UNIVERSITÀ DELLA CALABRIA

Dipartimento di Elettronica,
Informatica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi e Informatica
XXIII ciclo

Tesi di Dottorato

Proposal of new techniques for the efficient energy management in TDMA MAC protocols over MANETs

Annalisa Perrotta

UNIVERSITÀ DELLA CALABRIA

Dipartimento di Elettronica,
Informatica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi e Informatica
XXIII ciclo

Tesi di Dottorato

Proposal of new techniques for the efficient energy management in TDMA MAC protocols over MANETs

Annalisa Perrotta

Coordinatore
Prof. Luigi Palopoli

Supervisori
Prof. Salvatore Marano
Ing. Floriano De Rango

DEIS

DEIS – Dipartimento di Elettronica, Informatica e Sistemistica
Novembre 2010

Settore Scientifico Disciplinare: ING–INF/03

To my sister,
my parents
and my grandparents

Acknowledgements

I wish to express my gratitude and appreciation to prof Salvatore Marano and Floriano De Rango, whose supervision, encouragements and support have been invaluable for my work. I want to thank my Ph.D. coordinator, Luigi Palopoli for his precious suggestions and willingness. My deepest gratitude goes to my family and to all my friends for their unselfish help, for their unlimited support in hard times and for making my days pleasant.

Contents

List of Figures

List of Tables

List Of Publications

F. De Rango, A. Perrotta and S. Marano. QoS-CROMA: An On-demand Time-Slotted MAC Protocol with QoS support for Wireless Ad Hoc Networks. In *Proceedings of the Fourth IEEE International Symposium on Wireless Communication Systems (ISWCS)*, pages 706–710, 2007.

F. De Rango and A. Perrotta. Energy Evaluations of E-TDMA vs. IEEE802.11 in Wireless Ad Hoc Networks. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pages 273–279, 2010.

F. De Rango and A. Perrotta. Performance Evaluation of Two Slot Assignment Strategies under Distributed TDMA MAC protocol over Mobile Ad Hoc Networks. *Submitted to an international journal*, 2010.

F. De Rango and A. Perrotta. E-TDMA Analysis comparing two different slot assignment strategies: DASA and DCSA. *Submitted to an international journal*, 2010.

F. De Rango and A. Perrotta. Optimal setting of E-TDMA parameters for minimizing energy consumptions of DASA and DCSA slot assignments in a distributed TDMA MAC for MANETs*Submitted to an international journal*, 2010.

1

Introduction

1.1 MANETs and Medium Access Control

This thesis aims at proposing novel techniques and methods to provide an efficient management of energetic resources of mobile hosts at MAC (*Medium Access Control*) level in the context of Mobile Ad hoc NETworks (MANETs). A MANET is a wireless network without any fixed infrastructure or central coordination in which nodes exchange packets by sharing a common broadcast radio channel. Wireless networks differ enormously from wired networks; furthermore, ad hoc wireless networks have even more specific characteristics, such as node mobility, power constraints and routing functionality according to which each node can act as a router when the two end-points interchanging data are not directly within radio range of each other.

With the proliferation of wireless devices and the consequent growing use of mobile networks, a very large number of research studies focused on MANETs, and, particularly, on MAC and routing issues, that respectively belong to the Second and Third level of the ISO-OSI stack. As a matter of fact traditional routing and MAC protocols cannot work efficiently in MANETs' environment where the intrinsically random nature makes very challenging their design.

Our attention is focused on MAC. Due to the limitations of the wireless channel, the bandwidth shared among the nodes is limited. Therefore, one of the features of a MAC protocol is to provide an efficient usage of the bandwidth, allowing spatial reuse of wireless spectrum and concurrent transmissions for nodes that are sufficiently apart from each other, while attempting, on the contrary, to prevent collisions by avoiding that two or more interfering nodes access the medium at the same time (this is essential to the successful operations of a shared-medium network). Other issues of a MAC protocol for wireless ad hoc networks are related to fairness, node mobility, time synchronization, quality of service support, power consumptions and so on. Moreover, MAC operations are affected mainly by two problems, known as:

- *hidden* and

- *exposed* terminal problem.

The former happens when two nodes that are outside each other's range perform simultaneous transmission to a node that is within the range of both of them. This will cause a packet collision at receiver side. The latter consists in a transmission that could be safe without collisions at the receiver, but it is not started because another concurrent transmission is detected by the sender. There is not way for the sender to know if the receiver detects that transmission or not, so its transmission will be delayed even if it would never be sensed by the receiver.

Thus, the presence of hidden nodes causes an increased probability of collision at a receiver, whereas the channel access may be denied to exposed nodes which causes underutilization of the bandwidth resources.

Many solutions have been proposed in literature to tackle these problems. Protocols that implement them can be divided into two classes:

- *contention-based* and
- *collision-free* protocols.

The main difference between them is in providing a random or scheduled access to the network, respectively.

In contention-based protocols, the channel access policy is based on competition: a transmitter node contends to acquire the channel every time it has to send data packets.

In collision-free protocols one node is able to reserve the channel for a certain amount of time or data in which it can transmit (if protocol is *sender-oriented*) or receive (if protocol is *receiver-oriented*) its data packets without any conflict or collision with the transmissions (receptions) of other nodes. Every node usually has a own schedule that tells when they can access to the channel in reserved mode, thanks to a particular division of the channel, for example respect to time, frequency, or code, as in three main classes of collision-free protocols: TDMA, FDMA and CDMA, (*Time*, *Frequency* and *Code Division Multiple Access*, respectively).

Contention-based MAC protocols such as IEEE 802.11 are often applied in the context of wireless ad hoc networks for their easy deployment and reduced control structure that is typical of more complex medium access techniques based on the time-slot reservation or code selection. However, when traffic load is heavy, TDMA-based protocol can be essential not only in terms of higher throughput offered in the network but also for a reduction in the energy dissipation due to the capability of lowering the data packet collision.

1.2 Energy Issues at MAC layer

Energy consumption is another critical issue for MANETs. This happens because mobile nodes are battery-powered: their operational life is limited by their own power source. Efficient utilization of battery energy is essential, especially for small devices (sensors), because it can affect the overall network communication performance: when a node exhausts its available energy, it ceases to function. Besides, in MANETs mobile hosts act as routers, therefore a not reasonable or unfair usage of energetic resources of the network, involving, for example, central nodes, could result in a premature partitioning of the network, making unattainable some nodes and inhibiting communications. Thus, power consumption should be distributed on the nodes of the network and the overall transmission power for each connection should be minimized.

Energy issue is critical and challenging because it has to consider several variables: the network typology and topology, the ISO/OSI level at which to act, the possible QoS requests to satisfy, and so on. There is a trade-off between communication and energy performances because energetic consumptions are closely related to the basic functionality of a network, namely the information exchange. The implementation of energy saving techniques could degrade performances related to other metrics, like throughput, packet delays, etc., as well as satisfying some QoS requests could cause huge energy consumptions.

All the layers of communication are coupled in power consumption. Ad hoc routing and MAC protocols may consume different amounts of power and upper layers decisions may be conditioned. Many studies have proved that the radio activity of a wireless card consumes a huge part of the energy.

Several mechanisms, techniques and protocols have been proposed in literature to minimize energy consumptions at different ISO/OSI levels. It is possible to tackle the problem from different points of view: Physical, MAC and Routing Level. In this thesis MAC perspective is taken into account.

Existing literature proposals about energy-efficient techniques and mechanisms are related to the two main typologies of MAC protocols, contention-based and conflict-free: they present complementary or opposite problems: in contention-based protocols actually the energy consumption is very high because of the idle listening; in conflict-free protocols, instead, transmission schedules can reduce energy consumption due to collisions, even if the energy spent to maintain the transmission schedules consistent has to be taken into account. In general, we can classify proposed techniques concerning MAC level as:

- *energy-saving* techniques: by introducing sleep procedures, they reduce the time during which a node has to remain in idle listening, that is when the node does not transmit or receive but waits for possible future receptions or, in the worst case, packet overhearings that is the reception of packets that are directed to other nodes in the same transmission range and not to the node that has received it;
- *power-control* techniques: they regulate the transmission power to the minimum level needed to let a correct detection of a packet based on the

distance between transmitter and receiver node, without degrading communication performances, but also decreasing the number of possible overhearing nodes, depending on the power level chosen and node density.

- *energy-aware* techniques: their logic is deeply based on energy, that is they make decisions based on the residual energy of a node, or the average energy of the whole network, or the discharge rate of the nodes, and so on, and can change dynamically the intrinsic logic of a protocol.

It is also possible to implement more than one of these techniques (*hybrid* solution).

We have to note that is relatively simple to implement energy-saving or power-control techniques in collision-free protocols because each node knows in advance the exact schedule of its transmissions and receptions and can plan sleep procedures. The design of energy-aware mechanisms is more challenging and not so intuitive, but it is also very interesting. Indeed, energy-aware mechanisms are potentially more efficient, since they affect the intrinsic logic of the protocols. For this reason the research is focusing on this last type of solutions or hybrid solutions.

This thesis deals with this issue. Initially, we focus on the study of the energetic behavior of some MAC protocols. This first analysis is oriented to the individuation of how and how much the protocols could be affected by some of their own or scenario parameters, in terms of energy consumptions, in particular conditions of traffic, mobility or topology. All of these experimental discussions were aimed at proposing novel energy-aware techniques to minimize energy consumptions over MANETs.

1.3 Overview of the thesis

This thesis is divided into two parts.

The first part regards the experimental study of two MAC protocols from the energy and the communication point of view. In Chapter 2 related work is surveyed: several features, advantages and drawbacks of some traditional or energetic MAC protocols are presented; then the functional characteristics and the reasons of using NS-2, the discrete event simulator employed to deploy and test original and modified source codes of the chosen protocols and scripts describing our simulations scenarios in mobile ad hoc network context, are described.

The two subsequent chapters deal with an extensive experimental analysis of some interesting MAC protocols. The experiments of Chapter 3 focused on the comparison between the two main families of MAC protocols: contention-based and conflict-free. In particular we chose IEEE 802.11, that is the *de facto* standard for MANETs, among contention-based ones and E-TDMA (*Evolutionary Time Division Multiple Access*) among conflict-free ones. Several simulations were conducted to compare energetic and communication performances.

The protocols chosen for the comparison do not provide an efficient energy management, but they show good performances in terms of throughput, packet latency, scalability, and so on; for this reason we decided to study them also from an energetic point of view. The conducted experimental analysis shows that conflict-free TDMA protocols outperform contention-based MAC ones, not only from the communication but also from the energetic point of view, even without any energetic techniques, for each considered simulation scenario, both for low and for high traffic conditions, for which, in particular IEEE 802.11 has very huge energetic consumptions with respect to the other target protocol, E-TDMA.

The optimal results got from experimental analysis let the attention be focused on E-TDMA and generally on collision-free protocols. Chapter 4 presents three different scenarios of simulation and for each of them provides an exhaustive experimental campaign to study the effects of some E-TDMA parameters on the energetic efficiency, and to propose solutions to design possible energy-savings. It is very interesting how energetic performances of E-TDMA improve even if traffic load is heavier and heavier with respect to the allowed bandwidth. So we choose E-TDMA as representative collision-free TDMA protocols.

The second part of the thesis concerns the proposal of energy-aware mechanisms in order to minimize and efficiently manage energetic resources of nodes that run E-TDMA, or, in general, one similar TDMA MAC protocol. On the basis of the previous experimental study also upon tuning E-TDMA parameters, two analytical models for energetic consumptions are defined. These models could be valid for every TDMA system that presents an E-TDMA-like structure, with a control period for bandwidth request and reservation and a data-exchange period. These models constitute the basis for the design of the proposed energy-aware strategies.

We subdivided the problem in two distinct analysis: the Chapter 5 analyses data exchange phase while the Chapter 6 focuses on control phase and on broadcast slot assignment in a contention-based context.

As far as the E-TDMA data exchange phase is concerned, an analytical model for representing energetic consumptions is built. Through this model, which is valid for any conflict-free TDMA protocol, we propose an energy-aware modification that is related to the strategy followed to assign conflict-free temporal slots for data exchanges. Theoretical model shows the energetic efficiency of the proposed modification in reducing energetic consumptions. The energy-aware strategy was implemented and tested through an exhaustive plan of simulations that confirmed theoretical analysis, We report in details the related results in Chapter 5.

Finally, for control epoch we present an energetic modeling for the FPRP (*Five Phase Reservation Protocol*) which is used by E-TDMA in reservation phase, completed by the modeling of SU (*Schedule Update*) phase, during which nodes that won the previous FPRP contention, makes their bandwidth requests. Energy consumptions related to the E-TDMA control epoch are very huge and usually not balanced by information phase consumptions. Through the previous analysis we try to propose a novel energy-aware mechanism to further reduce control energy consumptions. This modification was integrated with that related to the E-TDMA information phase.

The new *energetic* E-TDMA protocol, in which we implements energy-aware and energy-saving techniques combined together in a hybrid fashion, was tested in our simulation environment: Chapter 6 reports these results.

The conducted analysis can be extended to many other protocols similar to E-TDMA. Moreover, the present thesis lays the foundation for further energy-aware strategies for TDMA protocols that could enable adaptive and dynamic changes based, for example, on traffic conditions, residual energy of the node, the average residual energy of the network, and so on.

2

MAC protocols and energy issues on MANETs

2.1 Introduction

In this chapter, some media access control layer protocols, for wireless ad hoc networks are presented. As previously stated, they can be mainly categorized as contention-based or collision-free MAC protocols, in particular we focus on "Time Division Multiple Access" (TDMA) ones.

The target protocols of this thesis are very popular in literature, but do not implement by themselves strategies to a reasonable and fair usage of energetic resources of the nodes on which they run. So they cannot be included among *energy-efficient* protocols. These last ones will be presented hereinafter. The choice was made considering the very attractive communication performance of them, while it may happens that metrics different from energy (throughput, delay, jitter) are not considered in experimental analysis of the energy-efficient protocols.

The main aim of our analysis was to study energetic performances of them as they are, comparing the two MAC category early mentioned and, then, to produce improvements through the design and implementation of energy-saving strategies, as many energy-efficient protocols do, and to think of novel energy-aware strategies.

Contention-based and collision-free MAC protocols are very different not only from radio access logic, but also from energetic point of view.

Contention-based protocols register an additional energy consumption for RTS and CTS packets, used to enable a collision-free communication and to prevent the hidden terminal problem. Furthermore, every packet needs several preamble bytes to tune the radio oscillator of a receiver before they can be transmitted. A node cannot listen to the medium during switching from listening to sending. Therefore collisions are possible if a RTS or CTS packet arrives during a transceiver state switch. So there is another energy consumption.

TDMA-based protocols (among collision-free) guarantees a collision-free communication without an energy-wasting contention period. Besides they require less transceiver state switches and are generally more energy-efficient than contention-based approaches. Disadvantages are a less flexible adaptation on actual network traffic load, the necessity of a precise time synchronization, a complex procedure to add new nodes to an existing network, and a limitation on the maximal number of nodes within a certain area.

Examples of contention-based protocols are: ALOHA (and all its variants), *Carrier Sense Multiple Access* (CSMA) [Muqattash and Krunz, 2003], the *Collision Avoidance* (CA) technique, used by many protocols like MACA [Karn, 1990] (*Multiple Access with Collision Avoidance*) and its variants, FAMA (*Floor Acquisition Multiple Access*) [Fullmer and Garcia-Luna-Aceves, 1995] and its variants, IEEE 802.11 (particularly the DCF mode), that is the *de facto* standard for MAC level in MANETs and uses the CSMA in union with CA, and some others that make use of CA through an out-of-band tone, like BTMA [Tobagi and Kleinrock, 1975] (*Busy Tone Multiple Access*) and its variants.

Examples of TDMA collision-free protocols are: FPRP [Zhu and Corson, 1998, 2001b], E-TDMA [Zhu and Corson, 2001a], ADHOC-MAC [F. Borgonovo and Fratta, 2004], CROMA [M. Coupechoux and Kumar, 2003a,b, 2004] and QoS-CROMA [F. De Rango and Marano, 2007], to mention only the most important ones.

We can classify energy-efficient contention-based or collision-free protocols according three main groups: *energy-saving*, *power-control* and *energy-aware* techniques. The first area includes all the mechanisms to reduce energy overhead in a protocol, like sleep procedures, the second suggests the usage of different power levels and clustering. The third area includes battery-aware and all the other strategies employed dynamically to fit with residual energy resources in the overall network or in a single node. *Hybrid* techniques combine all of these logics.

Recently, some researchers propose the use of multiple channels to overcome the limitations of single channel MAC protocols. Some of them try to minimize energy consumptions like Y-MAC [Youngmin Kim and Cha, 2008] that we will describe in the second paragraph. In particular Y-MAC has been proposed on a sensor network (WSN), where energy problem is very important. Most of the works in literature are designed for the WSNs. The following are only a little part of them: WiseMAC [El-Hoiydi and Decotignie, 2004], X-MAC [M. Buettner and Han, 2006], ELM-MAC [P. Yun, 2008] and A-MAC refCorradiniPPR10 among contention-based protocols, PEDAMACS [Coleri-Ergen and Varaiya, 2006], LMAC [van Hoesel and Havinga, 2004] sender-oriented, Crankshaft receiver-oriented [Halkes and Langendoen, 2007] and [X. Lu et al., 2008] among collision-free TDMA protocols.

It is relatively easy to implement the power-saving techniques in TDMA systems, where all the transmissions are scheduled in a pre-emptive period and nodes can enter sleep state when not involved in communication since they know in advance when they will send or receive data. Energy-aware strategies, hybrid or not, are the most challenging. there exists techniques to reduce idle periods varying dynamically frame dimension to adapt it with traffic conditions like in [Yahya and Benothman, 2008], using clustering logic and a TDMA super-frame structure like in [Tavli and Heinzelman, 2004], classifying packets or distinguish broadcast from unicast transmissions, like in [Xie and Wang, 2002], increasing sleep period and reducing state transitions, like in [van Hoesel and Havinga, 2004] and [Ma et al., 2009], using battery-aware logic like in [S. Jayashree and Murthy, 2004] [S. Jayashree and Murthy, 2008] or geographical informations like GMAC [Lessmann, 2007].

The reminder of the chapter is organized as follows. In the first section of this chapter we introduce some of traditional MAC protocols: IEEE 802.11 among contention-based, FPRP, E-TDMA and CROMA among collision-free.

In the second section we present Y-MAC, BAMAC, GMAC, and *continuous link scheduling* strategy.

Finally, in the last section we provide a brief description of the simulator used for all experimental analysis reported in this thesis: Ns-2.

2.2 Traditional MAC Protocol

2.2.1 IEEE 802.11

We will deal with the DCF (*Distributed Coordination Function*) of IEEE 802.11, because it is the broadly used in MANETs [IEEE].

The DCF mode of IEEE 802.11 is the standard MAC protocol for distributed ad hoc networks. It is based on the *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) mechanism. Every node, before delivering a data packet, has to detect for a short interval of time, called DIFS (that is *DCF Inter-frame Space*), if there is any other transmission on the wireless medium in its range. To minimize the occurrence of collisions, the node will start its transmission only if it senses the carrier idle for another randomized duration multiple of a slot time from 0 to $(CW - 1)$. CW (*Contention Window*) is a time counter owned by every node that is updated on the basis of an exponential back-off mechanism.

During the random back-off interval, if the station detects that the medium becomes busy, it pauses the decrement of CW and it will reactivate it when it senses the channel idle again for more than a DIFS. The initial value of CW, at the first transmission attempt, is equal to a pre-specified minimum contention window, CW_{min}, while, if the transmission fails, the value is doubled up to maximum contention window, CW_{max}.

Once the back-off timer expires, the transmitting node has two options to initiate the transmission:

- It can either transmit the data packet directly or
- transmit a short RTS (*Request-To-Send*) frame.

Usually broadcast packets and those that have a size smaller that a certain threshold, are delivered without the RTS–CTS handshake. For all the other packets, instead, the RTS-CTS exchange is used to prevent the *hidden terminal* problem. RTS–CTS handshake mechanism is described in Fig. 2.1.

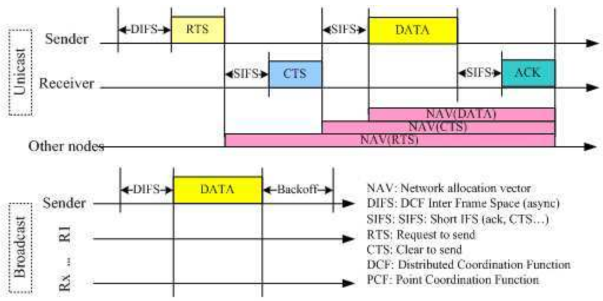


Fig. 2.1. Unicast and Broadcast packets transmission.

RTS and CTS packets contain the information of how long a node takes to transmit the data packets and so it is possible to discourage other stations by attempting to transmit during such period, setting the NAV counter (*Network Allocation Vector*). So RTS-CTS implements a virtual carrier sensing. Once the data packet is correctly delivered to the destination station, an *acknowledge* (ACK) frame will be sent back to confirm the transmission.

For every other new transmission, the node has to detect the channel again and waits until it will be idle to make its transmission. As it is possible to see, this MAC mechanism could present some drawbacks in terms of channel access delay, interference grade and energy consumption for high traffic load and when the nodes activity is high.

2.2.2 FPRP

FPRP [Zhu and Corson, 1998, 2001b] is another distributed TDMA MAC protocol that provides a 5-phases handshake (Fig. 2.2):

- RR: Reservation Request
- CR: Collision Report
- RC: Reservation Confirmation
- RA: Reservation Acknowledgement
- PP/E: Packing/Elimination

and then a collision-free data exchange. Every node that has to reserve one or more slot(s) sends a packet in RR-phase with a probability p. This nodes are called RN (*requesting nodes*). The nodes that have reported a collision in the RR-phase send a packet in CR phase. The RR-CR exchange is similar to RTS-CTS and eliminates for FPRP the *hidden terminal* problem.

The RN that receives a CR packet releases the contention, the others, instead, become TN (*transmitter nodes*) and send a RC packet. If no collision is detected by any neighbor node, each of them sends a RA packet to inform its own neighbors about a successful reservation for one of its neighbors. May be any RN detects a collision of RC packet. In this case no RA packet is sent. If a sender node does not receive any RA packet it will consider no longer itself a TN. In the PP/EE phase every node that is 2-hops away from a TN that has won its reservation, sends a PP packet to encourage its neighbors that are 3-hops away from the winner eventually to adjust own contention probability accordingly for that slot and then to speed up the convergence of the protocol. Also in this phase a TN sends an E packet with a probability of 0.5; so, in case of an isolated deadlock, a sender node will be still a TN only if it transmits and not receives an E packet.

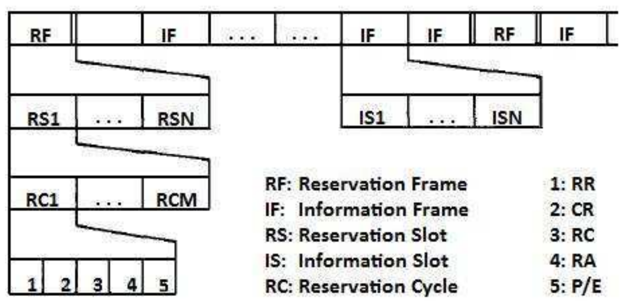


Fig. 2.2. Frame structure of the FPRP.

Each FPRP phase carries only a single logical bit packet, the meaning of which is always inferred in the context of the protocol, that is the particular FPRP phase is currently running. In FPRP two nodes will never be hidden each other, because possible collisions for a hidden terminal is detected at the receiver node that informs both transmitters in one of the FPRP-phases. So, unlike the CSMA/CA protocol, where the sender detects the collision at the receiver, a collision is always detected at the node where it occurs (then also at the receiver node). The sender analyzes collision reports from its one-hop receivers, based on which FPRP-phase is running, and makes the final decision. Thanks to the five-phase scheme FPRP protocol attempts to minimize the probability of collision in a robust and efficient way.

2.2.3 E-TDMA

E-TDMA [Zhu and Corson, 2001a] is a distributed TDMA MAC protocol. It is based on reservation and on TDMA schedule with a pseudo-periodic information frame that is shown in Fig. 2.3.

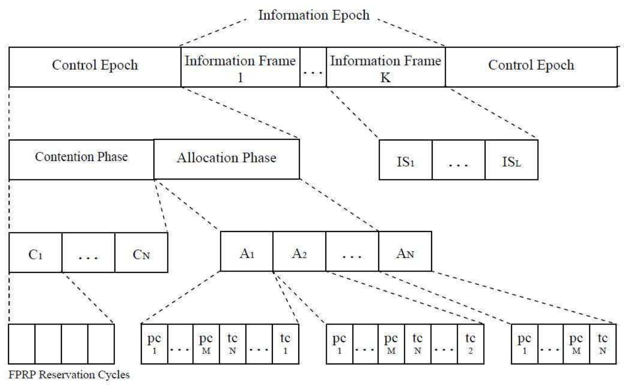


Fig. 2.3. E-TDMA pseudo-periodic frame.

