# CROMA: An Enhanced Slotted MAC Protocol for MANETs

Marceau Coupechoux, Bruno Baynat, Christian Bonnet, Vinod Kumar

# CROMA - an Enhanced Slotted MAC Protocol for MANETs

Marceau Coupechoux
*Institut Eurecom and Alcatel Research & Innovation, France*
*Route de Nozay, 91461 Marcoussis cedex, France*
*Tel: +33 1 69 63 43 59, Fax: +33 1 69 63 43 60*
(marceau.coupechoux@alcatel.fr)

Bruno Baynat
*University Pierre et Marie Curie, Paris, France*

Christian Bonnet
*Institut Eurecom, Sophia-Antipolis, France*

Vinod Kumar
*Alcatel Research & Innovation, Marcoussis, France*

**Abstract.** TDMA based MAC protocols can provide a very good utilization of the shared radio resources, especially at high input loads, in synchronized mobile ad hoc networks (MANETs). Global positioning systems like GPS or GALLILEO should provide a very good timing accuracy for synchronization of nodes. This paper presents a new medium access protocol for mobile ad hoc networks, called CROMA. CROMA is collision-free and receiver-oriented. It operates in a slotted environment, in a dynamic and distributed way. In this protocol, receivers act as local base stations and can manage one or several communications on a single slot. Thus, sophisticated functions are allowed at higher layers. Moreover, the hidden terminal as well as the exposed terminal problems are handled by CROMA. A theoretical analysis and extensive simulations show that CROMA can reach very high throughputs.

**Keywords:** Mobile ad hoc networks, MAC, conflict-free protocol, scheduling, dynamic slot allocation, TDMA.

## 1. Introduction

In recent years a lot of effort has been spent in the design of protocols for mobile ad hoc networks. Such packet networks are mobile and multi-hop and operate without any fixed infrastructure. This can be a low cost and easily deployable technology to provide high speed Internet access in a wireless environment, to organize networks of sensors, or to complement the coverage of future cellular networks.

In this paper, we pay special attention to the medium access control (MAC) sub-layer. It has a lot of impact on the system performance and its design is a very challenging issue. MAC should control access to the medium and share the channel between source-destination pairs and/or flows of data in a dynamic and distributed way. Some desirable features of the access protocol are: to be able to reuse the resources as efficiently as possible, to avoid congestion and collisions, to be fair, reliable, and energy efficient.

Many MAC protocols try to address these issues. In the literature two categories of schemes have been proposed: (i) the contention based schemes; (ii) the conflict-free schemes.

In the contention based protocols, the channel has to be acquired by the nodes for each packet to be transmitted. Examples of contention based schemes are CSMA/CA, MACA [18], MACAW [5], FAMA [14], IEEE 802.11 [1]. The latter seems to be very popular in most of the testbeds because the IEEE 802.11 family products are available off the shelf. Although IEEE 802.11 is flexible, robust and simple, a recent paper [29] claims that it may not do very well in a multi-hop environment. According to [29], 802.11 has still the hidden terminal problem, does not handle the exposed terminal problem at all and its backoff strategy leads to severe unfairness. In this family of protocols, MACA-BI [26] was the first one to be receiver oriented, i.e., the transmission of a packet is initiated by the receiver that sends a short control packet in order to reserve the channel and to invite the sender to transmit. As the receiver does not have the exact knowledge of packet queue at the sender, it must rely on a traffic prediction algorithm.

On the other hand, conflict-free protocols allow the reservation of the channel for a certain amount of time or data and transmissions are conflict-free. TDMA deterministic scheduling may be preferred for networks with heavy load, carrying mixed traffic and realizing sophisticated functions at higher layers. That is the reason why we propose in this paper a slot allocation protocol for mobile ad hoc networks.

Unfortunately, most of the scheduling problems are NP-complete. For example, Arikan [2] has shown that constructing an optimal schedule for the point-to-point scheduling problem to optimize throughput is NP-complete. And this is the same for the broadcast scheduling problem based on throughput optimization, as proved by Ephremides and Truong [12]. Consequently, MAC designers have focused on suboptimal, dynamic and decentralized solutions for the slot assignment problem.

A first class of scheduling protocols relies on the allocation of priorities to nodes. A given slot is assigned preferably to the node with the highest priority according to its offered traffic. Slots can be allocated by using a control channel, e.g. in [6]. Priorities of the neighbors are assumed to be known at each node and are allocated in a pseudo-random way as in [4]. Then different strategies can be applied for the allocation of the priorities in order to have a fair and efficient share of the channel (see e.g. [23]). However, some of these protocols suffer from a high overhead due to the control channel. Others do not address the problem of the distributed and dynamic assignment of priorities.

On the other hand, time-spread protocols seem to be very attractive because they are topology-independent (see e.g. [7] or [17]). However, the frame length makes them less scalable and this class of protocols also faces the problem of distributed and dynamic code assignment.

At last, the necessity to address the problem of mobility, topology changes, and scalability, gives rise to a family of protocols where the reservation of the slots is done via a random access, most of the time a handshaking, combined with a carrier sensing mechanism. FPRP [30] proposes a five-phase handshaking supported by a pseudo-Bayesian algorithm to enable a faster convergence of the reservation procedure. CATA [27] uses four mini-slots in each time-slot to enable unicast and multicast transmissions. The protocol proposed in this paper comes within this family of protocols. It tries to make use of the advantages of the most popular contention based protocols to a slotted environment in order to increase their efficiency. In particular, the aim of CROMA is to achieve a high slot utilization, i.e., a high capacity, at high input load thanks to an original reservation and polling scheme.

The paper is organized as follows. In section 2, we give a precise description of our proposed MAC protocol. We examine the correctness of this protocol in section 3. Section 4 gives an analytical study of the protocol in a fully connected network. At last, section 6 is the conclusion of the paper.

## 2. Protocol Description

The Collision-free Receiver-Oriented MAC (CROMA) is a medium access protocol for mobile ad hoc networks that schedules transmissions in a slotted environment. It is a dynamic and distributed protocol that operates on a single-frequency channel with omni-directional antennas. CROMA has been shortly presented in [8] and [9]. The present paper gives a full description of the protocol, integrates new advanced features, and provides an enhanced performance analysis.

In CROMA, time is divided into frames, each of them divided into a fixed number $L$ of time-slots. Each slot can be temporarily and locally attributed to the receiver of a communication link depending on topology changes and traffic patterns. When a receiver is occupying a slot, it is allowed to poll several senders among its neighbors. The number of current communications for each slot is however limited by the protocol to a pre-defined value $K$.

The polling packet sent by the receiver is used to reserve the channel and to invite a sender to send a data packet. In that sense, CROMA is a receiver-oriented protocol since a slot in the frame is associated to a single receiver.

CROMA doesn't rely on a traffic prediction algorithm at the receiver. Indeed, a requesting node has to reserve resources at its intended receiver during a random access phase. This reservation is needed only at the beginning of a packet train (or message). When a receiver has no longer traffic to poll, communications are released and the slot is free for another receiver.

### 2.1. Frame Structure

CROMA divides time into frames that are, in turn, divided into $L$ equal time-slots. All mobile nodes are assumed to be perfectly synchronized.

Synchronization is a very critical issue for CROMA as for all distributed TDMA systems. A possible solution, now at low cost, consists in making use of the GPS (Global Positioning System) that provides a global synchronization for all nodes. Also the european satellite navigation system, GALILEO, will provide a very good timing accuracy [13]. In this case, guard intervals have to be foreseen. Another way of research is local synchronization, where nodes try to synchronize themselves by exchanging beacons with their neighborhood [11] [10]. The algorithms proposed in the literature can be adapted in order to be used with CROMA. However, as in [30] and [27], this paper focuses on the protocol description and considers that synchronization is a realistic assumption.

Throughout this paper, the following terminology has been chosen: a *requesting node* is a node that has data packets to send but has not yet succeeded in the reservation phase, its *intended receiver* is the destination node of these data packets. A *sender* is a node that succeeded in the reservation phase and that transmits data packets when it is polled by the receiver. A *receiver* is a node that polls senders on a given slot. At last, we will clearly distinguish the sender/receiver pair of a communication as defined earlier from the source/destination pair of a packet, that can be different for control packets.

Each time-slot is divided into three parts: two mini-slots, called *REQ-mini-slot* (request) and *RTR-mini-slot* (ready to receive) for the signaling, and a data transmission phase, called *DATA-mini-slot* (see Figure 1).

The REQ-mini-slot is used by requesting nodes during the random access phase for sending a REQ to its intended receiver. The RTR-mini-slot is used by their intended receivers to acknowledge requests as well

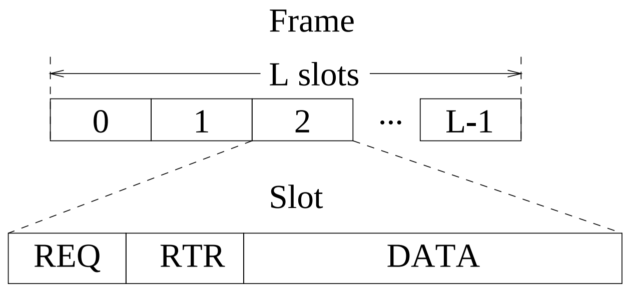



*Figure 1.* Frame structure of CROMA.

as previous data transmissions, and to poll one of the senders that previously managed a successful reservation. During the DATA-mini-slot, the sender that has been polled in the RTR-mini-slot transmits a data packet. These data packets are of fixed length. Indeed, it is assumed that a higher layer is responsible for fragmentation and reassembly.

