Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks xxx (2008) xxx–xxx
Ad Hoc Networks
www.elsevier.com/locate/adhoc

MERLIN: Cross-layer integration of MAC and routing for low duty-cycle sensor networks

Antonio G. Ruzzelli *, Gregory M.P. O'Hare, Raja Jurdak

Adaptive Information Cluster (AIC), School of Computer Science and Informatics, University College Dublin, Ireland

Abstract

Sensor network MAC protocols typically sacrifice packet latency to achieve energy efficiency. Such delays may well increase due to routing protocol operation. For this reason it is imperative that we attempt to quantify the end-to-end delay and energy consumption when jointly using low duty cycle MAC and routing protocols. In this paper, we present a comprehensive evaluation of MERLIN (MAC and efficient routing integrated with support for localization), a cross-layer protocol that integrates both MAC and routing features. In contrast to many sensor network protocols, it employs a multicast upstream and multicast downstream approach to relaying packets to and from the gateway. Simultaneous reception and transmission errors are notified by asynchronous burst ACK and negative burst ACK messages. A division of the network into timezones, together with an appropriate scheduling policy, enables the routing of packets to the closest gateway. An evaluation of MERLIN has been conducted through simulation, against both the SMAC and the ESR routing protocols (an improved version of the DSR algorithm). The results illustrate that the joint usage of both SMAC and ESR, in low duty cycle scenarios, causes extremely high end-to-end delays and prevents acceptable data delivery rate. MERLIN, as an integrated approach, notably reduces latency, resulting in nodes that can deliver data in a very low duty cycle, yielding a significant extension to network lifetime.

Keywords: Wireless; Sensor; Network; MAC; Routing; Protocol; Energy-efficient; Adaptive; Autonomous

1. Introduction

Distributed wireless sensor networks (WSNs) typically comprise large numbers of wireless devices deployed over a physical environment that actively cooperate in order to accomplish one or more tasks. Sensors are designed to support unattended operation for long durations, frequently in hostile environments. A sensing component forms an essential part of the device; popular examples include temperature, accelerometer, humidity, infrared light, pressure, and magnetic sensors, as well as chemical sensors. The wide range of potential WSN applications includes environmental monitoring [13], intelligent buildings [22], and object tracking [4]. Further novel applications envisage collaboration of sensors and radio frequency ID (RFID) tags with mobile devices, for example, in the field of logistics [4] and for military operations in [10].

* Corresponding author.
E-mail addresses: ruzzelli@ucd.ie (A.G. Ruzzelli), gregory.ohare@ucd.ie (G.M.P. O'Hare), raja.jurdak@ucd.ie (R. Jurdak).

1570-8705/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2007.11.012

Although WSN's have a wide and varied application space, common desirable features for most applications include robust and reliable communications, efficient energy consumption over the WSN's lifetime, scalability in terms of network size, dynamic programmability, dynamic network adaptivity in response to changes in the prevailing operating conditions and low unit cost per sensor. The latter invariably results in simple architectures, low processing capabilities and low memory capacities. Considering the often crosscutting design goals of a particular WSN application this poses significant difficulties that differ significantly from those encountered with wireless ad-hoc networks.

Recent studies on transmission radius [26], have demonstrated how protocols that are theoretically effective may perform poorly when deployed in realistic environments. Interference, multi-path effects and fading may cause a premature deterioration of the signal. In addition, nodes are not reliable and can often fail due to low cost hardware, leading to a dramatic reduction of the probability of receiving packets correctly. Furthermore, sensor network protocols often sacrifice packet latency in order to extend the network's operational lifetime. The result is a time delay that can exceed 1 s per hop [25]. MAC protocol latency is however not the only issue to be addressed as the routing protocol contributes to end-to-end packet delay. Quantifying the end-to-end delay for the joint operation of low duty cycle MAC and routing protocols is key to evaluating a WSN's performance. The definition of what constitutes acceptable performance varies in WSN's according to particular application domain requirements and the network size.

Traditional approaches have addressed energy efficiency and latency from either a medium access control or an effective routing perspective. This paper in contrast presents the MERLIN protocol (Mac and energy-efficient routing with localization support integrated), a cross-layer approach that jointly addresses energy efficiency and communication reliability issues through the close coupling of both MAC and routing layers. Integration of MAC and routing protocol features into a cross-layer architecture [5] can significantly improve sensor network performance, including: (1) a reduction of the end-to-end packet latency while preserving competitive energy efficiency for mutual node/gate communication; (2) an increase of the probability of packet reception at the gateway by providing controlled packet duplication to address sensor failure and lossy channels; (3) a more efficient usage of resources such as the memory capacity and processing capability of the sensor. MERLIN provides efficient and reliable communication for sensor network applications, targeting multi-hop topologies with designated data sinks rather than random peer-to-peer multi-hop communication with a random data sink.