As we can see an E-TDMA frame is divided in 2 parts: a *Control Epoch*, in which nodes exchange MAC control packets, and an *Information Epoch* in which nodes deliver their data packets without collisions, after having acquired the channel for one or more information slot(s) during the Control Epoch. The Control Epoch is, in turn, divided into *Contention* and *Allocation* phase.

In Contention Phase node contend for channel through a fixed number of FPRP reservation cycles. A TN reserves a slot after a certain number (say t) of successful FPRP cycles. In the following allocation phase, also called SU (*schedule update*) phase, every node informs its neighbors about its *information schedule*, based on its own *control schedule*: there are n permanent colors and m temporary colors; every group of FPRP cycles is used to reserve one of the m temporary colors (so there are m groups of t FPRP cycles), through which a node has the permission to request new permanent colors or information slots. Possessing a permanent color allows a node to transmit its control and information schedules. After a node has acquired a permanent color, it will transmit its schedule in it as long as a collision occurs. If a collision occurs, the node has to discard that color and it restarts a new contention for a temporary color. The requests of new slot or new permanent colors happens during t-color, at the end of permanent colors. There is an allocation phase for each temporary color, and they are placed in reverse order as Fig. 2.3 shows. At the end of allocation phases, every node can transmit its data depending on the information schedule updated in the last allocation phase. Then the cycle restarts.

Information Epoch has a certain number of identical information frames. in each of them every node knows in which slot it has to transmit or receive or remain idle, and form or to which one of its neighbor, or also if a slot is reserved for a broadcast transmission. then information epoch is the same as every traditional TDMA systems. Each node, besides, will maintain the same group of slots for a certain transmission till it want to, or a collision or other problem occurs, for example caused by node movements: these situations are detectable through SU packets of which stations modify opportunely information or control schedules.

Zhu and Corson [2001a] showed that E-TDMA logic is better than that of IEEE 802.11 for throughput, delay and served sessions. Other studies on E-TDMA were conducted in [V. Loscrì and Marano, 2004, Los] in which E-TDMA is integrated with a QoS-Routing, in a cross-layer approach. Good performances of E-TDMA lead us to select it as a target TDMA MAC protocol in our research. In the next chapter we will see that, in our conditions, E-TDMA behaves better than IEEE 802.11 also from an energetic point of view.

2.2.4 CROMA

CROMA [M. Coupechoux and Kumar, 2003a,b, 2004] is a time-slotted based medium access control protocol for MANET. It makes use of a single channel, omnidirectional antenna and it is receiver-oriented because each receiver is able to manage a greater number of connections on the same time-slot. Like all the TDMA MAC protocols, an exact synchronization is mandatory and for this purpose GPS or other synchronization protocols can be used. CROMA divides the time into frames that are also divided into a number N of time slots. Each slot is divided into three minislots such as depicted in Fig. 2.4.

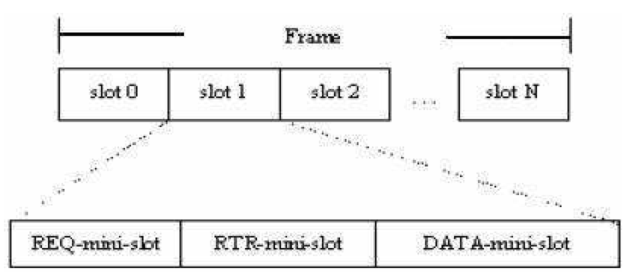


Fig. 2.4. Frame format of CROMA.

The slot reservation is performed by the sender through an on-demand policy. An example presented in Fig. 2.5 shows the mechanism of the on-demand request started by the sender.

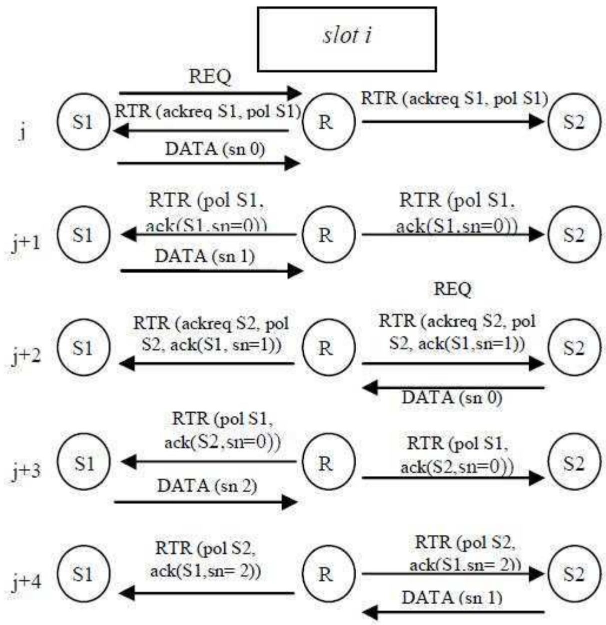


Fig. 2.5. Slot Reservation Phase of CROMA.

It is important to observe that receiver R can accept a prefixed number of connections. Moreover, the time slot choice on which to send a request is not arbitrary but is carried out following specific rules. First of all, the sender has to listen the channel for a frame time during the RTR-mini-slot to know which slot are busy and which ones are free. This phase is called reservation-phase. After classifying these slots on a frame basis, the sender chooses the slot, following a specific policy.

We worked on a modification in the original protocol, in order to support the QoS connections and to increase the slot utilization, are based on a definition of the slot utilization index and on the use of a real time scheduling to meet the deadline associated to the data packet transmissions. Moreover a sender-side slot selection policy has been proposed to reduce the latency time for the slot acquisition and to increase the system utilizations. More details could be found in [F. De Rango and Marano, 2007].

2.3 Energy-efficient MAC Protocols

A main strategy of the considered protocols to save energy is to kept the periods of active radio transceiver as short as possible and shut down unused radio transceivers immediately.

2.3.1 Y-MAC

Y-MAC is a TDMA-based multi-channel MAC protocol. Each node has assigned an exclusive send time slot in two hop neighborhood, collision-free access to the medium is guaranteed. Such a scheme is thus able to reduce energy wasted by contention and collisions. However, all nodes must wake up at every time slot so as not to miss incoming messages. This results in energy wastage due to idle listening and overhearing.

Fig. 2.6 illustrates the frame architecture of YMAC. Time is divided into several fixed-length frames, and each frame is composed of a broadcast period and a unicast period. Since the wake up times for nodes are dispersed, every node must wake up at the start of the broadcast period to exchange broadcast messages. If there are no incoming broadcast messages, each node turns off its radio until its own receive time slot to save energy. Determining the number of time slots is important because there is a trade-off between the number of time slots and the delivery latency. The more time slots we have, the more nodes we can allocate exclusive time slots to, but delivery latency increases due to the prolonged length of the frame period. One alternative approach is to increase the number of possible time slots using multiple channels. This requires complex operations.

Y-MAC separates broadcast traffic from unicast traffic. This makes broadcasting is more reliable than in the other MAC protocols that do not separate different types of traffic. If a node did not receive or send a broadcast packet in the broadcast time slot, the radio transceiver is turned off to conserve node energy. Each node changes its frequency to the base channel at the beginning of its own receive time slot, and potential senders also do so.

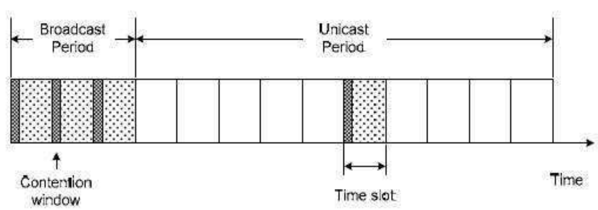


Fig. 2.6. Battery Discharge.

The medium access design of Y-MAC is based on synchronous low power listening. We define the time slot length to be long enough to receive one message. Therefore, only the contention winner can transmit a message to the destination node. Contention between potential senders is resolved in the contention window and a random back-off. The receiver wakes up at the end of the contention window to receive the data part of the message. If it receives no signal from neighboring nodes, the receiver node turns off its radio device.

Y-MAC was deployed in a real component and experimental results show that Y-MAC achieves low duty cycle under light traffic conditions, similar to other existing low power MAC protocols. We have proven that Y-MAC achieves effective transmission of bursty messages, under high traffic conditions, while maintaining low energy consumption.

2.3.2 BAMAC

Battery Aware Medium Access Control (BAMAC) is a contention-based protocol that considers nodes of the network, contending for the common channel, as a set of batteries and schedule them using a round-robin scheduler. It takes benefit of the chemical properties of the batteries to provide the longest life for the mobile nodes battery. Then it tries to increase the lifetime of the nodes by exploiting the recovery capacity effect of the battery.

When a battery is subjected to constant current discharge, it becomes unusable even while there exists a sizeable amount of active materials. This is due to the *rate capacity effect* of the battery. If the battery remains idle for a specified time interval, it becomes possible to extend the lifetime of the battery due to the recovery capacity effect. By increasing the idle time of the battery, the whole of its theoretical capacity can be completely utilized. This effect is found to be higher with higher remaining battery capacity and decreases with decrease in the capacity.

BAMAC protocol tries to provide enough idle time and uniform discharge of batteries for the nodes of MANETs by scheduling them in an appropriate manner. This can be effected by a *round-robin* scheduling (or *fair-share* scheduling) of these nodes. Each of them maintains a battery table which contains information about the remaining battery charge of each of its one-hop neighbors. The entries in the table are arranged in the non-increasing order of the remaining battery charges. The RTS, CTS, Data, and ACK packets carry the remaining battery charge of the node that originated the packet. A node, on listening to these packets, makes a corresponding entry in its battery table, in way that the nodes are scheduled based on their remaining battery capacities. The higher the remaining battery capacity, the lower the back-off period.

This ensures near round-robin scheduling of the nodes. Hence, a uniform rate of battery discharge is guaranteed across all the nodes. This guarantees alternate periods of transmission and idling of the node, that is, alternate periods of discharge and recovery of the battery, as illustrated in Fig. 2.7. Whenever a node gains access to the channel, it is allowed to transmit only one packet, giving rise to an average of (n - 1) idle slots, where n is the total number of nodes contending for the common channel and the slot duration corresponds to T_t. In each idle slot, the battery recovers one charge unit with a probability $R_{i,j}$. This improves the lifetime of the battery as it gains more idle time to recover charge.

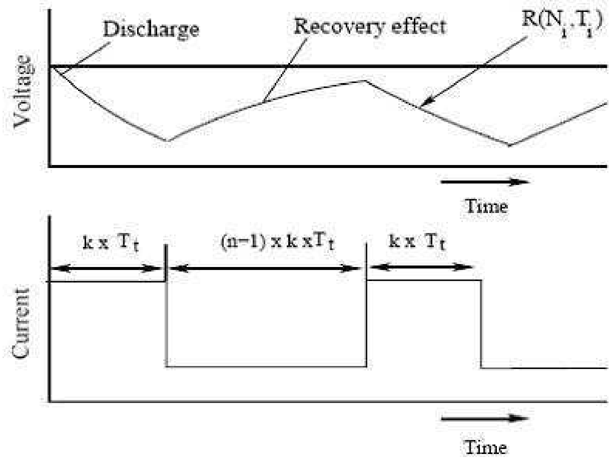


Fig. 2.7. Battery Discharge.

BAMAC provides a uniform discharge of batteries across nodes, extends the battery lifetime consuming 70nominal capacity of the battery per packet transmission and performs better, in terms of power consumption, fairness, and lifetime of the nodes, compared to IEEE 802.11 protocol. A discrete Markov chain was used to theoretically analyze the battery behavior of the protocol which was verified through extensive simulation studies. Markov model defines $R_{i,j}$ probabilities that we described before.

2.3.3 GMAC

GMAC (*Geographical MAC*) is a TDMA MAC protocol which assumes that nodes know the geographic positions of their two hop neighborhood and exploit this knowledge to construct a TDMA schedule. The schedule can be computed locally, no negotiation phase (and thus, no additional traffic) is needed. In order to determine the communication slots for a node, the latter must know the geographic positions of all nodes in its two-hop neighborhood.

For GMAC, any position data is sufficient, however inaccurate it might be, as long as it is consistent within each nodes two-hop neighborhood. GMAC divides time into frames called super-cycles divided into sub-frames divided into slots each node performs in each cycle one complete rotation in a geometric circle starting from the twelve oclock position.

Advancing in terms of time can thus be translated into advancing in terms of the current angle in the rotation. Hence, each point of time precisely corresponds to one specific angle in the rotation. Fig. 2.8 shows this relationship. Whenever the position of another node m is located on the line starting at the position of n along the current angle of n, then n may send data to m. Consequently, whenever the position of n is located on the line starting at the position of some other node m along the current angle of m (which is the same one as ns as all nodes are synchronized), then n might receive data from m. Reuse of the bandwidth is allowed, but in order to avoid packet collisions, the slot assignment of any TDMA MAC must be non-overlapping within the two hop neighborhood of each node. In the next super-cycle, all nodes broadcast the list of positions of their one-hop neighbors that they just acquired. Thus, after two super-cycles, all nodes have a consistent two-hop neighborhood information.

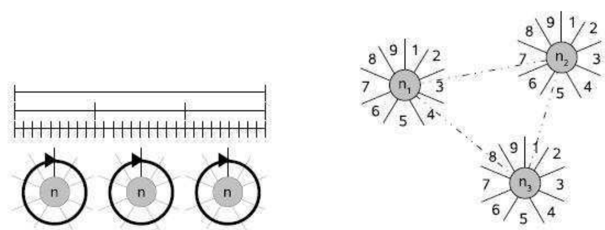


Fig. 2.8. (a) Slotting in GMAC and its equivalent in terms of angles ($c = 3$, $s = 9$). (b) TDMA slot assignment for n_1, n_2, n_3 ($s = 9$)

Although in low-traffic scenarios, the percentage of sleep time in GMAC might be high (since nodes can sleep when they do not want to send and no reception slot is scheduled for them), the nodes must constantly transit from sleep to idle and back when they reach a slot assigned to them for reception thus greatly reducing energy consumption. Additionally, GMAC guarantees a maximum delay per node which makes it suitable for data with real-time constraints.

2.3.4 Contiguous link scheduling

In this paragraph we report two interesting grouping scheduling techniques that will be very important to our proposal (see Chapter 5). We refer to [van Hoesel and Havinga, 2004] and [Ma et al., 2009].

Both run on TDMA systems. The first is designed for a centralized environment, in which a slot scheduler assigns bandwidth for connections. A schedule is broadcast to all mobiles so that they know when they should transmit or receive data. This schedule is called *Traffic Control Slot* (TCS). The slot scheduler is designed to preserve the admitted connections as much as possible within the negotiated connection QoS parameters.

The frame is divided basically into three different phases 2.9: uplink phases, downlink phases, and reservation phases. In downlink phases the base station transmits data to the mobiles, and in uplink phases the mobiles transmit data to the base station. In the reservation phase mobiles can request new

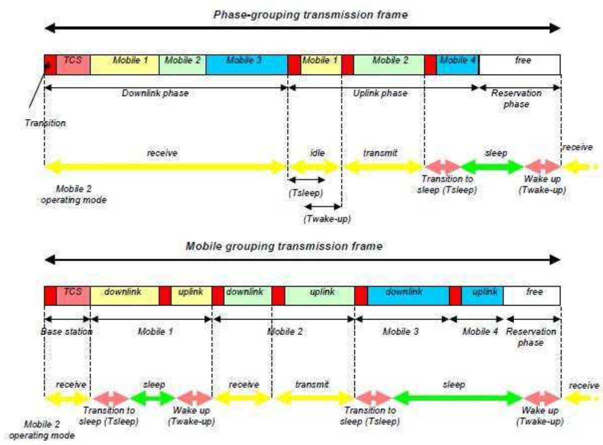


Fig. 2.9. Grouping strategies in a transmission frame.

connections. This is what the authors call *phase grouping*. They propose as an alternative a different approach with similar phases, but not grouped together in a frame according to the phase, but grouped together according to the mobile involved. The uplink and downlink phase of one mobile are grouped. So, as a result we will have in general multiple uplink and downlink phases in one frame. This is, instead, what the authors call *mobile grouping*.

Both strategies have positive and negative implications. Mobile grouping strives to minimise the power consumption, can reduce the number of state transitions, and thus achieve better energy efficiency, while phase grouping strives for an optimal channel utilisation.

In the mobile grouping strategy the slot-scheduler tries to group the transmissions and receptions of a mobile as much as possible according to the service classes, QoS and current load. Grouping of uplink and downlink traffic for one mobile implicates that there is some space between sending and receiving to allow the transceiver to switch its operating mode from sending to receiving. This has a negative effect on the capacity of the wireless channel. The advantage is that it allows the mobile (i.e. the radio device) to turn its power off for a longer period, and that it makes less operating mode transitions.

A same idea is designed and implemented in [Ma et al., 2009], but in this case authors put their focus on links In a TDMA sleep scheduling, each link l_{ij} is assigned a time slot, in which both sender node v_i and receiver node v_j should start up to communicate. After the allocated time slot, nodes v_i and v_j change to sleep. When using the traditional link scheduling algorithms which schedule the communication links one by one, node v_j may start up w_j times to monitor the channel in a period T, where w_j is the number of neighbors. As addressed before, the frequent startup would consume a large amount of extra energy and time.

A reasonable design is to assign consecutive time slots to all directed links incident to the same node, and then a node only needs to start up once to receive all the packets from its neighbors. We refer to such an interference-free scheduling as the *contiguous link scheduling*. A contiguous link scheduling is said to be valid if all the links incident to one node are assigned consecutive time slots.

Fig. 2.10 shows a sample of the contiguous link scheduling.

This second approach is more detailed and also proved through Markov chains. The main contribution is that scheduling scheme was implemented in a distributed context. The continuity of our proposal start from these results, even if distributed analysis is poor from energy point of view, because is focused on guaranteeing good performances in delays, because of the continuity of links used by flows.

The difference between our and their approach is that the continuity will not be dependent from the link , but from nodes' perspective and their activity of transmission or reception or idle.

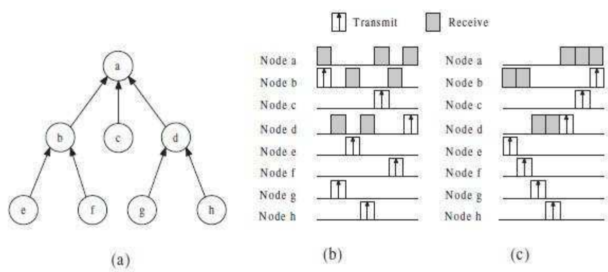


Fig. 2.10. Link scheduling and contiguous link scheduling: (a) Network topology, (b) Link scheduling, (c) Contiguous link scheduling.

2.4 The Network Simulator: ns2

The Network Simulator (ns) is a discrete event network simulation tool very popular for network research. It is an object oriented simulator. The choice of this software is dictated by the fact that it is Open Source (the source code of all versions is wholly available, that allows to have access and modify it without any restriction) and Freeware (the software can be downloaded for free from the official website and it should be compiled before execution).

The development of ns began in 1989; since then the simulator was permanently enhanced. Since 1995 ns is supported by the central research and development organization for the Department of Defense "DARPA" (*Defense*

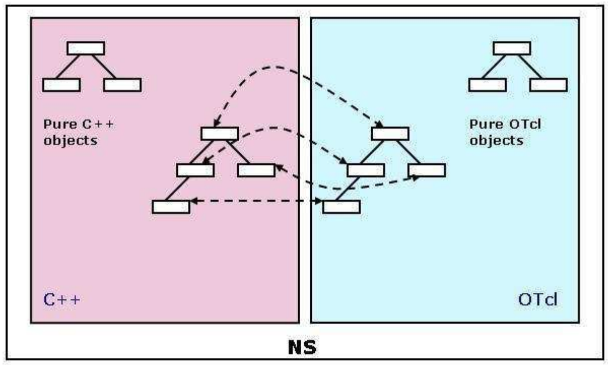


Fig. 2.11. Summary: Shows that NS consist of two languages (Split model). For object created in OTcl has a corresponding object in C++.

Advanced Research Projects Agency [1]). In 1996 version 2 of ns was introduced, containing some major architectural changes (see ns change log [The VINT Project. ns change log, 2008] for details), and the simulator thenceforward named ns-2. In 1997 the (initially for wired networks intended) simulator was extended by the Monarch project [Monarch, 1998] to support wireless networks. Today, ns-2 is the most widely used open source network simulation tool, not only for typical network simulations, but also for “resource allocation, real-time communication, energy issues in ad-hoc networks, transport protocols in wireless sensor networks, and control strategies for wireless robots” [Ivanov et al., 2007]. A review on MANET simulations [Kurkowski et al., 2005] showed that ns-2 holds the first place with 43.8% usage (from a review on wireless network research papers from ACM symposium based on 151 articles from a five-year-period).

The following information and descriptions are relating to ns version 2.33, released on 31st March 2008.

Ns-2’s architecture follows closely those of the Open Systems Interconnection (OSI) Reference Model. That means a simulated packet needs to pass the network layer, link layer, MAC layer and physical layer. On the lower layers, ns-2 currently supports Local Area Network (LAN), Wireless LAN (WLAN), and satellite networks. Static, dynamic, unicast, and multicast routing is supported, as well as several queueing techniques, such as First In, First Out (FIFO) algorithms or stochastic fair queueing. On the transport layer, ns-2 supports the ’usual’ protocols such as TCP, User Datagram Protocol (UDP), and Real-time Transport Protocol (RTP). Beside the provided models shipped with the ns-2 bundle, there exist a large repository of contributed code. About 120 contributions covering all OSI layers, as well as analysing or traffic generation tools can be downloaded from the contribution web page [ns 2: Con-

[1] www.darpa.mil

tributed code, 2008]. Those contributions are not documented within the ns-2 manual; any documentation and maintenance remains at the original author.

Although it is nice to choose from different models or analysing tools, the disadvantage is that different model implementations lead to different simulation results. For example, a comparison of three different ns-2 implementations for the *Optimized Link State Routing* protocol (OLSR) [Gunther and Reineck, 2008] showed significant differences in the results, although all implementations stated to be conform to the Request for Comments 3626 (OLSR). Also, different analysing tools might compute values (e.g. round trip times) differently. As consequence, it might be difficult to compare different studies, as one first has to figure out how each analysing tool calculates the values that are to be compared.

Taking a closer look at the implementation of ns-2, one first notice that the source code is split into two parts (this is one of the major changes from version 1):

- the main simulator is implemented in C++ as it provides sufficient execution speed;
- the configuration part is written in the object-oriented extension for the scripting language Tcl called OTcl.

When starting to simulate, the user has to write a Tcl script, and does not need to write and compile C++ programs. So the combination of the programming language C++ for the simulation itself and the scripting language Tcl as front-end offers a compromise between performance and usability. Only when adding new protocols or models the user needs to write C++ code.

There is a one-to-one correspondence between the modules written in OTcl and C++ in the meaning that they can be viewed and accessed by both languages, but obviously they are implemented in only one of them. User-tier mapping has done through tclcl commands generally embedded in a script; the logical structure of NS-2 is showed in Fig. 2.12.

The Fig. 2.13 shows a user view of Network Simulator.

As mentioned above, the first part when starting to simulate with ns-2 is to write a Tcl script. The script holds all information necessary to run the simulation, as the definitions of the network topology and protocols to be used, generation of network traffic or other time-controlled events, as well as commands to produce trace files.

In the second part, that is the analysis of the simulation output, these trace files are needed, as they basically define the simulation output. More precisely, during the simulation run ns-2 produces trace files for different purposes (in different formatting). One purpose would be the analysis of the trace file by hand. This trace file includes sent, received, forwarded or dropped packets of all nodes (amongst other information), ordered chronologically by the event times. The manual analysis can be very time-consuming, not only in large simulations but also in small simulations e.g. when creating aggregated statistics. A visualisation tool can provide a large amount of data in a man-

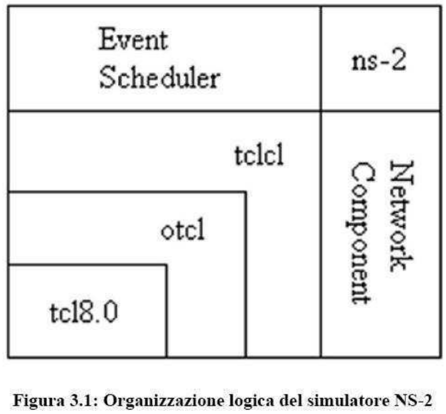


Fig. 2.12. Logic organization of ns2.

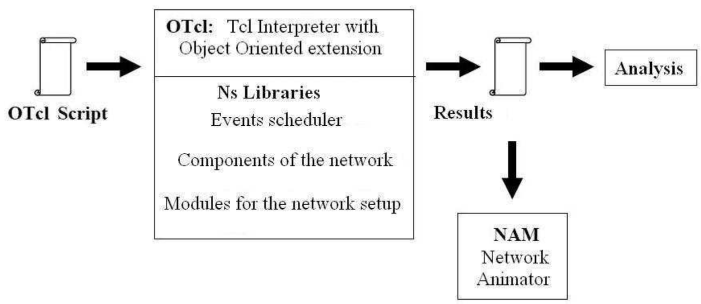


Fig. 2.13. User view of ns2 simulations.

ageable fashion, and determining the same information from textual traces is much more difficult. Several tools exist that help to visualise and analyse ns-2's simulation runs.

