On the interaction between multiple paths and Wireless Mesh Networks scheduler approaches

Valeria Loscrì
D.E.I.S. Department, University of Calabria
via P. Bucci, 42/c 87036 Rende, CS, Italy
e-mail: vloscri@deis.unical.it

Abstract— **Multi-path routing allows building and use of multiple paths for routing between a source-destination pair. This paper investigates the problem of selecting multiple routing paths to provide better reliability and load balancing in wireless mesh networks with stationary nodes. Previous work has investigated the use of additional data redundancy to improve the throughput of the network. In these specific cases, node disjoint-ness property of the multiple paths is required. In this work we investigate multipath routing without packet duplication, and no disjointed paths for achieving better performance in terms of packet delivery rate and low delay. We propose a very simple reactive on-demand distance vector routing protocol. Multiple paths built through this approach are loop-free. In order to better exploit resources redundancy (with the term resources redundancy we mean the possibility to exploit more nodes to send data packets), it is our belief that a routing protocol cannot be independent of the MAC layer. For this reason, we evaluated our routing protocol on four different MAC approaches specifically designed for Wireless Mesh Networks (WMNs). Firstly, we implemented the Coordinated Distributed Scheduler scheme of the Std. IEEE 802.16. Secondly, since some parameters have been left un-standardized in this scheme, we proposed an enhanced version of the CDS, in which a simple and dynamic criterion has been designed to set one of these parameters. Furthermore, we proposed two different scheduling schemes called Randomized-MAC (R-MAC) and Distributed Scheduling Scheme (DSS). We evaluated the impact of multiple paths in respect of the single path on all the scheduler schemes cited above. Results show as the simple routing approach is effective with every MAC protocol considered.**

Index Terms: WMNs, Multipath Routing;Coordinated Distributed Scheduling; R-MAC; Distributed Scheduling Scheme; MPRP;

I. INTRODUCTION

Wireless Mesh Networks (WMNs) have gained considerable attention as an alternative solution to applications such as community networks, enterprise networks, last mile access networks to the Internet, etc. [1, 2]. WMNs can be considered a special case of mobile Ad hoc networks where nodes have relative fixed positions and communicate to the Internet through one or more gateways [3]. In wireless mesh networks, traffic is primarily routed either towards the Internet Gateways (IGWs) or from the IGWs to the Access Points (APs). Thus, if multiple APs choose the best throughput path towards a gateway, the traffic loads on certain paths and mesh routers increase tremendously thereby deteriorating the overall performance of the network. Applications running in WMNs are predominantly Internet oriented and thus the traffic is either from the end users towards the Internet Gateway (IG) or vice-versa. In a WMN certain nodes or links can be heavily loaded while some nodes/links are seldom used. Due to the structure of WMNs, it is possible to exploit the redundancy offered in terms of multiple paths available from a certain node (e.g. Mesh Router, MR) to an IG or vice-versa. In this paper, we focus on the possibility to exploit multiple paths to improve the overall performance of the network. The novel Multipath Parallel Routing Protocol, MPRP [37], considered in this work is reactive and based on a distance vector routing approach. During the Discovery Phase, in which new paths are built, multiple paths between a pair of nodes, source-destination, will be discovered. The multiple paths are loop-free but not necessarely disjointed. Multiple paths are simultaneously used and a very simple criterion is used to split data traffic among the paths. In fact, we evenly distribute data traffic along the different paths. Although a simple routing approach is applied, we will show the effectiveness of multiple paths on different MAC protocols. Firstly, we implemented the Coordinated Distributed Scheduling (CDS) scheme of the Std. IEEE 802.16, designed for Wireless Mesh Networks. The CDS scheme is characterized with some limitations in some parts as we will see to the following. Indeed, some critical parameters have been left un-standardized and should be set from vendors or users. These parameters are critical because they affect the performance of the network. Another contribution of this paper is the definition of simple but effective criteria to set one of these parameters in a dynamic fashion. In practice, we propose an enhanced version of the CDS where the setting of some parameters is buffer queue based. This approach is called ProbApproach and we evaluated its performance when the MPRP is considered at the network layer. Moreover, two different scheduling schemes have been considered, the Randomized-MAC (R-MAC) [12] and the Distributed Scheduling Scheme (DSS) [13] and the impact of multiple paths of the MPRP have also been evaluated on both of these schemes. In practice, the main focus of this paper is to evaluating the impact of a simple multipath routing protocol on different MAC protocol designed for WMNs. Results show as the use of redundancy at the network layer allows better performance in terms of throughput and average end-to-end delay to be obtained in all the cases considered. The remainder of the paper is organized as follows. In Section II we give some details about the network model adopted. Section III describes the approaches developed at MAC layer. The new Multipath Parallel Routing Protocol (MPRP) is described in Section IV. Section V introduces the differences between some related works and our work. Extensive simulation results are sketched in Section VI. Finally, we conclude our paper in Section VII.

II. THE NETWORK MODEL

There have been many routing algorithms proposed for Mobile Ad hoc NETworks, MANETs, [4], [5], [6], [7], [8], [20]. However, the characteristics and requirements of WMNs are considerably different than those of general MANETs, such that the research community is focusing on new routing protocols that can significantly outperform the general MANET routing protocols. On the other hand, a fundamental aspect that has to be taken into account is the interaction between the routing protocol and the underlying MAC layer. Indeed, performance of the network derives from the synergic effect of both the MAC and network layers.

A. Some Assumptions

In this paper we consider a Wireless Mesh Network as composed of three distinct network elements (Figure 1):

- Network Gateway: one (or more gateway, Wired Internet Backbone) can be deployed to allow access to a different IP sub-network.
- Access Points (Mesh Routers (MRs) or Subscriber Stations (SSs)): the access points form a wireless backbone, providing connectivity in places otherwise difficult to access through traditional wired infrastructure. We assume the access points use IEEE 802.16 technology.
- Mobile Nodes (Mesh Clients or Wireless Clients): we consider as terminal user any device that can access the network gateway through direct or multi-hop communication (using the access points as relays). PDAs, laptops etc. can be considered available devices.

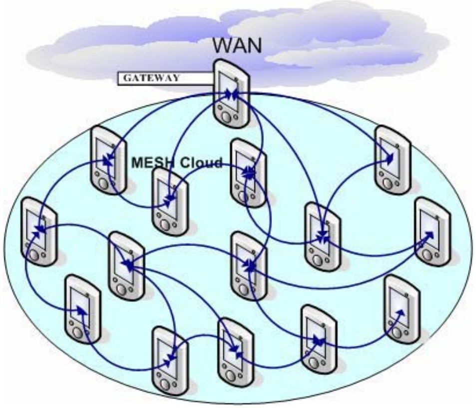

Figure 1. Wireless Mesh Network. The Gateway node provides Wide Area Network (WAN) connectivity to all other Mesh nodes.

We assume that Access Points are fixed and static in the network. Whereas each mobile node can possibly communicate with any other mobile node in the network, most expected traffic streams will occur between the mobile nodes and the network gateway, or Internet Gateway (IG) [18].

B. Node States description

In this paper we consider four different scheduling schemes, the CDS, an enhanced version of the CDS called ProbApproach, a distributed scheduling scheme called Randomized-MAC (R-MAC) and a Distributed Scheduling Scheme (DSS). Every scheme considered is based on a Time Division Multiple Access (TDMA) protocol at MAC (Medium Access Control) layer. With constraints required by conflict-free TDMA transmissions, the activity of a node n_i in a slot s can be classified into the following states:

- TX: Transmits to a set of neighbors R: (*state(s) = Transmit, target(s) = R*)
- RX: Receive from a neighbor n_j: (*state(s)=Recv, target(s) =* n_j)

If a node is not transmitting or receiving in this slot, it is in one of the following (passive) states:

- Blocked from transmitting because at least one of its neighbors receives from another node, and none of its neighbors transmits: *(state(s) Block_TX)*,
- Blocked from receiving because at least one neighbor is transmitting to another node, and none of its neighbors receives: (*state(s) = Block_RX*),
- Both Blocked from transmitting because at least one neighbor is receiving, and blocked from receiving because at least another neighbor is transmitting: (state(s) = *Block_TX_RX*),
- Experiencing a collision when it is supposed to receive from a neighbor (*state(s) = Collision*),
- Idle, when none of its neighbors transmits or receives in this slot: (*state(s) = Idle*).

Naturally, these states are mutually exclusive.

III. DISTRIBUTED SCHEDULER SCHEMES FOR WMNS

Firstly, in this Section we describe the IEEE 802.16 protocol for Wireless Metropolitan Networks that has been recently standardized to meet the needs of wireless broadband access [9, 10, 11], focusing on the Coordinated Distributed Scheduling scheme (CDS). Secondly, we suggest a simple criterion to set in a dynamic fashion some parameters of the Coordinated Distributed Scheduler scheme of the Std. IEEE 802.16 and we describe an enhanced version of the CDS. Thirdly, we describe two different scheduler schemes that represent extensions of the CDS scheme and allow better performance of the network to be obtained when single path is used as shown in [12, 13], respectively.