### 2.2. CROMA from an Example

Before going into more details in the protocol description, let us illustrate the key feature of CROMA that is to allow multiple reservations on the same slot. The receiver indeed maintains a list of senders that managed a successful reservation and will poll them in the successive frames. This feature is illustrated on Figure 2, which shows two successive reservations on the same slot $i$. In frame $j$, the REQ/RTR dialogue starts the connection between nodes A and B: A sends a REQ packet with its address. B sends back a RTR, that contains a field to acknowledge the reservation (ackreq), and a field to poll node A (pol). The RTR is also received by node C that is now aware of a communication on slot $i$ with B as receiver. During the data phase, A, that has just been polled by B, is allowed to transmit a packet to B with its address A and a sequence number (sn) 0. We say that B has got the floor on slot $i$. In frame $j+1$, C establishes a connection with B. With the RTR, node B acknowledges the reservation with the field ackreq, acknowledges the packet transmitted by node A in frame $j$, and polls node C. In frame $j+2$, B now polls A. With the RTR, it also acknowledges the data packet of C with sequence number 0. In frame $j+3$, node B polls node C and acknowledges the data packet of A with sequence number 1.

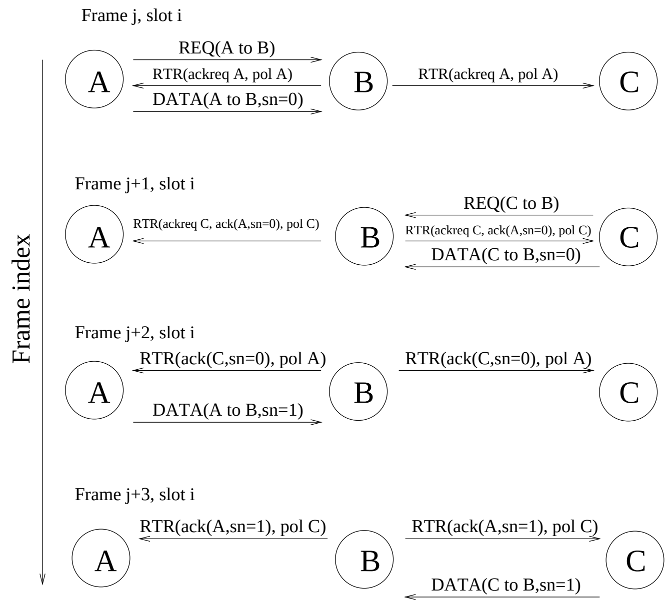



*Figure 2.* Example of two parallel connections on a slot with CROMA.

## 2.3. The choice of a receiver-oriented protocol

The choice of a receiver-oriented protocol is justified by the following arguments:

(i) this is a "natural" choice since only the zone that has to be secured with respect to collisions is the zone around the receiver, and thus, the spatial reuse of the radio resources is favored;

(ii) this choice allows the multiplexing of several communications on a single slot. That implies finer flow control and QoS negociation. If a slot is associated to a sender, it cannot easily multiplex communications with different receivers since they may not be available because of a hidden terminal;

(iii) if a slot is associated to a receiver, a current communication on a given slot does not prevent a random access on this slot. More bandwidth for the contention for the channel implies less collisions and interference. If a slot is associated to a sender, it has to send at each frame a control packet (RTS) to give the address of its intended receiver. Moreover, the receiver has to respond with another control packet (CTS) in order to avoid the hidden terminal problem. In CROMA, once the reservation has been done, the REQ is not used any more for the duration of the communication, and the REQ-mini-slot can be used for new reservations.

## 2.4. Packet Formats

This section describes the different packet formats and the MAC header of the data packets. It gives also the definition of all the MAC fields. Their signification will become clearer in the protocol description (sections 2.5, 2.6, and 2.7)

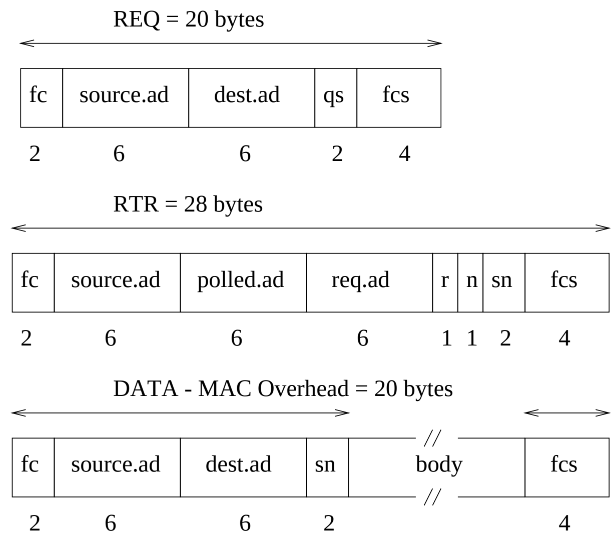



*Figure 3.* Packet formats of CROMA.

### 2.4.1. *Common Parts*

In Figure 3, the control packet formats and the MAC header of the data packets are shown. In all packets, generic information not described in this paper, like the protocol version, are given in the field *fc* that stands for *frame control*. The field *fcs* (frame check sequence) contains a CRC (cyclic redundancy code) calculated on all the fields of the MAC header and on the frame body. The field *source.ad* gives the ethernet address of the source of the packet.

Note that all packets, including data packets, have a fixed size, and each mini-slot is just long enough to allow the transmission of the associated packet. For example, the time to transmit a REQ including additional bits from the physical layer, the transmit-to-receive turn around time, as well as a small time interval to take into account the propagation delays equal the time of the REQ-mini-slot. Note also that it is preferred that the size of the control packets are short compared to the length of the data packets (e.g. 512 bytes).

2.4.2. *REQ Control Packet*

In a REQ, the field *dest.ad* gives the ethernet address of the destination of the packet (the intended receiver). The field *qs* is used by a requesting node to indicate to the intended receiver the requested quality of service for the communication. This field may be used by higher layers to negociate the QoS. It will be used in future versions of the protocol.

2.4.3. *RTR Control Packet*

A RTR has three different functions, as illustrated in section 2.2 and on Figure 2: respond to a REQ, poll the different senders on the current slot and acknowledge data packets.

In the RTR, the fields *req.ad* and *r* are used to reply to the requests sent on the same slot (during the REQ-mini-slot). If a request is correctly received and accepted, it is acknowledged by putting the address of the requesting node in the field *req.ad* and the value $ACK$ in the field *r*. If a request has been correctly received, but the communication can't be established, the field $r$ is set to $NACK$. This situation is possible if the requested QoS is not allowed or if the number of current communications has reached its maximum, $K$. If the receiver detects a collision of REQs, $r$ is set to $COL$. If the receiver didn't received any request, or if the request can't be decoded because of the channel conditions, $r$ is set to $NOTRECVD$. The values $NACK$, $COL$, and $NOTRECVD$ are useful information for the requesting nodes to reschedule their requests.

The field *polled.ad* is used by a receiver to poll a sender that previously managed to establish a connection on this slot. If a sender reads its address in the field *polled.ad*, it is allowed to send a data packet during the DATA-mini-slot of the same slot, just after receiving the RTR.

The acknowledgement of data packets is done thanks to the field *sn* that stands for *sequence number*. Each node maintains a counter that is incremented for each new data packet. Receivers keep the last received sequence number. If in time-slot $i$ of frame $j$, a receiver has received a data packet with sequence number $m$, it sets the field *sn* to $m$ in the RTR of the slot $i$ of frame $j+1$ and so, acknowledges the previous data packet.

The byte $n$ of a RTR gives information about the slot utilization. It is decomposed into seven bits that indicate the number $k$ of current communications, and one bit $t$ to inform that the receiver will not accept requests on this slot anymore. More details on the use of the bit $t$ for fairness are given in section 2.8. If $k$ has reached the maximum $K$

or if the bit $t$ is set to 1, no more request can be done on this slot.

#### 2.4.4. *Data Packets*

In data packets, the field *dest.ad* gives the address of the destination of the packet.

As previously explained, each sender maintains a counter that is incremented for each new packet. This sequence number is put in the field *sn* and is used by the receiver to acknowledge the reception of the packet. Let us recall that data packets have a fixed size, that results of a higher layer segmentation or aggregation.

### 2.5. Reservation

Any communication between two nodes must be preceded by a preliminary reservation phase. In the reservation phase, requesting nodes contend to get access to a receiver. This access is done in a random way during the REQ-mini-slots and consists of five sub-phases: listening of an entire frame, choice of a time-slot, transmission of the REQ on the chosen slot, listening of the RTR, and retry of a new reservation phase in case of failure (with or without random backoff). These five sub-phases are now detailed.

#### 2.5.1. *Frame Listening*

The first phase of the reservation consists in listening to the RTR-mini-slots during an entire frame, and maintaining for each slot in the frame the state of the slot. This listening process starts at the beginning of the reservation phase and lasts until the reservation has succeeded.

A slot can be in several states:

*FREE*: no activity has been sensed during the RTR-mini-slot, i.e., no receiver has got the floor on this slot. A request will be possible on this type of slot.

*OCC-NA*: i.e., occupied and not available. This is the case if a RTR has a *source.ad* different from the address of the intended receiver or if the requesting node has detected a collision during the RTR-mini-slot, or if it didn't managed to decode the field *source.ad* in the RTR, or if the requesting node is itself a receiver on this slot. This is also the case if the field $k$ of byte $n$ has reached the maximum number of communications on a slot or if the bit $t$ of byte $n$ is equal to 1. Note that a RTR collision detected on a slot does not necessary mean that the slot is free in a multi-hop situation. A request won't be possible on this slot.