This paper describes current developments with MERLIN, building on earlier work in [19] by introducing a novel transmission mechanism. A simulation of MERLIN is presented along with a comprehensive evaluation of its performance relative to SMAC [25] and the Eyes Source Routing [24] (ESR) protocols, as such protocols provide a useful benchmark for the comparative evaluation of MERLIN's performance. ESR, as the standard routing protocol in the EYES project, represents an improved version of the well known dynamic source routing [9] (DSR) protocol.

2. Related work

After a decade of research on wireless sensor networks, the scientific literature offers a profusion of energy-efficient protocols. Among them, Sensor-MAC (SMAC) [25] is one of the most well known sensor network MAC protocols, and serves as a useful benchmark against which to evaluate new protocols. The following discussion summarizes SMAC's main characteristics.

In SMAC, a node alternates periods of activity and sleeping. The sleeping period is further divided in three contention based access sections: the SYNC period dedicated to node synchronization update, the request to send (RTS) period, and the clear to send (CTS) period. Nodes periodically send a SYNC packet to synchronize the beginning of the active period with their neighbourhood. A node that has data to transmit firstly contends for the channel for the transmission of a SYNC packet and then follows the RTS/CTS/DATA/ACK handshake mechanism. Idle listening is high because even if no packets have been transmitted, nodes keep on listening until the end of the RTS period, or until they detect an RTS message addressed to another node. All packets include the duration of the transmission such that neighbouring nodes can set up a NAV timer and go into sleep mode so refraining from transmitting. To establish a communication, neighbouring nodes have to synchronize to the start of the active period. A node, named cluster head,

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

broadcasts periodically a SYNC packet for the rest of the nodes within the cluster. Bordering nodes have to keep synchronized with all their neighbours to allow communications between clusters. Therefore, they have to follow two or more synchronization timers simultaneously with an increase of node activity and hence energy consumption. Finally, the adaptive listening technique allows neighbouring nodes to take up passing the message a further hop away thus reducing the latency.

TMAC [2], which builds on SMAC, achieves improvements over SMAC in terms of energy consumption. TMAC forces nodes to start transmitting at the beginning of the active periods. Both TMAC and SMAC exhibit large end-to-end latency together with an idle listening at the receiver.

Another MAC protocol for WSNs is BMAC [16] which combines the mechanisms of low power listening [3,8] and clear channel assessment techniques in order to reduce idle listening at the receiver. A drawback of BMAC is however its poor performance under high contention scenarios due to the usage of a wake-up preamble. In the case of high channel contention, the wake-up preamble may be transmitted consecutively by several neighbouring nodes, which prevents the transmission of the actual data. SMAC, TMAC and BMAC use RTS/CTS, which causes high overhead due to the resultant small size of data payloads in sensor networks. Recently, Rhee et al. [18] developed Z-MAC that outperforms BMAC in high contention scenario by a combination of TDMA and CSMA access, while causing higher energy consumption for low contention scenarios. In a similar approach to MERLIN, DMAC [12] incorporates a data gathering tree to reduce latency. This technique is only suitable for unidirectional communication flow to a single gateway.

With respect to the routing protocol, the DSR [9] and AODV [15] algorithms are the most cited due to their flexibility and small resource footprint that facilitates their implementation upon computationally restricted sensors. In evaluating MERLIN within this paper, we use the eyes source routing [24] (ESR), which is based on the DSR protocol; ESR, whose main characteristics are subsequently described in this section, has been demonstrated to have an enhanced energy consumption profile to that of DSR. MERLIN fundamentally differs from the aforementioned protocols as they all address MAC and routing issues separately while MERLIN integrates both MAC and routing functionality.

Eyes source routing builds [24] (ESR) on dynamic source routing [9] (DSR) that is the de facto ad-hoc routing standard. The main functionalities of ESR are: (1) route setup, (2) route maintenance and (3) route re-establishment. A route setup packet contains four main fields: sourceID, destinationID, intermediate NodeID, and a hops to live (HTL) counter. The last is decremented every time a route setup packet is forwarded and it is used to limit the flooding to a maximum number of retransmission of the packet. In the route setup phase a source node that has a packet to transmit to a destination floods the network with a route setup message. A node that receives a route setup request stores the sourceID and the intermediate node ID (best neighbour), replaces the intermediate nodeID in the packet with its own ID and re-broadcasts the route setup message for other nodes. If a node receives a route setup request destined to it, then it is able to send back a route reply through its best neighbour that will forward it.

