affect the throughput calculation.

The terms $\bar{t}_{coll,C}$, $\bar{t}_{coll,R}$, and $\bar{t}_{coll,RO}$ are the average collision time of a single coded, regular, and opportunistic packet respectively. Note that $\bar{t}_{coll,R} = \bar{t}_{coll,RO}$, since the duration of data packet transmissions is the same.

But the collision probability for coded packets that was calculated in (1) is just the average value. It is not enough if we want to calculate the average collision time of a single packet transmission $\bar{t}_{coll,C}$. The reason is that a coded packet can only collide with its short RTS packet that is sent initially preceding any coded packet transmission. So the RTS packet that precedes any coded packet may collide with other RTS or data packets, and so there is a need to distinguish the collision probabilities of coded and opportunistic/regular packets and calculate the time spent during these phases. By following the same approach we can derive easily closed-form expressions for these probabilities. First, the probability that the RTS transmission from the relay j collides with another RTS transmission from another relay in the M_R attached clusters is calculated as follows

$$p_1(j) = \prod_{i=1}^{M_R} \frac{(C_i-1)\frac{\lambda_{C_i}}{\mu_{C_i}}}{(C_i-1)\frac{\lambda_{C_i}}{\mu_{C_i}} + R_i\frac{\lambda_{R_i}}{\mu_{C_i}} + RO_i\frac{\lambda_{RO_i}}{\mu_{C_i}}}. \quad (9)$$

The last terms in this equation, i.e. $R\frac{\lambda_R}{\mu_C}$ and $RO\frac{\lambda_{RO}}{\mu_C}$, express the number of regular and opportunistic packets that are transmitted during the service cycle μ_C of a single coded data packet. On the other hand, the probability that the RTS transmission from a relay collides with a regular or opportunistic transmission is

$$p_2(j) = \prod_{i=1}^{M_R} \frac{(R_i\frac{\lambda_{R_i}}{\mu_C(i)} + RO_i\frac{\lambda_{RO_i}}{\mu_{C_i}})}{(C_i-1)\frac{\lambda_{C_i}}{\mu_{C_i}} + R_i\frac{\lambda_{R_i}}{\mu_{C_i}} + RO_i\frac{\lambda_{RO_i}}{\mu_{C_i}}}. \quad (10)$$

Therefore, the average time spent on collisions for a single coded packet is $\bar{t}_{coll,C} = p_1 t_{RTS} + p_2 t_{data}$, where t_{RTS} and t_{data} are the transmission time of an RTS packet while the later is determined by the payload size and the various header overheads.

Regarding the collision of regular and opportunistic packets, since the colliding packets have similar size, and therefore the same impact on the collision time, there is no need to refine the calculation of p_R and p_{RO} respectively. Now in order to calculate the overall throughput achieved within a single cluster, we have to consider the transmission of regular, opportunistic, and coded packets.

C. Throughput

For deriving the final throughput formula we need also to identify the expressions for the service cycle of different packet types. By following the same reasoning with the previous paragraphs, the average cycle for the transmission of a regular packet in cluster i is

$$\begin{aligned}\frac{1}{\mu_R} &= \overline{W}_R + (R_i-1)\frac{\lambda_{R_i}}{\mu_{R_i}}T_R + RO_i\frac{\lambda_{RO_i}}{\mu_{R_i}}T_{RO} \\ &+ C_i\frac{\lambda_{C_i}}{\mu_{R_i}}T_C + T_R + \frac{1}{3}(R_i-1)\frac{\lambda_{R_i}}{\mu_{R_i}}\overline{T}_{coll,R} \\ &+ \frac{1}{3}RO_i\frac{\lambda_{RO_i}}{\mu_{R_i}}\overline{T}_{coll,RO} + \frac{1}{3}C_i\frac{\lambda_{C_i}}{\mu_{R_i}}\overline{T}_{coll,C}. \quad (11)\end{aligned}$$

The first line in this equation is the time consumed for the successful transmissions of the different packet types. T_R, T_{RO}, and T_C were calculated in Section III. In this formula we also account for the total average time spent in collisions, thus dividing by 3 different packet types, and the total average time in backoff. During this service cycle the MAC-layer throughput inside cluster i will be

$$G(i) = R_i\frac{\lambda_{R_i}}{\mu_{R_i}} + RO_i\frac{\lambda_{RO_i}}{\mu_{R_i}} + C_i\frac{\lambda_{C_i}}{\mu_{R_i}}. \quad (12)$$