*OCC-A-COL-k*: i.e., occupied, available, collision, and $k$ communications. In this case, the *source.ad* of the RTR is the address of the intended receiver, a collision has been detected by the receiver during the REQ-mini-slot ($r = COL$ in the RTR), and there are currently $k < K$ communications on the slot. A request will be possible on this slot.

*OCC-A-NCOL-k*: i.e., occupied, available, no collision, and $k$ communications. In this case, the *source.ad* of the RTR is the address of the intended receiver, no collision has been detected by the receiver during the REQ-mini-slot ($r \neq COL$ in the RTR), and there are currently $k < K$ communications on the slot. A request will be possible on this slot.

It is important to emphasize that the slot states are updated continuously during the whole reservation phase. In order to reduce the energy consumption, slot states updates can be however limited to a few frames before the reservation process.

### 2.5.2. *Choosing a Time-Slot*

The choice of the time-slot depends on the chosen scheduling policy. This policy may have several objectives. For example, it may maximize the slot utilization, limit the amount of interference in the network, establish connections that are robust to mobility. The impact of this choice is not detailed in this paper. We present here a simple policy that favors free slots first and therefore, aims at maximizing the slot utilization:

1. If there is at least one slot in state *FREE*,
   choose one randomly and exit, otherwise go to step 2;
2. If there is at least one slot in state *OCC-A*,
   select the slots having the lowest value of $k$. Among slots in this set:
   a) If there is at least one slot in state *OCC-A-NCOL*,
      choose one randomly and exit, otherwise go to step 2.b;
   b) Otherwise, choose one slot in state *OCC-A-COL* randomly and exit;

   Otherwise restart the reservation phase at the next frame.

### 2.5.3. *Transmission of the request and RTR generation*

On the chosen slot, the reservation is done by sending a REQ during the REQ-mini-slot. Two cases must now be considered:

(i) the sender has chosen a free slot. If the intended receiver can decode the REQ, it replies to the request by sending an RTR in the same slot and by using the fields *req.ad* and *r* of this packet, as explained in Section 2.4.3. Otherwise, the intended receiver doesn't reply. (Note however that the intended receiver may be aware that the slot is occupied, which can happen in a hidden terminal configuration. In this case, the receiver doesn't answer to the request. See Section 3 for more details.)

(ii) the sender has chosen a slot that is already occupied by the intended receiver. In this case, the intended receiver replies with an RTR whether it can decode or not the REQ.

### 2.5.4. *Listening of the RTR and decision*

A requesting node that has sent a REQ during the first mini-slot of the chosen slot listens to the following RTR-mini-slot. Table I gives a summary of the decisions of the requesting node after the RTR-mini-slot.

If the field *req.ad* has been set to its address and $r$ to $ACK$, the requesting node enters the transmission phase. If $r$ indicates a collision, the random backoff algorithm is started. In all other cases, the requesting node is allowed to restart the reservation phase at the next frame. The random backoff algorithm is thus only used when a high load is detected for the intended receiver.

Table I. Decision of a requesting node after listening to the RTR-mini-slot

| **Reception** | *req.ad* | $r$ | **Decision** |
|---|---|---|---|
| RTR decoded | my_address | $ACK$ | enter the transmission phase |
| | my_address | $NACK$ | retry on next frame |
| | not my_address | - | retry on next frame |
| | broadcast_address | $NOTRECVD$ | retry on next frame |
| | broadcast_address | $COL$ | start backoff algorithm |
| RTR not received or decoded | - | - | retry on next frame |

### 2.5.5. *Backoff Algorithm*

The backoff algorithm starts when a requesting node has been informed that a collision occurred. An integer $BO$ is randomly chosen between 1 and $BACKOFFWND$. This is a timer that is decremented at the beginning of each frame and each time the requesting node senses a slot in state *OCC-A* or *FREE*. As soon as $BO$ reaches 0, a slot is chosen on the forthcoming frame according to the scheduling policy for a new request. With this algorithm, the load on the available slots is taken into account.

The parameter $BACKOFFWND$ is increased by a multiplicative factor (1.5) at each successive retransmission and decreased by one at each success. However, there are a lower and an upper bound for it, called $BOmin$ and $BOmax$, e.g., 2 and 32.

### 2.6. Transmission

A sender whose request has been successful on a given slot starts its transmission phase. During a transmission phase, receivers of which resource has been reserved in the reservation phase, do a polling among their associated senders. When a sender recognizes its address in the field *polled.ad* of the RTR, it sends in the same slot a data packet during the DATA-mini-slot.

Each sender maintains a counter of its transmissions that is incremented at each new packet. This sequence number is copied in the field *sn* of the packet header. With this method, the receiver is able to acknowledge the last correctly received data packet. For that, a receiver copies in the field *sn* the sequence number of the last received packet. At the sender side, a sent data packet is stored until the receipt of the acknowledgement. If the next RTR is not received or if this RTR does not acknowledge the stored packet, a retransmission is necessary. After $M$ retransmissions the stored packet is thrown away. This loss can be treated by an upper layer.

Figure 4 shows an example of a transmission phase with a receiver and three senders. It only shows slots $i$ of successive frames. On the upper part of the figure, the RTRs of the receiver are represented with the fields *polled.ad* and *sn*. A cyclic polling is pictured for the scheduling of the senders and data packets are shown with their field *sn*.

It is clear that each receiver acts on a given slot as a local base-station with respect to its associated senders. Thus, the polling mechanism allows a high flexibility for the scheduling of different flows by higher layers and is a base for the implementation of QoS algorithms. Moreover, several parallel communications are possible on a given time-slot.

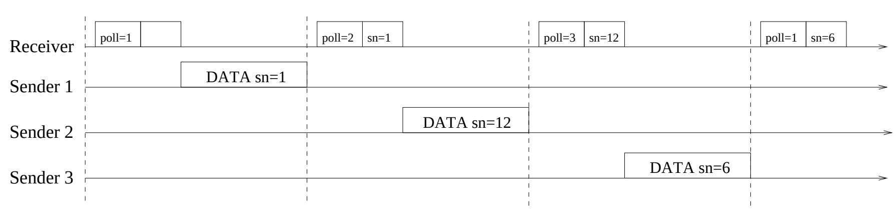



*Figure 4.* Polling during the transmission phase.

### 2.7. Release

An established communication can be interrupted in three cases:

(i) the sender informs the receiver that it sends the last packet of the communication by setting the field *sn* of the packet's header to the value *EOT* (end of transmission). If the last packet is correctly received, the receiver does not re-schedule the sender any more. However, it acknowledges the last packet with its next RTR, and this, even if it has no sender to poll;

(ii) if a receiver has polled a sender and doesn't receive any packet from the sender, a counter set to $W$ is decremented. When this counter reaches 0, the communication is released, and the receiver doesn't poll the sender any more. If after a poll, a packet is received, the counter is set again to $W$. After each polling, a sender starts a timer. If it doesn't receive any polling from the receiver when the timer expires, the connection is considered to be broken;

(iii) during a communication, a sender may receive several RTRs, i.e., there is a collision of RTRs. In this case, the sender considers that the current communication on this slot is released. Indeed, sending a data packet could imply a collision during the DATA-mini-slot. More precisions about this specific aspect are given in section 3.

### 2.8. Fairness Issue

CROMA includes a mechanism to assure a local fairness among data flows. On a given time-slot, fairness among incoming flows is assured by the receiver of the slot by means of the RTRs. By using different polling strategy, a receiver can easily give a fair allocation of the slot to incoming flows.

However, if the number of slots in the frame is small compared to the number of potential receivers, situations of unfairness can arise and flows can be completly starved. The bit $t$ included in the RTRs is used in order to avoid such situations.

A receiver having the floor on a given slot counts the number of consecutive full frames. A frame is full from the point of view of a receiver, if it senses ativity at each slot of the frame. In this case, it detects a potential blocking situation for pair of nodes that cannot communicate because there are no free slots any more. If the number of monitored full frames reaches $MAX\_FULLFRAMES$, the receiver sets the bit $t$ to 1 indicating that it will not accept new requests and that the current communications have to be released.

A sender detecting a bit $t$ set to 1, sets the field *sn* of its next packet's header to *EOT* and stops sending packets to the receiver. This release is done even if the sender have still packets to transmit. A requesting node detecting a bit $t$ set to 1 in a RTR update the slot state to *OCC-NA*.

This strategy aims at avoiding blocking situations that can lead to unfairness. Indeed, these cases are detected by the receivers that have to free their slot if the situation lasts.