A. Coordinated Distributed Scheduling (CDS)

The Medium Access Control (MAC) layer of the IEEE 802.16 has point-to-multipoint (PMP) mode and mesh mode. In the mesh mode, all nodes are organized in an Ad hoc fashion and use a pseudo-random function to calculate their transmission time. Almost all the existing works about the IEEE 802.16 focus on the PMP mode [14, 15]. The TDMA frame is divided into the control-subframe and the data subframe (Figure 2). While the slots of the data-subframe are mainly used for the transmission of data packets, the control sub-frame is used only for the transmission of control messages. Subframes are fixed in length and consist of transmission opportunities (TOs). The number of transmission opportunities in the control-subframe is a network parameter (MSH_CTRL_LEN) and can have a value between 0 and 15. Every TO consist in 7 OFDM symbols time. The data-subframe is situated after the control sub-frame in a frame and is divided into minislots. The minislot is the basic unit for resource allocation. In the CDS mechanism all the stations shall indicate their own schedule by sending a MSH-DSCH (Mesh Distributed Scheduling message) regularly. MSH-DSCH messages are transmitted during the Schedule Control sub-frame.

1) Parameters for Distributed Scheduling

There are two parameters used in Distributed Mesh Networks for scheduling: *NextXmtMx* (NXM) and *XmtHoldoffExponent* (XHE). These two parameters are contained within MSH-DSCH messages. Since in Distributed Scheduling there is not a Mesh Base Station (M-BS) which schedules and controls the transmission of each node, it is necessary a distributed manner to schedule the transmissions. The concept is based on communicating all nodes when any node is going to transmit (MSH-DSCH messages including the information of the neighbors) thus every station has the knowledge of the scheduling of its two-hop neighborhood.

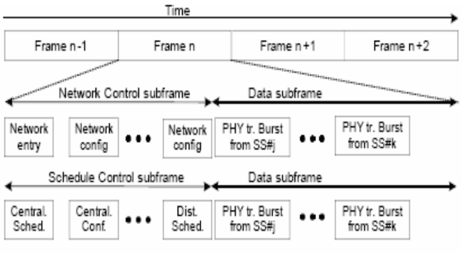

Figure 2. Frame structure in mesh mode.

In CDS the MSH-DSCH messages are scheduled in a conflict-free manner, there are not collisions. To the following we show how NXM and XHE are used to compute the XHT and the NXT values:

- XHE: in the standard *XmtHoldoffTime* (XHT) is the number of MSH-DSCH transmit opportunities after *NextXmtTime* (NXT) that this station is not eligible to transmit MSH-DSCH packets.

  $$XHT = 2^{(XHE+4)} \quad (1)$$

- NXM: in the standard is the next MSH-DSCH eligibility interval for this station

  $$2^{XHE} * NXM < NXT \le 2^{XHE} * (NXM+1) \quad (2)$$

For example, if NXM = 2 and XHE = 4 the station would be eligible between 33 and 48 transmission opportunities.

2) Distributed Election Algorithm

Every node calculates its NXT during the current transmission according to the distributed election algorithm defined in [16]. In this algorithm one node sets the first transmission slot just after the XHT as the temporary Next Transmission Opportunity (NXTO). In this instant, this node (let us to call this node as node A) shall compete with all the competing nodes in the two-hop neighborhood. There are different types of competing nodes (Figure 3) defined as follow:

- NXT includes the temporary transmission slot (Node B)
- EarliestSubsequenceXmtTime (ESXT, equal to NXT + XHT) is ≤ the temporary transmission slot (Node C)
- The Next Time is not known (Node D)

This algorithm is a pseudo-random function which uses the slot number and the Node's ID as the inputs and is executed at each node. It generates pseudo-random values depending on the input. The node wins when its result is the largest mixing value (Figure 4).

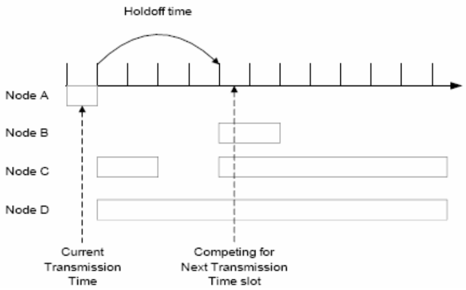

Figure 3. Competing Nodes for Next Transmission time slot.

When any node wins, it sets the temporary transmission opportunity as its next transmission time and logically it shall communicate this information to all the neighbors by sending the corresponding packet.

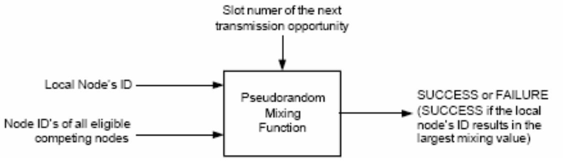

Figure 4. Pseudo-random Mixing Function..

In the case a node has not won, it chooses the Next Transmission Opportunity (NXTO) and repeats the algorithm as many times as it needs to win. The assignment of Transmission Opportunities (TOs) in the data-subframe is managed by a scheduling mechanism. The MSH-DSCH message of the distributed scheduling mechanism carries the Requests, Grants and Confirmations and all stations (Mesh Base Station, M-BS and Mesh Subscriber Stations, M-SSs) shall coordinate their transmissions in their two-hop neighborhood. MSH-DSCH messages are transmitted regularly by every node throughout the whole mesh network to distribute nodes schedules. As we already outlined, in this paper we focus on the CDS and analyze the transmission timing of the MSH-DSCH messages as this has much influence on the overall network performance. In order to evaluate the impact of the control schedule on the network performance, we developed a simplified data scheduler instead of considering a data scheduler based on the three-way-handshake mechanism. Since the simplified data scheduler implemented has been the same for every scheduling scheme considered in this work we reported its detailed description in the Appendix 1.

B. An enhanced version of the CDS: the ProbApproach

To avoid collisions of MSH-DSCH messages, each node must inform its neighbors about the next MSH-DSCH transmission time (or transmission opportunity). To save network resources and to reduce the control overhead mesh nodes do not broadcast the exact NXTO but only the *NextXmtTimeInterval* (NXTI) which is a series of one or more transmission opportunities. Therefore the IEEE 802.16 standard defines parameters NXM and XHE. Since each MSH-DSCH messages includes both node and all one-hop neighbors parameters, a generic node is able to calculate the NXTI of all nodes in the two hop neighborhood. Practically, a node has to wait a minimum of XHT (as defined in the previous sub-section) after the current XMT before it can send the next MSH-DSCH message. In [17] the authors developed an analytical model to evaluate the system performance of the distributed scheduler of the IEEE 802.16 mesh mode. Specifically, they developed methods for estimating the distributions of the node transmission interval and connection setup delay. Based on their analysis, the nodes with real time traffic shall have smaller hold-off exponents (XHE) because they need more chance to obtain data channel. However, too many nodes with small exponent value generate intensive competition that wastes system resource. A good reservation scheme should guarantee the bandwidth allocation fairness, improve the channel utilization and should adjust the exponent values of the nodes adaptively according to the competition node number variation. Based on these considerations we developed a probabilistic dynamic algorithm based on the data queue size of each node that adjust their hold-off exponents in a dynamic way. Loosely speaking, each node evaluates the data queue size and adjusts its XHE according to the queue size value: the greater is the size value (the greater is the number of data packets a node has to send) the smaller will be the XHE because, the node, shall have more chance to obtain data channel. Vice-versa, the smaller is the data queue size the greater will be the XHE. A pseudo-code that shows the fundamental operations of the dynamic approach is the following:

```
{const int Max_Buffer_Size;
var int queue_size;
//Node A is in the current transmission opportunity
Node A evaluates its queue_size;
//The number of data packets in the queue of the node
// A is higher than the half of the maximum of the
// queue size
if (queue_size > Max_Buffer_size/2) then
        XHE (Node A) = 0;
//The number of data packet stored in the A data
//queue is less than the half of the maximum size of the queue
else
        (XHE (Node A) = probabilistic value between 1, 2
        or 3);
//the probabilistic value is evenly chosen between 1, 2and 3}
```

This approach is probabilistic in the sense that, after a node evaluated its data *queue_size,* if this size is less than the half size (under-loaded node) of the entire capability of the queue, the node will set a XHE parameter between different values (1, 2 and 3) in an evenly distributed fashion. In this way, a node will choose one of the values with a probability p=0.333333. We adopted this probabilistic technique in order to avoid all the nodes to set the same value. With this probabilistic approach it is possible to "spread" the competition mitigating the collision probability.

On the contrary, when the number of data packets is higher than the half of the queue size (overloaded node), this means that a node need more data slots in a lesser time than other ones. For this reason this node will set the XHE equal to 0. The *Max_Buffer_Size* is the max number of data packets a node is able to store. This means that if we set *Max_Buffer_Size* equal to 50 (value used in simulations) and the buffer is full, a node will drop next data packets. The parameter *queue_size* is the number of data packets stored at the current instant that a node (*Node A*) has to send. We already said that queue size information is available at network layer. MAX_XHE is the maximum value of *XmtHoldoffExponent* that a node can acquire in our network. In [17] the authors argue that on choosing the MAX_XHE, four is large enough; otherwise, the connection setup latency will become too long. The threshold value of *Max_Buffer_Size/2* has been chosen as an intermediate starting point of analysis. Further study will be conducted to optimize this choice. We set MAX_XHE equal to 3 because higher values introduce an excessive latency for a node and even if a node does not have data packets to send, the information about its condition has to be known from its neighborhood. In practice, if a node has to wait longer time before to acquire a new opportunity to transmit (TO) it is not able to send its schedule information to the neighborhood. This impacts over latency because, in absence of update information, a node will be assumed as a competing node. It is worth to notice that the extension of the algorithm as developed does imply that the fairness of the scheduler scheme of the Std. is kept. In fact, a node can be delayed for some time to select its next opportunity to transmit (TO), but in a reasonable time each node will be able to select a new opportunity to transmit. This is due to the fact that the MAX-XHE has been set equal to 3 in our approach and the control slot acquisition delay is bounded in this way.