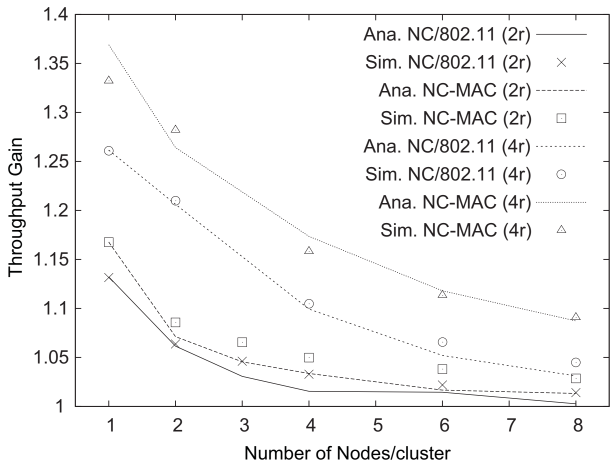
Recall that M_R is the number of clusters that the relay is interconnecting and therefore M_R is the number of packets that it can code and increase the throughput by this factor. Note also that the performance implications of using the relay for coding, and the impact of hidden nodes, are embedded in (12) since the contribution of each relay is calculated independently for each cluster that it interconnects. However, to solve this equations numerically we need the topology of the network so that we know the position of the relays in the system.

VII. Performance Evaluation

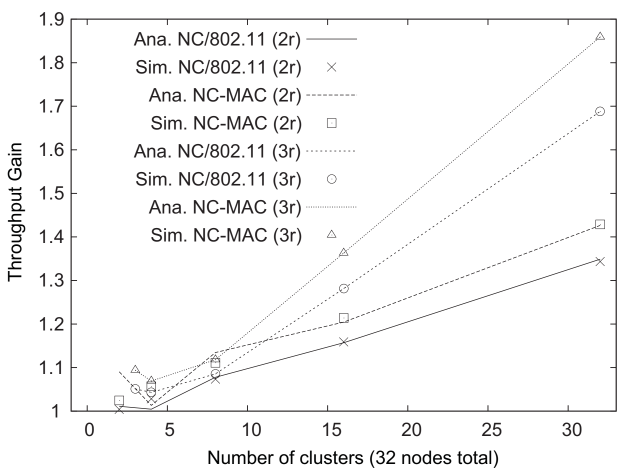
For evaluating the performance of the proposed protocol we defined different topologies that consist of inter-connected clusters of nodes. We have built our own simulation tool in C for NC-MAC and the IEEE 802.11 MAC protocol. We also implemented a simulation model for a coding scheme that adopts the main features of COPE [5] and includes all the basic features like pseudo-broadcast and separate data transmissions that contain reports for the stored coded packets. We call this protocol NC/802.11 to indicate that network coding is implemented independently of the 802.11 MAC protocol. We experimented with the number of CBR/UDP flows and the number of hops they traverse. Flows between two nodes in different clusters are generated randomly so that relay nodes are loaded asymmetrically and achieve different and more realistic coding gains. When multiple routes exist, shortest path routing is used for determining the end-to-end path of each flow. The channel access timing parameters are similar with 802.11 ($SIFS$=16μsec, $DIFS$=34μsec). For all our simulations we assumed contention-induced packet losses.

A. Number of Nodes per Cluster

Fig. 6 demonstrates the impact of assigning a different number of nodes to each cluster while the packet flows traverse just a single hop. In these experiments, the percentage of outgoing traffic load λ_{RO} is 60%. Fig. 6(a) presents simulation and numerical results for 4 clusters connected in chain and star topologies respectively. In a chain topology clusters are connected in tandem with a maximum of 2 relays between them (denoted as $2r$), while in the star topology all clusters are inter-connected together with a relay (denoted as $4r$). The first result we expect to see is the reduction in throughput as we increase the number of contending nodes within the clusters. The relay nodes gradually obtain a smaller fraction of the available bandwidth which prevents them from forwarding coded packets. This situation is reflected on the performance of relays and regular nodes that is different by at most 10%. Also another important observation is that the numerical results