## 3. Correctness

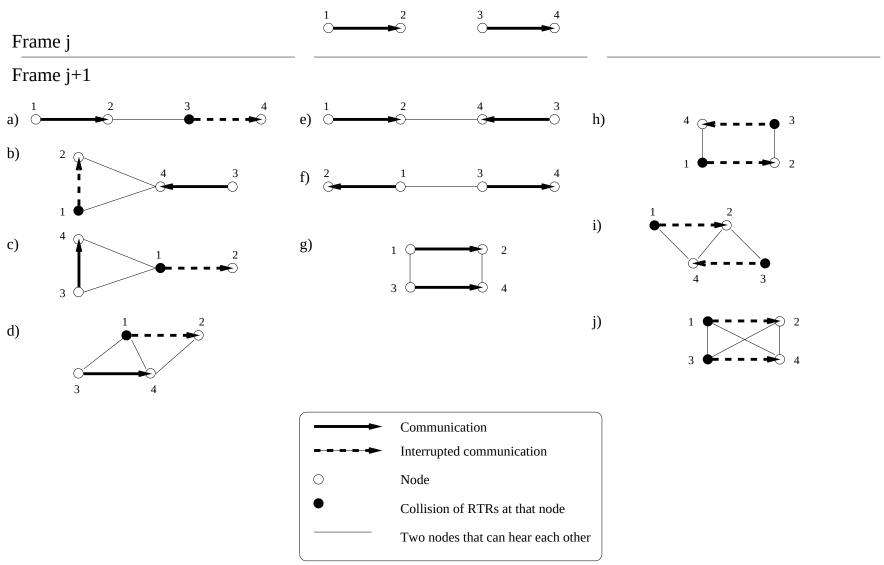



*Figure 5.* Interference between two communications sharing the same slot.

In this section, we will show that CROMA is correct, i.e., that it is collision-free in both fixed and mobile environment. The capture effect is not considered here, so this section shows that CROMA is collision-free in the common case provided that a sender releases its communication as soon as it detects a collision of RTRs.

Let us first consider a fixed and multi-hop topology. We now prove that two data packets cannot collide.

We suppose that a collision of two data packets occurs at a receiver $R_1$. These packets have been sent by two different senders, namely $S_1$ and $S_2$. During the RTR-mini-slot, $R_1$ specified the MAC address of the sender, say $S_1$, that was allowed to send its data in the current slot. As the MAC address is unique, a single colliding data packet is destined to $R_1$. Therefore, we know that the data packet of $S_2$ was destined to another receiver, $R_2$.

Now, as $R_1$ has received a data packet from $S_2$ and links are bidirectional, $S_2$ has received the RTR of $R_1$. Moreover, $S_2$ has also received a RTR from $R_2$, since it sent a data packet destined to $R_2$. Thus, $S_2$ has detected a collision of RTRs in the current slot without interrupting its communication with $R_2$. This is impossible. As a conclusion, no data collision can occur in a fixed topology.

Let us now consider the case of a dynamic topology. Two concurrent communications on a slot are shown on the top of Figure 5, from node 1 to node 2 and from node 3 to node 4. These communications are sharing the same slot in frame $j$ and they are far away enough, so that they don't interfere. In case of mobility and at the next frame $j+1$, node 3 can either stay out of range of nodes 1 and 2, enter the communication range of 1, 2, or both 1 and 2. Same alternatives can occur for node 4. Thus, after mobility, a total of 16 relative new positions are possible. Because of the symmetry of the problem, only 10 cases are shown on Figure 5.

The left hand side of Figure 5 shows situations, where a single communication is interrupted because the sender detected a collision of RTRs on the considered slot. For example, in case b, node 4 moved in the transmission range of nodes 1 and 2. In frame $j+1$, nodes 2 and 4 send simultaneously an RTR. Node 3 receives correctly the polling of 4, whereas node 1 senses a collision during the RTR-mini-slot. Node 1 decides to interrupt the communication with node 2 and does not send any data packet on this slot. If node 1 has still packet in its buffer, it has to enter a new reservation phase.

The central part of Figure 5 shows exposed-terminal topologies, where both communications can still share the same slot. In case e, node 4 moved in the transmission range of node 2. In frame $j+1$, node 1 (resp. 3) decodes the RTR of node 2 (resp. 4) because it is out of the transmission range of node 4 (resp. 2). Both nodes 1 and 3 can send data packet during the DATA-mini-slot.

The right hand side of Figure 5 shows topologies, where communications are released because both senders detected a collision of RTRs. Case j shows a configuration where the network of nodes is fully connected after mobility. Here, RTRs of nodes 2 and 4 collide at nodes 1 and 3. On detecting the collision, they decide to interrupt their communication.

So, in the common case, in both fixed and mobile environment, CROMA is collision-free. As in all protocols that rely on the exchange of short control packets, the capture effect may however affect this conclusion.

## 4. Analytical Study

In this section we calculate the approximate throughput, i.e., the slot utilization of the protocol CROMA in a fully connected network. Following [27], we claim that this topology is the worst case in terms of interference, contention, and spatial reuse because CROMA guarantees a collision-free transmission of data after reservation in a multi-hop environment.

### 4.1. Model for the Slot Utilization Analysis

First of all, we describe our analytical model for the slotted MAC protocol CROMA. From this model will be derived the slot utilization of CROMA as a function of the probability $p$ to send a REQ for a given source-destination pair. Let's enumerate the hypothesis of our model.

1. We consider a fully-connected network of $N$ synchronized nodes;

2. all packets are of constant length and are transmitted over an assumed noiseless channel;

3. there are $L$ slots per frame;

4. the maximum number of connections on a slot is $K$, i.e., when a receiver is already polling $K$ different senders on a slot, no new REQ is allowed;

5. a receiver can only be associated with a single slot. This hypothesis can be in practice relaxed, but for the sake of tractability of the model, we limit the analysis to this case;

6. a node can be a sender on several slots of the frame. While being in communication on a slot, a node can send a REQ on another slot of the frame to start another connection;

7. the traffic between any two nodes $s$ and $d$ is a ON/OFF traffic;

8. the ON periods are modeled by bursts of packets following a geometrical distribution. The length of a message follows a geometrical law with parameter $q$. Thus, the average message length (AML) is $1/(1-q)$;

9. the OFF periods are modeled by series of slots without transmission following a geometrical distribution. If a source $s$ doesn't communicate with a destination $d$, there is a probability $p$ that $s$ wants to communicate with $d$ at the next frame;

10. a non persistent policy is assumed for retransmissions after a failure. This hypothesis explains that we can consider a fixed probability $p$ to start a communication.

The system is described by the number of parallel connections on the slots at the end of the frame, $(a_0, a_1, ..., a_{L-1})$, where

- $a_i$ is the number of current connections on slot $i$,
- $0 \leq a_i \leq MIN(K, N-1)$ (see hypothesis 1 and 4),
- $S = \sum_{i=0}^{L-1} 1_{\{a_i>0\}} \leq MIN(N, L)$, (see hypothesis 3 and 5).

For the sake of simplicity, the states describe neither the receiver associated to each slot, nor the list of associated senders. The vector $(a_0, a_1, ..., a_{L-1})$ is a discrete-time stochastic process, whose state space is also discrete. Moreover, this process is independent of its history because the geometric law is memoryless. Consequently, this process is a discrete time Markov chain (DTMC). Since the state space is aperiodic and finite, the chain is always ergodic.

From a frame to another, we can have the following transitions on slot $i$:

- $a_i \rightarrow a_i + 1$ $(a_i < K)$: a reservation has been successful on slot $i$ AND no communication has come to the end,
- $a_i \rightarrow a_i$: (there is a successful reservation AND this is the end of a communication) OR (there is no successful reservation AND no message is ending),
- $a_i \rightarrow a_i - 1$ $(a_i > 0)$: there is no successful reservation AND this is the end of a communication.

A transition probability between the two states $(a_0, a_1, ..., a_{L-1})$ and $(b_0, b_1, ..., b_{L-1})$ is assumed to be the product of the transition probabilities associated to each slot:

$$P\left((a_0, a_1, ..., a_{L-1}) \rightarrow (b_0, b_1, ..., b_{L-1})\right) = \prod_{i=0}^{L-1} P(a_i \rightarrow b_i). \tag{1}$$

Results will show that this assumption is a good approximation.

### 4.2. One Slot Analysis

In this section, $L = 1$. In this simple case, we can derive a closed-form formula for the slot utilization.

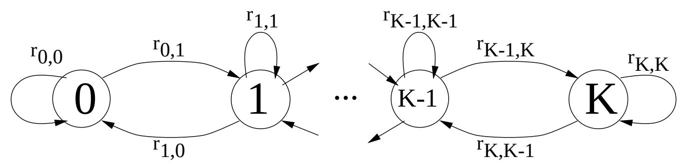



*Figure 6.* Discrete time Markov chain representing the state of the slot, for $K \leq N$.