Route maintenance provides a local route re-catch mechanism in case a link breaks. The re-catch mechanism is triggered by the node that realises that the best neighbour is no longer available, triggering a local route setup to establish an alternative route. The local route setup is controlled by the HTL parameter. Furthermore a route cut mechanism can shorten the route in case a node notices that its second order neighbour comes to within transmitting range. Finally the route re-establishment mechanism is necessary when local route requests fail and route maintenance is not able to recover the broken link. This phase is similar to the route setup with the difference that nodes can use information learnt during the route setup phase to recover another route quickly. The authors propose a directional and geographically limited flooding to reduce the energy required in the route-reestablishment. to illustrate MERLIN's performance benefits, Section 5.3 compares MERLIN against a system that couples ESR and SMAC.

With regards to cross-layer techniques, adaptive low power listening (ALPL) [6] provides a recent cross-layer approach that advocates richer information sharing between the MAC and routing layers. ALPL uses a cross-layer cost function whose terms depend on features from several layers across the stack, in order to optimize routing costs and BMAC low power listening modes. MERLIN's cross-layer approach adopts a closer coupling of MAC and routing functionality than ALPL, enabling finer

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

grained energy and delay optimizations, moving further away from layered architectures that have shown considerable end-to-end packet delay and packet relay failures in low duty-cycle networks.

3. The design of MERLIN

3.1. Deployment considerations

Before describing the details of the MERLIN protocol, it is prudent to consider some pertinent issues relating to sensor network deployment. An essential requirement is the presence of at least one data sink node, namely gateway, at which data transmitted from the distributed sensors converges. As such, it is preferable for the gateway to have higher energy and processing capabilities. Large scale sensor networks may necessitate the deployment of multiple gateways. Adequate coverage of the area of interest requires placement of the gateways in a manner that minimizes disparity between sizes of the subnetwork covered by each gateway. This is a network topology problem, which is beyond the scope of this paper.

In cases where multiple gateways are deployed, MERLIN requires that their clocks are synchronised, possibly by means of an alternative communication system like satellite, WLAN or WMAN. Gateway synchronization enables the initialisation process as described in Section 3.6. Prior studies in [7] have shown that MERLIN can support node location through the usage of a received signal strength indicator (RSSI) device, usually incorporated within standard off-the-shelf sensors.

3.2. MERLIN overview

Fundamental to the MERLIN protocol is the *timezone* concept, which subdivides the sensor network into time zones, as illustrated in Fig. 3. Time zone division commences during the initialisation phase with the broadcasting of an initial SYNC packet from the gateway to neighbouring nodes which, in turn, will synchronize their internal clock, will set their timezone, and will increment the timezone counter in the SYNC packet before forwarding it to surrounding nodes. At the end of the initialisation phase, all nodes will have been organised into appropriate time zones. In contrast to the message exchange scheme adopted by most other network protocols, nodes within MERLIN do not nominate a specific routing parent. Rather, MERLIN employs *multicast upstream* and *multicast downstream* transmissions to relay information both to the gateway and away from it. Exchange of synchronization packets and timezone updates occurs through periodic local broadcast. Asynchronous burst ACK[1] and negative burst ACK indicate successful reception and errors respectively. Controlled multi-path is employed as a means of reducing packet redundancy and ultimately transmission overhead. In MERLIN, each node locally stores a scheduling table that regulates its periodic activity.

3.3. Data traffic

MERLIN adopts a multicast transmission approach in communicating packets to adjacent time zones. In particular, it targets communication between nodes and gateways and vice versa. MERLIN supports two types of multicast: (1) upstream multicast towards the gateway; (2) downstream multicast away from the gateway. Furthermore, MERLIN provides a local broadcast data traffic for local exchange of information among neighbouring nodes (e.g. RSSI indicator, neighbor table, battery level, localization info etc.). In the case of downstream multicast and local broadcast, which are intended for all the nodes, the notification of at least one incorrect reception is sufficient to reschedule packet retransmission.

3.4. Scheduling

The purpose of the scheduling table is to enable time-slots allocation to network nodes in order to assign periods of node activity and inactivity. Scheduling allows the synchronizing of neighbouring nodes for transmission and reception. In MERLIN, *nodes in the same timezone use the same slot to transmit*. The scheduling table is usually transmitted by the gateway during the initialisation phase. Further work in [21] demonstrates how further tables can be opportunistically and dynamically injected into the network as application requirements dictate.