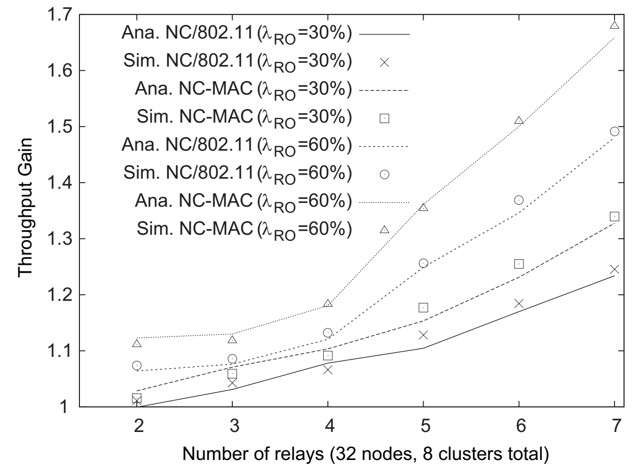


(a) Four clusters in star and chain topologies

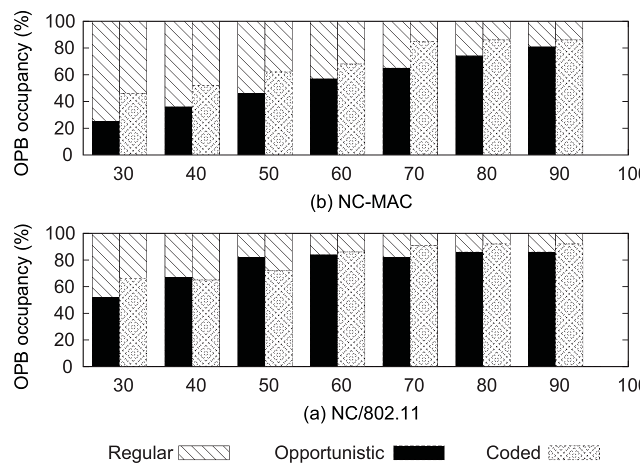


(b) Constant number of 32 nodes for different allocations into clusters.

Fig. 6. Simulation and numerical results with $\lambda_{RO} = 60\%$. The use of a star topology distributes the flows across more contention zones.



(a) Performance vs. different number of relaying nodes.



(b) Average system-wide MAC queue occupancy vs. outgoing traffic λ_{RO}. The two bars correspond to regular (left) and relay (right) nodes.

Fig. 7. Results for the throughput gain and the buffer requirements for different outgoing traffic load λ_{RO} .

follow closely the obtained simulations while the network is in the stable state with $\rho < 1$. Even for high load where the queue utilization ratio goes beyond 1, the model is still very accurate since the queuing modeling approach accounts for the dropped packets.

Since the results in Fig. 6(a) might be misleading in terms of the performance of both protocols under test, we evaluate the system performance when a constant number of nodes is allocated differently to clusters, adjusting thus the level of contention within each cluster. Fig. 6(b) presents these results for a chain and "3-relay" (denoted as $3r$) topologies, which means that each cluster was connected to 2 and 3 relays respectively. The use of more relays and the formation of different topologies results into even higher throughput increase with NC-MAC. These results represent a more realistic situation since in a wireless ad hoc network several neighbors of a node can act as relays. The numerical results follow the simulations even more closely when compared with the previous experiment, despite the existence of more clusters. The good match in this case is attributed to the fundamental features of NC-MAC that are embedded in the developed model which is primarily the differentiation of the transmitted packet types.

B. Number of Relay Nodes between Clusters

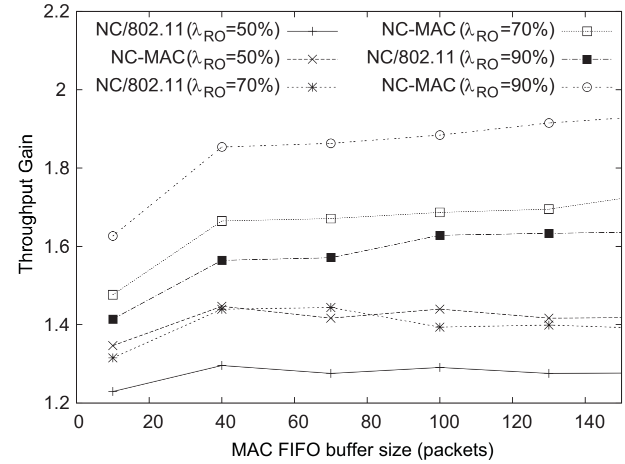
The results in the previous section indicate that higher performance is achieved when more nodes can act as relays and code packets. This is something that we expected but a more detailed look into the performance is provided now. We start with 32 nodes allocated to 8 clusters and we gradually increase the number of relays from 2 to the maximum possible of 7. Fig. 7(a) shows that the performance increase achieved by NC-MAC becomes more significant as the number of outgoing traffic λ_{RO} is increased. Our analytical model is also well-suited for modeling the implications of a higher outgoing traffic load. The separation between λ_{RO} and λ_R served precisely for modeling this scenario. As we will later see, the performance can continue to increase linearly until a certain point that depends primarily on the size of the MAC queue. Nevertheless, for a higher number of relay nodes that interconnect the clusters the importance of our protocol is even more significant as the results in Fig. 7(a) indicate. This is a result we expect since the outgoing offered load from the clusters λ_{RO}, can be distributed to more relays which correspond to an increased coding gain. The proposed NC-MAC can capitalize even more on the introduction of more relay nodes since the coding opportunities are increased. The reason behind the increase in the number of coding opportunities will become clear in the next set of simulations. Another important observation from the same set of results is that the proposed protocol is less sensitive to the number of nodes that contend for the medium within each cluster due to the low overhead of the OACKs. Even with light outgoing traffic load, and many opportunistic receivers, no unnecessary buffering of overheard packets takes place. At the same time, for NC/802.11 and even for light outgoing traffic load a node still suffers from contention-related losses for the out-of-band acknowledgments sent for coded packets. Furthermore, more buffer space is occupied at the opportunistic receivers. Finally, for high λ_{RO} we observed increased collisions because of hidden nodes. When we tried to obtain numerical results by not considering the effect of hidden nodes, we observed a significant mismatch with the simulation results.