Analysing Tools for ns-2

Three examples would be Network Animator *nam* [2], the trace file analyser *Trace graph* [3] and the visualisation and analysis tool for wireless simulations *iNSpect* [4].

Nam allows the user to see a simulation actually 'running'. Nam can visualise the network topology (wired and wireless) and can display packet flows and queues. Using a timeline and 'time actuator', the user can jump to

[2] www.isi.edu/nsnam/nam
[3] www.tracegraph.com
[4] toilers.mines.edu/Public/NsInspect

any point in time, accelerate or slow down the simulation. Nam is part of the ns-2 bundle.

Trace graph is a tool to graphically present ns-2's trace files. It has over 200 different 2- and 3-dimensional graphs that can display throughputs, round trip times, or node dependant information such as the number of forwarded packets. Beside the graphical output, the tool can also aggregate values, e.g. the sum of all TCP bytes received at a certain node. Trace graph in not part of the ns-2 bundle.

INSpect is another visualising tool, initially emerged from the fact that nam then was unable to visualise wireless traffic. INSpect can visualise wireless environments, and can display connectivity graphs, communication ranges, and coordinates of static and mobile nodes. It shows traffic flows and uses colours to indicate sent, received, forwarded or dropped packets.

3

E-TDMA vs. IEEE 802.11

3.1 Comparison of two logics

In the previous chapter we have described, among the others, the characteristics of IEEE 802.11 and E-TDMA. IEEE 802.11 is considered the *de facto* standard for wireless networks, but it could suffer of some drawbacks like access delay, interference grade, and so on, particularly for high traffic load. On the other hand, E-TDMA is characterized by good performances in terms of throughput, delays and served connections, as stated in [Zhu and Corson, 2001a], also for high traffic load. Each of them belongs to a particular MAC protocol typology:

- IEEE 802.11 belongs to *contention-based* MAC protocol family;
- E-TDMA belongs to *collision-free* MAC protocol family.

So they are very different, except for the RTS/CTS and RR/CR handshake, used to try to minimize the hidden terminal problem. In particular, in E-TDMA, each node can reserve a slot for its safe and collision-free transmissions, after a very short and trivial contention period (FPRP phases), to acquire the reservation. That slot will be reserved for all the others following frames, till the duration of the connection (in which case the node has to explicitly release that slot) or till an error occurs (for example caused by the movements of the nodes), however till the node releases it. In IEEE 802.11, instead, a node has always to contend for the permission for a new transmission but nodes do not suffer the cost of maintaining updated own schedules.

We selected each of them as representative of the MAC protocol typology to which it belongs. In this chapter we will offer an experimental comparison between the two protocols from energy point of view, which tries to answer to the following questions and propose possible solutions to minimize energy consumptions and to maximize network lifetime: which MAC protocol typology outperforms the other from energy point of view? How does the energy consumption change if we assume a certain amount of exchanged information? Which is the best MAC protocol in those conditions?

To evaluate performances of E-TDMA vs. IEEE 802.11 we used NS-2, that we introduced in the previous chapter. We report results about two different sets of simulations in the following paragraphs. Our aim is to compare the two target protocols in two different traffic conditions: the main difference between the two ones is, indeed, the generated data packet dimension (*pktdim*): 64 bytes in the first case, and 512 bytes in the second. This is not so trivial, because E-TDMA system uses packet fragmentation, so the packet bursts of the several flows will be longer. We will see that E-TDMA behaves better than IEEE 802.11 also in the second case, and also for high traffic conditions.

3.2 Simulation Environment

Simulation parameters that are common for E-TDMA and IEEE 802.11 and to both sets of simulations are described in the table 3.1.

Transmission Range	$250m$
Length of MAC Queue	$50pkts$
Routing Protocol	AODV
Propagation Model	Disk
Number of nodes	20
Duration of simulation	$300s$
System Bandwidth	$2Mbit$
Simulation Area	$700x700m^2$
Velocity	$v = 0m/s$
Type of flows	CBR
Number of flows	$6:2:14$
Fragment Length	$32bytes$
Packets per flow	$10,000pkts$
Starting time for flows	$80-150s$

Table 3.1. Simulation Parameters.

An ad hoc network is generated as follows: there are 20 nodes in the network, randomly placed in a squared area of 700 m by 700 m; the transmission range of every node is 250 m. Channel bandwidth is 2 Mbit and simulations last 300 seconds. For both protocol we use AODV routing protocol and the ideal DISK propagation model.

The difference between the two sets of simulations are related to packet dimension and transmission rate:

- *first simulation set*:
  - packet dimension = 64 bytes
  - transmission rate = 20 pkts/s
- *second simulation set*:

- packet dimension = 512 bytes
- transmission rate = 4 pkts/s

So for semplicity we will refer to two scenarios as 64-20 or 512-4 respectively.

Each CBR connection (or flow) has a different source node, but two different connections could have the same receiver node. So, 14 main connections is equivalent to say 14 sources in the network, that have to acquire channel for their transmissions and, for this reason, start RTS/CTS or FPRP handshake to acquire the permission and transmit or to reserve a slot/color, respectively.

A *Constant Bit Rate* source generates a number of packets per second (20 or 4, respectively): a target sender node x has a new 64-bytes packet to deliver to its target neighbor y every 0.05 seconds in the first case and a new 512-bytes packet every 0.25 seconds in the second case. Each flow starts at a time randomly chosen between the 80^{th} and 150^{th} second of each simulation and has to send 10,000 packets – in 220 seconds of simulation a node can deliver almost 4,400 (220*20) data packets — or 880 (220*4) in the second set of simulations —, so data exchanges finish by the end of simulation and never before. Thirty different traffic patterns have been generated and results shown are the average of them.

Nodes remain stationary for all the simulation, so a packet is dropped for reasons different by nodes movements.

We refer to the first 80 seconds of simulation as *setup phase* for E-TDMA. Other E-TDMA parameters are shown in the table 3.2.

Contention Phase	Cycles FPRP	c	8
	Number of Temporary Colors	tc	1
	Phases per cycle	f	5
	Total duration of FPRP	T_{FPRP}	$730\mu s$
Allocation Phase	Number of Permanent Colors	pc	15
	Total duration of SU	T_{SU}	$990\mu s$
Information Epoch	Number of INFO slot(*)	is	10
	Number of INFO frames(*)	if	4
	INFO slot duration	T_S	$154\mu s$
	Total duration of INFO	T_{INFO}	$616\mu s$
Total duration of E-TDMA		T_{ETDMA}	$2336\mu s$

Table 3.2. E-TDMA Parameters.

A control epoch uses 8 FPRP reservation cycles, 1 temporary color (for contention phase) and 15 permanent colors (for allocation phase). An information epoch consists of 4 information frames, with 10 information slots for each frame. The duration of every phase is intended for a guard time of $10\mu s$. With this E-TDMA parameters, considering fragment dimension of 32 bytes, we will have that:

- in the first set of simulation packets have 84 bytes after adding the header IP and are transmitted in 84/32 = 3 INFO slots;
- in the second set of simulation packets have 532 bytes after adding the header IP and are transmitted in 532/32 = 17 INFO slots.

This is the main cause of the differences of the two simulation environments for E-TDMA, that makes use of the packet fragmentation, but IEEE 802.11 will show poorer performances because of a longer bandwidth occupancy.

For energy evaluations we used parameters described in the table 3.3: we chose the amount of initial energy in a way that nodes do not consume all their own energy and they are active till the end of simulation.

Initial energy	E_{ini}	$200J$
Transmission power	P_{tx}	$0.6mW$
Reception power	P_{rx}	$0.3mW$
Idle power	P_{idle}	$0mW$
Sleep power	P_{sleep}	$0mW$

Table 3.3. Energetic Parameters.

Nodes do not consume energy in idle mode, so we can evaluate energy due only to data and signaling traffic and it is possible to note the differences between the two protocols regarding to the energy consumption due only to their basic working.

Besides, $P_{idle} = 0$ is similar to say that a node enters immediately in sleep state when it does not take part to any transmission, and the node does not consume energy in transmission or in the sleep state. The reality is quite different: P_{idle} and P_{sleep} are not equal to zero; in fact P_{idle} is one of the major source of energetic overhead, because during the lifetime of a node most of time is usually spent in idle mode, waiting data to transmit or to receive or detecting the channel.

Besides, transition between two different functional states can be also quite consistent for a node, in a particular phase of its battery life, as we will see in chapter 5. For these reasons, setting P_{idle} and P_{sleep} to 0 mW equals to implement ideal and perfect energy-saving strategies.

In the following paragraphs we compare E-TDMA and IEEE 802.11 performances. For every graphic we compare the two protocols for a vaying number of main end-to-end connections (or flows, mc = *main connections*) in the network, from 6 flows to 14, with a step of 2.

First of all, we will show *Packet Delivery Ratio*, as the percentage of the number of received data packets to the number of sent data packets, to measure communication performance of the two protocols. Then, we have considered the following energetic metrics:

- *Average Network Energy Evolution*: the average trend of residual energy of all the nodes during the simulation;
- *Energy Goodput*: ratio of total data correctly delivered to the total energy consumed;
- *Detailed Energy Consumptions*: the average total energy consumptions of all nodes during all the simulation; we particularly focus on energy consumptions due to:
  - state: *transmission* and *reception* - we will not consider *sleep* and *idle* states, but only *active* states;
  - phase: *data* and *control* packet consumptions.

3.3 Experimental Results: 64-20 scenario

3.3.1 Data Packet Delivery Ratio

In figure 3.1 we show communication performance of IEEE 802.11 vs E-TDMA for different number of active flows. Each point shows the number of data packets received over the total sent packets till that time.

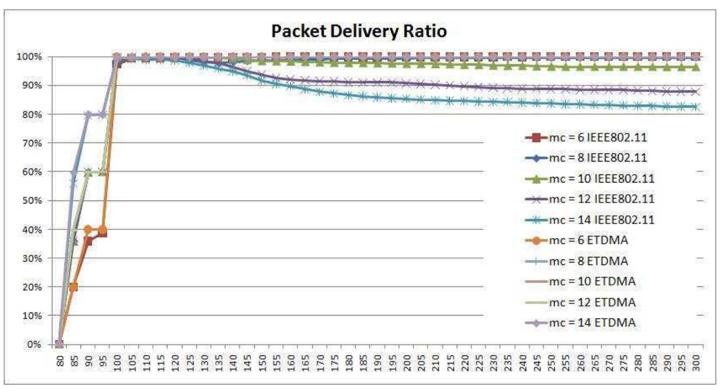


Fig. 3.1. Data Packet Delivery Ratio E-TDMA vs. IEEE 802.11.

Data sessions start between the 80^{th} and the 150^{th} second of simulation. Between t = 100 and 120 s, for both protocols, for every number of active flows there is a value of delivery ratio about 100%. Curves related to IEEE 802.11, except for 6 and 8 flows, begin to behave worse already at 120-130 seconds. E-TDMA, instead, gets optimal results (100%) for every case till the end of simulation (300 seconds). In table 3.4 it is possible to compare the last values of delivery ratio registered in the simulations for both protocols.

Our results are consistent with [Zhu and Corson, 2001a] and, even with many connections (related to the total number of nodes) E-TDMA behaves much better than IEEE 802.11 and indeed it is scalable in the number of connections.

	mc = 6	mc = 8	mc = 10	mc = 12	mc = 14
E-TDMA	100%	100%	100%	100%	100%
IEEE 802.11	100%	99.72%	96.41%	87.79%	82.56%

Table 3.4. Data Packet Delivery Ratio.

3.3.2 Energy Evolution

In Fig. 3.2 we compare the evolution of residual energy of nodes when they run E-TDMA or IEEE 802.11.

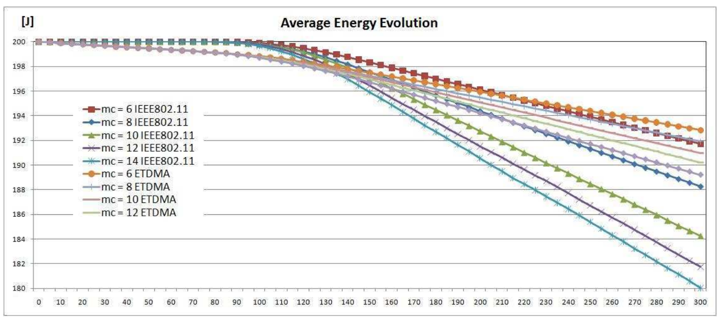


Fig. 3.2. Energy Evolution: E-TDMA vs. IEEE 802.11.

Again, E-TDMA behaves better than IEEE 802.11. In particular we can note that curves change their own trend at about 100 seconds of simulation, because data packet exchanges begin to increase in the network.

We can easily distinguish two groups of curves, the trend of which are quite different:

- E-TDMA curves have a constant consumption of energy among 0 and about 100 seconds, due to control epoch and setup phase; after 100 seconds they become more inclined; nodes consume more energy with more active connections.
- IEEE 802.11 curves do not consume energy till the start of flows; in the second part they behave worse than E-TDMA's. Besides that increasing mc the differential consumptions is less and less, varying from about 3.5-4 J to 1.71 J.

We can also note that E-TDMA with $mc = 6$ and 8 behaves better than IEEE 802.11 with $mc = 6$. In table 3.5 we report results at the last second of simulation.

	mc = 6	**mc = 8**	**mc = 10**	**mc = 12**	**mc = 14**
E-TDMA	192.83	191.91	190.95	190.21	189.20
IEEE 802.11	191.71	188.26	184.26	181.74	180.03

Table 3.5. Energy Evolution $[J]$ $(at = 300s)$.

3.3.3 Energy Goodput

Good performance concerning energy evolution does not show the real energetic goodness of a protocol. The fact that spent energy is low could happen also when throughput is low due to few transmissions. Then we need a way to correlate energy consumption to data delivery and compare adequately the two protocols from the energy point of view.

An efficient metric to calculate this is the Energy Goodput (EG), that could be stated as follows:

$$EG = \frac{\text{total transmitted data}}{\text{total consumed energy}} \tag{3.1}$$

The higher value of EG, the higher energy performance a protocol has, because with it we can count how many data the system can correctly deliver for a unit of spent energy. Its inverse value could be also interesting, showing how much energy is spent to correctly deliver a unit of data.

We have already seen that throughput and energy performance of E-TDMA are very good respect to IEEE 802.11, so we expect good results also in EG, and we can see them in Fig. 3.3. But it is interesting to study the EG registered by both protocols.

As for energy evolution also for EG we can easily distinguish two groups of curves, one for E-TDMA and one for IEEE 802.11: at the beginning IEEE 802.11's EG is better than E-TDMA's. This is due to the fact that IEEE 802.11 delivers initially data more quickly, and then delivers more data than E-TDMA, that tries to acquire a unicast slot for each new flow.

The other curves for IEEE 802.11 are mainly constant. We can see, instead, that E-TDMA not only performs better than IEEE 802.11, but when the number of active flows increases also EG increases. In fact, we have the best value registered for EG; that is to say that the protocol consumes less energy to transmit the same quantity of data, or exchanges more data with the same quantity of energy.

We will see that EG curve has a characteristic shape depending on the particular MAC that is running in the nodes.

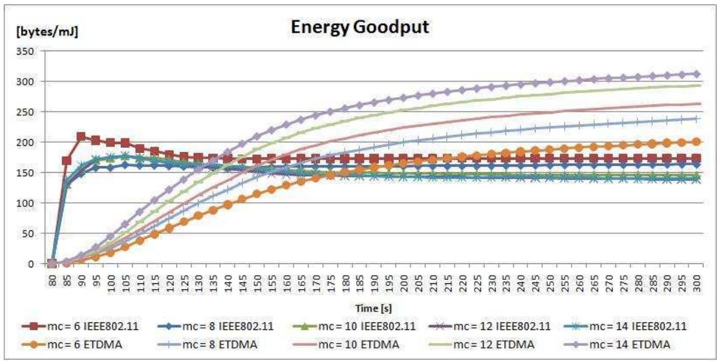


Fig. 3.3. Energy Goodput: E-TDMA vs. IEEE 802.11.

In table 3.6 we show the last averaged values of EG at the end of simulations.

	mc = 6	**mc = 8**	**mc = 10**	**mc = 12**	**mc = 14**
E-TDMA	200.80	238.54	263.48	293.59	312.88
IEEE 802.11	173.60	164.02	146.07	138.38	139.59

Table 3.6. Energy Goodput $[b/mJ]$ $(at = 300s)$.

3.3.4 State Energy Consumptions

In the two following figures (3.4 and 3.5) we compare energy consumptions related to the type of active state for both protocol: transmission and reception state, comparing them also with the total consumed energy.

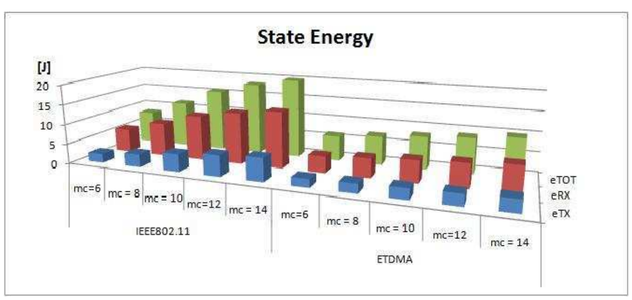


Fig. 3.4. Transmission (E_{TX}) and Reception (E_{RX}) energy consumptions: E-TDMA vs. IEEE 802.11 (without setup phase).

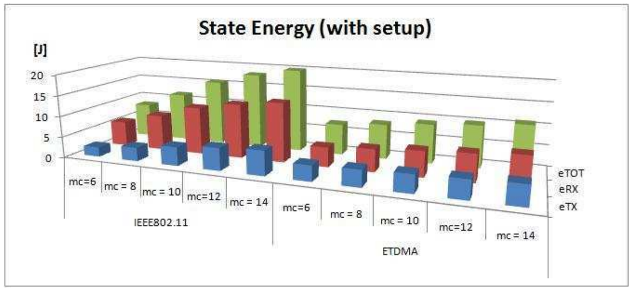


Fig. 3.5. Transmission (E_{TX}) and Reception (E_{RX}) energy consumptions: E-TDMA vs. IEEE 802.11 (considering setup phase).

In Fig. 3.4 we show only energy consumptions due to data delivery period (that is to start evaluations at the 80^{th} second of simulation, then without considering setup consumptions). We can see that energy consumption for E-TDMA is very low respect to IEEE 802.11's. We have to remember that, in our network and conditions, without idle consumptions, IEEE 802.11 does not consume energy till the beginning of information flows (and then of routing procedures to find paths from the several sources and destinations).

In Fig. 3.5 we show the over-all energy consumption, considering setup phase. We can note that the first 80 seconds energy consumptions are quite considerable for E-TDMA; actually, let us have a look at the transmission energy (eTX): we can see that in case of $mc = 6$ and 8 energy consumptions for IEEE 802.11 are less, and became a little more than E-TDMA's if the number of connections increases. It is impossible to note this only considering the Fig. 3.4.

So for a little number of connections, E-TDMA behaves worse than IEEE 802.11; then, increasing the number of connections, energy consumptions do not increase as much as for IEEE 802.11 (from 2.10 to 3.12 J spent in transmission state). This happens because, with a few connections, all the nodes has to acquire at least a broadcast slot, also when it is not needed (without any data to deliver). When flows are more, the probability that a node has to transmit data packet is higher and, then, signalling overhead is balanced with the data consumptions. Energy consumptions depend on the length of setup phase. It is important to find a trade-off between the duration of the setup phase (for energy consumptions) and the communication performance of the protocol. Setup phase is basic to let each node make the reservation of a slot for its broadcast transmissions (usually used for routing packets) and for the convergence of the protocol.

Even if the power to receive a bit is less than the one needed to transmit it (we set $P_{tx} = 2P_{rx}$), we can see that the consumptions in reception state are much more than those related to transmission state; particularly for IEEE 802.11 they are twice the ones relative to E-TDMA. This is because IEEE 802.11 handshake is very expensive due to collisions and retransmissions of signalling packets. In E-TDMA the contention phase is less expensive than in IEEE 802.11, indeed: rarely collision happens among SU packets, that are signalling packets relative to the allocation phase, and FPRP packets are so small: in the Table 6.8 we will see that FPRP consumptions are very low. In general, the energy spent in reception state encounters also the one due to packet overhearing. Energy saving strategies could reduce this consumptions, as we will see in the chapter 5.

3.3.5 Phase Energy Consumptions

The two following figures (3.6 and 3.7) are related to the energy consumptions due to data and control packets delivery.

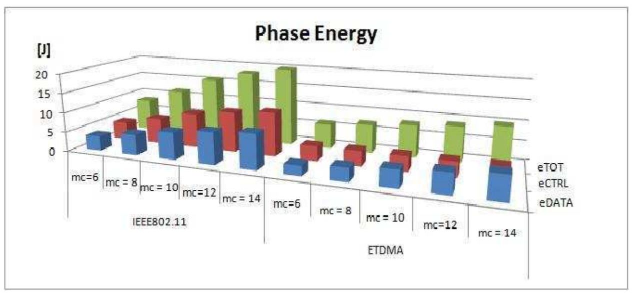


Fig. 3.6. Data (*eDATA*) and Control (*eCTRL*) energy consumptions: E-TDMA vs. IEEE 802.11 (without the setup phase).

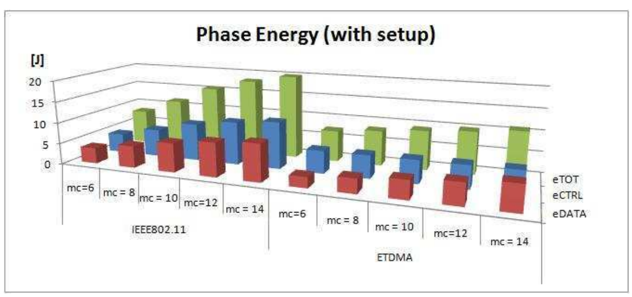


Fig. 3.7. Data (*eDATA*) and Control (*eCTRL*) energy consumptions: E-TDMA vs. IEEE 802.11 (considering the setup phase).

We can note that the energy spent in IEEE 802.11 for signalling is not balanced by that spent for data delivery. In E-TDMA, instead, already for 10 active connections, *eDATA* overcomes *eCTRL*. We are referring particularly to Fig. 3.6, in which we compare only energy due to data delivery. Adding the first setup consumptions, *eCTRL* overcomes *eDATA* in every case.

We can see that energy consumptions due to data packet, in IEEE 802.11 is greater than in E-TDMA. This can be caused by high number of retransmissions. We can see, besides, that energy consumptions due to control packets is heavy. This is caused by the fact that the RTS-CTS handshake is expansive in terms of computation respect to the size of the packets.

3.4 Experimental Results: 512-4 scenario

3.4.1 Data Packet Delivery Ratio

In Fig. 3.8 we compare IEEE 802.11 vs. E-TDMA communication performances.

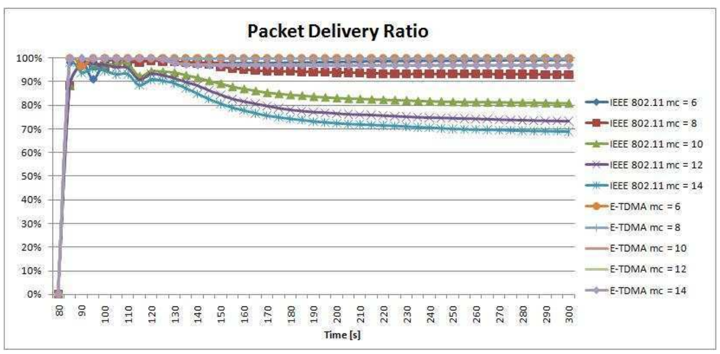


Fig. 3.8. Data Packet Delivery Ratio E-TDMA vs. IEEE 802.11.

As we had seen in the previous set of simulations, E-TDMA gets optimal results (100%) even for high traffic conditions till the end of simulation, while performances of IEEE 802.11 are good only with 6 main connections, but for a bigger number of flows begin to degrade till 70%, and then are even worse than in the previous scenario (20 pkts/s of 64 bytes) results.

In table 3.7 we compare the last values of the two scenario. Also in this case our results are consistent with [Zhu and Corson, 2001a] and E-TDMA is shown to be scalable in the number of connections.

3.4.2 Energy Evolution

In Fig. 3.9 we show the evolution of residual energy of nodes with E-TDMA or IEEE 802.11.

	scenario	mc = 6	mc = 8	mc = 10	mc = 12	mc = 14
E-TDMA	**64-20**	100%	100%	100%	100%	100%
	512-4	100%	100%	97.01%	97.01%	97%
IEEE 802.11	**64-20**	100%	99.72%	96.41%	87.79%	82.56%
	512-4	100%	92.87%	80.81%	73.31%	68.93%

Table 3.7. Data Packet Delivery Ratio.