The system is described by the number of parallel connections on the considered slot at the end of the frame (the DTMC is shown on Figure 6). Let's now compute the transition probabilities $r_{i,j}$ of this Markov chain. Remember that the probability for a source-destination pair to enter a ON period is $p$. Thus, the probability that a node sends a request on a free slot is the probability that this node has a request for at least one of the destinations:

$$p' = 1 - (1-p)^{N-1} . \tag{2}$$

Thus, on a free slot, a successful reservation occurs iff only one single node among $N$ is sending a request during the REQ-mini-slot. Consequently the probability to have a successful reservation on a free slot is

$$\theta(0) = \binom{N}{1} p'(1-p')^{N-1} . \tag{3}$$

On an occupied slot with $n$ connections, a receiver has got the floor on the slot and successively polls $n$ senders that managed to reserve resources. Here, a successful reservation occurs iff only a single node among the $N-(n+1)$ nodes not currently in connection is sending a request. Therefore, the probability to have a successful reservation on an occupied slot is

$$\theta(n) = \binom{N-(n+1)}{1} p(1-p)^{N-(n+1)-1} . \tag{4}$$

In state $0 \leq n < K$, there is a transition to state $n+1$ iff a successful request is received and this is not the end of the current communication. The transition state $r_{n,n+1}$ is thus given by:

$$r_{n,n+1} = \theta(n)q . \tag{5}$$

In state $0 < n < K$, there is a transition to state $n-1$ iff there is no successful request and this is the end of a communication, so

$$r_{n,n-1} = (1-\theta(n))(1-q) . \tag{6}$$

From these two equations, we obtain directly $r_{n,n}$ for $0 < n < K$:

$$r_{n,n} = 1 - r_{n,n+1} - r_{n,n-1} \ . \tag{7}$$

In state 0, the slot is free and so $r_{0,1} = \theta(0)$ and $r_{0,0} = 1 - r_{0,1}$. In state $K$, $r_{K,K} = 1 - r_{K,K-1}$. The transition matrix is given by:

$$P = \{r_{i,j}\}_{0 \leq i,j \leq K} \ . \tag{8}$$

The stationary probabilities are obtained by solving the steady state equations $\vec{\pi} = \vec{\pi}P$, that enable to express all the probabilities in function of $\pi_0$:

$$\pi_n = \frac{\pi_0}{1-q}\left[\frac{q}{1-q}\right]^{n-1} \prod_{k=0}^{n-1} \frac{\theta(k)}{1-\theta(k+1)} \ , \tag{9}$$

for all $n \in \{1, \cdots, K\}$. The system is totally described with the following equation: $\sum_{n=0}^{K} \pi_n = 1$. At last, the slot utilization of the protocol is given by $U = 1 - \pi_0$:

$$U = 1 - \frac{1}{1 + \sum_{n=1}^{K} \frac{1}{1-q}\left[\frac{q}{1-q}\right]^{n-1} \prod_{k=0}^{n-1} \frac{\theta(k)}{1-\theta(k+1)}} \ . \tag{10}$$

Figure 7 shows the slot utilization of CROMA, $U$, as a function of the probability $p$ for $K = 3$, $N = 5$ and different average message length ($AML = 2$, 10 and 100 packets). Dotted curves have been obtained by simulations. These simulations reproduce the hypothesis of our model. We can see on the one hand that the approximations of the analysis have a small impact on the performance evaluation. On the other hand, it is clear that CROMA can achieve a very high slot utilization provided that the average message length is high.

From the DTMC, the average number of connections, $N_c$ on the slot can also be derived:

$$N_c = \sum_{n=0}^{K} n\pi_n \ . \tag{11}$$

Figure 8 shows the average number of connections for different AML values. This mean number is clearly related to the delay of transmission of a burst because the higher the number of connections on a slot is, the smaller is the resource allocated to a single connection. Thus, a trade-off has to be made between slot utilization and delay.

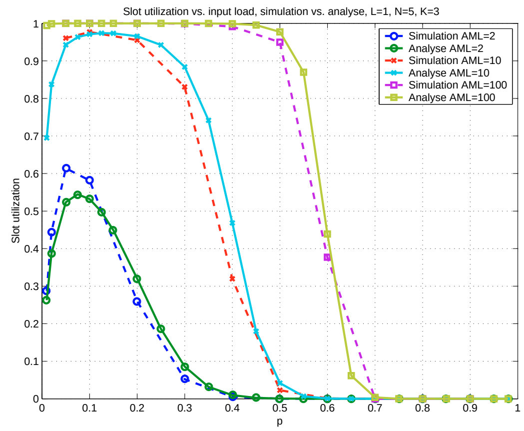



*Figure 7.* Slot utilization vs. input load, L = 1, N = 5, K = 3.

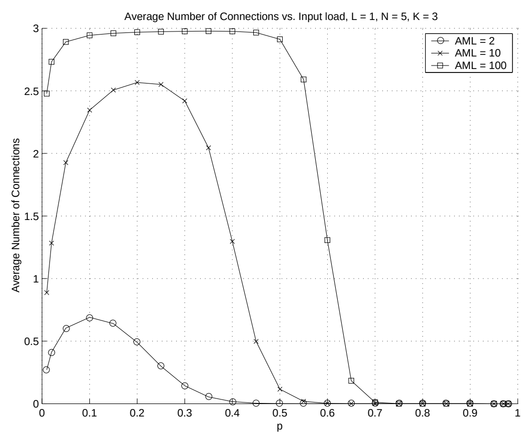



*Figure 8.* Average number of connections vs. input load, L = 1, N = 5, K = 3.

## 4.3. Multi-slot Analysis

In this section, we extend the previous result to the general case with $L$ slots. We first compute the transition probabilities, while distinguishing an occupied slot, a free slot and a full slot. For the sake of readability, we only consider the case $K \leq N$.

Let's consider a slot $i$ occupied by the receiver $d$ (this is the case, where $0 < a_i < K$).
The number of nodes that are likely to send a REQ to $d$ are nodes that are currently not in communication with $d$, their number is $N - 1 - a_i$. The probability for such a node $s$ to send a REQ on slot $i$ is $p$ (see

hypothesis 9). Thus, the probability of a successful reservation is:

$$\theta_i = \binom{N-1-a_i}{1} p\,(1-p)^{(N-1-a_i)-1} \ . \tag{12}$$

Note that if $a_i = N-1$, all nodes have a connection with the considered receiver, so that there is no REQ on this slot, and $\theta_i = 0$. Now the probability that a message is ending is (see hypothesis 8): $1-q$. We can now derive the transition probabilities for slot $i$:

$$\begin{aligned} P(a_i \rightarrow a_i + 1) &= \theta_i q & (13)\\ P(a_i \rightarrow a_i) &= \theta_i(1-q) + q(1-\theta_i) & (14)\\ P(a_i \rightarrow a_i - 1) &= (1-\theta_i)(1-q) & (15) \end{aligned}$$

Let's now consider a free slot $i$ ($a_i = 0$). There are $S = \sum_{i=0}^{L-1} 1_{\{a_i>0\}}$ occupied slots in the frame, i.e., $S$ receivers, since a receiver is associated to a single slot (see hypothesis 5).

On the considered free slot $i$, $N$ senders are likely to send a REQ for $N-S$ possible receivers. Indeed, a node is allowed to send traffic to several receivers in parallel on different slots, so all nodes are likely to start a new communication on $i$. Moreover, requests on $i$ can be addressed to any of the $N-S$ nodes that are not receivers on another slot because $i$ is not attributed.

Let's consider a node $s$. The probability that $s$ has $n$ REQ for the $N-S$ possible receivers is

$$p_1(n) = \binom{N-S}{n} p^n (1-p)^{N-S-n} \tag{16}$$

if $s$ also belongs to the $S$ receivers, and

$$p_2(n) = \binom{N-S-1}{n} p^n (1-p)^{N-S-n-1} \tag{17}$$

otherwise. Thus, the probability that $s$ has $n$ requests is:

$$p(n) = p_1(n)\frac{S}{N} + p_2(n)\frac{N-S}{N} \ . \tag{18}$$

Now, the probability that $s$ sends a REQ on the free slot $i$ is:

$$\begin{aligned} \beta &= \sum_{n=1}^{N-S} Pr[\text{s sends a REQ on i}|\text{s sends n REQ}]p(n) & (19)\\ &= \sum_{n=1}^{N-S} min\left(\frac{n}{L-S}, 1\right) p(n) \ . \end{aligned}$$

At last, there are $N$ possible senders like $s$, so the transitions probabilities for $i$ are:

$$P(0 \to 1) = \binom{N}{1} \beta(1-\beta)^{N-1} \tag{20}$$

$$P(0 \to 0) = 1 - P(0 \to 1) \tag{21}$$

Let's at last consider a full slot ($a_i = K$). The transition probabilities are obvious:

$$P(K \to K) = \theta_i(1-q) + q(1-\theta_i) \tag{22}$$

$$P(K \to K-1) = 1 - P(K \to K)\,. \tag{23}$$

The steady state equations $\vec{\pi} = \vec{\pi}P$ are solved using any numerical method, e.g., the iterative method of Gauss-Seidel (see [3] or [25]).

Figure 9 shows the slot utilization of CROMA as a function of $p$ for different average message lengths. Analysis and simulations (dotted lines) are compared and the figure shows a good adequation of the two methods. As for $L = 1$, we can see that CROMA can achieve very high slot utilization provided that the AML is high. Note that values of $p$ near 1 are not realistic in a real implementation because of the backoff algorithm. Simulations show that the point of operation of a highly loaded CROMA network with backoff is always for $p < 0.5$. Figure 10

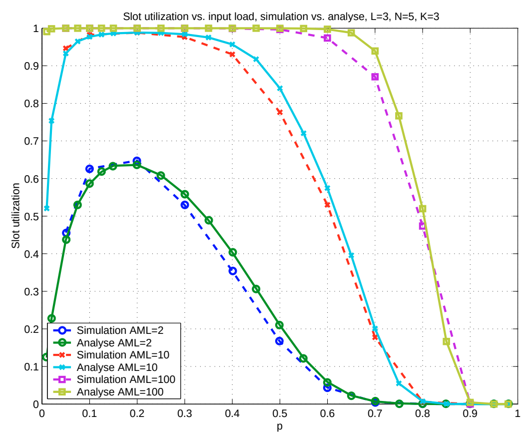



*Figure 9.* Slot utilization vs. input load, L = 3, N = 5, K = 3.

shows the influence of $K$ on the system performance. There is a clear gain of channel utilization as $K$ increases. However, this is obtained at the cost of higher delays. This is shown on Figure 11, where the average number of connections per slot is plotted. A higher number of connections per slot implies a higher delay for the burst transmissions.