Fig. 1 shows the MERLIN scheduling table, referred to as V-table (due to the V-shape communication flow). The length of the table is equal to the length of one frametime and comprises four timez-

[1] A burst ACK, also referred to as burst tone, is a short signal that does not contain any encoded information.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

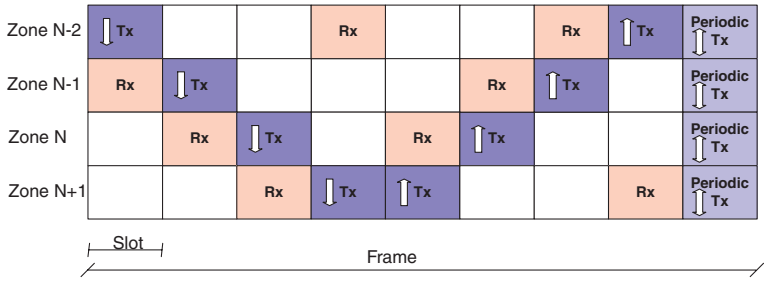

Fig. 1. The table of scheduling with periodic local broadcast.

ones. Each small rectangle represents a slot. When a timeslot is allocated for transmission to a particular timezone, the adjacent zone owns the slot for reception while nodes in further timezones are in sleeping mode. Note that the V-table performs fast upstream and downstream multicasts by forwarding a packet to four timezones towards the gateway or in the opposite direction within the same time frame. Appending the same table can allocate further zones, while flanking the same table provides the scheduling for further frames.

The 4-zone V-table allows potential parallel transmission of nodes that are located four zones apart. Appending the scheduling table for further zone transmission shows that an increase of the number of zones in a table results in fewer timezone parallel transmissions. Therefore, it is clear that the smaller the number of zones in a table results in higher number of parallel zone transmissions. Although the theoretical minimum size schedule is a 3-zone table, in this case the irregularity of the timezones due to random node locations causes significant collisions at the zone in between the two parallel transmissions. Our empirical results in [20] have proven that the 4-zone V-Table provides the highest number of parallel retransmission and therefore achieves the minimum number of collisions.

The last column of timeslots in the V-table is dedicated to local broadcast. In order to avoid inter-zone collisions due to simultaneous local broadcast by nodes the following formula applies:

$$\mathrm{Mod}(\mathrm{frameN},4) = \mathrm{Mod}(\mathrm{myZone},4)$$

where frameN is the frame counter (which is the ratio between the global time and the frame length) and it is the incremental number of frames that is cyclically repeated from 0 to 100; myZone is the node timezone. In this way nodes in the same timezone can contend for the slot for local broadcast only once each four time frames. Although not depicted in Fig. 1, it is intended that when a zone is scheduled for local broadcast, nodes within the same and adjacent zones are in listening mode.

3.5. Transmission mechanism

Within the MERLIN time slot structure (see Section 3.4), each time slot includes a *contention period* (CP), located at the beginning of the time slot, as illustrated in Fig. 2. We now consider the cases of the receiving and transmitting nodes in turn:

Receiver nodes: MERLIN adopts the *clear channel assessment* (CCA) mechanism through *low power listening* (LPL), an approach that is effectively utilized in BMAC [16] and deployed in TinyOS motes. With the LPL technique, the radio wakes up and samples the channel for a short time period of 4 ms [16] referred to as CCA period. In MERLIN, nodes scheduled for reception activate their radio for a CCA period in order to listen to the channel. If no channel activity is detected, the node goes into sleep mode; otherwise it receives the transmitted packet. The adoption of the CCA mechanism significantly reduces the idle listening time at the receiver.

Transmitter nodes: As shown in Fig. 2, a node that wishes to transmit initially chooses a random time within the contention period and wakes up at that time to sense the channel for a CCA period. If nothing is detected then the channel is assumed free of carriers and the node immediately commences transmitting the packet preamble, the duration of which is T_C equal to whole CP length. This guarantees that the preamble transmission reaches the end of the CP. The data packet is transmitted immediately after the preamble.