For the same experiment we present in Fig. 7(b) the allocation of the available buffer space between the three types of data packets both at regular nodes and the relays. Note that no acknowledgments are stored since they are processed immediately by updating the hash table. The results are very encouraging since they show that the previous performance is achieved with the storage of the minimum number of opportunistic packets. A subtle point that is not shown, is that even though for high inter-cluster load the percentage of opportunistic packets in the buffer is nearly the same for both protocols, NC-MAC stores only packets that are actually needed for coding. The same holds for coded packets at the relays where several of them are unnecessarily coded by NC/802.11.

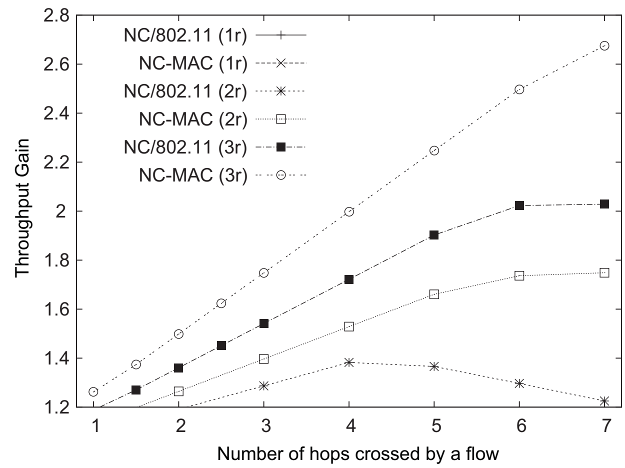
C. Sizing the MAC Queue

One of the questions that naturally comes up at this point is the effect of the MAC queue size on the performance of the relays and regular nodes. Contrary to what we would expect, increasing the size of the MAC queue both at individual regular nodes and at the relays does not always lead to improved performance. In general, with increased queue size a relay has more coding opportunities both with NC/802.11 and NC-MAC[4]. However, this is only true for symmetric traffic between clusters for which the simulation results are presented in Fig. 8(a). Numerical results are not presented to avoid clogging the presentation. In this figure we observe that after a certain buffer size, further increase of the buffer size does not enhance the performance even for higher offered load at the relay since these packets are dropped at very high rate, i.e. $\rho > 1$. This is one important conclusion from several simulations that we performed. NC-MAC is able to maintain an increasing throughput gain even for a higher offered load. When trying to interpret this result our analysis provided a very interesting insight. With NC-MAC a coded packet before it is transmitted it replaces in the output queue of the relay two packets, and more buffer space is saved when the cross-cluster

[4]The NC/802.11 protocol also stores coded packets at regular nodes contrary to NC-MAC.



(a) Throughput gain vs. buffer size for clusters and different percentages of outgoing traffic load λ_{RO}.



(b) Throughput gain vs. number of hops flows that are traversed by a flow. 32 nodes allocated to eight clusters. 50% of flows are outgoing.

Fig. 8. Simulation results for the throughput gain.

flows are increased. But the removal of regular packets from the relay queue after they are coded, has minimal impact on their reliable delivery because of the RTS/CTS mechanism. However, for NC/802.11 even after several regular packets are coded, they have to be maintained in the buffer. This is necessary since out-of-band group acknowledgments must be received for each of the regular packets that is contained in the coded packet. Finally, it is easy to see that a higher degree of symmetrical packets flows would work in favor of NC-MAC.