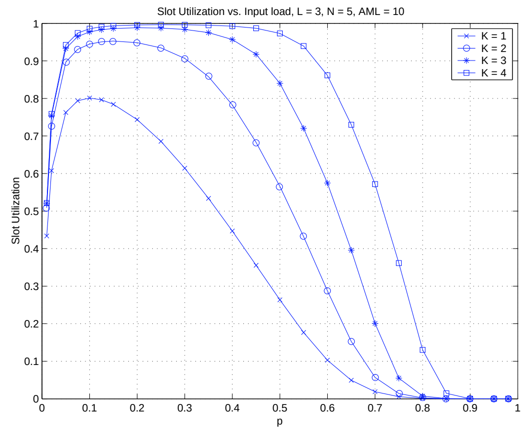



*Figure 10.* Slot utilization vs. input load, influence of K, L = 3, N = 5, AML = 10.

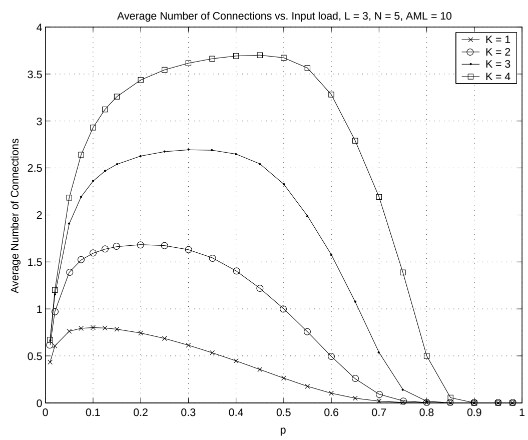



*Figure 11.* Average number of connections vs. input load, influence of K, L = 3, N = 5, AML = 10.

## 5. Performance Analysis in a Multi-hop Environment

In this section, we provide simulation results and the performance of CROMA and of the standard IEEE 802.11 (DCF mode) are compared.

### 5.1. Methodology

Studying MAC protocols in a multi-hop environnment leads to the problem of chosing an appropriate node's topology. Literature on ad hoc networks has solved the problem by considering on the one hand typical networks, like the string network, or the grid network, and on the other hand randomly generated networks. In this paper, we adopted part of the two approaches by running CROMA over a classical and challenging network and over a random network.

We will now describe the metrics used to evaluate the performance of the MAC protocols.

**End-to-end delay**: this is the average time spent by a packet from the traffic generator of a source to the reception module of the destination.

**End-to-end delay jitter**: this is the standard deviation of the end-to-end packet delay.

**Aggregate throughput**: this is the average number of bits successfully received by all nodes in the network per second. The input load is the average number of bits transmitted by all nodes per second.

**Fairness index**: this is the widely used index, $f$, defined in [16]. If a system allocates resources to $n$ contending entities, such that the $i^{\rm th}$ entity receives an allocation $x_i$, then:

$$f(x) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2} . \tag{24}$$

If all entities get the same amount, i.e., $x_i$'s are all equal, then the fairness index is 1 and the system is 100% fair. The choice of the metric depends upon the application. In our case, we will consider that the entities are the flows of data between source-destination pairs $(i, j)$ and the metric is their throughput, $T_{i,j}$.

## 5.2. Performance in a Challenging Environment

### 5.2.1. *Throughput and Delay Analysis*

In order to evaluate the performance of CROMA in ad hoc networks, we considered a very simple multi-hop situation that has been used in the literature for the evaluation of MAC protocols, e.g. in [15]. Nodes are assumed to be static, the traffic is ON/OFF with exponential distributions, and the packet size is set to 512 bytes. Moreover, the channel is supposed to be perfect with a physical data rate of 2 Mbps. The transmission area of a node is a disk of radius $R$. Outside of the transmission area no communication is possible. Simulations have been done using the Network Simulator v2 (ns2, see [21]). The simulation parameter values are presented in Table II. Note that the mean OFF time is fixed and that the mean ON time will vary in simulations.

In this configuration, eight nodes form a regular topology, flows of data are shown on Figure 12. Four end-to-end communications are running in parallel: 0-1-2-3, 0-5-2-7, 7-6-5-4, and 3-6-1-4, so that several nodes have to receive and/or to relay several flows of data. A solid line without arrow between two nodes means that they are in the communication range of each other, i.e., the transmissions from one of them can be successfully decoded by the other one. A solid line with arrow means that at least one flow of data is using this link.

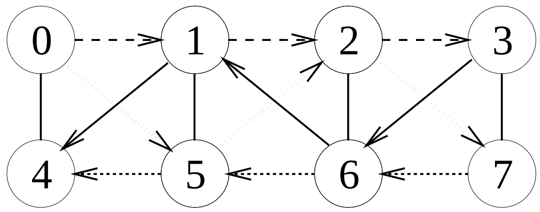



*Figure 12.* A mulithop topology, the "squares topology".

Table II. Main Parameter Values for Simulations

| **Parameter** | **Value** |
|---|---|
| DATA Packet size | 512 bytes |
| BOmin | 2 |
| BOmax | 64 |
| K | 3 |
| W | 3 |
| M | 7 |
| MAX_FULLFRAMES | 30 |
| Inter-mini-slot time | 10 $\mu$s |
| PHY overhead | 24 bytes |
| PHY Data Rate | 2 Mbps |
| ON distribution | Exponential |
| OFF distribution | Exponential |
| Peak Rate | 256 Kbps |
| Mean OFF time | 0.5 s |
| Simulation time | 200 s |
| Number of simulations per point | 10 |

This configuration is interesting for several reasons:

(i) it exhibits a lot of hidden terminal situations. For example, nodes 6 and 2 are hidden from node 0, nodes 7 and 3 are hidden from node 5;

(ii) spatial reuse is possible and there are situations of exposed terminal. For exemple, nodes 1 and 2 are exposed. Several flows can share the same slot, e.g., 1-4 and 2-7, or 4-0 and 7-3;

(iii) nodes and flows experience different contention situations, nodes 0, 3, 4, and 7 have three neighbors, while nodes 1, 2, 5, and 6 have five neighbors.

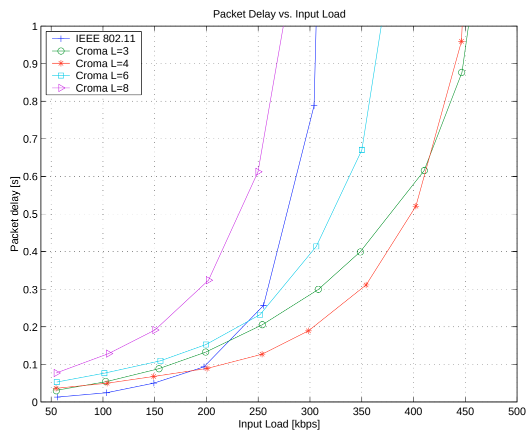



*Figure 13.* End-to-end delay vs. input load, squares topology.

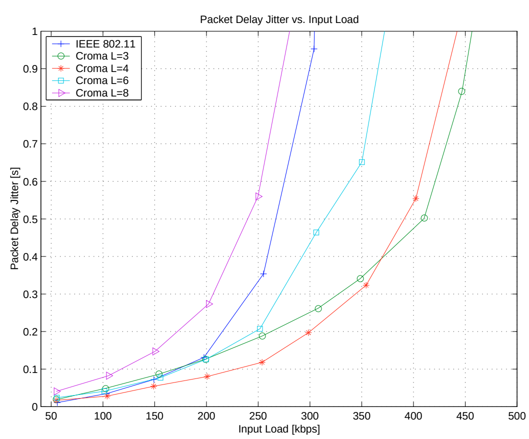



*Figure 14.* End-to-end delay jitter vs. input load, squares topology.

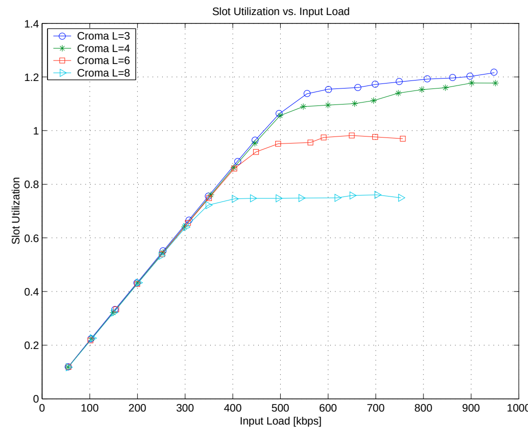



*Figure 15.* Slot utilization vs. input load, squares topology.

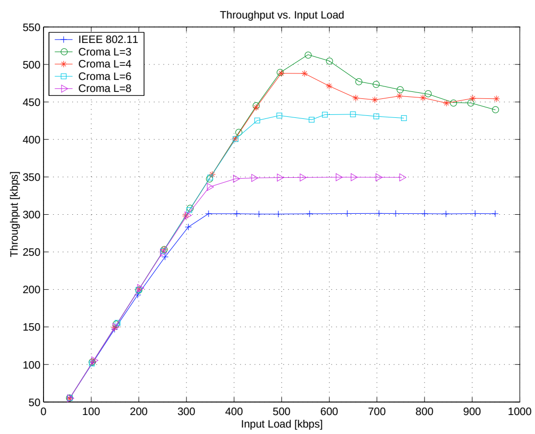



*Figure 16.* Throughput vs. input load, squares topology.

Figures 13 and 14 show the end-to-end packet delay and jitter as function of the input load for IEEE 802.11 and CROMA. The different curves for CROMA assume different number of slots per frame.