C. The Randomized-MAC scheduling scheme

Based on the description of the Coordinated Distributed Scheduling scheme CDS of the Std. IEEE 802.16 (CDS), every node competes for the channel access and tries to broadcast its scheduling information periodically (a node can transmit its schedule and can try to reserve new data slots in the current TO). The channel contention result is correlated with the total nodes number, exponent value and network topology. A key factor in a similar scenario is represented by the capacity to use control slots in an effective fashion and some parameters as *XHE* and *NXM* have to be opportunistically set in order performance to be improved. We know that the usage of control slots of the CDS is not optimal [12]. In fact in [12] the authors measured the number of unassigned control slots in each frame. This parameter permits to understand the dependence of the topology of the network with the scheduling scheme.

1) Details of the proposed Randomized-MAC (R-MAC)

The R-MAC scheme differs from the IEEE 802.16 CDS scheme in the selection of the NXTO. In fact, a node transmitting in the current TO runs a Random Function to select the NXTO instead of considering the Hash Function (MeshElection) of the 802.16 standard. The frame structure is shown in Figure 5. The scheduling mechanism for a Local Node (LN) is described in Figure 6. In Figure 6 Random Function is a function that randomly picks-up an available slot. A slot is available if the state in the next frame of the Local Node is IDLE. The Redistribution Function is applied if and only if a node did not find an available TO slot in the previous frame or it lost it for some reason as shown in Fig. 7. Redistribution Function is a Random Function in which a node, that needs a TO for the current frame, analyzes a set of available slots (the state of the Local Node is IDLE in this slot) and randomly selects one of the available slots. In this way all the neighbors that does not have a valid TO slot in the current frame will apply the same random function and it can happen, above all when density network increases, that two 1 or 2 hop neighbors select the same control slot. In this case there will be a collision which will be resolved in the next frame, in which each node will compute a new control slot.

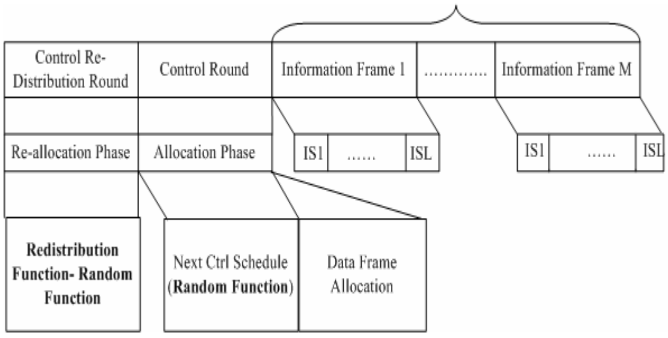

Figure 5. Frame Structure of the Randomized-MAC (R-MAC). At the beginning of each control frame, Re-Distribution Round is applied, it will be applied only for the nodes that did not maintain a valid *XmtOp* for the current frame. The Redistribution Function considered is the same Random Function used in the Allocation Phase. In the Control Round each node computes the next *XmtOp* applying a Random Function and sends updated information in the current *XmtOp*. Data slots allocation takes place in a different portion of the frame.

The main difference between R-MAC and CDS is in the assignment mechanism of control slots. In CDS, based on the setting of specific parameters, a node has to wait for a certain amount of time before to compete anew. R-MAC generates schedules not necessarily conflict-free, but the assignment of control slots is realized in a more "aggressive" fashion that permits to use almost all control slots in each frame.

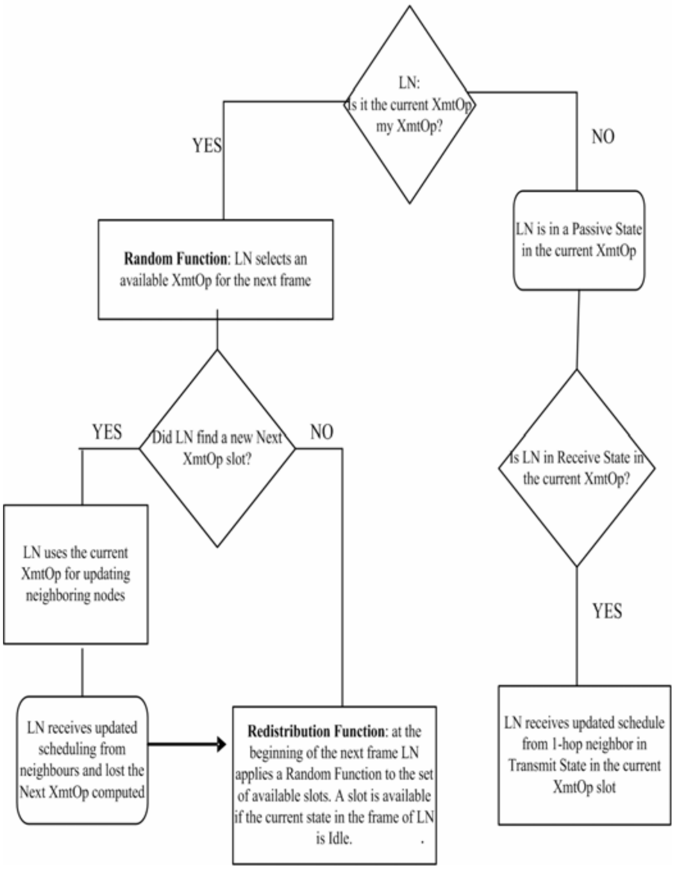

Figure 6. Flow Chart of Local Node (LN) scheduling mechanism of R-MAC.

Changing network topology implies that the parameters as XHE and NXM have to be set anew in CDS. To the contrary we have to set no parameters in R-MAC.

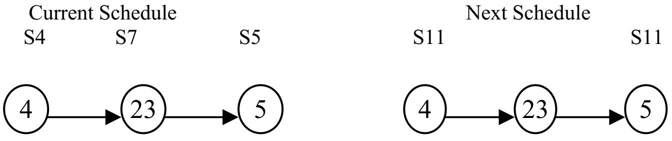

Figure 7. Current Schedule: node 4 is transmitting in the slot S4, node 23 is transmitting in the slot S7 and node 5 is transmitting in the slot S5. Node 4 reserves the slot S11, in the current frame for the next frame, as *XmtOp*. This reservation takes place in the slot S4. Node 4 transmits the updated schedule to its neighborhood. Node 23 is notified with the updated schedule. In the next slot, S5, node 5 reserves the same slot S11 (selected randomly). Node 5 will be notified that this slot has already been reserved by another neighbor node (node 4) in the slot S7 by node 23. Node 5 will set its state in the slots S11 as state(S11) = Block_RX.

Each node considers a number of TOs that is 16 as in the IEEE 802.16 standard and this is the number of opportunities to transmit (control slots used to update the neighbours and make new data slots requests). The channel is partitioned in two sub-portions: a Control Round where the schedules are updated and Data Round where user data transmission takes place (Figure 5). The state (one of the states considered in Section II) of the nodes in each slot is updated during the control round when each node in the current TO computes the

NXTO and sends the updated information to the neighbors. The info-schedule (or data schedule) will be updated at the beginning of each new frame (Data Frame Allocation); in this way different transmission requirement nodes can be accommodated. Two different data are considered in the Control Round: 1) current information (or current schedule) and 2) next information (or next schedule). The current schedule is the actual schedule used by node to transmit and to compute another NXT (in accordance with the IEEE 802.16 notation that is the NXTO). After the Random Function is called and the NXT is computed a node updates its schedule and sends this information (the updated schedule) to its neighborhood (1 and 2-hop away nodes) through the message (MSH-DSCH) where the request of a certain bandwidth (data slots requests) is contained. Let us assume that during a frame (F_{curr}) a node tries to acquire a new slot (TO). Once a node receives information from the neighbors it updates neighbors information and it can happen that the slot it tried to reserve has already been reserved from another node. In these conditions the Local Node (LN) sets its state in this slots as *Block_RX* (see Figure 7). Once the situation as described happens LN remains without an opportunity to transmit in the Frame ($F_{next\text{--}curr}$) even if there are some unassigned slots. Based on these considerations we developed a scheduling scheme in which a Re-Distribution Control sub-phase is implemented. In this way we try to use a higher number of control slots. We would outline that our approach does not introduce any additional different overhead packet with respect to the standard approach (CDS). In fact, the scheme developed is based only on a different computation of the TO. By the way, it can happen that some more control MAC packets are sent using R-MAC due to a higher slots assignment. An estimation of the potential increment of overhead is given in Appendix II.