The random starting time allows asynchronous transmissions in MERLIN to notify multiple correct receptions by means of short *burstACK*, referred to as *BACK*. We stress the fact that such bursts do not carry any coded information and that

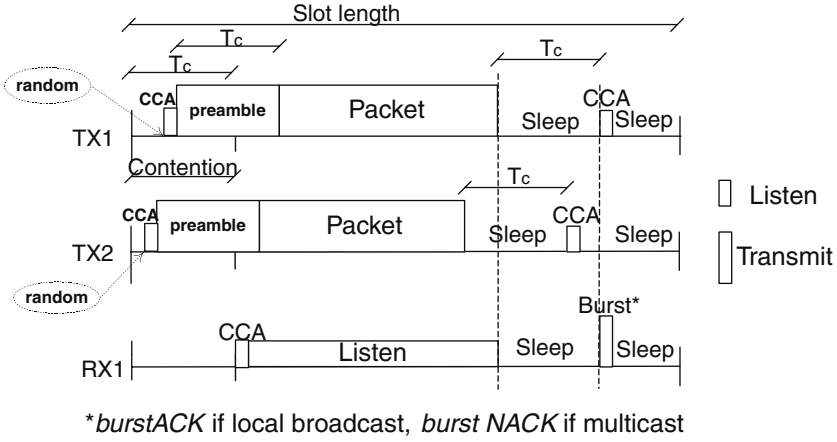

Fig. 2. Transmission mechanism for collision avoidance of MERLIN.

multiple overlapping bursts are identified at the receiver as a single burst. For *upstream multicast transmission, one burstACK is sufficient to notify the receiver of correct reception of at least one node in the zone closer to the gateway*. In fact, when multicasting upstream, the transmitter categorically does not need to know all nodes participating in the forwarding process. Rather, the notification of at least one correct reception is enough to consider a packet has been forwarded. The scheduling table ensures the packet has been transmitted in the correct direction. In the case of upstream multicast, only nodes in the neighbouring time zone that is closer to the gateway have their radio in listening mode due to the scheduling scheme. This is not applicable for both local broadcast and downstream multicast packets which carry information for all neighbours or from the gateway to the network respectively.

In the case of downstream multicast or local broadcast transmission, which occurs within dedicated slots in the scheduling table in Fig. 1, the burst tone identifies a reception error. This is referred to as *negative-burstACK* (*BNACK*). Because local broadcasts are intended for all neighbouring nodes, the transmitter reschedules the packet in case it receives a BNACK. A shortcoming of the BNACK mechanism is that a node that is momentarily unreachable (e.g. if a vehicle is passing by) might not be aware of a packet transmission hence it does not send back a BNACK. The problem is alleviated if the transmitter identifies packets with an incremental ID and make a backup of the last few sent packets. Packets can be uniquely identified by a combination of node ID and incremental packet ID. If a further transmission occurs, the receiver can identify the missing packets by a mismatch in the incremental ID number, and subsequently requests a copy of these. The packet recovery procedure described, which is applied for downstream and local broadcast, results in an increase of transmission delay and the number of backups is limited by the memory constraints of the device.

In essence, there is no difference in the signal transmitted by the two burst tones. BACK and BNACK can be simply identified by means of the slot in which the transmission occurs. BACKs are transmitted in both slots dedicated to upstream multicast while BNACKs are transmitted in downstream multicast and local broadcast. In both multicast and local broadcast, if an error is detected, the packet will be rescheduled after a random exponential back-off procedure. The burst tone is usually provided in standard WSN radios. So as not to interfere with other ongoing transmission, the burst transmission is delayed by exactly `Tc` after the end of the maximum packet length allowed. A primary consequence of this is the elimination of the possibility of accidentally corrupting nearby transmissions. In addition, the burst transmission is asynchronous, thus ensuring that the transmitter can return to the sleep state after the transmission is complete, and can wake-up after `Tc` to detect either a BACK or BNACK in response to its transmitted packet.

3.6. Initialisation

Nodes start the network initialisation phase by listening for a SYNC packet that contains *timing information*, *sender ID* and *sender timezone*. Gate-

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

ways, that are synchronized to the same time reference, start initialising the network by broadcasting SYNC packets. Gateways set their timezones to 0. Sensor nodes in the vicinity of the gateway that receive the SYNC packet will use it to synchronize their internal clock. As they are one hop away from the gateway, these nodes set their timezone value to 1. As depicted in Fig. 3, timezone one nodes will then forward the SYNC packet to more distant nodes. The transmission mechanism for collision avoidance described in Section 3.5 ensure proper forwarding activity. All nodes receiving SYNC packets from nodes in timezone 1 set their timezone to 2. This procedure is repeated until all nodes have set their timezones. In the case of multiple gateways, the gateways start flooding the network simultaneously by sending a SYNC packet to neighbouring nodes. Upon receiving SYNC packets from two different gateways, nodes can compute their timezones to each gateway and select their timezone with respect to the closest gateway. At the end of the initialisation phase, *the timezone number is equal to the number of minimum hops a packet needs to reach the closest gateway*.