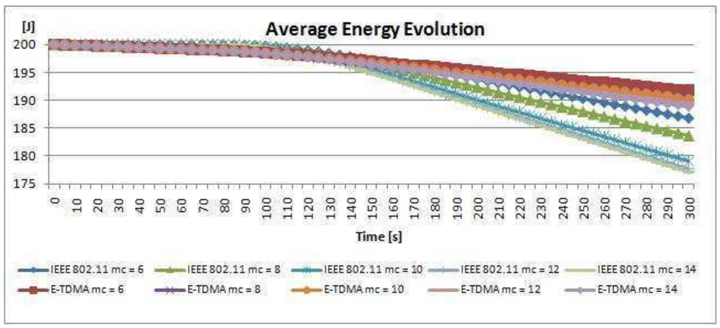


Fig. 3.9. Energy Evolution: E-TDMA vs. IEEE 802.11.

As we have seen before in Fig. 3.2, when traffic conditions get stable (IEEE 802.11 indeed does not consume energy before the beginning of the flows), E-TDMA gives better results than IEEE 802.11 not only for the same value of *mc* but each E-TDMA's point overcomes IEEE 802.11's. This is the main difference related to 64-20 scenario. Also here, two groups of curves are evident, that belong to E-TDMA and 802.11 respectively and vary their own slope at about 100-120 s (when almost all the flows begin): 802.11's curves have a bigger slope and are more spaced out each other than E-TDMA's. In table 3.8 we show results at the end of simulation, compared with those of 64-20 scenario. In the second scenario there is an amount of packets bigger than in the first of a factor equal to 1.6 and for this reason we have more energy consumptions but the gap between two corresponding IEEE 802.11's values is very bigger then that between two E-TDMA's: it varies from 5.00 to 3.00 for IEEE 802.11 and from 1.04 to 0.27 for E-TDMA.

	scenario	mc = 6	mc = 8	mc = 10	mc = 12	mc = 14
E-TDMA	**64-20**	192.83	191.91	190.95	190.21	189.20
	512-4	191.79	190.88	190.17	189.45	188.93
IEEE 802.11	**64-20**	191.71	188.26	184.26	181.74	180.03
	512-4	186.75	183.59	179.05	177.68	177.03

Table 3.8. Energy Evolution $[J]$ $(at = 300s)$.

3.4.3 Energy Goodput

Average Energy Evolution gave very good results for E-TDMA, and Energy Goodput, that is shown in Fig. 3.10, confirms them. Also in this case we need to study in detail the EG registered by both protocols.

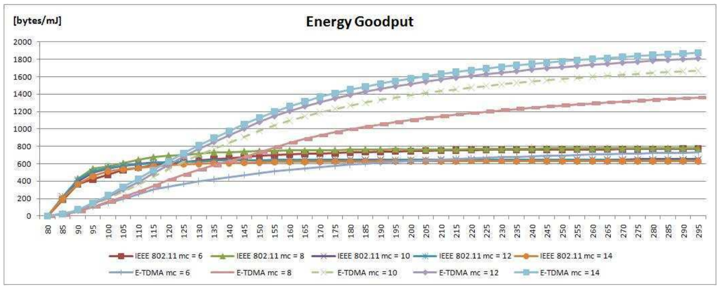


Fig. 3.10. Energy Goodput: E-TDMA vs. IEEE 802.11.

IEEE 802.11's and E-TDMA's EG are respectively recognizable through the different shapes, as we already saw in 3.3.

In this case, after the non-steady period, for 6 connections E-TDMA's performances are almost the same as IEEE 802.11's. When the number of main connections grows, E-TDMA EG grows but the gap between two adjacent curves is less and less. IEEE 802.11 results are almost the same for all the connections.

In table 3.9 we show the last average values of EG at the end of simulations, comparing 64-20 and 512-4 scenarios.

	scenario	**mc = 6**	**mc = 8**	**mc = 10**	**mc = 12**	**mc = 14**
E-TDMA	**64-20**	200.80	238.54	263.48	293.59	312.88
	512-4	770.95	779.31	652.18	639.53	630.43
IEEE 802.11	**64-20**	173.60	164.02	146.07	138.38	139.59
	512-4	734.69	1369.82	1682.40	1822.13	1887.67

Table 3.9. Energy Goodput $[b/mJ]$ $(at = 300s)$.

3.4.4 State Energy Consumptions

In Fig. 3.11 are showed the energy consumptions related to the type of active state for both protocol: transmission and reception state, compared also to the total consumed energy.

Energy consumption for E-TDMA is very low related to IEEE 802.11's. For a little number of connections, E-TDMA behaves worse than IEEE 802.11

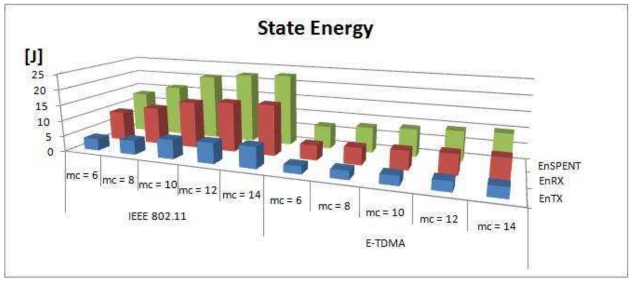


Fig. 3.11. Transmission (E_{TX}) and Reception (E_{RX}) energy consumptions: E-TDMA vs. IEEE 802.11 (without setup phase).

(adding in the figure consumptions due to setup phase); then, increasing the number of connections, energy consumptions do not increase as much as for IEEE 802.11. This happens because, with a few connections, all the nodes has to acquire at least a broadcast slot, also when it is not needed (without any data to deliver). When flows are more, the probability that a node has to transmit data packet is higher and, then, signalling overhead is balanced with the data consumptions.

Even if the energy consumption to receive a bit is less than the one needed to transmit it (we set $P_{tx} = 2P_{rx}$), we can see that consumptions in reception state are much more than those related to transmission state; particularly for IEEE 802.11 they are twice the ones relative to E-TDMA. This is because IEEE 802.11 handshake is very expensive due to collisions and retransmissions of signalling packets. In E-TDMA the contention phase is less expensive than in IEEE 802.11, indeed: rarely collision happens among SU packets, that are signalling packets relative to the allocation phase, and FPRP packets are so small: in the Table 6.8 we will see that FPRP consumptions are very low.

3.4.5 Phase Energy Consumptions

Fig. (3.12 shows the energy consumptions due to data and control packets delivery.

As in the other scenario, also here the energy spent in IEEE 802.11 for signalling is not balanced by that spent for data delivery. In E-TDMA, instead, already for 10 active connections, *eDATA* overcomes *eCTRL*. We are referring particularly to Fig. 3.6, in which we compare only energy due to data delivery. Adding the first setup consumptions, *eCTRL* overcomes *eDATA* in every case.

Energy consumptions due to data packet is greater in IEEE 802.11 than in E-TDMA, perharps due to collision and retransmissions, because the network seems to be a bit congested.

What it's important to note, as in the corresponding graphic for 64-20 scenario, is that energy consumptions due to control packets is already heavy even if with 512-4 scenario data packets are bigger and the RTS-CTS hand-

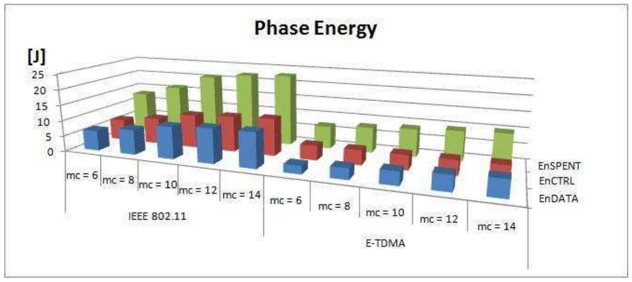


Fig. 3.12. Data (*eDATA*) and Control (*eCTRL*) energy consumptions: E-TDMA vs. IEEE 802.11 (without the setup phase).

shake could be balanced to the size of the packets. In the chapter 6 we will analyse control energy consumptions related to setup phase and to all the E-TDMA signalling.

3.5 Conclusion

In this chapter we presented a comparison between two different MAC protocols, in particular the *de facto* standard IEEE 802.11, belonging to *contention-based* MAC protocol family, and E-TDMA, belonging to *collision-free* MAC protocol family, to analyse communication and energy performance of the two MAC strategies. To better study energetic behaviour of the protocols we tried to correlate energy consumptions to the real amount of correctly received data packets, through the Energy Goodput metric. Simulation results are showed for different traffic conditions, for a varying number of main connections and of packet dimensions generated by flows. We have seen that E-TDMA overcomes IEEE 802.11 in all the cases, both for low and for high traffic conditions, from energy and, in many cases, also from communication point of view, accordingly to the results presented in [Zhu and Corson, 2001a] and [V. Loscrì and Marano, 2004]. For this reason we choose *collision-free* protocols, and in particular E-TDMA, as a target to analyse and propose new techniques and solutions for a good management of energetic resources over mobile ad hoc networks.

4

Energetic Analysis of E-TDMA

4.1 Introduction

Thanks to the very good energetic and communication performances of E-TDMA, E-TDMA was chosen to be the target protocol to be analysed and in which to implement solutions and techniques to face the energetic problem over MANETs, at MAC level.

A possible drawback could be the energetic overhead due to the E-TDMA control phase, that is very heavy in some network conditions, as we have seen, and is not balanced to energy consumptions related to data exchange. What we could do to improve E-TDMA's efficiency is to act on E-TDMA's information phase and verifying the effects of this.

There are three ways to improve bandwidth efficiency of E-TDMA:

- increasing the number of information slots (*is*) in a information frame;
- increasing the number of information frames (*if*) in a super-frame;
- increasing the amount of bytes (*frag*) to send during an information slots, that is equivalent, for E-TDMA logic, to say increasing the length of the maximum fragment allowed on the links of the network, that then means enlarging the dimension of an information slot.

In this chapter we deal with a deep analysis on effects of these three parameters on E-TDMA's energy consumptions.

4.2 Simulation Environment

Wee have conducted three sets of simulations; in each of them we make evaluations varying one of the parameters that we just said before:

- number of information slots;
- number of information frames;
- size of the slots.

Transmission Range	$250m$
Length of MAC Queue	$50pkts$
Routing Protocol	AODV
Propagation Model	Disk
Number of nodes	20
Duration of simulation	$300s$
System Bandwidth	$2Mbit$
Simulation Area	$700x700m^2$
Velocity	$v = 0m/s$
Type of flows	CBR
Number of flows	$6 : 2 : 14$
Generation rate of flows (r)	$4 : 4 : 16pkt/sec$
Number of flows (mc)	10
Packet size	$512bytes$
Fragment Length(*)	$32bytes$
Packets per flow	$10,000pkts$
Starting time for flows	$80 - 150s$

Table 4.1. Simulation Parameters.

All the other parameters are common for all the sets of simulations. Generic parameters are listed in the table 4.1. Almost all are the the same of the table 3.1 that was already described in the previous chapter, but in particular three of them are noteworthy:

- generation rate of flows;
- number of flows;
- packet size.

We decided to have a fixed number of flows ($mc = 10$) and evaluate energy consumptions with a varying rate of transmission from 4 to 16 pkts/s with a step of 4, while the value of the packet size is 512 bytes as the last set of simulations in the previous chapter.

Also E-TDMA and energy parameters (in tables 4.2 and 4.3, respectively) were already explained in the previous chapter.

Some parameters in the tables were marked with (*): each of them (number of information slots and of information frames in table 4.2 and fragment length in table 4.1) will have a varying value in the three following paragraphs, respectively, to evaluate how much each of those parameter affects E-TDMA consumptions.

Contention Phase	Cycles FPRP	c	8
	Number of Temporary Colors	tc	1
	Phases per cycle	f	5
	Total duration of FPRP	T_{FPRP}	$730\mu s$
Allocation Phase	Number of Permanent Colors	pc	15
	Total duration of SU	T_{SU}	$990\mu s$
Information Epoch	Number of INFO slot(*)	is	10
	Number of INFO frames(*)	if	4
	INFO slot duration	T_S	$154\mu s$
	Total duration of INFO	T_{INFO}	$616\mu s$
Total duration of E-TDMA		T_{ETDMA}	$2336\mu s$

Table 4.2. E-TDMA Parameters.

Initial energy	E_{ini}	$200J$
Transmission power	P_{tx}	$0.6mW$
Reception power	P_{rx}	$0.3mW$
Idle power	P_{idle}	$0mW$
Sleep power	P_{sleep}	$0mW$

Table 4.3. Energetic Parameters.

4.3 Experimental Results varying the number of information slots (is)

We consider the E-TDMA hypothesis that each node has reserved at least one information slot (for broadcast transmissions, like we said in the chapter 2). Each packet has a length of 512 bytes, that become 532 with IP header (that has a length of 20 bytes), so each packet will be divided in 17 fragments and lasts 5 frames, while in an E-TDMA superframe there are only 4 frames. When a flow is assigned one and only one slot a frame we will have the values reported in the table 4.4 in which:

- is is the number of information slots in a information frame;
- T_{ETDMA} is the duration of an E-TDMA superframe;
- $T_{TOT-INFO}$ is the total duration of an information period in an E-TDMA superframe;
- T_{pkt} is the time that a node takes to deliver a packet in seconds;
- BW_{EFF} is an adimensional index of bandwidth efficiency of E-TDMA, calculated as fraction of the total E-TDMA bandwidth really used to exchange data and not for signalling, that is equal to:

$$BW_{EFF} = \frac{T_{TOT-INFO}}{T_{ETDMA}}. \tag{4.1}$$

is	T_{ETDMA}	$T_{TOT-INFO}$	T_{pkt}	BW_{EFF}
10	0.0080 s	0.0062 s	0.04 s	0.77
13	0.0099 s	0.0080 s	0.049 s	0.81
15	0.0112 s	0.0092 s	0.056 s	0.82
20	0.0144 s	0.0123 s	0,057 s	0.85

Table 4.4. Changes in parameters caused from variation of the number of *information slots*.

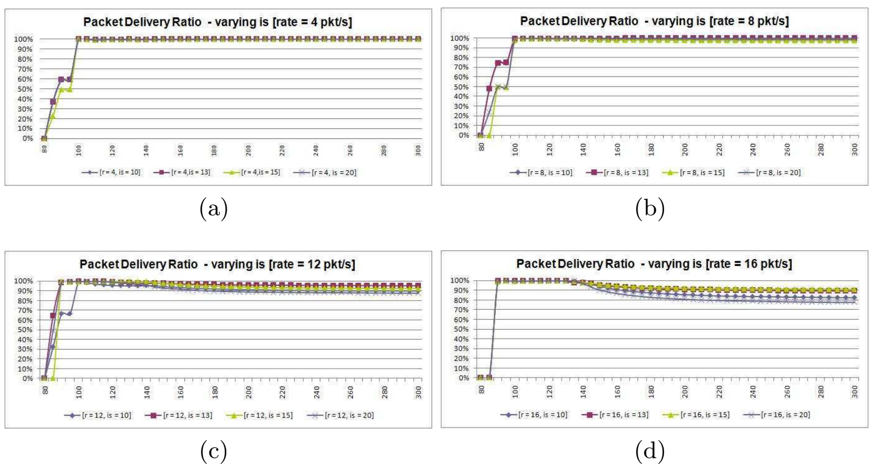


Fig. 4.1. Packet Delivery Ratio varying number of *information slots*.

As we can see from the table 4.4 the duration of a packet, increases when *is* increases, but with more information slots a node has also more chances to acquire more bandwidth for its flows. In this paragraph we will show improvements or drawbacks of energetic and communication performances of E-TDMA when the number of information slots and the transmission rate vary.

4.3.1 Data Packet Delivery Ratio

In Fig. 4.1 we show results related to communication performances of E-TDMA, when the number of information slots varies from 10 to 20 and transmission rate varies from 4 to 16 pkts/s, with a step of 4.

For low values of rates (Figg. 4.1(a) and 4.1(b)) almost all the curves register 100% of data packet delivery ratio. For higher values performances degrade till 90% and 80% respectively for 12 pkts/s (Fig. 4.1(c)) and 16 pkts/s (Fig. 4.1(d)). This happen because rate of generation of the flows is not faced up by a coherent increase of the number of temporary colours to make easier the acquisition of more bandwidth for a flow. However it is interesting the fact

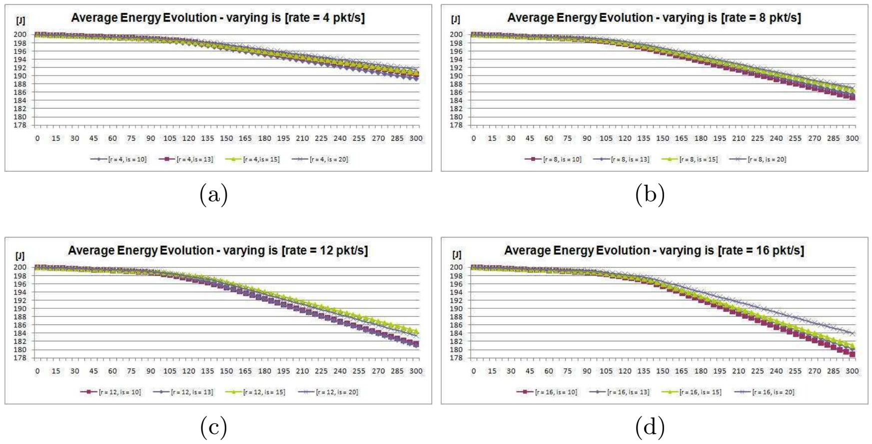


Fig. 4.2. Average Energy Evolution varying number of *information slots*.

that, in the last two cases, the results with *is* = 10 is similar to those with *is* = 20 while the other two curves show the best results.

4.3.2 Energy Evolution

In Fig. 4.2 we show energy evolution in the network.

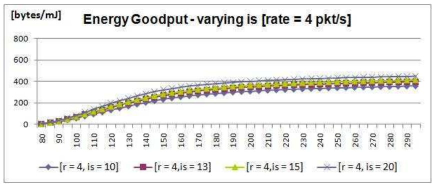
Energy consumptions increase when rate grows and decrease with less information slots. This could seem a good result but, as we just said in the previous chapter and especially for cases in which we did not register good communication performances, to efficiently analyse energetic behaviour of E-TDMA we have to refer to Energy Goodput plots, that are shown in the next paragraph.

4.3.3 Energy Goodput

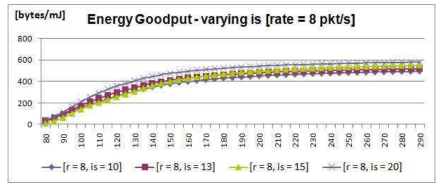
Comparative performance energy-communication is shown in Figg. (4.3).

EG is better when rate and the number of information slots grow. Values go from about 300-400 (Fig. 4.4(a)) to about 600-700 bytes/mJ (Fig. 4.4(b)).

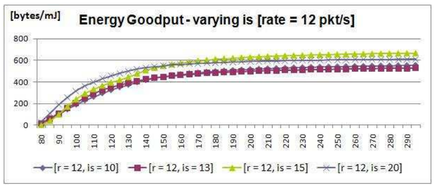
We deduce that is very convenient to increase the number of information slots from energy point of view, but when traffic conditions are too high E-TDMA throughput and communication performances degrade without a contemporary increase of the number of temporary colors, that could let nodes to take advantage and really use a bigger bandwidth, that is more information slots reserved for their own flows.



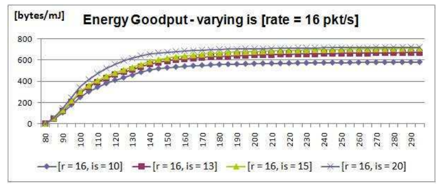
(a)



(b)

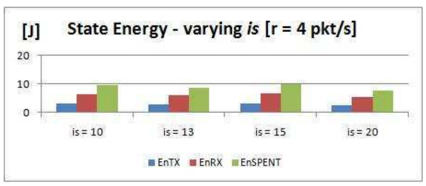


(c)

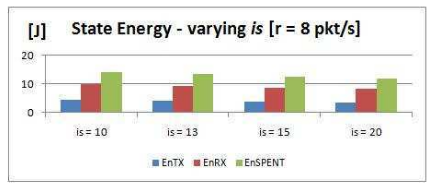


(d)

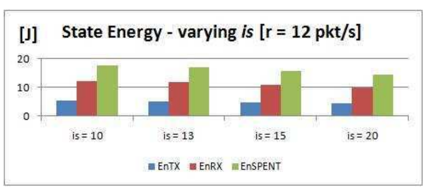
Fig. 4.3. Average Energy Goodput Evolution varying number of *information slots*.



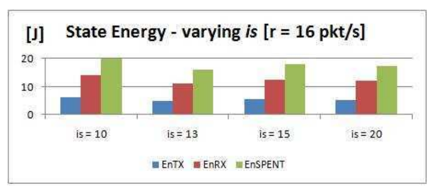
(a)



(b)



(c)



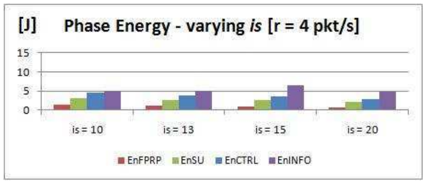
(d)

Fig. 4.4. Total Phase Energy at the end of simulation, varying number of *information slots*.

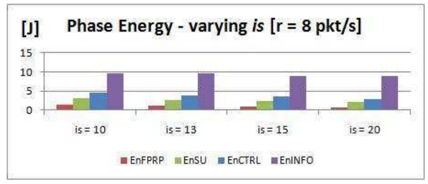
4.3.4 State and Phase Energy Consumptions

We report in Figg. 4.4 and 4.5 results on energy consumptions related to the active states (transmission, reception, total) and phases (control and information) of the protocol, respectively.

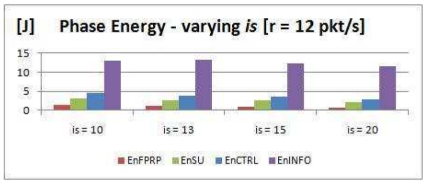
Energy consumptions grow if rate grows while get lower when there are more information slots. Particularly from Fig. 4.5 we can say that data exchange consumptions compensates more and more control phases'.



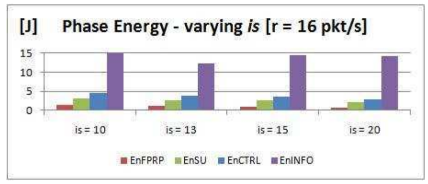
(a)



(b)



(c)



(d)

Fig. 4.5. Total Phase Energy at the end of simulation, varying number of *information slots.*

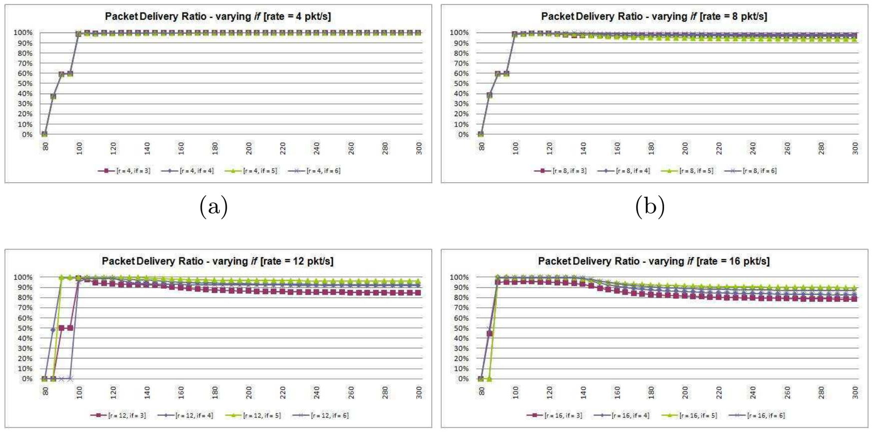
4.4 Experimental Results varying the number of information frames (*if*)

In this paragraph we show results when all parameters have standard value but the number of information frames that varies from 3 to 6 with a step of 1. We report changes on the system caused by the variation of the number of information frames, in the table 4.5 , where *FxPkt* is the number of frames that a node spend to deliver a packet (for the other symbols see the paragraph 4.3.

if	T_{ETDMA}	$T_{TOT-INFO}$	T_{pkt}	BW_{EFF}	*FxPkt*
3	0.0065 s	0.0046 s	0.039 – 0.045 s	0.71	6/7
4	0.0080 s	0.0062 s	0.04 s	0.77	5
5	0.0095 s	0.0077 s	0.038 s	0.81	4
6	0.0111 s	0.0092 s	0,0333 s	0.83	3

Table 4.5. Changes in parameters caused from variation of the number of *information frames.*

Having one more information frame means that there is one more slot in each information period for each node. That's why T_{pkt} decreases with more information frames. In the following subsections we analyse energy behaviour of E-TDMA due to a varying number of information frames and a varying *transmission rate* (from 4 to 16 pkts/s, with a step of 4).



(a) (b)

(c) (d)

Fig. 4.6. Packet Delivery Ratio varying number of *information frames*.

4.4.1 Data Packet Delivery Ratio

As in the previous paragraph, also in this case packet delivery ratio of E-TDMA in almost all cases for low traffic conditions (*rate* = 4 and 8 pkts/s, in Figg. 4.6(a) and 4.6(b), respectively) is equal to 100% (Fig. 4.6). Only for *rate* = 8 pkt/s and *if* = *5* frame there is a worse value registered. When traffic conditions are higher (*rate* = 12 and 16 pkts/s, in Figg. 4.6(c) and 4.6(d), respectively), values decrease till 90% and 80% respectively and the results with *if* = *5* frames seem to be the best, with about 90% of packet delivery ratio.