D. Details of the proposed Distributed Scheduling Scheme(DSS)

In this section we describe some potential enhancements to mesh deployment of 802.16 that are well within reach. The proposed advances should increase the average throughput and the robustness of the Coordinated Distributed Scheduling scheme of 802.16. We do not modify the structure frame of the standard and we do not add complexity, so the hardware of the standard must not be changed. The proposed scheme differs from the IEEE 802.16 CDS scheme in the selection of the new TO. In fact, a node transmitting in the current TO uses a Random Function instead of the hash function (the MeshElection function defined in the IEEE Std 802.16) to compute the NXTO. The frame structure is the same of the R-MAC shown in Figure 5. The only difference consists in the Redistribution Function of the Re-allocation Phase. In fact, R-MAC considers a random function to re-allocate unassigned control slots while DSS applies a hash function. The advantage to use a hash function instead of using a random function is that in DSS no collisions are possible in this phase and we can affirm that the whole mechanism of the DSS allows the conflict-free schedules to be generated. In R-MAC the random function applied at the beginning of each frame does not ensure the schedules are conflict-free, but on the other hand allows a greater number of control slots to be assigned. The DSS scheduling mechanism for Local Node (LN) is described in Figure 6. In Figure 6 Random Function randomly picks-up an available slot. A slot is available if the state in the next frame of the Local Node is IDLE. The access is conflict-free because the Local Node uses the current TO slot computed in a conflict-free fashion in the previous frame. Whether a node (Local Node, LN) did not find an available TO slot in the previous frame or lost this slot for some reasons as show in Figure 5 it will apply a Redistribution Function. This function is a Hash Function in which a node, that needs a TO for the current frame, analyzes a set of available slots (that is, the state of the Local Node is IDLE in this slot). The Hash Function will be applied to the 1-hop neighbors of the Local Node. In this way, all the neighbors that do not have a valid TO slot in the current frame will apply the same function with the same information (slot number, neighbor ID and address of the node applying the function) and only one node will win the competition. The main advantage of the DSS mechanism is that no parameters must be set, and in this way the protocol is robust for different topologies and network scenarios. Each node considers a number of TOs that is 16 as well as in the IEEE 802.16 Std and this is the number of opportunities (in the control frame) to transmit updated schedule, send data slots reservation requests and reserve another TO. The channel is partitioned into two sub-portions: a Control Round where the schedules are updated and Data Round where user data transmission takes place (Figure 5). The state of the nodes in each slot is updated during the control round when each node in the current TO computes the NXTO and sends the updated information to the neighbours. The data schedule will be updated at the beginning of each new frame; in this way different transmission requirements nodes can be accommodated. As we can observe the DSS scheme is very similar to the R-MAC scheme. In fact, as we already outlined, the only difference consists in the type of function implemented at the beginning of each control-frame.

IV. MULTIPATH PARALLEL ROUTING PROTOCOL (MPRP)

In the literature, there is much research on multi-path routing for ad hoc networks [19, 21, 22, 23]. There are several philosophies that approach the problem of multi-path in different ways. In this paper we use all multiple paths at the same time and packets are split among these. This is because the Network Model as considered in the previous section is static or quasi-static and although some topological changes can happen they are very low with respect to network in which mobility is supported. For this reason it seems more useful to considering paths used simultaneously that permit a load balancing to be obtained. In fact, the use of multiple paths as backup paths seems more appropriate for the fault-tolerance in a mobile context.

A. Multipath Parallel Routing Protocol (MPRP)

The Multiple Parallel Routing Protocol [37] is a simple distance vector routing protocol that allows multiple paths for a single couple of nodes source and destination to be built. In the MPRP, each mobile host maintains a multiple path routing table. Each entry of the table contains following information: destination address, next hop address, hop count, sequence number and a pointer to the list of multiple paths (route-list).

As well as in Ad hoc On-demand Distance Vector routing protocol (AODV) the value of sequence is used for determining the freshness of a route. Each element of the multiple list contains next hop address, hop count and Route Expiration Time (REXP).

1) Computation of Multiple loop-free paths: Route Discovery Phase

In the original version of the Ad hoc On-demand Distance Vector routing protocol (AODV), duplicated Route Request packets (RREQs) are discarded. In MPRP, all duplicated RREQ copies should be processed. However, using all duplicated route copies to obtain multipath, may cause routing loop. In MPRP all duplicated copies are examined, but only those which permit the preservation of the loop-freedom property are considered in building multiple paths. The integration between the CDS, the ProbAppr, the R-MAC and the DSS and the MPRP permits the interference between the paths to be eliminated. This is due to the fact the all the scheduler schemes considered are based on a TDMA approach and compute in a distributed fashion conflict-free schedules. The only scheme that can generate some collisions is the R-MAC, but we will evaluate this characteristic in the Performance Evaluation Section and we will see that even if the schedules are not totally conflict-free, good performance in terms of throughput and delay are obtained. To the best of our knowledge this is the first work considering the integration between totally distributed scheduler schemes, specifically designed for WMNs and a Multipath routing protocol. In this paper we are interested to evaluate the impact of using multiple paths to split data traffic. In fact, we do not consider any specific policy or link quality measure to select a path. Simply, we randomly select an available path to split data traffic. We borrowed the concept of advertised-hopcount from the Ad hoc On-demand Multipath Distance Vector routing protocol (AOMDV) [23]. The advertised-hopcount of a node *i* for a destination *d* represents the "maximum" hopcount of the multiple paths for *d* available at *i*. "Maximum" hopcount is considered, as the advertised hopcount can never changes for the same sequence number. The protocol only allows accepting alternate routes with lower hopcounts. This invariance is necessary to guarantee loop-freedom property. The advertised-hopcount is initialized each time the sequence number is updated. A node *i* updates its advertised-hopcount for a destination *d* whenever it sends a route advertisement for *d*. It is updated as shown in Figure 8.

$$\textit{Advertised-hopcount}_i^d := \max_k\{\textit{hopcount}_k \mid (\textit{nexthop}_k\ ,\ \textit{hopcount}_k) \in \textit{route-list}_i^d\}.$$

The same rule as in AOMDV is used in order the *loop-freedom* property to be guaranteed as shown in Figure 4.

```
if $(\text{seqnum}_i^d < \text{seqnum}_j^d)$ then
    $\text{seqnum}_i^d := \text{seqnum}_j^d$;
    if $(i \neq d)$ then
        $\text{advertised-hopcount}_i^d := \infty$;
        $\text{route-list}_i^d = \text{NULL}$;
        insert(j, $\text{advertised-hopcount}_j^d + 1$)
            into $\text{route-list}_i^d$;
    else
        $\text{advertised-hopcount}_i^d := 0$;
else if $(\text{seqnum}_i^d == \text{seqnum}_j^d)$ &&
        $((\text{advertised-hopcount}_i^d) > (\text{advertised-hopcount}_j^d))$
    insert(j, $\text{advertised-hopcount}_j^d + 1$)
        into $\text{route-list}_i^d$;
endif
```

Figure 8. Route update rules. This is used whenever a node *i* receives a route advertisement to a destination d from a neighbor *j*. The variable seqnum_i^d, $\text{advertised-hopcount}_i^d$ and route-list_i^d represent the sequence number, advertised-hopcount and route-list for destination *d* at node *i* respectively.

Let us to consider an example of MPRP works. In Figure 9 a route-discovery phase is shown. The RREQ packet is sent out from the source node S to its neighbors. Suppose that node N_1 receives the RREQ packet from the node S and the node P_1 receives the RREQ packet from the node N_1 before to receive it from the node S.

Example:

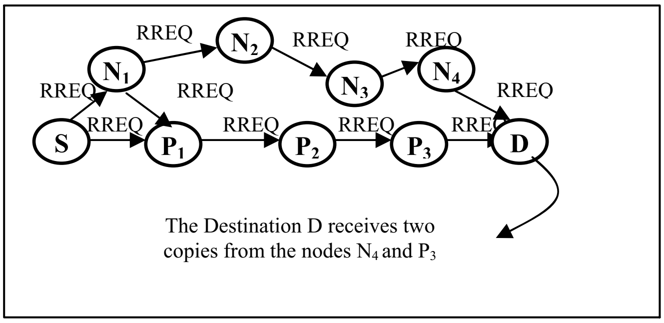

Figure 9. The RREQ packet is propagated from the source node S to the destination node D in the Route Discovery Phase.

The RREQ packet will be propagated in the network until it will reach the destination D. Each intermediate node can process a new RREQ packet and the destination node D receives two RREQ packets, the first one from the node N_4 and the other from the node P_3. In this case two paths with a common link will be built: the S-N_1-N_2-N_3-N_4-D path and the S-N_1-P_1-P_2-P_3-D path (see Figure 10).

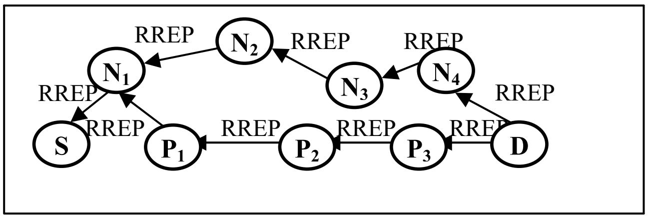

Figure 10. In this phase the RREP packet is propagated.

Note that the rule used to guarantee the loop-freedom property excludes the second path found to be greater (that is, with a higher number of hops) than the first one found. Assume a data packet is sent from source node S to N_1 for the destination D. N_1 has two different paths for the destination D, the first one has the next-hop N_2 and the other has the next-hop

P_1. N_1 randomly will select the path without considering any link quality measure or queue length.

***Lemma**: two parallel paths for the same couple source-destination (SD) do not interfere to each other even if they are simultaneously used.*

This is because a conflict-free schedule scheme is used at MAC layer that permits nodes belonging to the two paths have different control slots and cannot interfere to each other.