During the initialisation, it is possible that a node is momentarily classified in a higher timezone. For example, should a node in a timezone N fail to access the channel, or if its packet collides, the node re-applies the channel contention procedure in the successive frame. In the meantime, any node waiting for initialisation can receive a SYNC packet through a third party, causing it to temporally select a higher timezone. Such a transient timezone assignment does not prevent the node from communicating with neighbouring nodes, though this may result in a longer route to the gateway. This situation is rectified when the node receives the timezone N SYNC packet resulting in a path with a fewer number of hops to the gateway. In the case of nodes being equidistant from two gateways the node will select a timezone according to the first SYNC received. The total initialisation time depends on the scheduling table in use. In particular, previous studies on the performance of the scheduling of MERLIN [20] showed that the V-table, described in Section 3.4, enabled the initialisation of 14 timezones in a network in approximately 9 s.

3.7. Synchronization

The poor hardware sophistication of sensor nodes requires repeated gateway synchronization. In addition to synchronizing the starting moment (offset) of the slots among nodes, receivers must continually compensate for the frequency skew of each node's individual clock. Both of these issues are solved by including timestamps in each transmitted packet. All receiver nodes can then estimate

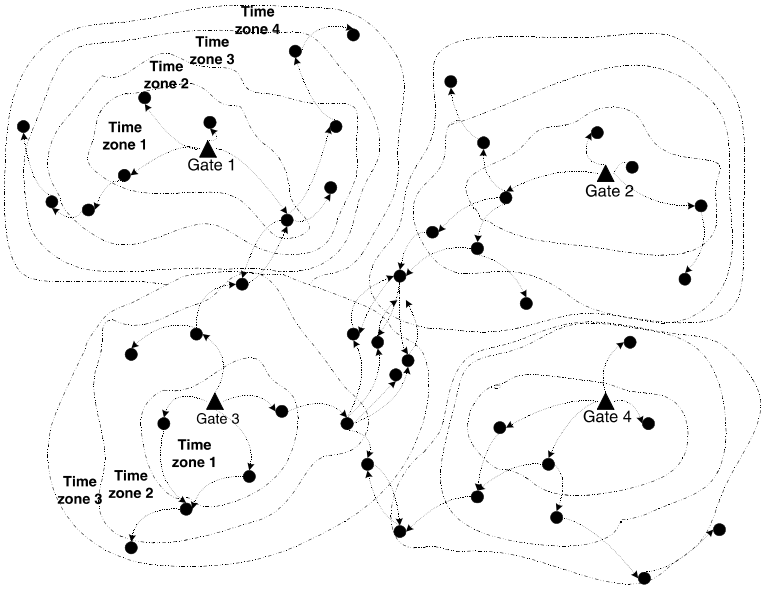

Fig. 3. Division of the network in timezones following the simultaneous network flooding by gateways of a SYNC packet. Every node sets its zone and forward the packet to more distant nodes.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

the start of the slot according to the sender, and synchronize their clock offset. Furthermore, a node can calculate its clock skew by comparing over time the observed difference between a sender's clock and its own clock. In MERLIN, *nodes update their clock synchronization from the nodes belonging to the lower timezone* that are closer to the gateway. In fact, such nodes hold a finer synchronization, through similar reasoning to the stratums used in NTP [14]. With respect to scalability, new nodes can join the network by simply listening to any packet. The packet information contained such as sender timezone, packet transmission time and type (e.g. upstream multicast) allow the new node estimating its position and synchronization. The node is then in position to join the network.

If a new gateway joins the network, it firstly joins as a new node. Then, it announces its presence through local broadcast of a SYNC packet. Nodes compare their old timezone number with the new one and forward the SYNC packet. In case the new gateway is closer, a node changes its timezone at the beginning of the successive frame only after the transmission of the SYNC to has been successful. This avoid temporary network disruption caused by neighbouring nodes synchronized to different gateways.

3.8. Packet format

In MERLIN a packet represents a collection of messages that are assembled when a node is required to forward a number of messages received from other nodes. A node generates a message containing protocol information such as sourceID, destID, forwardID, forwardZone, msgID, msgType, and, of course, the DATA payload. All messages are uniquely identified by the msgID, which is a combination of sourceID and private message number ID.

In the process of forwarding packets to the gateway, a node can receive messages from several nodes which are to be forwarded to nodes on lower time zones. Since sensor data is usually only a few bytes, for example, temperature, pressure, chemical data and so on, MERLIN concatenates the messages and then transmits them as a single packet. Thus, a node that has more than one message to send, aggregates and sends the messages in one packet during the same time slot, which both saves energy and reduces transmission overhead. Several factors such as application type, transceiver data rate, and channel condition dictate the maximum number of messages in a packet. For example, small packet size are favourable in lossy wireless channels. In the MERLIN standard configuration, the packet size does not exceed 100 bytes (maximum packet length) although the simulations consider a larger range of packet sizes were considered.