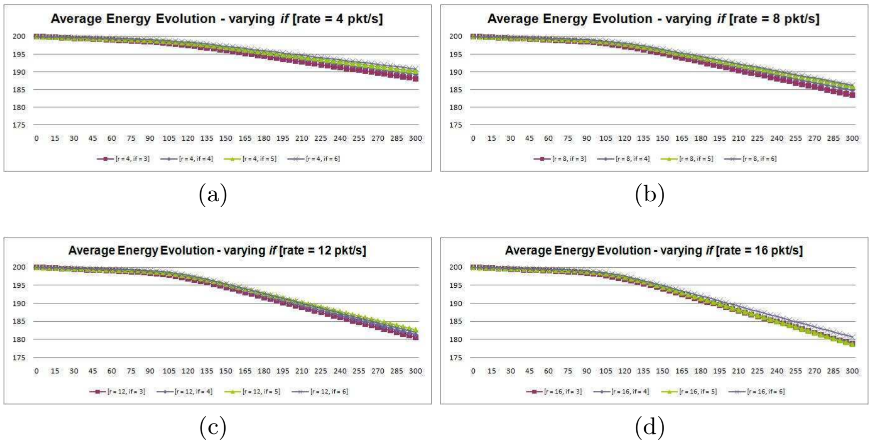
4.4.2 Energy Evolution

Average Energy Evolution of the nodes in the network is shown in Fig. 4.7.

When transmission rate grows, we have more energy consumptions: the gap between rate = 4 pkts/s (Fig. 4.8(a)) and rate = 16 pkts/s (Fig. 4.8(d)) is equal to about 10 Joule. For the same value of rate, instead, E-TDMA behaves with a different value of information frames is quite the same: in particular energy consumptions are less with *if* = 6 frames for every value of *rate* but 12 pkts/s, for which curve with *if* = 5 frames registers the best values.

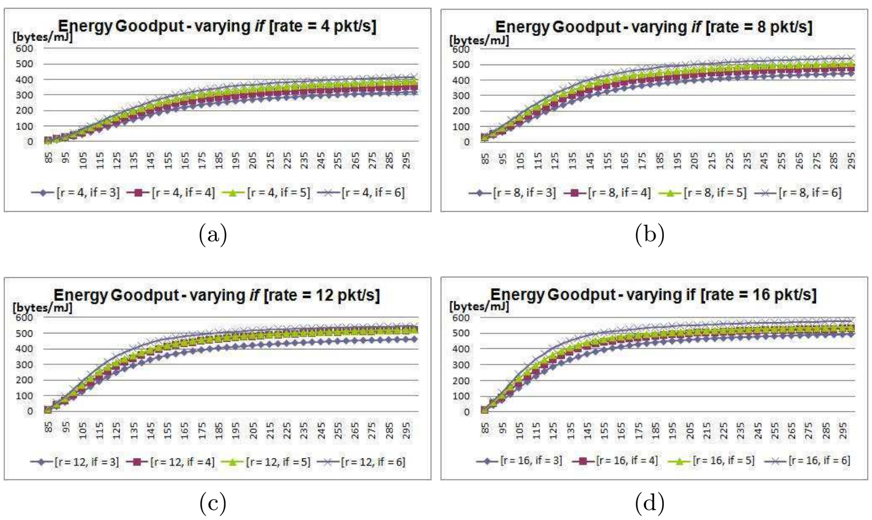
4.4.3 Energy Goodput

Also in this case, because of poor results in packet delivery ratio, EG is essential to better characterize energy behaviour of E-TDMA. We show its average evolution in Figg. 4.8.



(a) (b) (c) (d)

Fig. 4.7. Average Energy Evolution varying number of *information frames*.



(a) (b) (c) (d)

Fig. 4.8. Average Energy Goodput Evolution varying number of *information frames*.

We can clearly see that for all values of *rate* EG grows when also the number of information frames grows, so the fact of increasing the number of information frames causes more energy savings. EG values go from about 300-400 (in Fig. 4.8(a)) to 500-600 bytes/J (in Fig. 4.8(d)), then caused energy savings are low than those possible with the increase of the number of information slots, that we saw in the previous paragraph.

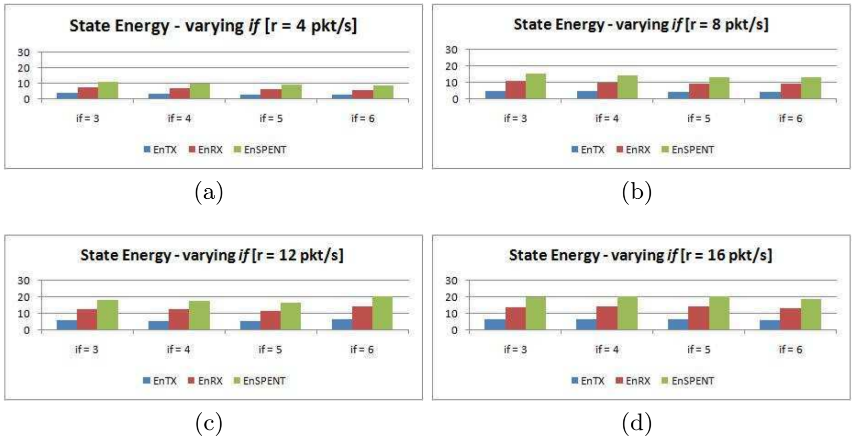


Fig. 4.9. Total Phase Energy at the end of simulation, varying number of *information frames*.

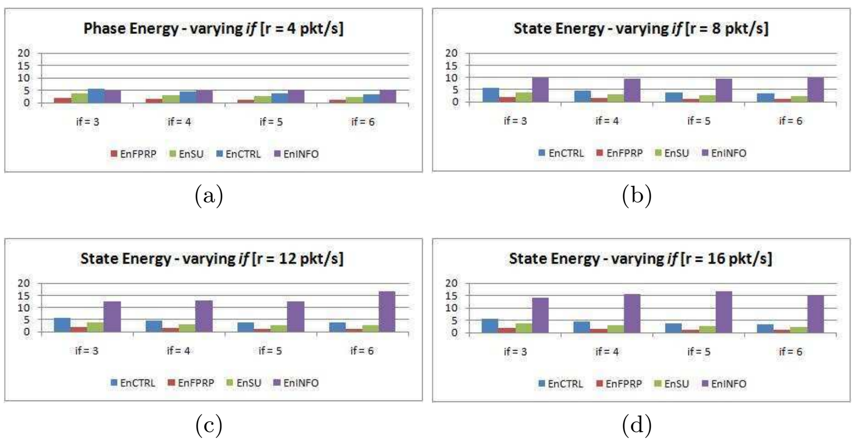


Fig. 4.10. Total Phase Energy at the end of simulation, varying number of *information frames*.

4.4.4 State and Phase Energy Consumptions

In Figg. 4.9 and 4.10 we show particular energy consumptions detailed respect to the states of functioning and the E-TDMA phases.

State energy consumptions grow with the transmission rate and get lower when the number of information frames, and the same thing happens also for phase energy consumptions; but it is interesting to note that control energy consumption is lower with more information frames and information energy consumptions grow and overcomes more and more control energy.

4.5 Experimental Results varying fragment length (*frag*)

In this paragraph we will analyse E-TDMA behaviour for a varying fragment length, then, for a varying duration of the information slots, that is directly related to it. In table 4.6 we report changes on the system when the fragment length varies from 32 to 64 bytes with a step of 32. *SxPkt* is the number of slot that a node takes to deliver a packet; the other symbols were described in the previous paragraphs.

frag	T_{ETDMA}	$T_{TOT-INFO}$	T_{pkt}	BW_{EFF}	*FxPkt*	*SxPkt*
32	0.0080 s	0.0062 s	0.04 s	0.77	5	17
48	0.0105 s	0.0087 s	0.0315 – 0.042 s	0.83	3/4	12
64	0.0131 s	0.0112 s	0.0393 s	0.86	3	9

Table 4.6. Changes in parameters caused from variation of *fragment length*.

Information period is a bit changed, and we have to note that, with the hypothesis of one slot a frame for a flow, T_{pkt} could have variable values for the same case: this is evident in the case of *frag* = 48 bytes, in which T_{pkt} could be smaller or longer than the other two cases, if a packet is sent through 3 or 4 E-TDMA frames (that is *SxPkt* = 3 or 4). This is the reason for fluctuations in analysis that we report in the following subsections.

4.5.1 Data Packet Delivery Ratio

In Figg. 4.11 we show communication performances for varying values of transmission rate (from 4 to 16 pkts/s with a step of 4) and fragment length (from 32 to 64 bytes with a step of 32).

Also in this case, with lower traffic conditions (Figg. 4.11(a) and 4.11(b)) values are always equal to 100%. For higher traffic conditions performances degrade till 90% and 80%, but, differently from the two previous paragraph this is the only case in which standard curve with high traffic conditions (particularly in Fig. 4.11(d)), that is that with *frag* = 32 bytes, show the worst behaviour.

4.5.2 Energy Evolution

Average Energy Consumptions (Fig. 4.12) grow with the *rate* while has a different behaviour between low (Figg. 4.12(a) and 4.12(b)) and high rate (Figg. 4.12(c) and 4.12(d)): the *frag*-64 curves are the best for the first two cases, while the *frag*-48 curves are the best for the last two cases. However the values are very near each other.

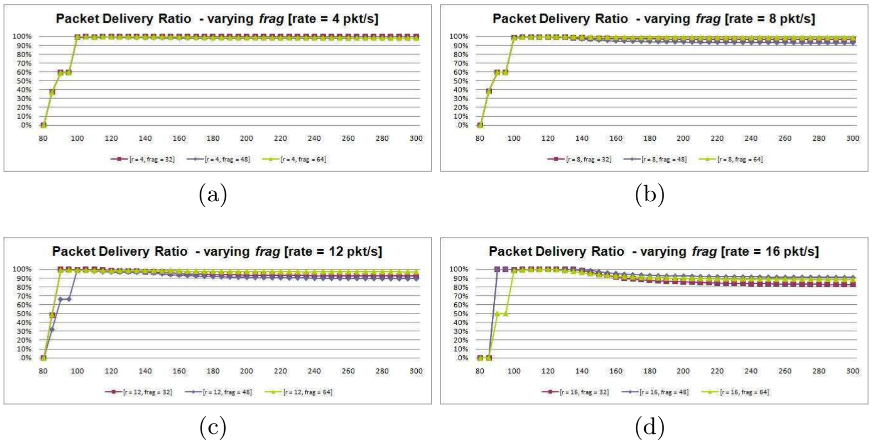


Fig. 4.11. Packet Delivery Ratio varying *fragment length*.

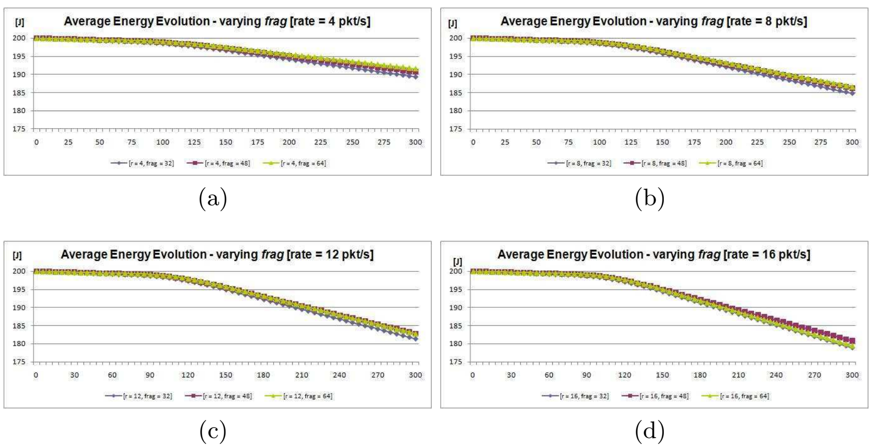
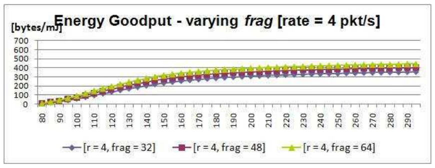


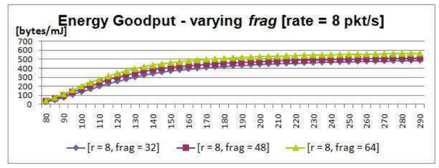
Fig. 4.12. Average Energy Evolution varying *fragment length*.

4.5.3 Energy Goodput

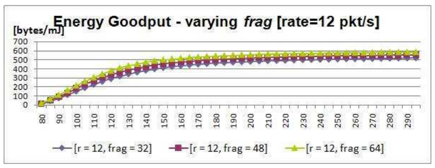
In Fig. 4.13 we show the real energetic performance of the protocol, through the average EG evolution. EG grows when *rate* and fragment length grow, from about 350-450 to about 500-600 bytes/Joule, then, also in this last case, we can say that it is energetically efficient to increase the fragment length. And it is very similar to the previous case (variation of the number of information frames).



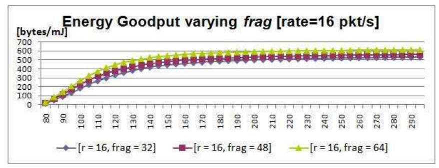
(a)



(b)

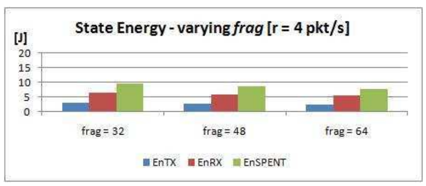


(c)

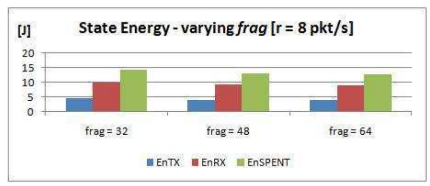


(d)

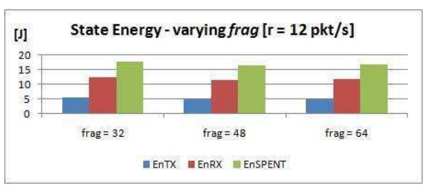
Fig. 4.13. Average Energy Goodput Evolution varying *fragment length*.



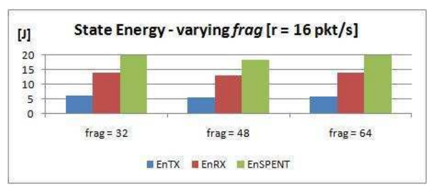
(a)



(b)



(c)



(d)

Fig. 4.14. Total Phase Energy at the end of simulation, varying *fragment length*.

4.5.4 State and Phase Energy Consumptions

In Figg. 4.14 and 4.15 we compare energy consumptions of E-TDMA detailed respect to state and phase. Relating to state, we can note that, even if the overall consumptions grow with the transmission rate, at lower traffic conditions ($rate = 4$, Fig 4.14(a) or 8 Fig. 4.14(a)) energy consumptions get low when fragment length grow, but, with higher traffic conditions ($rate = 12$, Fig 4.14(c) or 16 Fig. 4.14(d)) the best value are reported by the 48-bytes curve.

Relating to phase, energy consumptions grow with *rate* and with *frag*, but also in this case, for high traffic conditions (Figg. 4.15(c) and 4.15(d)), information consumptions for $frag = 48$ show the best values.

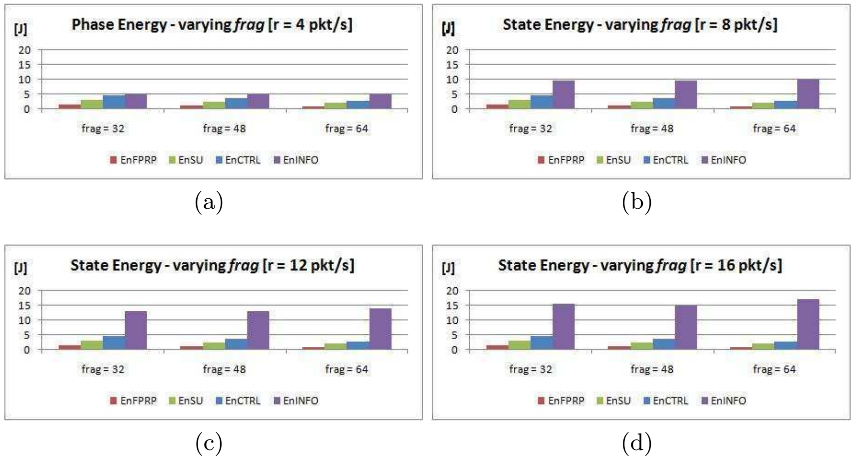


Fig. 4.15. Total Phase Energy at the end of simulation, varying *fragment length*.

4.6 Conclusion

In this chapter we described communicative and energetic behaviour of E-TDMA in particular network conditions, trying to increase bandwidth efficiency, through three sets of simulations with the aim of realizing how and how much three parameters, that are crucial for E-TDMA, could affect this protocol. Through a deep analysis we deduced that, respect to this functioning conditions, increasing E-TDMA bandwidth it's also possible to get more energy savings in the network, with the cost of poorer communication performances. In particular at high traffic conditions performances get lower but changing some parameters we could get some improvements (for example increasing also the number of temporary color, as we said before). Then it could be real that real improvements can be got through the modification of more than one parameter, in particular traffic condition. We have, however, to note that the network with 512-bytes packets and 14 main connections is very stressed, related to the scenario 64-20. In the next chapter we will introduce some first energy saving and energy aware strategy over E-TDMA systems. Also in that condition we will analyse the effects of the increase of the system bandwidth, but only through a varying fragment length, that in this chapter seemed to give the better results, but for a 64-20 simulation scenario. This will be also the scenario for all the following experimental simulations.

5

E-TDMA Information Phase: Models and Energy-Aware Strategies

In this paragraph we are going to illustrate the analysis conducted on E-TDMA and on TDMA systems in general and then we will present the proposed energy-aware strategy to better manage energy resources of wireless nodes in a MANET.

5.1 Energetic Modeling

Time division multiple access (TDMA) method divides the shared medium in frames subdivided in turn into slots, as in Fig. 5.1, where:

- T_f is the time duration of a frame;
- S is the number of slots in a frame;
- T_s is the time duration of a slot and is equal to T_f/S.

In traditional TDMA systems each slot hosts conflict-free possibly concurrent transmissions through a slot reservation phase that comes before data exchanges (*information*) phase, in which nodes can make slot requests and possibly obtain conflict-free slots for their transmissions or receptions, in sender– or receiver–oriented logics respectively. *Reservation* and *information* phases are usually incorporated in a super-frame periodically repeated. In one information phase a frame with the same list of transmissions can be repeated a number of times F that usually is fixed and could depend by QoS requests (more information phases could increase bandwidth efficiency), by hosts mobility rate or by traffic conditions, and so on. After a reservation phase every node knows the exact schedule of transmissions that will occur within its transmission range, during the subsequent correspondent information phase. The analysis in this chapter will involve only information phase.

Nodes in their lifetime spend very much time in idle state [3] but in such a system nodes can easily adopt energy saving strategies to reduce energy consumptions: each node knows when it is free from communications and

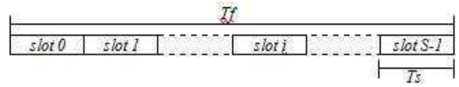


Fig. 5.1. A typical TDMA system.

can switch off its radio opportunely. Generally the transition from one functional mode to another causes additional costs to nodes, from energy and time points of views. These costs are not always mentioned because they are usually negligible, but time and energy costs related to *sleep-active* and *active-sleep* transitions: T_{a-s}, E_{a-s}, T_{s-a} and E_{s-a} respectively. Let us consider a case in which node i has to remain idle for n slots and could plan sleep procedures. If it enters in sleep mode its scheduler will become as depicted in Fig. 5.2: at t_0 the active-sleep transition starts; after T_{a-s}, at $t_1 = t_0 + T_{a-s}$ the node enters sleep mode; after $T_{sleep} = t_2 - t_1 = nT_s - (T_{s-a} + T_{a-s})$, at t_2 sleep-active transition starts and at $t_3 = t_0 + nT_s$ is active to transmit. Sleep procedures could be inhibited or even stopped

- because of time overheads: time during which the node could stay off has to be enough to let the node make active-sleep and sleep-active transitions and be ready to transmit or receive at the beginning of the appropriate slot;
- because of energy overheads: transition could be undesirable for poxible required energy gains.

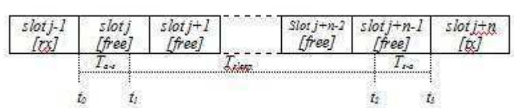


Fig. 5.2. Scheduler of a node entering in sleep mode.

So, if n is the number of free slots for a node, we have to consider the constraints in Eqq. 5.1 and 5.2 to let the node enter in sleep mode:

$$n > \frac{T_{s-a} + T_{a-s}}{T_s} \tag{5.1}$$

$$\begin{cases} n > \frac{E_{s-a}+E_{a-s}+E^*}{T_s P_{rx} - (T_s - T_{s-a} - T_{a-s}) P_{sleep}} & \text{if } T_s > T_{s-a} + T_{a-s} \\ n > \frac{E_{s-a}+E_{a-s}+E^*}{T_s P_{rx}} & \text{if } T_s = T_{s-a} + T_{a-s} \end{cases} \tag{5.2}$$

where P_{sleep} and P_{rx} are power quantities spent in sleep and reception state, respectively. We used P_{rx} instead of P_{idle}, that is the power spent in idle state, to encounter also packets possibly overheard by nodes from other communications in their own range. Nominal values of energetic parameters

P_{tx}	$1.65W$
P_{rx}	$1.4W$
P_{idle}	$1.15W$
P_{sleep}	$0.05W$

Table 5.1. Power quantities for each functional mode of a wireless card.

T_{a-s}	$800\mu s$
T_{s-a}	$800\mu s$
E_{a-s}	$1.4mJ$
E_{s-a}	$0.2mJ$

Table 5.2. Time and energy overheads in transitions active-sleep (a-s) and sleep-active (s-a) for a wireless card.

are described in tables 5.1 and 5.2 [2] [3], in which P_{tx} is the power spent in transmission state.

Also in the following mathematical analysis we will use P_{rx} instead of P_{idle}. E^* is the desired value of energy gains in sleep procedures.

Time overhead of transitions is important to better set parameters of a TDMA system, in particular T_s, to let a node enter in sleep mode as many times as it is needed to get enough energy savings. For the following analysis we will suppose a slot duration much higher than transitions ($T_s >> T_{s-a}+T_{a-s}$) and energetic overhead negligible related to idle consumptions ($T_sP_{rx} >> E_{s-a} + E_{a-s}$) with $E^* = 0$. For these conditions a node can enter in sleep mode even only for 1 slot.

In a network with N nodes, E_i is the value of residual energy of the node $i(0 \leq i \leq N1)$. During the reservation phase a node makes a QoS request of s_i slot/s in f_i frame/s $q_i = q(s_i, f_i)$. We will consider only request for which $s_i \leq f_i$: a request of $s_i > f_i$ could be divided into several $s_i \leq f_i$ requests. If node i makes a QoS request of $q_i = q(3, 4)$ in the contention phase, after a successful reservation, node i gets an information phase schedule like the one in Fig. 5.3.

Frame 0					Frame 1					Frame 2					Frame 3				
sleep	sleep	tx	sleep	sleep	sleep	sleep	tx	sleep	sleep	sleep	sleep	tx	sleep	sleep	sleep	sleep	sleep	sleep	sleep

Fig. 5.3. Transmission schedule of node i.

Total energetic cost $C_{tot}(s_i, f_i)$ [J] of information phase of the request $q_i = q(s_i, f_i)$ of the node i will be:

$$C_{tot}(s_i, f_i) = s_i T_s P_{tx} + s_i(E_{s-a} + E_{a-s}) \\ + [(f_i S - s_i)T_s - s_i(T_{s-a} + T_{a-s})]P_{sleep} \quad (5.3)$$

got by the sum of transmission (P_{tx} is the power spent to transmission state), transition and sleep cost, respectively. Note that the relation is true even if the hypothesis ($T_s >> T_{s-a} + T_{a-s}$) is removed. Without sleep procedures, total energetic cost [J] is:

$$C'_{tot}(s_i, f_i) = s_i T_s P_{tx} + (f_i S - s_i)T_s P_{rx} \quad (5.4)$$

Energetic gain got through sleep procedures is:

$$G_e(s_i, f_i) = (f_i S - s_i)T_s(P_{rx} - P_{sleep}) \\ + s_i(T_{s-a} + T_{a-s})P_{sleep} - s_i(E_{s-a} + E_{a-s}) \quad (5.5)$$

This formulas are valid only for schedule with transmission and free slots. Adding reception slots means adding also the same term for reception energy both in 5.3 and in 5.4, that will not be in the energy gain of 5.5, while the transition cost would be increased or not depending on the distance between neighbor slots: so slot assignment could be important to get more energy gains and this is another reason that led us to propose another slot assignment logic introduced in the next paragraph.