V. PERFORMANCE EVALUATION

A. Simulation Environment

As we already outlined, the main focus of this paper is the evaluation of the impact of a simple multiple routing approach over different MAC protocols based on distributed schedulers suited for WMNs. In order to do that, we implemented the different schedulers described above in this paper and the Multiple Parallel Routing Protocol, MPRP, in a well-known simulator, ns2 [36]. We generated different simulation scenarios in order to compare the unipath version of our routing protocol and the MPRP when two or three multiple paths are built. We limited the total number of paths to three, because we observed that in the specific scenarios considered, the increasing of the number of paths does not significantly change performance of the network. Parameters used in the simulation are summarized in Table I. It is worth to "justify" some choices regarding these parameters. We evaluated the synergic effect of the different MAC approaches and the MPRP by changing the density of the network, because every scheduler scheme considered in this work is TDMA-based . In practice, control slots are re-assigned to different nodes whether nodes are sufficiently far away from each other (in this work we limited the distance to re-assign the same slots to two hops) and performance of the network are affected from the density of the network. Generally, the increasing of the density of nodes in a network implies degradation in terms of performance. On the other hand, when a multiple routing approach is considered, the higher is the number of nodes in the network the higher is the number of potential multiple paths. For this reason we decided to vary the density of nodes in the network by varying the number of nodes and keeping the size of the grid. Another interesting aspect that affects the performance of the network is the main size of the paths found through a routing protocol. We took into account this aspect by changing the size of the simulation area from a square grid of 1000X1000 to a rectangular grid of 1200X600. We introduced a number of sources equal to 40 and we considered different transmission rates, ranging from 5 pkts/sec. to 500 pkts/sec., and we did that because certain nodes are more data traffic loaded than other nodes in this way. This is a very interesting aspect concerning WMNs. In fact, we do not have topological changes due to the mobility of the users, but we need dynamicity of the protocols considered (they have to be very reactive) because the traffic could be extremely variable from a user to another and for the same user during the time. In this work we evaluated some significant parameters for wireless mesh networks:

1) *Throughput*: is the number of data packets correctly delivered to the destination over the number of data packets sent;

2) *Average end-to-end data packet delay*: the time that takes into account all the delay encountered in the network as buffering delay, transmission delay and propagation delay of data packets;

3) *Data Packets Lost*: is the average number of data packets not delivered at the destination. A data packet is considered not correctly delivered when it is corrupted or it is not delivered when there is not sufficient space in the data buffer (data buffer full).

TABLE I. SIMULATION PARAMETERS

Input Parameters	
Simulation area	1000x1000 meter, 1200 x 600 meter
Number of traffic sources	40
Number of nodes	30, 40, 50, 60, 70
Size of data packets	64 bytes
Transmission range	250 m
Simulation Time	500 s
Data Buffer Size	50
Data traffic scenario	
Number of Traffic sources	40
Sending rate	5, 50, 500 pkts/sec
10 sources	Rate 5 pkts/sec
15 sources	Rate 50 pkts/sec
15 sources	Rate 500 pkts/sec
Simulator	
Simulator	NS-2 (version 2.1b6a)
Medium Access Protocol	802.16 (CDS), R-MAC, ProbAppr, DSS
Link Bandwidth	1 Mbps
Confidence interval	95%
#run	10
#Control Slots (TOs)	16
Slot	32 bytes
Frame	64 slots
Slot duration	1.54e-04 sec.
Frame duration	9.86e-03 sec.

B. Results

In this sub-section we show simulation results. We compare results of the unipath version of our routing protocol and the MPRP with two and three multiple paths used in a parallel fashion over all the scheduler schemes implemented at MAC layer. As we already outlined, in this work we did not introduce any specific criteria to split data traffic, but we evenly distributed data packets on the different available paths. In Figure 11 we evaluated Data Delivery Ratio when the CDS scheme of the Std. is considered. Results show how multiple paths allow increased Data Delivery Ratio to be achieved. As we can observe, the positive effect of multiple paths increases when the density of the network increases too. We can justify this behavior because the possibility to find more available paths between a pair of nodes increases when we have more nodes in the network. On the other hand, the resources as the number of control slots and their re-usability become less effective when the density of the network increases. For this reason it is interesting to evaluate the effect of the two layers together, the network and the MAC layers. In fact, we can observe that the throughput increases in correspondence of 50 nodes and starts to decreasing in correspondence of 60 and 70 nodes. We explain this behavior considering the two opposite effects of the routing and the MAC protocols. Another important observation that we can do is that this behavior is similar when unipath routing is considered. Probably, this is due to the fact that, when a smaller number of nodes are considered in the network the average length of the paths is greater. This reasoning is confirmed when delay plots are considered as we will see to the follow. Similar behaviors can be observed when the R-MAC is considered in Figure 12. We can observe that the distance between the different curves in Figure 12 is smaller than in Figure 11. This means that multiple paths impact is smaller in the case of R-MAC is considered. This is due to the fact that R-MAC allows a better control slots distribution to be realized as observed in [12] and performance in terms of throughput are better when the unipath is considered. The same reasoning is available when the DSS is considered. In fact, we realize a better distribution of control slots that is confirmed from better results obtained (DSS allows to receive from to 2 to 5 % of data packets more than the 802.16), but the percentage of unassigned control slots, that is not reported in this paper (due to the lack of the space [13]), is higher in DSS than R-MAC and for this reason the impact of multiple paths is stronger in DSS. Multiple paths have a positive effect also when the ProbAppr is considered (Figure 14). However, also in this case we have to observe as the data delivered are higher when the unipath approach is considered than the unipath version of the 802.16 and for this reason the impact of multiple paths is not strong as in Figure 11. Figure 15 shows the throughput when a different scenario in terms of simulation area size is considered. The rectangular are of 1200X600 meters implies that the average number of hops for the path found is higher than 1000X1000 square meters. In fact, we can assist to a general degradation of the performance and the percentage of delivered data packets is smaller in this case. This represents a good scenario to apply a multipath routing approach.

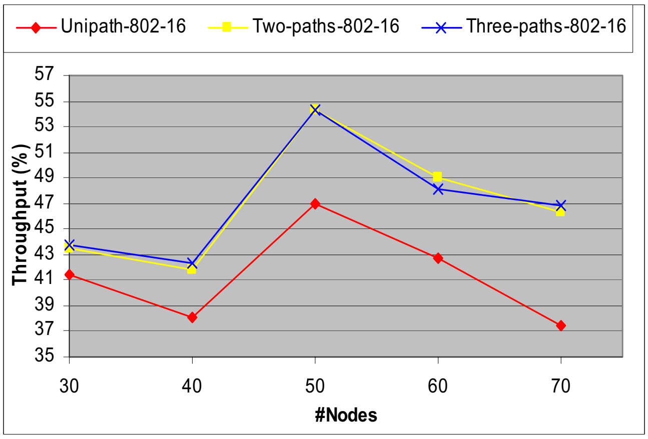

Figure 11. Throughput (%)(1000 X 1000 grid).

Concerning the average end-to-end data packet delay we can observe as this parameter increases for unipath version on all the scheduler schemes considered. In fact, in Figure 16 the delay for the CDS scheme is shown and the best value is obtained in correspondence of 50 nodes. As we already observed for throughput is also available for the delay. In fact, 50 nodes seem to represent a good trade-off between the two opposite behaviors.

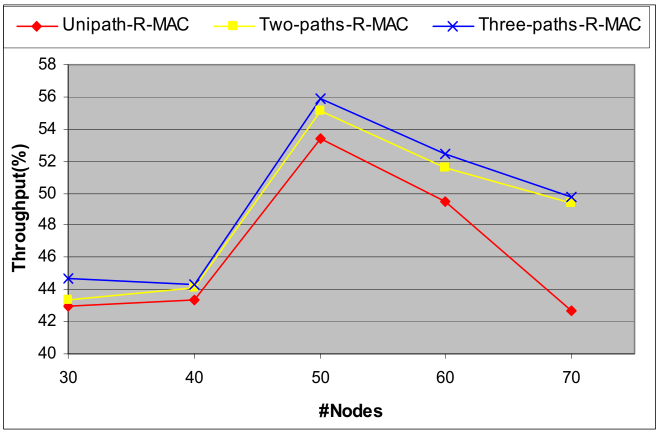

Figure 12. Throughput (%)(1000 X 1000 grid).

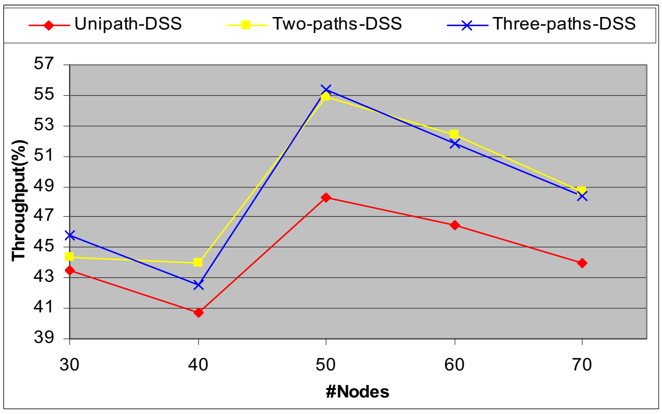

Figure 13. Throughput (%)(1000 X 1000 grid).

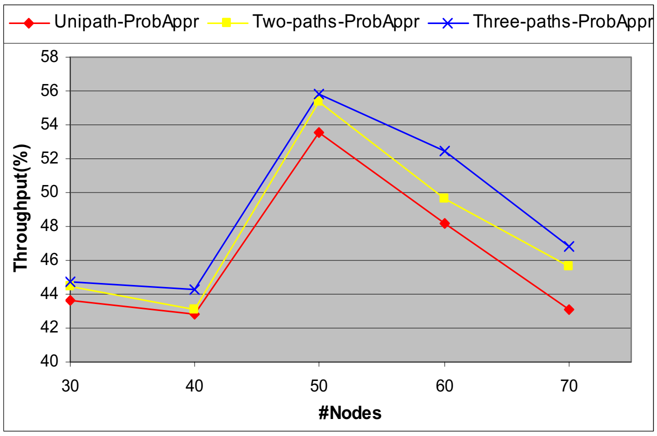

Figure 14. Throughput (%)(1000 X 1000 grid).