In the case of low input load, IEEE 802.11 outperforms CROMA because the low level of contention implies a small number of collisions and small backoff windows. At this level of load, the network cannot fully take advantage of the reservation scheme because trains of packets are small.

In the case of higher input load, IEEE 802.11 nodes experience more contention, and thus more collisions and wider backoff windows: the access delay increases drastically. On the other side CROMA takes advantage of packet bursts to reduce the number of requests per transmitted packet. If a flow has made a successful reservation, long trains of packets can be transmitted without contention. Delays and jitters of CROMA $L = 8$ remains however always above IEEE 802.11's performance. It is clear that CROMA $L = 8$ is not well dimensionned for the topology. Actually, the number of slots is too high and the resource is not fully exploited, as it is shown on Figure 15. To overcome this problem, a higher layer can split a link layer connection into two seperate CROMA connections. The slot utilization of CROMA $L = 8$ does not exceed 0.75. This is much less than CROMA $L = 6$ that reaches 0.97. CROMA $L = 3$ and 4 fully exploit spatial reuse and exceed 1.1.

The reservation scheme, the synchronization, and the ability of CROMA to handle the exposed terminal problem allow the network to achieve high throughputs. Figure 16 shows aggregate throughput as a function of the input load. IEEE 802.11 saturates at a throughput of 300 Kbps. In comparison, CROMA $L = 8$ achieves a maximum throughput 350 Kbps, although we have seen that it is obviously badly dimensionned for the topology. CROMA $L = 6$ reaches a maximum throughput of 425 Kbps. For less slots per frame, a problem of stability of the throughput arises. Although CROMA $L = 3$ and 4 achieve resp. 475 and 510 Kbps, the throughput decreases for input loads higher than 525 Kbps. Indeed, the small number of slots implies a slight instability with the considered topology. However, curves show a slow decrease leading to acceptable values even at high input load.

#### 5.2.2. *Fairness Analysis*

Without any fairness strategy and without the use of the bit $t$, blocking situations can lead to severe unfairness. This is particularly the case when the input load is high and the number of slots per frame is small for the considered topology/traffic pattern. For example in the topology of the Figure 12 with $L = 4$, if node 1 hears the RTRs of node 2 on slot 0, node 5 on slot 2, node 6 on slot 3, and sends RTRs on slot 1, 1 cannot send any REQ since the frame is full. In case of low input load, this situation is transient and has a low impact on the long term fairness. In case of high input load however, the connection 3-4 is completly starved leading to severe unfairness.

Figures 17 and 18 shows the benefit of use of the bit $t$ with the aforementionned fairness strategy. The fairness index of CROMA $L = 3$ and $L = 4$ are compared to the index of IEEE 802.11. For the IEEE standard and for CROMA without the use of the $t$, the index is close to 1 for low to moderate input load. After a threshold, the increase of input load leads to a drop of the index. This threshold is 350 Kbps for

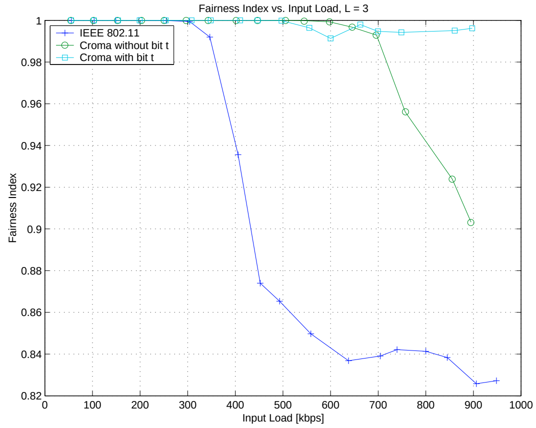



*Figure 17.* Fairness index vs. input load, L = 3, squares topology.

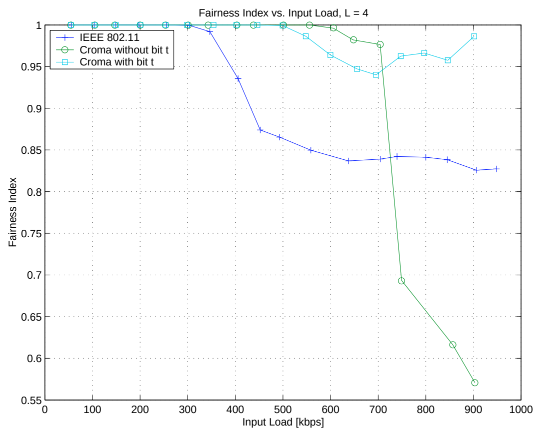



*Figure 18.* Fairness index vs. input load, L = 4, squares topology.

IEEE 802.11, and approximatly 700 Kbps for CROMA. With the use of the bit $t$, the fairness index of CROMA remains always above 0.95 for both $L = 3$ and $L = 4$.

### 5.3. Performance in a Random Network

In the previous section, we compared IEEE 802.11 and CROMA over a simple and pre-defined multi-hop topology. In this section, we consider a random connex network. 30 nodes are drawn at random in a 1000mx1000m square area, each node having a transmission range of 250 m. This network is shown on Figure 19. 10 connections are established between 10 random pairs of nodes. The traffic is assumed to be exponential ON/OFF with the same parameters as in the previous section. Figure 20 shows the aggregate throughput of the network as

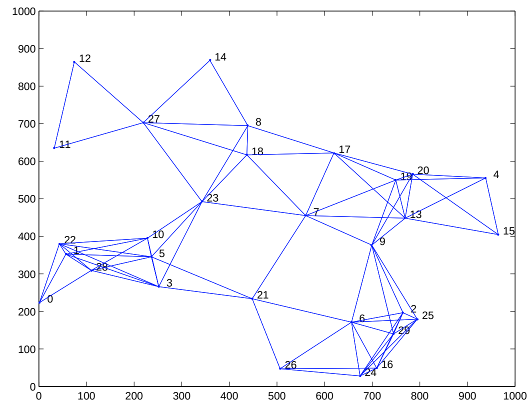


*Figure 19.* Random topology with 30 nodes in a 1000mx1000m area.

a function of the input load. While IEEE 802.11 and CROMA $L = 8$ saturate at a load of approximatly 500 Kbps, CROMA $L = 6$, $L = 4$, and $L = 3$ reach resp. 600, 700, and 750 Kbps.

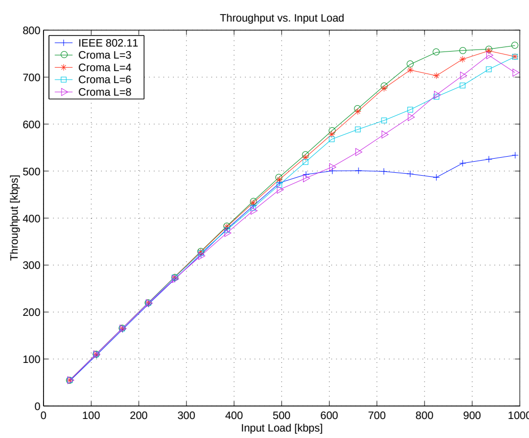



*Figure 20.* Throughput vs. input load, random topology.

Figure 21 shows the mean end-to-end delay of the data packets as a function of the input load. It is clear that the better performance of CROMA in term of throughput is obtained at the expense of higher packet delays and jitters at low input load (see Figure 22). In this case, IEEE 802.11 outperforms CROMA. However, CROMA allows to extend the area of acceptable delay and jitter by one third. For exemple, CROMA $L = 6$ still exhibits delays under 600 ms at an input load of

700 Kbps. Note also that at low input load, the frame length of CROMA has little influence on the end-to-end delay.

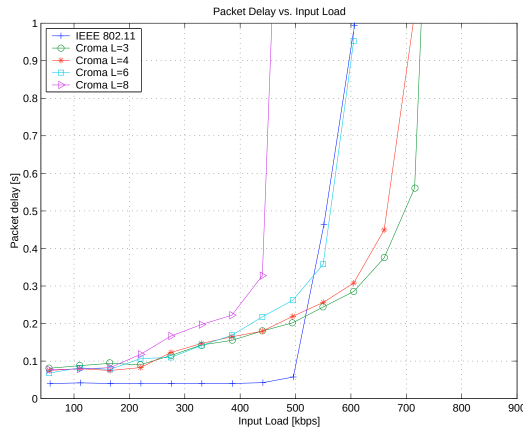



*Figure 21.* End-to-end delay vs. input load, random topology.

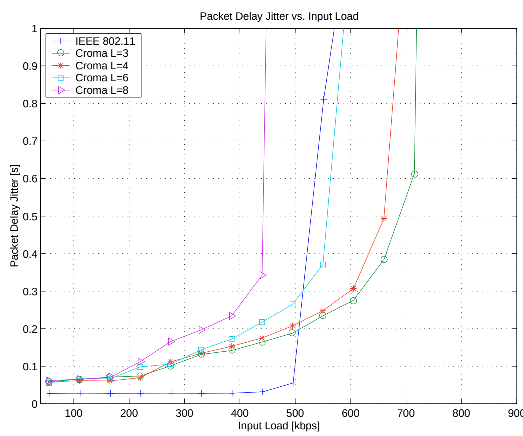



*Figure 22.* End-to-end delay jitter vs. input load, random topology.

In term of fairness, CROMA still outperforms IEEE 802.11 in a random topology as shown on Figure 23. Note that it is very difficult to get statiscally satisfying results over random topologies because of the simulation time. However, ten different random connex networks (not shown here) have been simulated and provide similar results.