5.2 Energy-Aware Proposed Strategies

In traditional TDMA systems, as we said, a frame is repeated F times in the information phase with the same list of transmissions. So, if a node reserves, for example, the third slot of a frame, it is assigned always the same slot of all the following information frames, also in the next superframes, till it holds the request of it, that is if it has still data to send or no collision or other problems have occurred. We will refer to this slot assignment logic as DASA (*Dynamic Alternated Slot Assignment*). Nodes that adopt sleep procedures will enter in sleep mode every time it is possible. An example of DASA schedule is in Fig. 5.3. Entering in sleep mode produces very high energy savings for the node, but this configuration requires, besides transmission and sleep costs, an overhead in time and energy for active-sleep and sleep-active transitions.

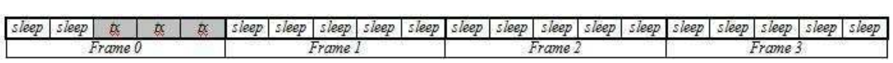


Fig. 5.4. Transmission schedule of node i with DCSA: three contiguous slots in the first frame.

DCSA (*Dynamic Contiguous Slot Assignment*) logic assigns all the requested slots in a contiguous fashion to let a node enter in sleep mode at most twice per flow. With DCSA a possible schedule for the request $q_i = q(3,4)$ could be the one in Fig. 5.4. With DCSA configuration for a generic request we will have:

$$C_{totDCSA}(s_i, f_i) = s_i T_s P_{tx} + (E_{s-a} + E_{a-s}) + [(f_s S - s_i)T_s - (T_{s-a} + T_{a-s})]P_{sleep} \quad (5.6)$$

We encounter only once transition overheads $T_{s-a}+T_{a-s}$ and $E_{s-a}+E_{a-s}$. Cost difference between the two slot assignments will be:

$$G_{eDCSA}(s_i, f_i) = C_{tot}(s_i, f_i) - C_{totDCSA}(s_i, f_i) = (s_i - 1)(E_{s-a} + E_{a-s} - (T_{s-a} + T_{a-s})P_{sleep}) \quad (5.7)$$

These are energy savings related to DCSA assignment when a node is involved only in one transmission. In this case it does not care which of the four frames the requested slots are assigned in. When there are more requests DCSA logic assigns slots as contiguous as possible. If the number of occupied slots grows performance of the two assignment logics could be similar. In this case it is important which slot to assign to each node, in both logics (which group of slot if the assignment chosen is DCSA). In this paper allowed requests are only those with $s_i = f_i$. Future works will analyze solutions to implement QoS request in which $s_i \leq f_i$ and provide energy savings with one or the other logic or let each requesting node choose at runtime the best slot assignment depending on the traffic conditions and its own schedule.

Energy goodness of one logic rather than the other in a TDMA system is strictly related to the minimum quantity n (Eqq. 5.1 and 5.2) to let a node enter sleep mode: a large n naturally lets DCSA perform better than DASA with normal traffic conditions; for a n too small performances could be quite similar and, in some cases, also not so good. So the dimensioning of T_s is very challenging, considering not only traffic and communication issues but also energy issues. T_s is determined by the maximum value of packet fragmentation allowed in the network. For an adequate value of n, DCSA logic is quite equivalent to have one slot bigger, with the same value of packet fragmentation.

We conducted two groups of simulations, that will be reported in the two following paragraphs. In the first [REFER] one we have compared the two slot assignment logics for the same value of fragment length, equal to $32 bytes$. In the first one we have compared the two slot assignment logics for the same value of fragment length, equal to $32 bytes$. According to the equation 5.1 a node will enter sleep mode when it would stay idle for a number n of slots such that:

$$n > \frac{T_{s-a} + T_{a-s}}{T_s} = \frac{500}{298}\mu s = 1.68\mu s \Rightarrow n \geq 2 \quad (5.8)$$

With a bigger fragment length, and then T_s, n could be lower: still 2 with $frag = 48$ and 1 with $frag = 64$. In the second group of simulations we have consideredthe sensitivity of the the two logics to the fragment length and, in particular, we analyse how it affects the energy performance, like control energy results, (in the next Chapter 6 we will report energy consumptions of E-TDMA control epochs): with a bigger fragment length, so with bigger data slots and data frames bandwidth efficiency increases. Simulations will evaluate how much fragment (then n) variation affect E-TDMA in DASA and in DCSA case, considering also a varying number of data connections. As we will see, a good choice for fragment length has to consider also the number of main connections in the network.

5.3 Experimental results

In this paragraph we will show result about the first set of simulations. To evaluate performance of DASA vs. DCSA slot assignment logic both were implemented on NS2 over E-TDMA. Simulation and E-TDMA parameters are described in tables 5.3 and 5.4, respectively. Traffic pattern and all other scenarios parameters follow the same rules described already in the paragraph 3.2 to which we remand to better understand the analysis.

Energetic parameters considered in our simulation are shown in table 5.5. As for all the other simulations, the amount of initial energy is very high: the nodes do not consume all their own energy and they are active till the end of simulation. But now the values are different by the table 3.3 because we insert energy-saving procedures in the E-TDMA protocol. For simplicity we consider the duration of active-sleep transition equal to the reverse sleep-active transition: $T_{tr} = T_{a-s} = T_{s-a}$, so as the power spent in the two processes: $P_{tr} = P_{a-s} = P_{s-a}$. Generally this is not true for a real node, as we saw in the previous paragraph; the assumption was made because of NS Energy model: we have chosen the average of the values considered.

As in the previous experimental paragraphs, we will show results about:

- *Packet Delivery Ratio*;
- *Average Network Energy Evolution*;
- *Average Energy Consumptions* and in particular:
  - *Idle Energy*
  - *Residual Energy*
  - *Information-frame Energy* and
  - *Reception Energy*;
- *Energy Goodput.*

5.3.1 Data Packet Delivery Ratio

In Fig. 5.5 we compare communication performance of traditional DASA versus the new proposed slot assignment DCSA. Between the 80^{th} and the 100^{th}

Transmission Range	$250m$
Length of MAC Queue	$50pkts$
Routing Protocol	AODV
Propagation Model	Disk
Number of nodes	20
Duration of simulation	$300sec$
System Bandwidth	$2Mbit$
Simulation Area	$700x700m^2$
Velocity	$v = 0m/sec$
Type of flows	*Constant Bit Rate*
Generation rate of flows	$20pkt/sec$
Number of flows	$6:2:14$
Packet size	$64bytes$
Fragment Length	$32bytes$
Packets per flow	$10,000pkts$
Starting time for flows	$80^{th} - 150^{th}sec$
Min No. of Slots to Sleep(n)	2

Table 5.3. Simulation Parameters.

Contention Phase	Cycles FPRP	c	8
	Number of Temporary Colors	tc	1
	Phases per cycle	f	5
	Total duration of FPRP	T_{FPRP}	$1.04ms$
Allocation Phase	Number of Permanent Colors	pc	15
	Total duration of SU	T_{SU}	$2.336ms$
Information Epoch	Number of INFO slot	is	15
	Number of INFO epoch	if	4
	INFO slot duration	T_S	$17.88ms$
Total duration of E-TDMA		T_{ETDMA}	$21.256ms$

Table 5.4. E-TDMA Parameters.

Initial energy	E_{ini}	$500J$
Transmission power	P_{tx}	$1.6mW$
Reception power	P_{rx}	$1.4mW$
Idle power	P_{idle}	$1.3mW$
Sleep power	P_{sleep}	$0.08mW$
Transition duration	T_{tr}	$250\mu s$
Transition power	P_{tr}	$0.2mW$

Table 5.5. Energetic Parameters.

second we can note an oscillation among the results, because flows start. After the 100^{th} second there is no more difference among curves: each one shows 100% of data delivery, consistently with [Zhu and Corson, 2001a] and [**?**]. This is because slot assignment has impact only for energetic issue: each node has the same number of slots during which it can send or receive data in each information phase with DASA or with DCSA. So the amount of exchanged useful data is the same.

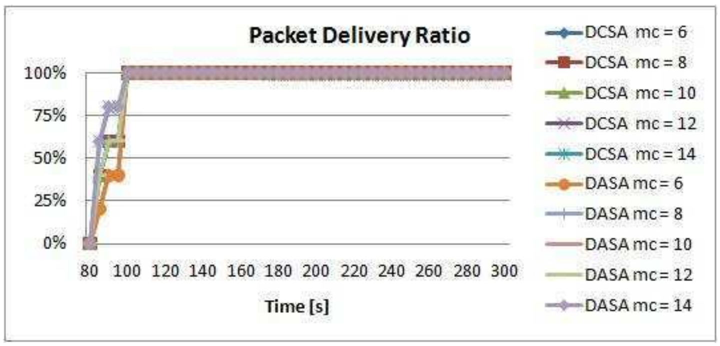


Fig. 5.5. Delivery ratio (%) for DASA vs. DCSA (80-300 s).

5.3.2 Energy Evolution

In Figg. 5.6 and 5.7 we compare residual energy of nodes when they run DASA or DCSA slot assignment strategy, in particular we show the evolution of it and its values at the end of simulation. Unlike the previous figure, now we can see the better performance got through the proposed slot assignment (DCSA). For both slot assignment logic, as the number of main connections increases, the energy spent increases a bit, within the same slot assignment. For a clearer comparison in Fig. 5.6 we have shown only results of mc=6 and mc=14 because values are very similar.

5.3.3 Energy Goodput

EG (*Energy Goodput*) is an index for energy/throughput performance together. We have already seen that DCSA logic performs better than DASA in energy evolution, while both have the same results related to throughput. So we expect good performance in EG (Fig. 5.8). EG grows with the number of flows. As we expected DCSA EG outperforms DASA EG. It is interesting to compare the two logics. In fact EG for DASA/8 (that is DASA with 8 main connections) is similar but still lower than DCSA/6 and for DASA/12 and DASA/14 EG is only higher respectively than DCSA/8 and DCSA/10. Going

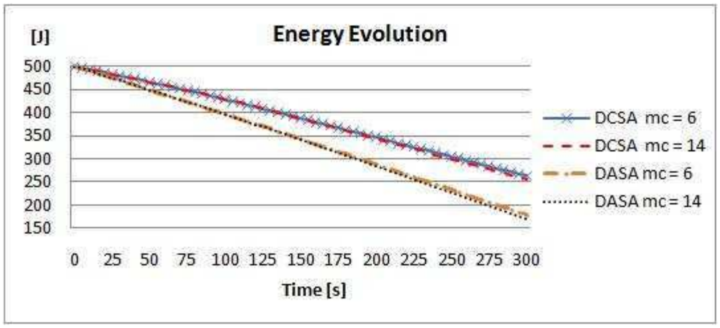


Fig. 5.6. Energy Evolution [J] for DASA vs. DCSA (0-300 s).

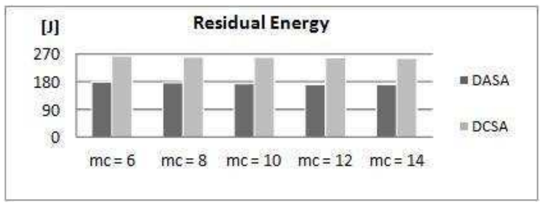


Fig. 5.7. Residual Energy [J] for DASA vs. DCSA (at 300 s).

on, difference between the two logics is higher and higher. As we can also note from table V, EG for DCSA logic increases of about 15 bytes/mJ adding two new flows, while EG for DASA increases less (11.91, 10.62, 9.38) except for mc=14 (14.08), in which we could note little and proportional improvements also for energy evolution or throughput, respect to DCSA. But EG also for DCSA/14 is higher. That is to say DCSA logic consumes less energy to transmit the same quantity of data, or exchanges more data than DASA with the same quantity of energy.

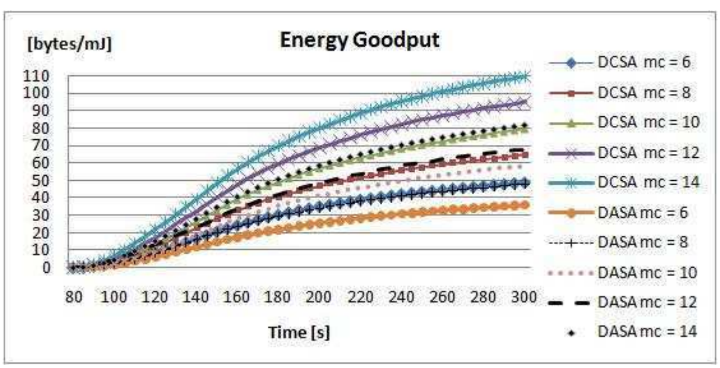


Fig. 5.8. EnergyGoodput [bytes/mJ] for DASA vs.DCSA (80-300 s).

5.3.4 Average Energy Consumptions

In this paragraph we will show detailed results about energetic consumptions during reception, and idle states and information phase of E-TDMA, comparing the two chosen assignment logics. There are negligible differences in the energy spent in transmission, like also for amount of delivered packets, due only to random nature of simulations: as we have already seen before, communication performances are quite the same for both logics.

When nodes run DASA, they enter in sleep mode less often than DCSA. This causes a higher amount of energy spent in idle mode (Fig. 5.9), but also in receiving mode (Fig. 5.10) due principally to the overheard energy. Even if the amount of correctly received data is the same as DCSA (as we saw for throughput), being unable to enter in sleep mode and staying idle could let a node overhear packets not properly addressed to itself. The amount of overheard energy grows with the number of connections, varying from 0.827 J to 2.887 J. Besides, DCSA energy gain in idle energy varies 82.454 J for 8 connections to 86.641 J for 12 connections.

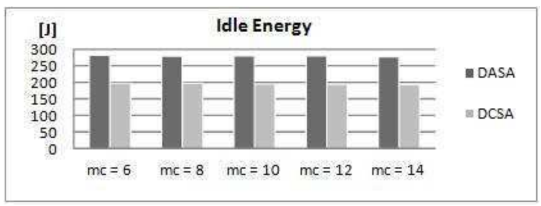


Fig. 5.9. Idle Energy [J] for DASA vs. DCSA (at t=300 s).

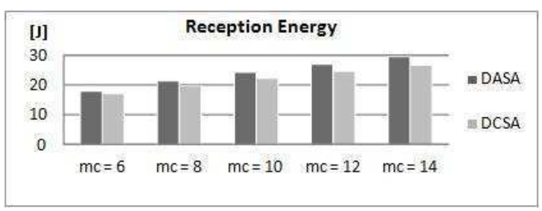


Fig. 5.10. Reception Energy [J] for DASA vs. DCSA (at t=300 s).

As the use of a different slot assignment logic involves information phase of E-TDMA the effective energetic earnings got through DCSA are in Fig. 5.11 that shows the total averaged energy spent in this phase: DCSA improvements go from 77.4 J to 80.75 J.

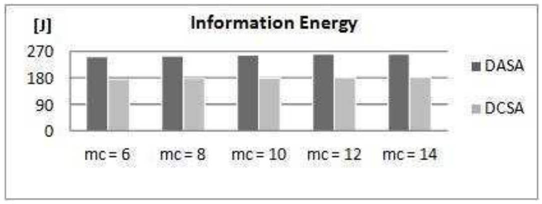


Fig. 5.11. Information Energy [J] for DASA vs. DCSA (300 s).

5.4 Experimental results varying FRAG

Simulations reported in this paragraph will evaluate how much fragment variation (T_s and then n) affects E-TDMA in DASA and in DCSA case, considering also a varying number of data connections. Simulation, E-TDMA and Energy parameters are the same as in the previous paragraph (see tables 5.3, 5.4 and 5.5).

In table 5.6 the effects caused by different fragment dimensions used are reported.

		Fragment length (*frag*)		
		32	48	64
Total duration	INFO slot	$298\mu s$	$426\mu s$	$554\mu s$
	INFO frame	$4.47ms$	$6.39ms$	$8.31ms$
	INFO period	$17.88ms$	$25.56ms$	$33.24ms$
	E-TDMA frame	$21.256ms$	$28.936ms$	$36.616ms$
Bandwidth Efficiency		0.84%	0.88%	0.91%
Fragments Per Packet		$2.625 \Rightarrow 3$	$1.75 \Rightarrow 2$	$1.3125 \Rightarrow 2$
Packets Per INFO Frame		$1.33 \Rightarrow 1$	2	2
Min No. of Slots to Sleep(n)		$1.68 \Rightarrow 2$	$1.173 \Rightarrow 2$	$0.9025 \Rightarrow 1$
Packet time (T_{pkt}) (Best – worst [ms])	DASA	$9.238 - 12.614$	$6.816 - 10.192$	$8.864 - 12.24$
	DCSA	$0.894 - 20.958$	$0.852 - 28.084$	$1.108 - 35.508$

Table 5.6. Effects on parameters of variation of fragment length.

T_{pkt} is intended for the assignment of one and only one slot per frame to a connection. For both case we can see that $frag = 48$ has the best results, even if only for the first value in DCSA. Variance of DCSA in packet time duration could have many effects on queue analysis that will be presented in a future work. All the differences in throughput in DASA and DCSA are caused by these values.

We have compared DASA and DCSA on the same metrics introduced in the previous paragraph, to which we have added *Control Energy* that is the percentage of energy spent in signalling and control traffic (we refer to *contention* and *allocation* phases of E-TDMA) with reference to the total energy spent in the whole simulation.

5.4.1 Data Packet Delivery Ratio

DASA and DCSA present good results in packet delivery ratio. At about $t = 100s$ delivery ratio values raise till about 100% for all flows (Fig. 5.12). As *mc* and *frag* grow, delivery ratio gets lower but only 4 of them are under 99% even if they are always above 96%. This is due to the fact that there are too many connections in our network. However packet delivery ratio is always above 96% with DASA, and with our proposed logic DCSA is always above 98%. Little differences are due to randomness of the network.

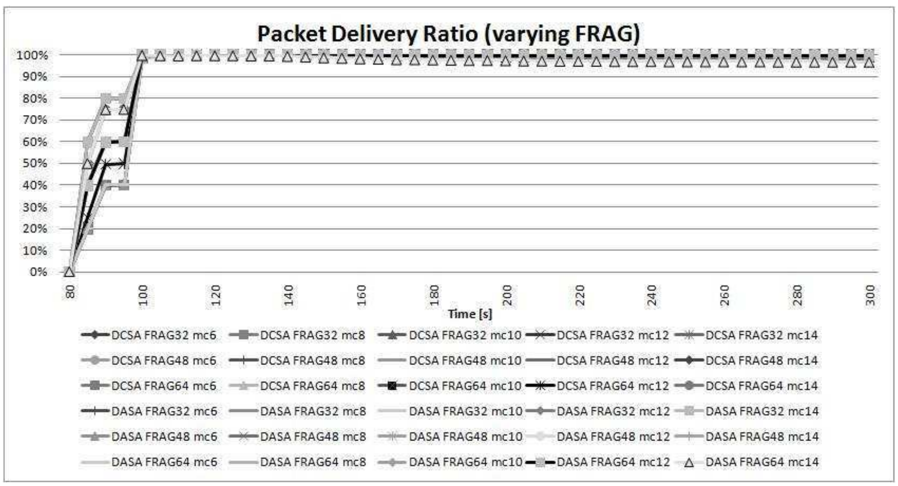


Fig. 5.12. Packet Delivery Ratio (%received over sent data) as fragment length changes (DASA vs. DCSA).

5.4.2 Energy Evolution

Energy evolution results (Fig. 5.13) are averaged for different values of *mc* because they are almost similar for the same value of *frag* and for the same slot assignment logic. Values grows from DASA *frag32* (fragment length=32bytes) to DCSA *frag64*, so it is evident that DCSA outperforms DASA. In Fig. 5.14 we show detailed results for residual energy in the last second of simulation: residual energy for both logics is lower within the same *frag* when *mc* grows and each value with DCSA is not only better than the related one with DASA but even all the DCSA values outperform the others with DASA. Besides, a bigger fragment length effects more DASA than DCSA.

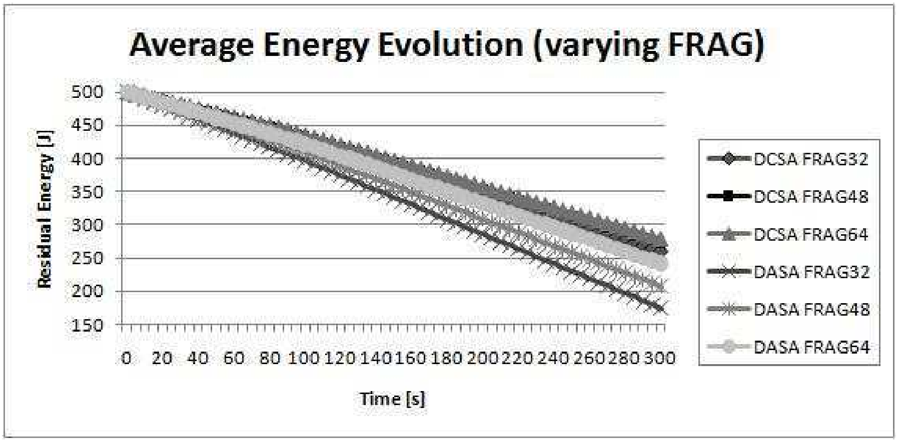


Fig. 5.13. Average Energy Evolution varying fragment length (DASAvs.DCSA).

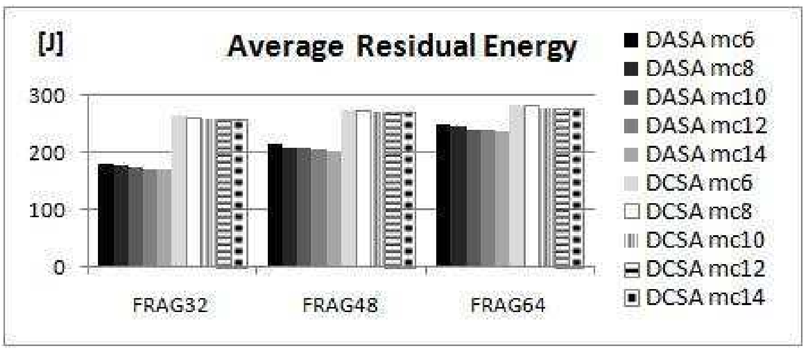


Fig. 5.14. Average Residual Energy varying fragment length and no. of main connections (DASA vs. DCSA) at t=300s.

5.4.3 Energy Consumptions

We can clearly see improvements of DCSA slot assignment through energy saving related to idle state (Fig. 5.15). With DASA energy savings grows more than DCSA ones when *frag* grows: in DCSA values are quite similar but we can note a light improvement when *mc* grows within the same *frag*. However there is not a considerable improvement in idle spent energy even if in the information phase there are more transmissions with a bigger *frag*. This happens because idle periods occur also during control phase where nodes cannot enter in sleep mode .

In Fig. 5.16 we show energy spent by nodes during receiving operations. As we expect, reception energy grows when mc grows for the same value of *frag*. As quantity of exchanged data is about the same, differences represent the amount of energy spent for overheard packets. When *mc* varies DCSA values

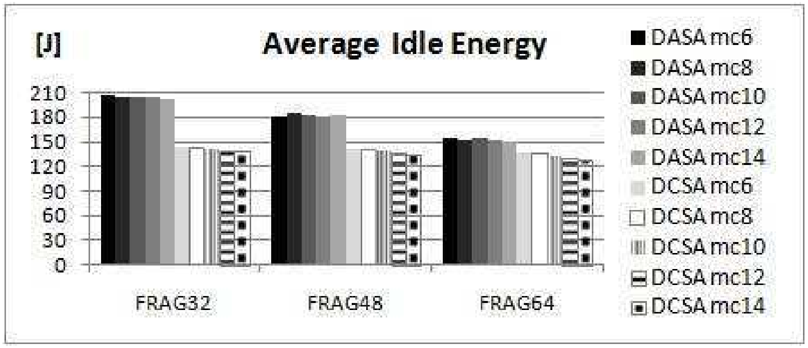


Fig. 5.15. Average Idle Energy varying fragment length and no. of main connections (DASA vs. DCSA).

are better than corresponding DASA ones and the improvement is about the same. However values for frag48 are better than for frag64 and this improves when *mc* grows. This happens because with frag64 many nodes have to stay idle because connections are many and they cannot enter in sleep mode.

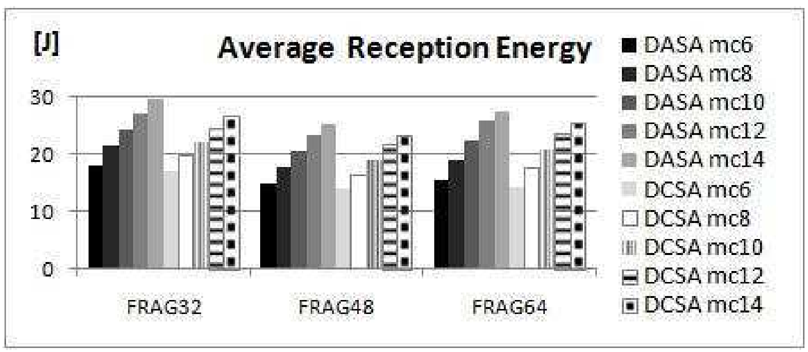


Fig. 5.16. Average Reception Energy varying fragment length and no. of main connections (DASA vs. DCSA).