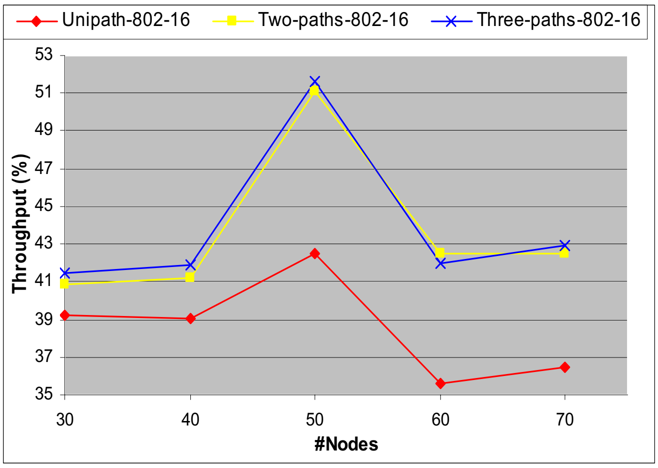

Figure 15. Throughput (%)(1200 X 600 grid).

The TDMA-based protocols react better when the density of the network is smaller because the resources are enough for all the users, but if the number of nodes is too small it is difficult to find "good" paths (in terms of number of hops) and a node can be loaded for different simultaneous transmissions. On the other hand when the density increases the resources of the networks are not sufficient for simultaneous requests and a node can be "delayed" to reserve slots and to send data packets, but it is possible to find better paths and above all it is possible to find more available paths that work in a simultaneous fashion. In Figure 17 the delay of the R-MAC is shown and we can observe that the distance between the curves is small and this slope of the curves confirms the reasoning we considered for the throughput (Figure 12). In Figures 18 and 19 we can observe that the minimum delay is obtained when 50 nodes are considered as in the other cases. The ProbAppr (Figure 19) presents a more linear behavior in respect of the R-MAC (Figure 17) or the DSS (Figure 18). Maybe this behavior is related with the dynamicity of the protocol. In fact, the ProbAppr adjusts its average latency time (with the term latency we mean the average interval that occur between the acquisition of a control slot and the next for a node), in respect of the different data traffic load, that is the ProbAppr take into account the different needs of the nodes in different times. However, ProbAppr does keep the fairness because nodes that have a smaller number of data packets to send in a certain moment are only delayed for a certain time but they will acquire control slots in a finite time (no starvation in the system is generated). The effect of the two opposite behaviors is much more visible when the grid of 1200 X 600 meters is considered (Figure 20). In Figure 21 we evaluated the percentage of lost data packets in the network because the data buffer is full over the total number of lost data packets. This parameter is very interesting because is a kind of measure of the load balancing obtained through the multiple approach. In fact, we expect that data packets are frequently lost due to the full buffer when the multiple paths have a small impact. When multiple paths allow to distributing data packets on different nodes we obtain more data packets delivered to the destination and data buffers will be regularly "emptied" and the percentage of data packets lost for full buffer will decrease.

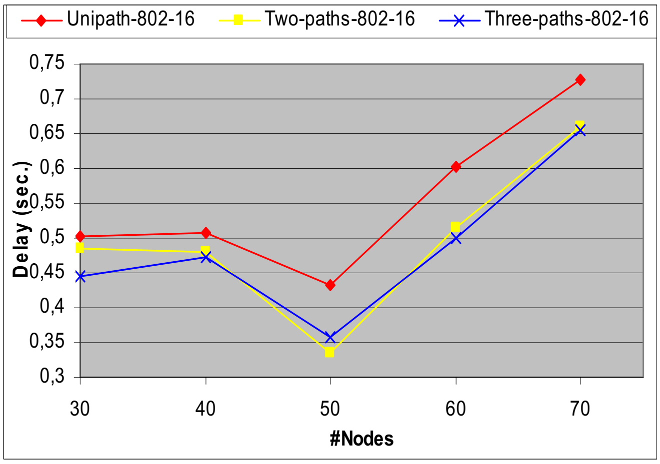

Figure 16. Average end-to-end data packet delay (sec.)(1000 X 1000 grid).

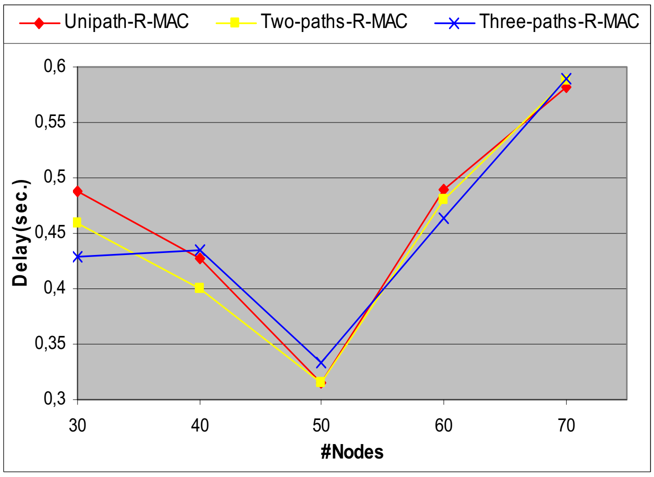

Figure 17. Average end-to-end data packet delay (sec.)(1000 X 1000 grid).

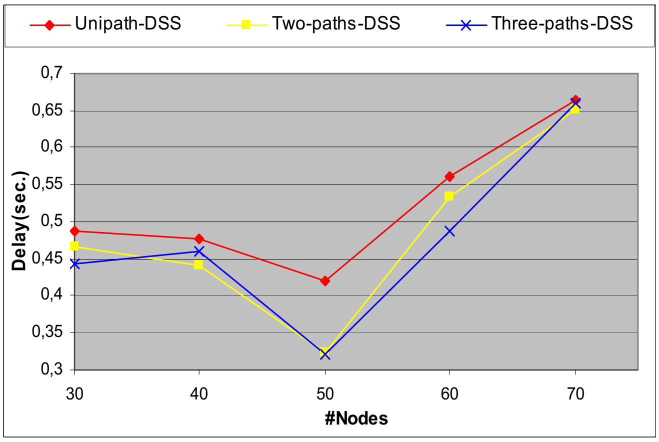

Figure 18. Average end-to-end data packet delay (sec.)(1000 X 1000 grid).

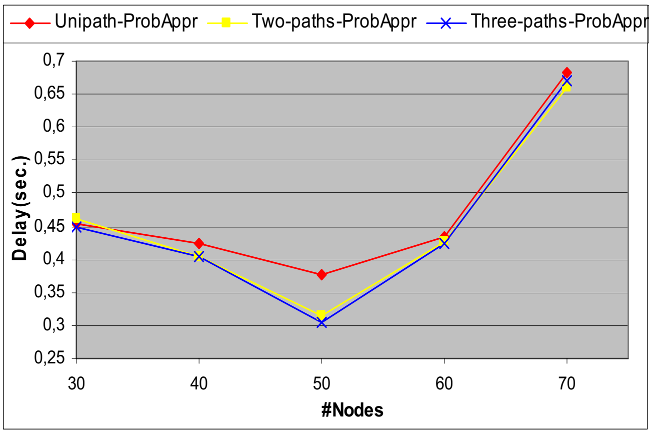

Figure 19. Average end-to-end data packet delay (sec.)(1000 X 1000 grid).

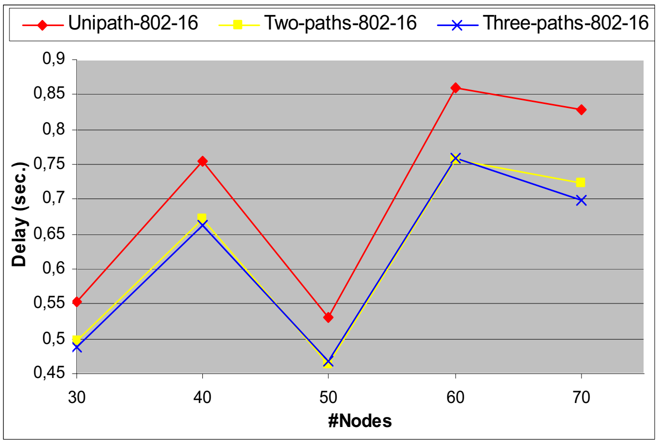

Figure 20. Average end-to-end data packet delay (sec.)(1200 X 600 grid).

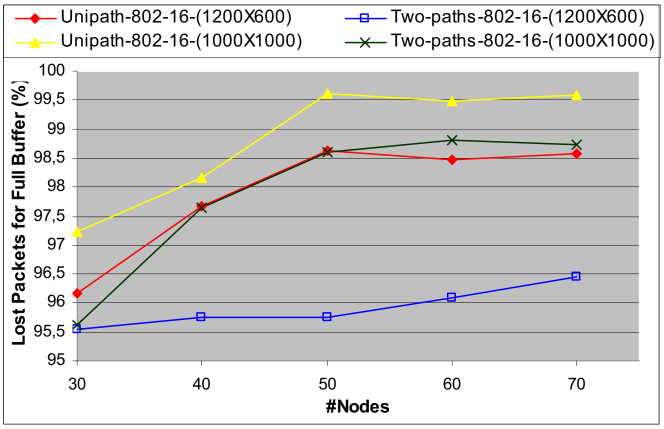

Figure 21. Lost data packet for full data buffer over Lost data packets (%).