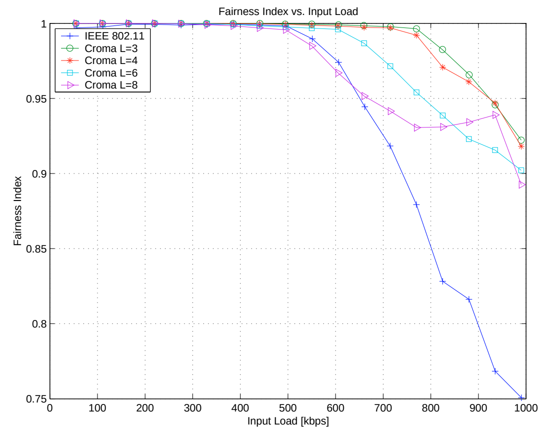



*Figure 23.* Fairness index vs. input load, random topology.

## 6. Conclusion

In this paper, a new MAC protocol, called CROMA has been proposed for mobile ad hoc networks. CROMA operates in a slotted environment, it is collision-free and receiver-oriented. The reservation of the resources is made through a random access phase on each slot of the frame. The transmission is done thanks to a polling by the receivers. Thus, receivers of a connection act as local base-stations and sophisticated functions at higher layers can be easily implemented.

The correctness of CROMA has been proven. Even with a dynamic topology, CROMA handles both the hidden and the exposed terminal problems.

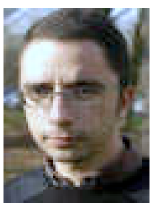
Theoretical performance analyses and extensive simulations show that CROMA can reach very high throughput in a fully connected network provided that the average message length is large. Moreover, CROMA outperforms IEEE 802.11 at high input load thanks to a better channel utilization.

## References

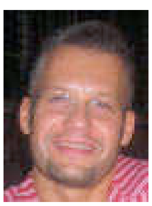
1. ANSI/IEEE Std 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE, 1999.
2. E. Arikan, Some Complexity Results about Packet Radio Networks, IEEE Trans. on Inform. Theory, vol. IT-30, NO. 4, pp 681-685, Jul. 1984.
3. B. Baynat, Théorie des files d'attente, des chaînes de Markov aux réseaux à forme produit, Hermes Science Publications, Paris, 2000.

4. L. Bao and J. J. Garcia-Luna-Aceves, A New Approach to Channel Access Scheduling for Ad Hoc Networks, Proc. of ACM/IEEE MOBICOM'01, pp 210-221, Jul. 2001.
5. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, MACAW: A Media Access Protocol for Wireless LAN's, Proc. of ACM SIGCOMM, pp 212-225, Aug. 1994.
6. I. Cidon and M. Sidi, Distributed Assignment Algorithms for Multihop Packet Radio Networks, IEEE Trans. on Computers, vol. 38, NO. 10, pp 1353-1361, Oct. 1989.
7. I. Chlamtac, A. Faragó, and H. Zhang, Time-Spread Multiple-Access (TSMA) Protocols for Multihop Mobile Radio Networks, IEEE/ACM Trans. on Networking, vol. 5, NO. 6, pp 804-812, Dec. 1997.
8. M. Coupechoux, C. Bonnet, and V. Kumar, CROMA - a New Medium Access Protocol for Mobile Ad hoc Networks, Proc. of WTC'02, Sept. 2002.
9. M. Coupechoux, B. Baynat, C. Bonnet, and V. Kumar, Modeling of a Slotted MAC Protocol for MANETs, Proc. of MADNET'03, Mar. 2003.
10. J. Elson, L. Girod, and D. Estrin, Fine-Grained Time Synchronization using Reference Broadcasts, Proc. of the 5th Symposium on Operating System Design and Implementation, Dec. 2002.
11. J. Elson and D. Estrin, Time Synchronization for Wireless Sensor Networks, Proc. of the 15th Inter. Parallel and Distributed Processing Symposium, pp 1965-1970, Apr. 2001.
12. A. Ephremides and T. V. Truong, Scheduling Broadcasts in Multihop Radio Networks, IEEE Trans. on Communications, vol. 38, NO. 4, pp 456-460, Apr. 1990.
13. European Commission, esa, Galileo Mission High Level Definition v3.0, available on http://europa.eu.int/comm/dgs/energy_transport/galileo/index_en.html, Sept. 2002.
14. C. L. Fullmer and J. J. Garcia-Luna-Aceves, Solutions to Hidden Terminal Problems in Wireless Networks, Proc. of ACM SIGCOMM'97, pp 39-49, Sept. 1997.
15. J. J. Garcia-Luna-Aceves and C. L. Fullmer, Floor Acquisition Multiple Access (FAMA) in Single-Channel Wireless Networks, ACM Mobile Networks and Applications (MONET), vol. 4, NO. 3, pp 157-174, 1999.
16. R. Jain, D. Chiu, and W. Hawe, A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems, DEC Research Report TR-301, Sept. 1984.
17. J.-H. Ju and V. O. K. Li, An Optimal Topology-Transparent Scheduling Method in Multihop Packet Radio Networks, IEEE/ACM Trans. on Networking, vol. 6, NO. 3, pp 298-306, Jun. 1998.
18. P. Karn, MACA - a New Channel Access Method for Packet Radio, Proc. of ARRL/CRRL, Apr. 1990.
19. Y. H. Kwon and D. C. Lee, An Uplink Packet Relay Protocol for CDMA Cellular-like Systems, Proc. of MILCOM'02, vol. 2, pp 940-945, Oct. 2002.
20. H. Luo, S. Lu, and V. Bharghavan, A New Model for Packet Scheduling in Multihop Wireless Networks, Proc. of ACM/IEEE MOBICOM'00, pp 76-86, Aug. 2000.
21. ns2 web page, http://www.isi.edu/nsnam/ns.

22. L. C. Pond and V. O. K. Li, A Distributed Time-Slot Assignment Protocol for Mobile Multi-Hop Broadcast Packet Radio Networks, Proc. of IEEE MILCOM'89, vol. 1, pp 70-74, Oct. 1989.
23. S. Ramanathan, A Unified Framework and Algorithm for (T/F/C)DMA Channel Assignment in Wireless Networks, Proc. of IEEE INFOCOM'97, vol. 2, pp 900-907, Apr. 1997.
24. R. Ramaswami and K. K. Parhi, Distributed Scheduling of Broadcasts in a Radio Network, Proc. of IEEE INFOCOM'89, vol. 2, pp 497-504, Apr. 1989.
25. W. J. Stewart, An Introduction to the Numerical Solution of Markov Chains, Princeton University Press, New Jersey, 1994.
26. F. Talucci, M. Gerla, and L. Fratta, MACA-BI (MACA by invitation) - A Receiver Oriented Access Protocol for Wireless Multihop Networks, Proc. of IEEE PIMRC'97, vol. 2, pp 435-439, Sept. 1997.
27. Z. Tang and J. J. Garcia-Luna-Aceves, A Protocol for Topology-Dependent Transmission Scheduling in Wireless Networks, Proc. of IEEE WCNC'99, vol. 3, pp 1333-1337, Sept. 1999.
28. N. H. Vaidya, P. Bahl, and S. Gupta, Distributed Fair Scheduling in a Wireless LAN, Proc. of ACM/IEEE MOBICOM'00, pp 167-178, Aug. 2000.
29. S. Xu and T. Saadawi, Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?, IEEE Comm. Magazine, vol. 39, NO. 6, pp 130-137, Jun. 2001.
30. C. Zhu and M.S. Corson, A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks, Proc. of IEEE INFOCOM'98, vol. 1, pp 322-331, Mar. 1998.

Marceau Coupechoux received M.Sc. in Mathematics from the University Pierre et Marie Curie (Paris) in 1998 and a double Engg. Degree from the Ecole Nationale Supérieure des Télécommunications (ENST), Paris, in 1999, jointly with the University of Stuttgart in 2000. He joined Alcatel Research & Innovation in 2000 and is Ph.D. student since 2001 at the Institut Eurecom, Sophia-Antipolis. He is the author or co-author of 10 conference papers and 5 patents.



Bruno Baynat received the MS degree from the Institut National Polytechnique de Grenoble in 1988 and the Ph.D. degree from the University Pierre et Marie Curie in 1991. Presently, he is Maître de Conférence (Associate Professor) at the University Pierre et Marie Curie. His research interests are presently in the development of models for the performance evaluation of communication systems, with applications to wired networks (Multicast, QoS) and wireless networks (Ad-Hoc, GPRS/EDGE/UMTS, Wi-Fi).

Christian Bonnet joined Institut EURECOM as an associate professor in 1992. Since 1998 he has been at the head of the Mobile Communications Department of EURECOM. He teaches distributed and real-time systems, mobile communication systems, wireless LANs and protocols for mobility management. His main areas of research are Mobility Management protocols, wireless access to IP Networks and data communications in mobile networks including Mobile Ad Hoc networks. He is currently participating in research projects related to UMTS in the field of QoS and Ipv6.

Vinod Kumar received MS, and Ph.D from Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, in 1977 and 1980, respectively. He is Director in Alcatel Research & Innovation for projects related to Wireless Communications. He has more than 20 years of experience in digital mobile communications, signal processing and performance evaluation of higher protocol layers, and he has more than 20 patents, and more than 30 publications in technical journals, books and conference proceedings in above mentioned areas. He is Associate Professor in University of Marne La Vallée, and Visiting Professor in several other institutes in France. He is Officiating Secretary of the Wireless World Research Forum (WWRF).