In Fig. 5.17 we plot energy spent during information phase of E-TDMA. DCSA results are better than DASAs even if the amount of traffic is more or less the same. For DCSA the improvement is less relevant than for DASA. Looking also at the residual energy plot in Fig. 5.14, we can say that information and idle energy (a part of which is encountered also in the first quantity) determine the general evolution of energy in the network.

In Fig. 5.18 we plot the percentage of control energy above the total that is spent in the overall network during the whole simulation. The worst values are due to lower quantity of spent energy in information phase, that is the E-TDMA period affected by introduction of DCSA logic; while the energy spent

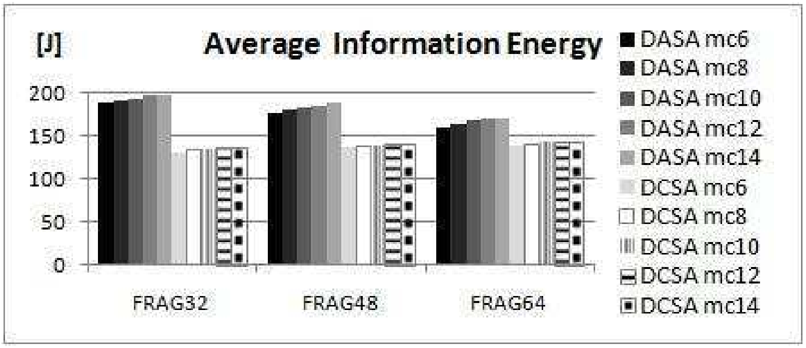


Fig. 5.17. Average Information Energy varying fragment length and no. of main connections (DASA vs. DCSA).

for control period is quite the same for the two considered slot assignment logics. The gap between DASA and DCSA reduces clearly when *frag* grows and lightly when *mc* grows. Besides, the percentage of a certain *mc* and *frag* of DCSA is quite the same as the related one in DASA with the same *mc* and a little *frag*. We have to note that we used high power values and however we could have better result with a longer simulation, or with some energy-aware technique in control period. This will be investigated in future works.

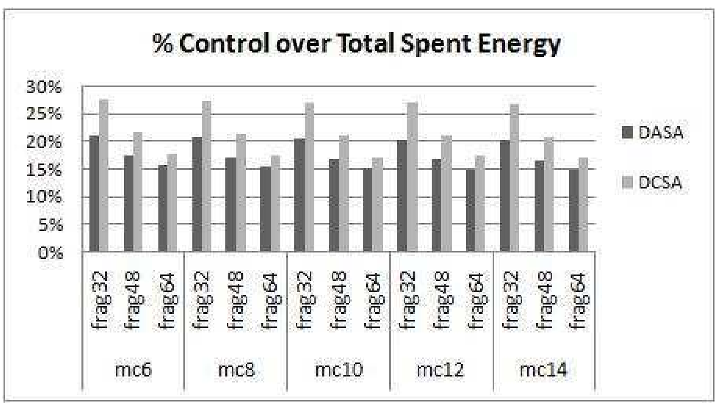


Fig. 5.18. Percentage of Control vs. Total Spent Energy varying fragment length and no. of main connections (DASA vs. DCSA).

5.4.4 Energy Goodput

DCSA registers always better results than DASA, and then we expect good results also in EG. As we saw in [12] the trend of *EG* is specific of the used

MAC protocol (E-TDMA in this case), so the two slot assignment logics have a similar EG trend (Figg. 5.19 and 5.20: for both energetic performance is when *frag* and *mc* grow. The main difference between DASA and DCSA is that, with more connections, DCSA has always an EG better whatever is *frag*, while in DASA there are better results with *frag64* than the curve with next *mc* and *frag32*. This is basically due to *n*, because also for one only slot (with *frag64*) DASA node can enter in sleep mode. Then may be reasonable choosing fragment length depending on number of main connections. Improvement in DCSA curves when *mc* grows is very bigger than in the related DASA ones. Finally we can say that DCSA outperforms DASA from energy point of view.

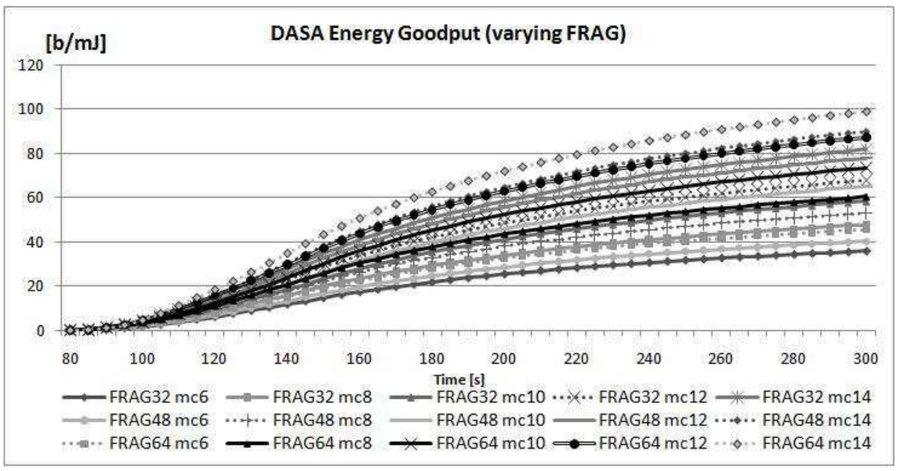


Fig. 5.19. Energy Goodput varying fragment length and no. of main connections (DASA).

5.5 Conclusions

In this chapter we described analysis and simulation results about a new energy-aware strategy for TDMA MAC protocols. The new logic is applicable to the data-packet exchange phase not only of the target protocol (that is the E-TDMA) but also to each TDMA MAC protocol that makes use of the same kind of information frames. the new strategy was compared with the traditional TDMA system and reported very well results for different simulation scenarios and traffic condition without degrading communication performance of the protocols that implements it.

In the last chapter we will conduct another analysis related to the contention-phase of TDMA protocols, to be integrated with the one presented in this chapter.

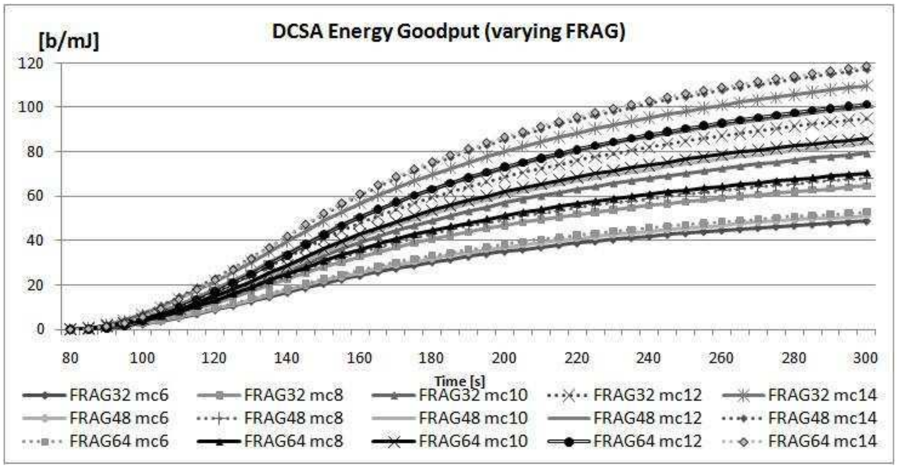


Fig. 5.20. Energy Goodput varying fragment length and no. of main connections (DCSA).

6

E-TDMA Control Phase: Models and Energy-Aware Strategies

6.1 Introduction

In this chapter we introduce two models of the control epoch that come before the information epoch in E-TDMA protocol.

Control epoch (see the Chapter 2.2.3) is composed by

- a *contention* phase, in which nodes run *Five Phase Reservation Protocol* (FPRP), and
- an *allocation* phase, in which nodes update their schedule and send it to their neighbors; we call this phase *Schedule Update*, SU.

We will derive the best and the worst costs of FPRP and of SU protocols, from energetic or communication point of view: usually they are not coincident, rather they follow opposite basic concepts, that we will explain in detail.

After the description of the two energy consumptions models, we will present the energy-aware strategy proposed to improve further energy behaviour of E-TDMA and that has effects on control or information epoch consumptions, depending on being or not in E-TDMA setup-phase.

6.2 FPRP: Energetic Modeling

Before beginning to describe our study about FPRP phase, let us define some symbols we are going to use forward:

- P_T = power spent when a node is in transmission state
- T_{Tb} = time to transmit one-bit packet
- P_R = power spent when a node is in reception state
- T_{Rb} = time to receive one-bit packet.

An FPRP packet is composed by only one bit so we can write that:

- E_{T-FPRP} = energy spent to transmit one FPRP packet = $P_T T_{Tb}$
- E_{R-FPRP} = energy spent to receive one FPRP packet = $P_R T_{Rb}$
- T_{s-FPRP}= duration of an FPRP slot
- T_{FPRP} = duration of an FPRP cycle
- c_{FPRP} = number of FPRP cycles.

To better quantify the cost of an FPRP slot we have to consider the consumptions in idle mode. We have to make a distinction:

- When a node is transmitting it cannot receive any other packet. So the total energetic cost of an FPRP transmission slot is

$$E_{TX-FPRP} = E_{T-FPRP} + P_{idle}(T_{s-FPRP} - T_{Tb}) \tag{6.1}$$

- When a node is in reception mode it could receive a packet in any instant of a slot (for the propagation delay). If a node receives more than one packet during a single slot, protocol detects a collision. So there is a little difference in the energy spent for receiving an FPRP packet with or without a collision. When there could be potentially a collision, we have to consider the worst case in which the node could potentially receive the FPRP packet in each time of a slot. We will have:

$$E_{RXcoll-FPRP} = P_R T_{s-FPRP} \tag{6.2}$$

$$E_{RX-FPRP} = E_{R-FPRP} + P_{idle}(T_{s-FPRP} - T_{Rb}) \tag{6.3}$$

The cost of a slot in idle listening slot is equal to:

$$E_{IDLE-FPRP} = P_{idle} T_{s-FPRP} \tag{6.4}$$

Two are the possible roles for a node in an FPRP slot (Fig. 6.1):

- RN = if it has data to send and tries to contend for acquiring an FPRP slot/color;
- iH = if it has not data to send or doesn't contend in that slot, and there is a RN node i-hop away from it,

with only one RN, and

- RN_i = i-th node RN
- iH_{xy} = i-hop neighbor from RN_x and RN_y .

when there are more than one RN. As we said in 2.2.2, FPRP phases are the followings:

- RR = Reservation Request
- CR = Collision Report
- RC =Reservation Confirmation
- RA =Reservation Acknowledgement

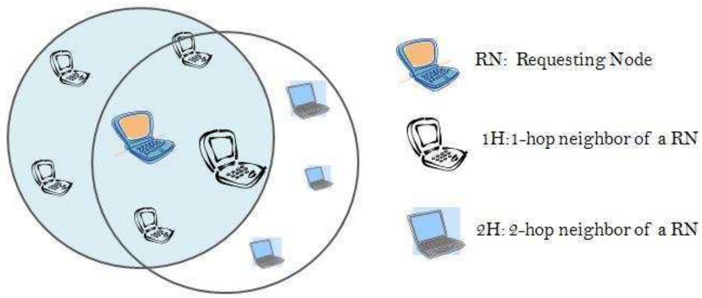


Fig. 6.1. A portion of a network with one RN and some 1H and 2H nodes.

- PP= Packing Packet
- PE = Packet Elimination

The resulting states (Res) in which a node could be at the end of an FPRP cycle are the following ([]):

- I= idle
- T = transmitting
- R = receiving
- B = blocked

Besides, p_{EP} is the probability that a RN sends the EP packet. Depending on traffic and connectivity conditions, and then on the actions made in the several FPRP slots, that are:

- tx: transmission of an FPRP packet
- rx: reception of an FPRP packet
- rx-coll: reception with collision of an FPRP packet
- tx(p_{EP}) or rx(p_{EP}) transmission or reception of an FPRP packet depending on the probability p_{EP}.

RNs or iHs will register different energy consumptions. We are going now to consider poxible contending cases that could happen running FPRP.

FPRP Contention Cases for nodes

We will have the following cases:

Case A: no contending nodes

Nobody begins to contend. Energy spent for an FPRP cycle will be equal to:

$$E_{FPRP} = 5E_{idle} = T_{FPRP}P_{idle} \quad (6.5)$$

Case B: only 1 contending node

Two scenarios are possible in this case:

1. *Failure*: a node tries to contend for a color (to became TN) but it has no neighbor. In table 6.1 we show the actions made by the only present node, that is RN. At the end of the cycle it consider itself as Idle.

	RN
RR	tx
CR	
RC	tx
RA	
PP	
PE	
Res	I

Table 6.1. Case B1: Failure with only one contending node (isolated RN).

2. *Success*: a non-isolated RN tries to acquire a color and is the only RN in a range of 2 hops: he wins the contention (table 6.2).

	RN	$1H$	$2H$	$3H$
RR	tx	rx		
CR				
RC	tx	rx		
RA	rx	tx	rx-coll o rx	
PP		rx-coll o rx	tx	rx-coll o rx
PE	tx(p_{EP}) o rx	rx(p_{EP})		
Res	**T**	**R**	**B**	**I**

Table 6.2. Case B2: success with only one contending node.

Case C: two or more contending nodes

1. *Failure*: there are one or more one-hop neighbor in the 2-hops range of more than one RN that contend for the same color: FPRP packets collide at 1-hop receivers ($1H_{12}$) and contention is lost by each RN (table 6.3).
2. *Success*: more than 1 RN contend within 2-hops range but they don't have either one common neighbor, so all RN could win the contention, depending on the probability of sending a PE packet.

	RN_1	RN_2	$1H_{12}$	$1H_1$,$1H_2$	$2H_1$,$2H_2$
RR	tx	tx	rx-coll	rx	
CR	rx o rx-coll	rx o rx-coll	tx	rx o rx-coll	
RC					
RA					
PP					
PE					
Res	**I**	**I**	**I**	**I**	**I**

Table 6.3. Case C1: Failure with two or more contending nodes.

	RN_1	RN_2	$1H_1$,$1H_2$	$2H_1$,$2H_2$,$2H_{12}$	**3H**
RR	tx	tx	rx		
CR					
RC	tx	tx	rx		
RA	rx	rx	tx	rx o rx-coll	
PP			rx o rx-coll	tx	rx o rx-coll
PE	tx(p_{EP}) o rx	tx(p_{EP}) o rx	rx(p_{EP})		
Res	**T or R**	**T or R**	**R**	**B**	**I**

Table 6.4. Case C2: success with two or more contending nodes.

6.2.1 Energy consumptions of an FPRP slot

Now we will conduct a simple energetic study about FPRP phases merging the states described in the precedent paragraph. We have to make the following distinctions:

- FPRP point of view
  - BC_{FPRP}: the best case from FPRP point of view is the case B2 in which only a node in a 2-hops range tries to contend, then it wins the contention and it has not to contend anymore in the following FPRP cycles to acquire that temporary color. In this case at the end of the FPRP cycle the RN is the only transmitter, one-hop nodes (1H) are receivers of only that node and two-hop nodes (2H) are blocked. This is also the best case from network utilization point of view because other nodes that are not involved in RN transmission (namely not in a range of two hops from RN) could try to contend for the same temporary color, in the same FPRP frame.
  - WC_{FPRP}: the worst case from FPRP point of view is the case C1, in which two or more nodes try to contend, their requests collide and nobody wins the contention, but each node has to retry to contend. No transmission will be allowed in this case.
- Energy point of view: the best case (BEC_{FPRP}) is the cheapest (when nobody transmits or receives) and the worst case (WEC_{FPRP}) is the most

expensive, energetically speaking; so it depends on the energy consumptions of every node based on its own role, and particularly on how many times a node transmits, receives, detects a collision or stays in idle. The functioning states of a node that we will consider are the following: RN, 1H and 2H. For each of them we have:

$$BEC_{FPRP} = 5E_{IDLE} \tag{6.6}$$

that is related to the cost of the case A.

6.2.2 FPRP roles

Each FPRP role has a particular energetic cost.

Requesting Node

In table 6.5 we show all the possible contention cases for RN.

	Table						
	6.1	6.2	6.3	6.4		**Merge**	WEC_{FPRP}
RR	tx	tx	tx	tx		tx or idle	tx
CR		rx o rx-coll				idle o rx o rx-coll	rx-coll
RC	tx	tx		tx	⇒	tx o idle	tx
RA		rx o rx-coll		rx o rx-coll		idle o rx	rx-coll
PP						idle	
PE		tx(p_{EP}) o rx		tx(p_{EP}) o rx		tx o idle o rx o rx-coll	tx/rx-coll
Res	I	T	I	T			
Cost		BC_{FPRP}	WC_{FPRP}	*			

Table 6.5. RN energy consumptions.

From FPRP point of view, we will have the following energetic costs for the RN role:

$$E_{BC-FPRP-RN} = 3E_{TX} + E_{RX-coll} + E_{IDLE} \tag{6.7}$$

$$E_{WC-FPRP-RN} = E_{TX} + E_{RX-coll} + 3E_{IDLE} \tag{6.8}$$

We can obtain WEC merging the four results and considering the most expensive cost, we will have that for a RN the consumptions of FPRP are the following:

$$\begin{cases} E_{WEC-FPRP-RN} = 3E_{TX} + 2E_{RX-coll} \text{ if } E_{TX} > E_{RX-coll} \\ E_{WEC-FPRP-RN} = 2E_{TX} + 3E_{RX-coll} \text{ if } E_{TX} < E_{RX-coll} \end{cases} \tag{6.9}$$

Note that the *Cost* of case described in table 6.4 and reported in the table 6.5 is not BC_{FPRP} because in the subsequent cycle that node could lose the contention, because of the nearby RN.

1-Hop away from the Requesting Node

For nodes that are one hop away from the contending node (1H), we will have:

	Table		
	6.1	6.2	6.3
RR	rx	rx-coll o rx	rx
CR		rx o tx	
RC	rx		rx
RA	tx		tx
PP	rx o rx-coll		rx o rx-coll
PE	rx(p_{EP}) o rx-coll		rx(p_{EP}) o rx-coll
Res	R	I	R
Cost	BC_{FPRP}	WC_{FPRP}	

$\Rightarrow$

Merge	WEC_{FPRP}
idle o rx o rx-coll	rx-coll
tx o idle o rx o rx-coll	tx/rx-coll
rx o idle	rx
tx o idle	tx
idle o rx o rx-coll	rx-coll

Table 6.6. 1H energy consumptions.

We have the following result for energy consumptions, related to BC and WC:

$$E_{BC-FPRP-1H} = E_{TX} + 2E_{RX} + E_{RX-coll} + E_{IDLE} \tag{6.10}$$

$$E_{WC-FPRP-1H} = E_{RX} + E_{RX-coll} + 3E_{IDLE} \tag{6.11}$$

Considering, finally, the merging of the three cases, and the most expensive state of functioning, we will have that, for a 1-hop neighbor of one or more RNs, the energy consumptions of FPRP are the following:

$$\begin{cases} E_{WEC-FPRP-1H} = 2E_{TX} + 2E_{RX-coll} + E_{RX} & \text{if } E_{TX} > E_{RX-coll} \\ E_{WEC-FPRP-1H} = E_{TX} + 3E_{RX-coll} + E_{RX} & \text{if } E_{TX} < E_{RX-coll} \end{cases} \tag{6.12}$$

2-Hops away from the Requesting Node

Nodes that are two hops away from the RN, register energetic consumptions that we show in table 6.7:

For a 2H node, BC and WC show the following energy consumptions:

$$E_{BC-FPRP-2H} = E_{TX} + E_{RX-coll} + 3E_{IDLE} \tag{6.13}$$

$$E_{WC-FPRP-2H} = E_{BEC-FPRP} \tag{6.14}$$

The case A (Table 6.1) and C1 (Table 6.3) are BEC for 2H. The worst energetic consumptions related to two-hop neighbors of one or more RNs could be expressed as:

	Table		
	6.1	6.2	6.3
RR			
CR			
RC			
RA		rx o rx-coll	rx o rx-coll
PP/PE		tx	tx
Res	I	B	B
Cost	WC_{FPRP} e BEC	BC_{FPRP}	

$\Rightarrow$

Merge	WEC_{FPRP}
idle	idle
idle	idle
idle	idle
rx o rx-coll	rx-coll
idle o tx	tx

Table 6.7. 2H energy consumptions.

$$E_{WEC-FPRP-2H} = E_{TX} + E_{RX-coll} + 3E_{IDLE} \tag{6.15}$$

We saw that energy and communication points of view are quite different in FPRP context. Each of this cost could be used in a node to decide if it has to take or not part to a new contention for the request of more bandwidth for one or more of its connections, or possibly became a 1H or 2H node of someone else, depending on its residual battery-life. A possible evaluation of the total probabilistic cost of a FPRP cycle for a single node could be got combining all of the previous costs:

$$\begin{aligned} E_{WEC-FPRP} \approx\; & P_{RN}E_{WEC-FPRP-RN} \\ & + P_{1H}E_{WEC-FPRP-1H} + P_{2H}E_{WEC-FPRP-2H} \end{aligned} \tag{6.16}$$

where:

- P_{RN} = probability to be a RN
- P_{1H} = probability to be a 1-hop neighbor of a RN
- P_{2H} = probability to be a 2-hop neighbor of a RN

Each of these probabilities depends on the values of:

- P_{send} = probability to have data to be transmitted.
- P_C = probability to contend for a slot, that is the contention probability and is equal to: $1/n_c$

where n_c is the number of contending nodes for a target FPRP cycle/color, and for the RN we have:

$$P_{RN} = P_C P_{send} \tag{6.17}$$

This last cost could be used to overestimate FPRP energy consumptions and design opportunely an energy-aware strategy, energetically improving the FPRP pseudo-bayesian algorithm.

6.3 SU Phase Energetic Modeling

The analysis of Schedule Update (SU) phase of E-TDMA protocol is very simpler than FPRP, because is deterministic, and, thanks to the FPRP previous contention, is also collision-free with a high probability. So collisions happen usually only when node mobility is very high related to super-frame duration.

Each node, to send a packet with its own schedule during SU phase, must have acquired a slot/color during one of the previous FPRP phase, in the same super-frame, or in one of the precedent. At most a node could have at most one permanent color and one temporary color in control schedule.

We will use the following symbols for our study:

- B_{SU} = number of bits of a SU packet
- T_{s_SU} = duration of a SU slot
- T_{Tsu} = time to send a SU packet
- P_T = power needed to transmit a bit (for a particular range)
- P_R = power needed to receive a bit
- E_{Tsu} = energy spent to send a SU packet = $T_{Tsu}P_T$
- E_{TXsu} = energy spent in a slot SU when the node sends its packet = $E_{Tsu} + P_{IDLE}(T_{SU} - T_{Tsu})$
- E_{Rsu} = energy spent to receive a SU packet = $T_{Rsu}P_R$
- E_{RXsu} = energy spent in a slot SU when the node receives SU packets = $E_{Rsu} + P_{IDLE}(T_{SU}T_{Rsu})$
- TC = number of temporary colors
- PC = number of permanent colors
- n_{PC_SU} = number of SU slots related to PC colors = $TCPC$
- n_{TC_SU} = number of SU slots related to TC colors = $PC(PC+1)/2$ (*Gauss number*)
- n_{SU} = number of SU slots = $n_{PC_SU} + n_{TC_SU}$
- nc_{SU} = number of SU cycles = TC
- n_{TXsu} = number of slots in which a node will send a SU packet
- n_{RXsu} = number of slots in which a node will receive a SU packet =
- E_{SU}= total energy spent in SU phase

As for FPRP, also for SU we have to distinguish between communication and energy point of view:

- SU point of view
  - BC_{SU} happens when each node has its own permanent color to send own transmission schedule, and all of the temporary color are occupied by nodes who need to reserve new information slots;
  - WC_{SU} happens when not even a SU slot is occupied from any node.
- Energy point of view: as for FPRP, the best and the worst energetic (BEC and WEC respectively) case are the cheapest or the most expensive respectively. Further we will explain those in detail.