VI. RELATED WORK

Multipath routing allows building and use of multiple paths for routing between a source-destination pair of nodes. It exploits the resource redundancy (considered as the possibility to use different paths to send data packets) and diversity in the underlying network (that is the use of multiple channels through a TDMA approach) to provide benefits such as fault tolerance, load balancing, bandwidth aggregation and improvement in QoS metrics such as delay, throughput. Although WMNs are similar to ad hoc networks in some respects, such as both being multi-hop wireless networks, there are a few important distinctions that warrant different routing strategies. Firstly, since mesh routers are stationary, mobility is no longer a problem. This means network topology change is less frequent than in ad hoc networks. Secondly, the traffic distribution in a WMN is generally skewed. This is because most user data traffic is directed towards/from Internet Gateways (IGs) or applications servers on the network [24]. Thirdly, WMNs demand better scalability, robustness and a range of other metrics in order to effectively provide backbone infrastructure services. There are hundreds of proposed routing protocols for MANETs, but as we already outlined the few differences between this kind of networks and WMNs requires new routing solutions. The WMN companies are using a variety of routing protocols to satisfy their needs. Some of these are proprietary and held secret [9], while other use well-known routing protocols as Firetide [25] that uses a Topology Broadcast based on reverse-path forwarding [26]. Other companies rely on the IEEE 802.1 spanning tree protocol for routing at layer 2 [27]. In [28] authors propose extensions of existing routing protocols such as the AODV[4], DSR [5] and SOAR [29], such that they optimize access to a set of nodes called netmarks (similar to the nodes called gateways in WMNs). All the protocols considered above for WMNs, build a single path between a pair of source-destination. In our work we exploit the possibility to use multiple paths with a minimum additional overhead that allows higher reliability and load balancing to be achieved. In [30] authors build a credit-based, opportunistic forwarding mechanism capable of using multiple routes at short-time scales and building long-term reliable routes at a cost of some redundancy. It is shown that for WMNs with significant variability in link quality, ROMER can significantly improve the throughput. The authors of ROMER considered that each node communicate with all its neighbors by sector antennas using 802.11g/a radios. They did not implement the MAC standard for WMNs. In our work we only designed and considered MAC protocols more suitable for WMNs. In [31] authors address the problem of high-throughput routing. They design a Multi-Radio Link-Quality Source Routing (MR-LQSR) as a source routing protocol and add a new path metric called Weighted Cumulative Expected Transmission Time (WCETT) that reflects the loss rate and link bandwidth. Another paper in which a different routing metric has been introduced is [32]. Also, Iannone et al. [33] propose a new metric. In [32] and [33], the authors show that non-conventional metrics may result in an increased capacity in WMNs. In this paper we do not propose any new routing metric. Instead, we exploit the possibility to use a very simple mechanism to build multiple paths and evaluate this approach routing on MAC protocols suited for WMNs. In [34] and [35], authors apply the concept of routing for multihop wireless routing. The design of their protocols exploits the feature of probabilistically independent reception of transmissions at different nodes. They do not leverage advantages of using multiple paths in an explicit fashion.

VII. CONCLUSIONS

In this work we considered a simple multipath routing approach the Multipath Parallel Routing Protocol, MPRP. MPRP allows loop-free paths to be created and multiple paths are used in a simultaneous way. In order to evaluate the impact of multiple paths we considered 4 different schedulers designed for WMNs that are TDMA based and totally distributed. Through simulation results we observed as the multiple paths allow better performance to be achieved in terms of throughput and delay in all the schemes, but some of the schemes considered permits to have better performance than the unipath version. The results obtained in this paper suggest that a cross-layer approach between the network and MAC layer would be a good direction to obtain better performance. In fact, we outlined as the routing protocol and the TDMA-based schedulers can behave in opposite ways. A cross-layer approach should be taken into account this aspect. Moreover, it seems that exist an optimal point that permits a trade-off between the two protocols to be achieved and this aspect would be analytically investigated.

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang. “Wireless mesh networks:a survey,” *Computer Networks,* vol. 47, pp. 445-487, 15 March 2005.

[2] B. Schrick and M. Riezenman, “Wireless broadband in a box,” *IEEE Spectrum Magazine*, pp. 38-43, June 2002.

[3] J. Garcia-Luna-Aceves, C. Fullmer, E. Madruga, D. Beyer, and T. Frivold, “Wireless internet gateways (WINGs),” in Proc. of IEEE MILCOM ’97,(Monterey, California), Nov. 1997.

[4] C.Perkins, E. Royer, S.Das, “Ad hoc On-Demand Distance Vector (AODV) Routing ,“ Internet Draft, 17 Feb. 2003.

[5] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networs,” in *Mobile Computing* (Imielinski and Korth, eds.), vol. 353, Kluwer Academic Publishers, 1996.

[6] E. Royer and C. Toh, “A review of current routing protocols for ad-hoc mobile wireless networks,” *IEEE Personal Communication*, Apr. 1999.

[7] A. J. Raju and J. Garcia-Luna-Aceves. “A comparison of on-demand and table-driven routing for ad-hoc wireless networks,” in Proceedings of *IEEE ICC*, June 2000.

[8] S. R. Das, C.E. Perkins,and E.E Royer, “Perfomance comparison of two on-demand routing protocols for ad hoc networks,” in Proceedings of *IEEE INFOCOM*, pp. 3-12, 2000.

[9] “MeshNetworks website.” http://www.meshnetworks.com.

[10] IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001), “IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Acces Systems,” 2004.

[11] URL: http://wirelessman.org/tga/contrib/C802.16a-02_30r1.pdf.

[12] V. Loscri’, “A new distributed scheduling scheme for wireless mesh networks,” in the Proceedings of the 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC’07).

[13] V. Loscri’, “MAC protocols over wireles mesh networks: Problems and Perspectives,” Elsevier Journal of Parallel and Distributed Computing, vol.68, Issue 3, pp. 387-397, March 2008.

[14] Guosong Chu, Deng Wang, and Shunliang Mei. A QoS architecture for the MAC protocol of IEEE 802.16 BWA System. IEEE International Conference on Communications Circuits & System and West Sino Expositions, vol.1, pp. 435-439, China 2002.

[15] Kitti Wongthavarawat and Aura Ganz. IEEE 802.16 based last mile broadband wireless military networks with quality of service support. IEEE Milcom 2003, vol.2, pp. 779-784.

[16] IEEE 802.16-2005, “ IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems for Mobile Users,” Dec. 2005.

[17] Min Cao et al. “Modeling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode”, *Proc. of the 6th ACM international symposium on Mobile ad hoc networking and computing,* pages 78-89, New York, NY, USA,2005, ACM Press.

[18] Ashish Raniwala and Tzi cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. *In Twenty-Fourth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, March 2005.

[19] Kui Wu, Janelle Harms, “On-Demand Multipath Routing for Mobile Ad Hoc Networks,” Computer Science Department University of Alberta, AB, Canada, T6G 2E8.

[20] Y. Ko and N. H. Vaidya, “GeoTORA: A protocol for geocasting in mobile ad hoc networks,” in *Proc. of the 8th International Conferenceon Network Protocols (ICNP)*, (Osaka), Nov. 2000.

[21] Valeria Loscrì, Floriano De Rango, Salvatore Marano” Mobile Networks Ad hoc on demand multipath distance vector routing (AOMDV) over a distributed TDMA MAC protocol for QoS support in wireless ad hoc networks:integration issues and performance evaluation, in European Transaction on Telecommunications, DOI:10.1002/ett.1117, 2006.

[22] Valeria Loscri’, Salvatore Marano “A New Geographic Multipath Protocol for Ad hoc Networks to Reduce the Route Coupling Phenomenon”, *In Proc. of Vehicular Technologies Communications-Spring(VTC)*, 2005.

[23] S.R.Das, et al. “On-demand Multi-path Distance Vector Routing for Ad Hoc Networks”, ICNP 2001.

[24] A. Raniwala and T Chiueh, “Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh networks,” in *Proceedings of Mobihoc’05*, May 2005.

[25] “Firetide website.” http://www.firetide.com.

[26] R. Ogier, F. Templin, and M. Lewis, “Topology dissemination based on reverse-path forwarding (TBRPF).” RFC 3484, Feb. 2004.

[27] “Mesh Dynamics website.” http://www.meshdynamics.com.

[28] S. Roy and J.J. Garcia-Luna-Aceves, “ Node-centric hybrid routing for ad-hoc wireless extensions of the internet,” in Proceedings of IEEE Globecom, 2002.

[29] S. Roy and J.J. Garcia-Luna-Aceves, “Using minimal source trees for on-demand routing in ad hoc networks,” in *Proceedings of IEEE INFOCOM*, Anchorage-Alaska, Apr. 2001.

[30] Y. Yuan, H. Yang, S. H. Y. Wong, S. Lu, and W. Arbaugh, “ROMER: resilient opportunistic mesh routing for wireless mesh networks,” in *Proceedings of the First IEEE Workshop on Wireless Mesh Networks (WIMESH’05)*, Santa Clara, CA, Sept. 2005.

[31] R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *Proc. of Mobicom’04*, (Philadelphia, PA), sep 2004.

[32] J. Tang, G. Xue, and W. Zhang, “Interference-aware topology control and QoS routing in multi-channel wireless mesh networks,” in*Proc. of Mobihoc’05*, May 2005.

[33] L. Iannone, R. Khalili, K. Salamatian, and S. Fdida, “Cross-layer routing in wireless mesh networks,” in *Proc. of the 1st InternationalSymposium in Wireless Communication Systems*, (Mauritius), Sept. 2004.