D. Multihop Flows

The effect of flows spanning multiple hops is examined next and Fig. 8(b) presents the related simulation results. These results correspond to a chain topology of 32 nodes allocated to 8 clusters. In this case the offered load at each relay is increased because the relays have to forward traffic from their neighboring relays. Therefore, the percentage of traffic that the relays must forward is higher while the locally generated load is decreased. At the same time, the coding opportunities are also increased. Nevertheless, there is a critical point that is different for each protocol and depends on the number of relays and the forwarding offered load. When the forwarding load is increased more than a certain value, the increased contention and buffer overflows at the relays disallows them from forwarding additional coded packets. In our simulations we observed that increasing the buffer size could only increase performance when the number of hops is low while at the same time the packet flows across clusters are highly symmetric.

VIII. Conclusions

In this paper we proposed a set of incremental enhancements for distributed MAC protocols that target improved performance when wireless network coding is employed. Our protocol called NC-MAC, does not specify the network coding algorithm itself but instead defines a set of fundamental mechanisms that allow any opportunistic network coding algorithm to increase its performance. The proposed scheme focuses on ensuring correct coding decisions at each network node, while it requires no cross-layer interactions. This problem is solved in an integrated fashion first with the use of opportunistic and adaptive acknowledgments at the MAC layer, and second with the use of virtual opportunistic data packet buffers. In the case of the IEEE 802.11 MAC, the above mechanisms can be easily implemented by simple incremental extensions that do not affect the capabilities of legacy 802.11 nodes. Simulations of the proposed protocol for various static ad hoc network topologies demonstrate a performance improvement over existing schemes by 30-40%.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[2] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE International Symposium on Information Theory*, 2003.

[3] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, "Wireless network coding: opportunities and challenges," in *Proc. Allerton Conference*, 2007.

[4] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, "The importance of being opportunistic: practical network coding for wireless environments," in *Proc. Allerton Conference*, 2005.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *Proc. SIGCOMM*, 2006.

[6] F. Xue and S. Sandhu, "PHY-layer network coding for broadcast channel with side information," in *Proc. Information Theory Workshop (ITW)*, 2007.

[7] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in opportunistic wireless routing," in *Proc. SIGCOMM*, 2007.

[8] J. Liu, D. Goeckel, and D. Towsley, "Bounds on the gain of network coding and broadcasting in wireless networks," in *Proc. IEEE INFOCOM*, 2007.

[9] S. Sengupta *et al.*, "An analysis of wireless network coding for unicast sessions: The case of coding-aware rotuing," in *Proc. IEEE INFOCOM*, 2007.

[10] Y. E. Sagduyu and A. Ephremides, "On joint MAC and network coding in wireless ad hoc networks," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, pp. 3697–3713, Oct. 2007.

[11] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proc. MobiCom*, 2007.

[12] W. Chen, K. Letaief, and Z. Cao, "Opportunistic network coding for wireless networks," in *Proc. IEEE International Conference on Communications (ICC)*, 2007.

[13] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "On MAC scheduling and packet combination strategies for practical random network coding," in *Proc. IEEE International Conference on Communications (ICC)*, 2007.

[14] J. Le, J. C. Lui, and D. M. Chiu, "Towards coding-efficient link-scheduling and coding-aware routing in wireless networks," in *Proc. International Conference on Network Protocols (ICNP)*, 2007.

[15] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Prentice-Hall, 1987.

[16] "Part 11: Wireless LAN medium access control and physical layer (PHY) specifications," IEEE 802.11-1999 Standard, Aug. 1999.

[17] Y. E. Sagduyu and A. Ephremides, "Crosslayer design for distributed MAC and network coding in wireless ad hoc networks," in *Proc. IEEE International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005.

[18] L. Kleinrock and F. A. Tobagi, "Packet radio in radio channels, part 1: Carrier sense multiple access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. 23, no. 12, pp. 1400–1416, Dec. 1975.

[19] O. Tickoo and B. Sikdar, "A queueing model for finite load IEEE 802.11 random access MAC," in *Proc. IEEE International Conference on Communications (ICC)*, 2003.

Antonios Argyriou (S'00, M'07) received the Diploma in electrical and computer engineering from Democritus University of Thrace, Greece, in 2001, and the M.S. and Ph.D. degrees in electrical and computer engineering as a Fulbright scholar from the Georgia Institute of Technology, Atlanta, USA, in 2003 and 2005, respectively.

Currently, he is a Research Scientist at Philips Research, Eindhoven, The Netherlands. From 2004 until 2005, he was also a research engineer at Soft.Networks, Atlanta, GA, and was involved in the design of MPEG-2 and H.264 video codecs. His current research interests are in the areas of wireless communications and networking. He has published more than 30 papers in international journals and conferences. He is a member of IEEE.