The BEC for a node is when it and nobody else in its range has a color, then nobody sends any packet. In this case we will have:

$$E_{BEC-SU} = n_{SU} P_{IDLE} T_{SU} \tag{6.18}$$

This value corresponds to the cost of SU phase when the network is in setup phase, when there are no transmissions yet, or at the end of the simulation and there are no more transmissions. We can easily note that this cost correspond to the WC for SU, so we will have:

$$E_{WC-SU} = E_{BEC-SU} = n_{SU} P_{IDLE} T_{SU} \tag{6.19}$$

The WEC happens when there are more than one temporary color and the target node has got the last one (this one is assigned just at the end of the SU phase), and all of the temporary and permanent colors are assigned to its neighbor nodes. In every SU cycle a node has to send a SU packet during its permanent color, and another one during its temporary color, while in the other slots he has to receive SU packets from neighbors. We will have:

$$N_{TXsu} = 2n_{cSU} \tag{6.20}$$

$$n_{RXsu} = n_{SU} - n_{TXsu} \tag{6.21}$$

$$E_{WEC-SU} = E_{TXsu} n_{TXsu} + E_{RXsu} n_{RXsu} \tag{6.22}$$

Temporary colors are used more when there are new requests of bandwidth, then, we can imagine this is the case in which we are at the beginning of the transmissions or when permanent color are all assigned already but more information slots are needed from nodes. Permanent colors are used, instead, to confirm the possession of one or more colors (temporary or permanent) and of information slots. So, an average case could be the situation in which there are transmissions and receptions only during the slots corresponding to the permanent colors. We will have:

$$n_{TXsu} = n_{cSU} \tag{6.23}$$

$$n_{RXsu} = n_{SU} - n_{TXsu} - n_{TC_SU} \tag{6.24}$$

$$E_{AVG-SU} = E_{TXsu} n_{TXsu} + E_{RXsu} n_{RXsu} + P_{iDLE} n_{TC_SU} T_{sSU} \tag{6.25}$$

At this point is easy to derive the cost of BC of SU, consindering in 6.25 the transmissions during temporary colors:

$$E_{BC-SU} = E_{TXsu}(n_{TXsu} + n_{TC_SU}) + E_{RXsu} n_{RXsu} \tag{6.26}$$

6.4 Energy-Aware Strategy

The energy-aware strategy described in the Chapter 5 gave huge improvements in energy consumptions, but, as we saw in the paragraph 5.4.3, Control Energy (Fig. 5.18) consumptions sometimes are above the 25% of the whole spent energy during simulation. In the two following tables (6.8 and 6.9) we plot in details Control Energy consumptions of E-TDMA in the two scenarios presented in the Chapter 3, 64-20 and 512-4, in which we emphatized setup-phase consumptions.

Energy type	$t = 0..80s$ [J]	$t = 80..300s$ [J]					**% of total**
	setup phase	mc=6	mc = 8	mc = 10	mc=12	mc=14	
E_{FPRP}	0.47	1.19	1.21	1.22	1.23	1.27	27–28%
E_{SU}	1.11	2.59	2.60	2.61	2.61	2.63	30%
E_{CTRL}	1.58	3.78	3.81	3.84	3.85	3.89	29%

Table 6.8. Energy consumptions of ETDMA in the setup phase, and equivalent percentage over the consumptions of the whole simulation.

Energy type	$t = 0..80s$ [J]	$t = 80..300s$ [J]					**% of total**
	setup phase	mc=6	mc = 8	mc = 10	mc=12	mc=14	
E_{FPRP}	0.48	3.11	3.12	3.13	3.13	3.14	13.32%
E_{SU}	1.13	3.11	3.12	3.13	3.13	3.14	26.57%
E_{CTRL}	1.61	4.58	4.62	4.65	4.66	4.68	25.77%

Table 6.9. Energy consumptions of ETDMA in the setup phase, and equivalent percentage over the consumptions of the whole simulation.

As many times we have said, data transmissions start in an instant of time randomly chosen between the 80^{th} and 150^{th} second of simulation. When nodes run E-TDMA, instead, they need to reserve a broadcast slot, for routing control packets even if there are no data transmissions, to prevent unbearable delays in case of arrival of any data packet [Zhu and Corson, 2001a].

The energy consumption of setup phase is almost constant for every number of flows: it only depends on the number of the nodes in the network and on their mutual disposition. The major energy consumptions are during SU phase, because control packets have to be sent after having acquired the broadcast slot. But we cannot either ignore FPRP costs if we think that is related only to the exchange of one bit per phase.

The energy consumption of the setup phase for E-TDMA is equal to less than one third of the total control energy consumed during the whole simulation. Then, the initial acquisition of a broadcast slot, without any data transmission, is not so good from energy point of view, and could be justified only for high traffic.

All these results led us to design a simple new energy-aware strategy to reduce control energy consumption. It consists in suppressing some phases when they are not needed without invalidating the functioning of the protocol. In particular our modification is composed by two parts, the former related to setup-phase period, and the latter related to the rest of the simulation.

During setup phase E-TDMA cares only about the reservation of broadcast slots: all the information epochs are energy-wasting, especially considering that protocol logic requires that each node is active at least for the broadcast slot it has reserved, and for all the other broadcast slots that its neighbor reserved, even if it has not to transmit or receive nothing, till the end of setup-phase. But the highest consumptions come from the rest of simulation, especially from SU phase. SU phase consumptions are not so cautious and sometimes unnecessary, in a context in which topology or traffic conditions change are not so frequent. So we proposed two policies to apply in implementing our modification:

- to switch-off network interfaces when to remain idle would be uselessly energy-wasting because not needed;
- to suppress unnecessary phases.

Both are equivalent from protocol point of view, because we inhibit some phases, but the second is better from energy point of view because eliminates all costs that are due to energy state transitions, and further it can reduce the duration of the setup phase in terms of time, even if there is the same amount of control data exchanged. The implementation of our modification consisted in acting:

- on the information epochs during all the setup-phase;
- on the control epochs during the rest of simulation

Inhibiting all the information epochs does not invalidate E-TDMA functioning, so we can inhibit all of them, while we cannot do the same with control epochs, while during the simulation both FPRP and SU are still important to establish new connections, to require and acquire more bandwidth for new flows and to assure collision-free transmissions as more as possible. But we could think to a periodic inhibition, with a periodicity that depends on mobility, traffic conditions, and other network parameters, without degrading communication performances but getting sometimes very high energy gains.

The modification was implemented in ns-2 over E-TDMA in DCSA mode: we will evaluate improvements respect to DCSA results. In the next paragraph we will show simulation results.

6.5 Experimental Results

Simulation, E-TDMA and energetic parameters are described in tables 6.10, 6.11 and 6.12, respectively. Traffic pattern and all other scenarios parameters follow the same rules described already in the paragraph 5.3 to which we remand to better understand the analysis.

Transmission Range	$250m$
Length of MAC Queue	$50pkts$
Routing Protocol	AODV
Propagation Model	Disk
Number of nodes	20
Duration of simulation	$300sec$
System Bandwidth	$2Mbit$
Simulation Area	$700x700m^2$
Velocity	$v = 0m/sec$
Type of flows	*Constant Bit Rate*
Generation rate of flows	$20pkt/sec$
Number of flows	$6 : 4 : 14$
Frequency of Inhibition	$0; 50; 100; 200$
Packet size	$64bytes$
Fragment Length	$32bytes$
Packets per flow	$10,000pkts$
Starting time for flows	$80^{th} - 150^{th}sec$
Min No. of Slots to Sleep(n)	2

Table 6.10. Simulation Parameters.

Contention Phase	Cycles FPRP	c	8
	Number of Temporary Colors	tc	1
	Phases per cycle	f	5
	Total duration of FPRP	T_{FPRP}	$1.04ms$
Allocation Phase	Number of Permanent Colors	pc	15
	Total duration of SU	T_{SU}	$2.336ms$
Information Epoch	Number of INFO slot	is	15
	Number of INFO epoch	if	4
	INFO slot duration	T_S	$17.88ms$
Total duration of E-TDMA		T_{ETDMA}	$21.256ms$

Table 6.11. E-TDMA Parameters.

The metric we have considered are:

- *Packet Delivery Ratio*;

Initial energy	E_{ini}	$500J$
Transmission power	P_{tx}	$1.6mW$
Reception power	P_{rx}	$1.4mW$
Idle power	P_{idle}	$1.3mW$
Sleep power	P_{sleep}	$0.08mW$
Transition duration	T_{tr}	$250\mu s$
Transition power	P_{tr}	$0.2mW$

Table 6.12. Energetic Parameters.

- *Average Network Energy Evolution*;
- *Average Energy Consumptions*, and in particular:
  - *Idle Energy*
  - *Residual Energy*
  - *Information-frame Energy* and
  - *Reception Energy*
- *Energy Goodput*

All of these were evaluated for a varying number of connections, from 6 to 14 with a step pof 4, and for a varying frequency of inhibition (f in the following figures): 0 (that is the case in which energy-aware strategy was not implemented), 50, 100 and 200, that is every f cycles we inhibited one control epoch.

6.5.1 Data Packet Delivery Ratio

We refer to the Fig. 6.2. Between the 80^th and the 100^th second we can note an oscillation among the results, because flows start. After the 100^th second the differences among curves become nearly imperceptible: only for 14-connections curve values are a bit less of 100% of data delivery, so our modification does not have consistent implications on communication performances. Also the amount of exchanged data is the same.

6.5.2 Energy Evolution

In Fig. 6.3 we compare the evolution of the average residual energy of nodes. We can clearly note the improvements of applied energy-aware strategies, both in setup-phase (till the 80^{th} second of simulation) for which consumptions are less and less, and in the rest of simulations. As we saw in the previous chapter, for about all the connections energy consumptions are quite similar, so we can note two groups of curves, one related to simple DCSA (with $f = 0$) and the other related to the other frequencies, as we can also note in detailed Fig.6.4.

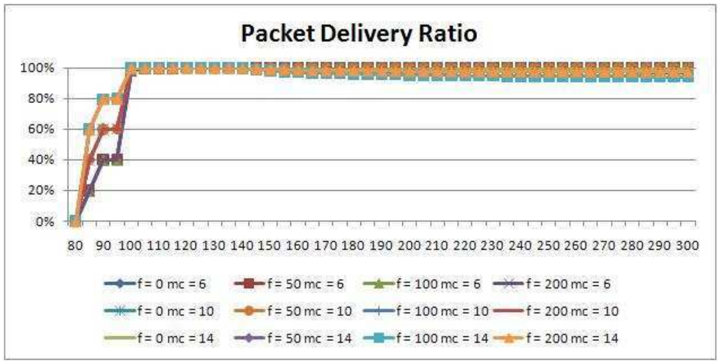


Fig. 6.2. Delivery ratio (%).

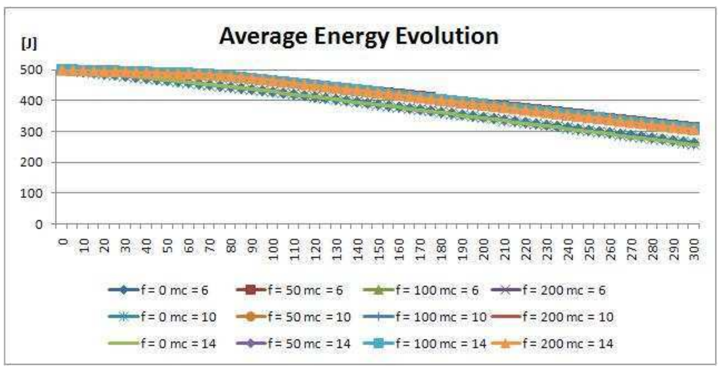


Fig. 6.3. Energy Evolution [J].

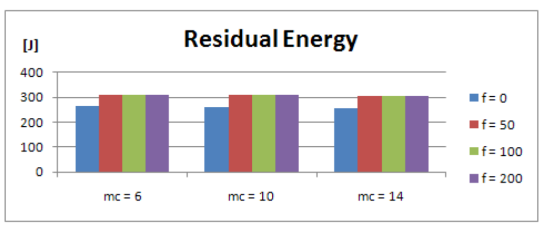


Fig. 6.4. Residual Energy [J].

6.5.3 Energy Goodput

We have already seen the improvements caused from the introduction of our modification. So we expect good performance also in EG (Fig. 6.5). EG grows with the number of flows and we can clearly see that modification outperforms the simple DCSA, and improvemetn is bigger and bigger as mc grows. That is to say DCSA-f50 (or f100 or f200) logic consumes less energy to transmit the same quantity of data, or exchanges more data than DCSA simple with the same quantity of energy.

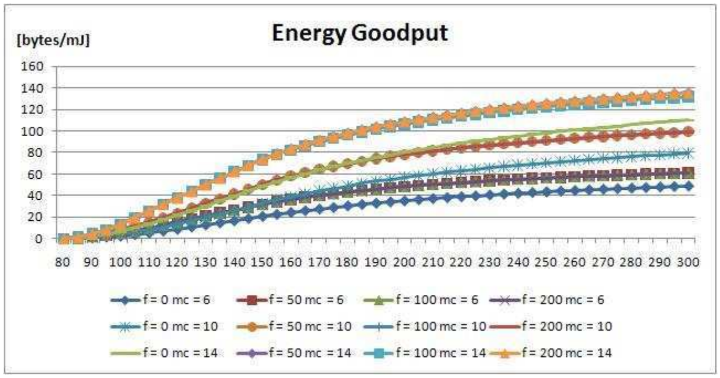


Fig. 6.5. EnergyGoodput [bytes/mJ].

6.5.4 Average Energy Consumptions

In Figg. 6.6, 6.7 and 6.8 we show detailed results about energetic consumptions during reception, and idle states and information phase of E-TDMA. There are a bit difference among curves related to DCSA with inhibition frequency, but, for the same amount of exchanged data, we can see that the best result are given by DCSA-f50, that is the case in which we inhibit more cycles. Simple DCSA give always the poorest results.

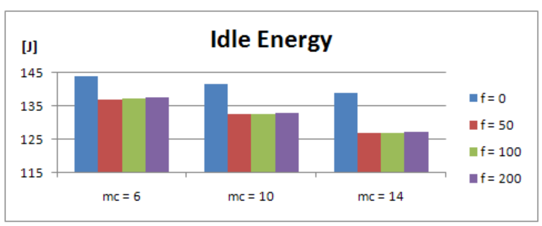


Fig. 6.6. Idle Energy [J].

6.6 Conclusion

The aim of this chapter was to study in detail Control Epoch of E-TDMA, subdividing it in two parts respectively related to FPRP and SU phases.

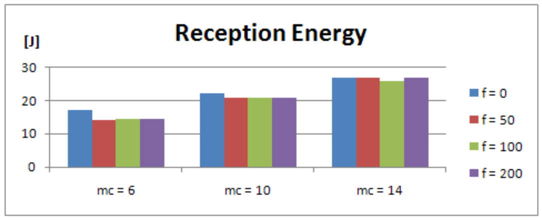


Fig. 6.7. Reception Energy [J].

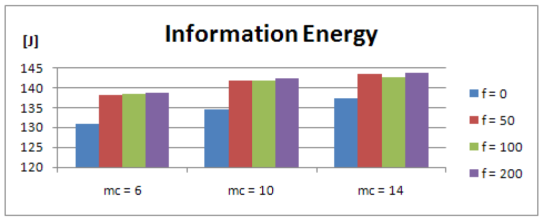


Fig. 6.8. Information Energy [J].

Analysis of them provided two models that could be used to deploy energy-aware strategies in E-TDMA reducing its energy consumptions. It was so proposed a new energy-aware strategy for the E-TDMA control phase, that were implemented and tested, comparing it with the original DCSA mode. The fusion of the two energy-aware strategies introduced in this and in the previous chapter led us to the proposal of a new energy-aware TDMA MAC protocol for MANETs, that will be presented in a future work.

Conclusions

In this thesis the energy issue over Mobile Ad Hoc NETworks (MANETs) was addressed.Wireless nodes have limited battery-resources and a bad energy management could inhibit their life and, then, the life of their own network. Energy issue is a very relevant research topic and a lot of efforts have been made in these recent years to propose efficient and effective solutions to it, at different levels of ISO/OSI protocol stack. This thesis focused mainly on the MAC level, planning to provide an accurate analysis of some well-known protocols from energetic point of view and then to design and propose techniques to improve their energetic performances guaranteeing an efficient management of their energetic resources.

Literature subdivides MAC protocols mainly into two typologies: *contention-based* and *collision-free* protocols. They are very different from each other not only from communication point of view (channel access, bandwidth usage, and so on), but also from energy point of view when they do not provide some strategy to prevent implicit huge consumptions. In literature three are the typologies of strategy proposed to reduce energy consumptions: the well known *energy-saving* and *power-control* techniques and the more challenging *energy-aware* techniques. Usually approaches implemented in protocols vary depending on its MAC typology. Some other protocols, instead, called *energy-efficient*, integrate into themselves energy strategies.

In this context this thesis provides several contributions.

The first consists in the energetic comparison of two protocols, each belonging to a different MAC protocol family: IEEE 802.11, that is the most known and used both in industry and in every day life, among contention-based, and E-TDMA (*Evolutionary Time Division Multiple Access*), among collision-free protocols. Simulations were conducted over ns-2 simulator.
Several metrics were investigated in the comparison, not only related to energy consumptions: the first considered was always the communication performance to validate the goodness of MAC protocols in their main aim, that is the correct exchange of data. Communication performance was evaluated through the *Data Packet Delivery Ratio*, that is the percentage of the number

of received data packets to the number of sent data packets. The metrics that we used to study in particular the energetic behaviour of the two protocols are the following: the *Average Network Energy Evolution*, that is the evolution of the average residual energy of the nodes and the *Average Energy Consumptions* in which we encounter detailed consumptions related to some certain states (transmission, reception, idle or sleep) and phases (data or control); but to better evaluate the energy efficiency of protocols, we used the *Energy Goodput* metric, that correlates throughput to residual energy consumptions, and gives an index of how many data could be correctly delivered in a unit of energy.

The very good performances of E-TDMA respect to IEEE 802.11 in terms of energy consumptions led us to select it and study in details its energetic behaviour depending on the values of its basic parameters: we considered varying values for *information slot*, *information frames* and *fragment length*. We noted that the fact of increasing the bandwidth of E-TDMA and, then, the values of those parameters, cause a more balanced energy consumptions in control epochs respect to information epochs, as well as give very good results in energetic performances, also for high traffic scenarios.

All the other contributions are related to the novel energy-aware techniques that we proposed to enhance the energetic behaviour of E-TDMA. We divided analysis into two parts.

The first one was related to the E-TDMA information epochs, and is, for this reason, extendible to all TDMA collision-free protocols, in which there is the same frame format as E-TDMA. We applied energy-saving strategies and sleep procedures to information epochs and provided a simple analysis to model them. The proposed energy-aware modification involved the intrinsic logic of the traditional TDMA systems, that is their slot assignment strategy, trying to reduce the transitions between active and sleep states. We called traditional TDMA systems DASA (*Dynamic Alternated Slot Assignment*), opposed to DCSA (*Dynamic Contiguous Slot Assignment*) that is the new proposed slot assignment. We compared DASA vs. DCSA through several simulation sets: DCSA's performances overcome DASA's in terms of energy consumptions, confirming results provided by the theoretical model, and without wasting communication performances that are almost similar, as we expected.

Reducing energy costs related to data exchanges, made even more huge energetic control spent energy. So the second analysis involved the E-TDMA control epochs. They are composed by two different phases: FPRP (*Five Phase Reservation Protocol*) phase, during which the possible transmitter nodes contend to acquire new information slots, and allocation phase (*Schedule Update* or *SU* phase). After a simple mathematical modelling of these two phases, we proposed an *inhibiting* logic with the aim of reducing control spent energy. Finally E-TDMA was modified to include DCSA and the novel *inhibiting* energy-aware strategy was compared with the original E-TDMA DCSA.

Simulation sets showed the high energetic improvements of the *enhanced* E-TDMA, without degrading its communication performances.

This work might be further extended along several directions. First of all the new energy-aware strategies presented for E-TDMA control and information epochs might include other simulation parameters or be implemented in other TDMA schemes. Second, the simple theoretical analysis could be the basis for the design of more efficient energy-aware strategies related to the actual energy consumptions or the energy drain rate of nodes to dynamically change intrinsic protocol behaviour. Third, an energy efficient Routing protocol might be added to MAC energy-aware strategies, in a cross-layer approach, to improve further management of energetic resources of the network.

References

S. Coleri-Ergen and P. Varaiya. Pedamacs: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Trans. on Mobile Computing*, 2006.

A. El-Hoiydi and J.-D. Decotignie. Wisemac: An ultra low power mac protocol for multi-hop wireless sensor networks. In *Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)*, 2004.

M. Cesana F. Borgonovo, A. Capone and L. Fratta. Adhoc mac: a new mac architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services. *Wireless Networks*, 10(4):359–366, 2004.

A. Perrotta F. De Rango and S. Marano. Qos-croma: An on-demand time-slotted mac protocol with qos support for wireless ad hoc networks. In *IEEE International Symposium on Wireless Communication Systems (ISWCS)*, 2007.

C. L. Fullmer and J. Garcia-Luna-Aceves. Floor acquisition multiple access (fama) for packet-radio networks. In *ACM SIGCOMM*, pages 262–273, 1995.

Michael Gunther and Karten M. Reineck. Leistungsanalyse des olsr-protokolls in drahtlos vermaschten netzen mit ns-2, 2008. URL `http://www.projektiv.net/karsten/Leistungsanalyse des OLSR Protokolls in drahtlos vermaschten Netzen mit s-2.pdf`.

G. Halkes and K. Langendoen. Crankshaft: An energy-efficient mac-protocol for dense wireless sensor networks. In *EWSN*, 2007.

IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard 802.11, June 1999.

Svilen Ivanov, Andr Herms, and Georg Lukas. Experimental validation of the ns-2 wireless model using simulation, emulation, and real network. In *In 4th Workshop on Mobile Ad-Hoc Networks (WMAN07*, pages 433–444, 2007.

P. Karn. Maca - a new channel access method for packet radio. In *ARRL/CRRL Amateur Radio Computer Networking Conference*, 1990.

Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. MANET simulation studies: the incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(4):50–61, October 2005. ISSN 1559-1662.

J. Lessmann. Gmac: A position-based energy-efficient qos tdma mac for ad hoc networks. In *IEEE International Conference on Networks (ICON)*, pages 449–454, 2007.

E. Anderson M. Buettner, G. Yee and R. Han. X-mac: A short preamble mac protocol for duty-cycledwireless networks. In *SenSys*, 2006.

C. Bonnet M. Coupechoux, B. Baynat and V. Kumar. Modeling of a slotted mac protocol for manets. In *International Workshop on Mobile Ad Hoc Networking and Computing (MADNET)*, 2003a.

C. Bonnet M. Coupechoux, B. Baynat and V. Kumar. Croma a slotted mac protocol for manets with multislot communications. In *WNCG Wireless Networking Symp.*, 2003b.

C. Bonnet M. Coupechoux, B. Baynat and V. Kumar. Croma an enhanced dynamic slot allocation protocol for manets. *MONET, Special Issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks*, 2004.

J. Ma, W. Lou, Y. Wu, X.-Y. Li, and G. Chen. Energy efficient tdma sleep scheduling in wireless sensor networks. In *IEEE INFOCOM*, 2009.

CMU Monarch. *The CMU Monarch Project's Wireless and Mobility Extensions to NS*, 1998.

Muqattash and M. Krunz. Cdma-based mac protocol for wireless ad hoc networks. In *MOBIHOC*, 2003.

The VINT Project. ns 2: Contributed code. http://nsnam.isi.edu/nsnam/index.php/contributed code, 2008.

S. Kim P. Yun, C. Kim. Elm-mac: Energy efficient link management by using optimal transmission power in mac protocol. In *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2008.

B. S. Manoj S. Jayashree and C. Siva Ram Murthy. On using battery state for medium access control in ad hoc wireless networks. In *MOBICOM*, 2004.

B.S. Manoj S. Jayashree and C. Siva Ram Murthy. Network lifetime driven mac protocols for ad hoc wireless networks. *WINET Journal*, 14(6), 2008.

B. Tavli and W. B. Heinzelman. Mh-trace: Multi-hop time reservation using adaptive control for energy efficiency. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 22(5), 2004.

ns: Change history The VINT Project. ns change log. http://www.isi.edu/nsnam/ns/changes.html, 2008.

F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part iithe hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Trans. Commun.*, 23:1417–1433, 1975.

F. De Rango V. Loscrì and S. Marano. Tuning the parameters of e-tdma mac for an efficient multipath-aodv over wireless ad hoc networks. In *Vehicular Technology Conference (VTC-Fall)*, pages 5001–5005, 2004.

L. van Hoesel and P. Havinga. A lightweight medium access protocol (lmac) for wireless sensor networks. In *INSS*, 2004.

Matt Spear X. Lu, K. Levitt, N. S. Matloff, and S. F. Wu. A synchronization attack and defense in energy-efficient listen-sleep slotted mac protocols. In *International Conference on Emerging Security Information, Systems and Technologies*, 2008.

Maotao Xie and Xiaoli Wang. An energy-efficient tdma protocol for clustered wireless sensor networks. In *ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM)*, 2002.

B. Yahya and J. Benothman. An energy efficient hybrid medium access control scheme for wireless sensor networks with quality of service guarantees. In *IEEE GLOBECOM conference*, 2008.

Hyojeong Shin Youngmin Kim and Hojung Cha. Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks. In *International Conference on Information Processing in Sensor Networks*, 2008.

C. Zhu and M. S. Corson. A five-phase reservation protocol (fprp) for mobile ad hoc networks. In *IEEE INFOCOM*, pages 322–331, 1998.

C. Zhu and M. S. Corson. An evolutionary-tdma scheduling protocol (etdma) for mobile ad hoc networks. Technical Report 17, 2001a.

C. Zhu and M. S. Corson. A five-phase reservation protocol (fprp) for mobile ad hoc networks. *Wireless Networks*, 7(4), 2001b.