[34] Sanjit Biswas and Robert Morris, “ExOR: Opportunistic Multihop Routing for Wireless Networks,” in *Proceedings of SIGCOMM*, 2005.

[35] Sanjit Biswas and Robert Morris,” Opportunistic routing in multi-hop wireless networks,” In Proceedings of ACM Computer Communications, 2004.

[36] UCB/LBNL/VINT Network Simulator - ns (Version 2). http://www-mash.cs.berkeley.edu/ns/, 1998.

[37] V. Loscrì “A Routing Protocol for Wireless Mesh Networks” in *Proceedings of IEEE 66th Vehicular Technology Conference (VTC 2007-Fall)*, Oct. 2007, Baltimore, MD, USA

VIII. APPENDIX 1: DATA SCHEDULER

In this section we describe a data scheduler mechanism that is not based on the three-way-handshake mechanism of the Std. IEEE 802.16. Before of describing the new mechanism, we sketch some property of a data scheduler based on a three-way-handshake mechanism.

A. A three-way-handshake Data Schedule

The data scheduler of the Std. IEEE 802.16 has been left un-standardized. This means that the criteria used to assign data slots to nodes have not been defined. However, the Std. establishes that the data scheduler is based on a three-way-handshake mechanism as shown in Figure 22.

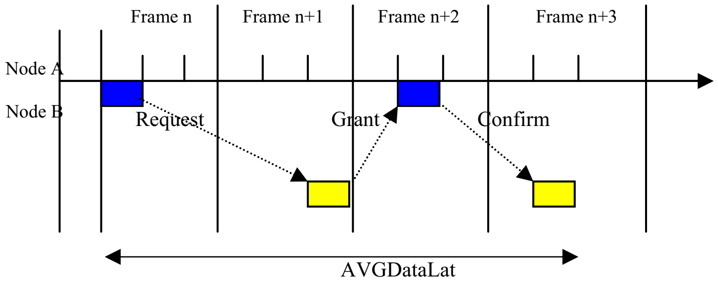

Figure 22. A data scheduler based on a three-way-handshake mechanism.

Requests, Grants and Acks are transported by messages transmitted in dedicated control slots, accessed according to the election procedure that we described in the previous section. As we are interested in the evaluation of the performance of the CDS of the Std. 802.16 we estimate the minimum interval time between a Request a Grant and finally the Confirm to use data slots. Let call the average time a generic node has to wait before a TO, to be obtained as AVGLat. Node A sends the Request of data slots in its current TO within the MSH-DSCH message, in a conflict-free way (within the MSH-DSCH node A sends its schedule to its neighbors). Imagine the request is addressed to a specific neighbor (1-hop neighbor, i.e. node B in Fig.4). We have to add the average time node B has to wait in order a TO, to be reserved and we call this time AVGLat. If data slots are available node B will grant node A (after a time equal to 2*AVGLat) and finally node A will wait an AVGLat time before using data slots and it will confirm the reservation in the moment of using data slots. As conclusion, a node A has to wait in average 3*AVGLat before to transmit and this is when data slots are immediately available. A three-way-handshake data schedule, allows conflict-free data schedules to be built, but data latency (that is, the average time between the request of data slots and their reservation) could be excessive. On the other hand, this work is to evaluate the impact of the CDS control schedule based on the MeshElectionFunction and its XHE parameter. For that, we developed a simpler data schedule and we deal simulations with this simplified data schedule.

B. A simplified Data Schedule

In the simplified data schedule node A (Figure 23) reserves a data slots S_A in the current TO and uses it immediately in the current data frame. The best case is for node A when no others neighbor nodes try to reserve the same data slot. Unfortunately, it can occur that two or more neighbor nodes (two-hops neighbors) randomly select the same data slot. In this latter case, the data slot will be not used from any nodes.

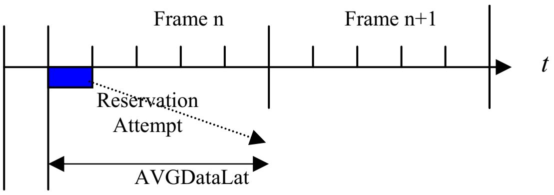

Figure 23. A simplified data scheduler.

We make an estimation of the average latency to acquire a data slot in both the data schedulers considered. Let call the number of data slots that are in the IDLE state (and are available to be selected for a new transmission in the data frame) AvailableDataSlots and ControlSlots is the number of control slots in a frame (that is 16 in the standard). In order to estimate this latency time we introduce the probability P, that two nodes A and B select the same slot:

$$P((\text{node A} \to S_A) \text{ and } (\text{node B} \to S_A)) = \frac{1}{\#\text{AvailableDataSlots}} \quad (1)$$

Let call P_{coll} the probability that at least two generic nodes selects the same data slot in the same control frame.

$$P_{coll} = P(\text{two nodes select the same data slot}) = \frac{\#\text{ControlSlots} - 1}{\#\text{AvailableDataSlots}} \quad (2)$$

Of course, we are supposing that in both the schedulers the selection of a data slot is a random selection. Moreover, we consider the following conditions:

- every node in the control frame cannot be updated about the recent reservation (this situation occurs if each node in the control frame is a two-hop neighbor of the other nodes in the current control-frame or two generic nodes that are two-hops neighbors cannot be updated to each other through a common one-hop neighbor for some reasons, i.e. the one-hop neighbor transmitted in a previous slot in the same current control-frame);
- Every node activates a data flow in the same time (that is, each node needs a data slot in the same time of every two-hop neighbor);
- At least two-hop neighbor nodes randomly select the same data slot.

Let call the average time between a request for a data slot and the real reservation of this AVGDataLat. In the case of the three-way-handshake mechanism we obtain:

$$\text{AVGDataLat}_{\text{three-way-handshake}} = 3 * \text{AVGLat} + 3 * (\text{Pcoll} * \text{AVGLat}) \quad (3)$$

In the standard, the control frame is 16 control slots and the data frame we considered is 64 data slots. The AVGDataLat of the simplified data scheduler will be:

$$\text{AVGDataLat}_{\text{simplified}} = \text{AVGLat} + (\text{Pcoll} * \text{AVGLat}) \quad (4)$$

Of course, the difference between the two data schedulers is that the data scheduler based on the three-way-handshake mechanism is totally conflict-free and the simplified data scheduler is not necessarily conflict-free. On the other hand, the conditions we considered above are rarely verified and in this context, in which we are interested in the evaluation of the performance of the control scheduler, we prefer to consider a simplified mechanism to reserve data slots above all when considering that the data scheduler reservation (that is the policy to assign data slots has been left un-standardized).

IX. APPENDIX II

As far as additional control overhead, at MAC layer, introduced by the R-MAC, is concerned, we roughly evaluated it. In practice, we need to individuate the worst case that is in correspondence of a number of nodes equal to 30. This value has been obtained on a simple observation of the results obtained in [12]. In fact, when 30 nodes are considered, the percentage of unassigned control slots for the CDS 802.16 is the highest value while the percentage of unassigned control slots for the R-MAC is at minimum value. This means that the number of control packets sent out by the R-MAC is at a maximum level while the number of control packets sent out by the CDS is at a minimum level. It is worth to notice that in this case we are evaluating the overhead considering that the average number of control slots CDS scheme assigns during a simulation is 84 % of the total available control slots. This estimation is not so accurate because some slots could be assigned to different nodes that are not neighbors to each other. In practice, some control slots could be re-used in a conflict-free fashion in CDS and R-MAC could assign different slots even if nodes are far away to be assigned the same control slot. Let us to compute how much more information R-MAC sends in the worst case. In [11] the MSH-DSCH message format is indicated. The constant number of bits that a node sends in an *XmtOp* is given by the sum of the different fields in the MSH-DSCH message (see Figure 18). The variable field MSH-DSCH-Scheduling_IE() varies in dependence with the number of neighbors. As we already seen we can use two different approaches to evaluate the mean number of neighbors for each node. We choose to consider the simulated approach in which the mean number of neighbors with 30 nodes in the networks has been estimated to be 3.5.

The variable field of the MSH-DSCH message format we are considering is the MSH-DSCH-Scheduling-IE(). It depends of the number of neighbors and is

16 bits + 24 * #Neigh (No_SchedEntries)

So in the worst case we have additional overhead information to send, in terms of bits of

Total Information = 220 * 2.5 = 550 bits

Management Message Type = 41	8 bits
Coordination Flag	1 bit
Grant/Request Flag	1 bit
Sequence Counter	6 bits
No. Requests	4 bits
No. Availabilities	4 bits
No. Grants	6 bits
Reserved	2 bits
MSH-DSCH-Scheduling_IE()	*variable*
MSH-DSCH-Request_IE()	16 bits
MSH-DSCH-Availability_IE()	32 bits
MSH-DSCH_Grant_IE()	40 bits

Figure 24. MSH-DSCH message format.

The rough estimation of the additional overhead introduced permits to conclude that there is no significant additional overhead with R-MAC in respect of the CDS scheme, above all considering better performance in terms of throughput and delay that we have obtained.

BIOGRAPHY

Valeria Loscri' received the Laurea Degree in Computer Science and the Ph.D. in Information Science, from the University of Calabria (UNICAL), Italy, in 2003 and 2007 respectively. Currently she is with the Department of Electronics, Computer Science and Systems (D.E.I.S.) of the University of Calabria as Research Fellowship with the Telecommunication group. Her current research interests include Mobile and Wireless Networks and Performance Evaluation of Communication Networks. From January 2006 to July 2006 she has been visiting researcher in the Rice Network Group at the Rice University of Houston (TX) and her supervisor has been Prof. Edward Knightly.