In order to facilitate the concatenation process, MERLIN provides three small buffers for *local broadcast, upstream multicast and downstream multicast*. Such buffers are used to temporarily store a node's own messages as well as messages received from other nodes for transmission. Every time a node receives a packet of a certain type, it compares all messages contained within it to those in the appropriate buffer and deletes any duplicate messages. The buffer is then updated with the new set of messages. Forming packets for transmission follows the FIFO principle. The transmission process of the different type of packets is regulated by the scheduling table, described in Section 3.4.

3.9. Routing characteristics

The division in timezones together with the scheduling and type of data traffic allow packets to be routed to and away from the closest gateway. Recall that MERLIN does not address a specific forwarding node. This may cause duplication of packets during forwarding activity. However, the packet generation is controlled through a mechanism of *overhearing*. This simple but effective routing is further improved by way of a mechanism of on demand zone maintenance.

3.9.1. Controlled multi-path

Messages can be concatenated to form a packet (see Section 3.8). When a packet is formed, a msg-index is generated. The msg-index contains all msgIDs of messages within the packet. The msg-index, or the msgID in the case of a single message, is located at the beginning of the packet, thus allowing neighbouring nodes of the same timezone overhear message transmissions. A start frame delimiter can allow identifying the beginning of the msg-index. On identifying messages in its buffer that have already been transmitted by a neighbouring node in the same time zone, a node immediately deletes these messages from the buffer as shown in Fig. 4. This mechanism is used for multicast upstream communications as it is only necessary that one instance of the message reaches the gateway. Down-

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

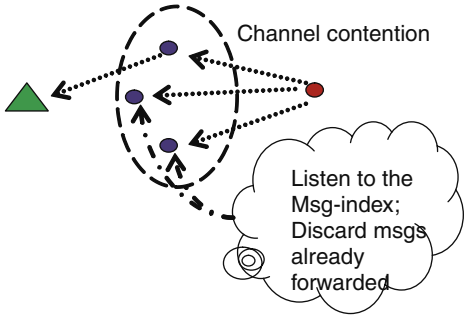

Fig. 4. The controlled multi-path mechanism through neighbouring nodes overhearing.

stream transmissions do not apply this mechanism, as deleting a message causes some nodes not to receive communication from the gateway, such as periodic network updates.

3.9.2. Timezone maintenance

Should a node in zone N not receive the periodic timezone update from any neighbouring node in zone $N-1$, it transmits a timezone update request (TUR) through upstream multicast. In case of no answer, the node will assume that the connection with zone $N-1$ has failed. The node then tries to re-establish connection with any node within the same timezone through local broadcast of TUR. If the node receives at least a burstACK, it changes its timezone to $N+1$. Otherwise, the node assumes a failed connection also with nodes within the same zone. As a result, it tries to re-establish a connection through a downstream multicast transmission of TUR. At this stage, a reception of a burstACK means a change of timezone to $N+2$. For all cases, the node repeats a TUR transmission twice before assuming a failed route connection.

4. Analytical model of transmission in MERLIN

An important aspect to study is the number of packets generated by a multicast transmission in a multihop environment. In MERLIN, a node transmits packets to the gateway without specifying a forwarder in the lower timezone. Multiple copies of the same messages are then reduced through the controlled multi-path mechanism that allows nodes within the same timezone to overhear then discard messages transmitted by neighbouring nodes. In this section we use an ideal mathematical model in order to study the impact of the MERLIN transmission with respect to the location of a node source and the gateway destination. In order to understand the behaviour of the protocol, we calculate the upper bounds of the total number of duplicated packets in an ideal environment thereafter implementing the model within Matlab. The results can be generalized to any transmission mechanism that employs a directed broadcast and a message overhearing mechanism.

4.1. Model assumptions

Suppose a certain area is densely populated with sensor nodes and among them a destination D. In mathematical modelling for communication, a transmitter/receiver node is a point while, an omni directional transmission in $2D$ is a circle of transmission radius r centred at the transmitter. Therefore, at the end of the MERLIN initialisation, timezones $1,2,3,\ldots,n$ divide the network area into concentric annuli $A_1, A_2, \ldots, A_n$ such that $A_i \cap A_j = 0$ $\forall i,j = 1,\ldots,n$. As shown in Fig. 5, ideal timezones are modelled as concentric annuli centred at the destination D and node transmissions are modelled as circles centred at the source node (s) or forwarding node (f).

Given a certain timezone i, the corresponding ith annulus has inner radius of $(i-1)*r$ and outer radius $i*r$. The circle A_1 is an annulus of inner radius equal to 0 and outer radius equal to r.

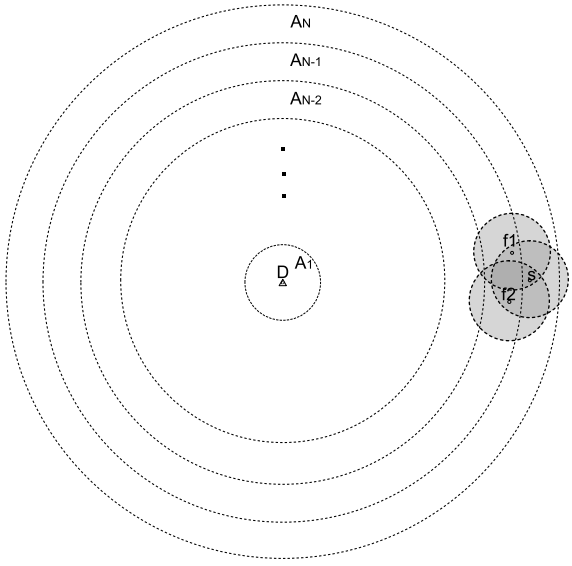

Fig. 5. In an ideal environment timezones are modeled as concentric annuli centred at the gateway and transmission as circles.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

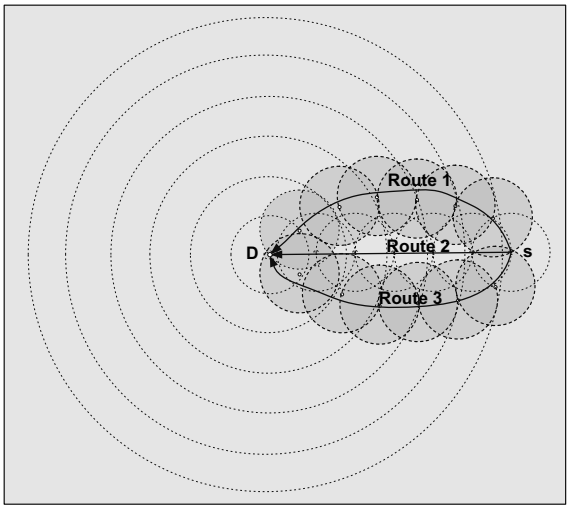

Fig. 6. Multiple routes generated by the forwarding activity of MERLIN from source s to destination D.

Consider a node source s, which has a packet to transmit, located at some point in timezone N. An upstream packet transmission of s is received within a circle c centred at s of radius r.[2] The intersection of s and the annulus A_{N-1} creates a region $I_{N-1} = c \cap A_{N-1}$. The region I_{N-1} contains all potential forwarding nodes of s's packet for the annulus A_{N-1}. Potential forwarders in region I_{N-1} will compete to forward the packet resulting in m number of re-transmissions to the annulus A_{N-2}. In MERLIN, a node transmits a packet only if no other nodes within its transmission radius r transmits the same packet. This limits the number of forwarding nodes in the region I_{N-1}. As a result, the distance between any two forwarding nodes must be greater than r. Each forwarding node α covers a transmission circle c^{α}_{N-1} that intersects the lower annulus A_{N-2} in at least one point.[3] The union of such intersections originate the region I_{N-2}, defined by the following expression:

$$I_{N-2} = \left(\bigcup_{\alpha=1}^{m} c^{(\alpha)}_{N-1} \right) \cap A_{N-1} \quad (1)$$

[2] As MERLIN considers nodes and gateway transmission power to be equal.

[3] The distance between the inner and the outer radius of an annulus is equal to the node transmission radius. This ensures that a transmission can always reach the lower annulus. The same is achieved in real scenarios where a node places itself in timezone n only if it receives and acknowledges correctly from and to timezone $n-1$.

where next potential forwarders are located in region I_{N-2}. The process of forwarding a packet from s in the Nth annulus to D can be characterized by a set of regions $I_{N-1}, I_{N-2}, \ldots, I_1$.

4.2. Bounds on the number of packet re-transmissions

In MERLIN the forwarding activity can generate multiple copies of the same packet travelling along different routes from s to D as shown in Fig. 6. In a plane densely populated with nodes, the number of forwarding nodes of the jth annulus is somehow proportional to the size of the region I_j. Therefore, the first step in computing the upper bound of packet re-transmissions is to identify the maximum size for all I. It is important to notice that the portion of I_{j-1} covered by a single forwarding node f located in annulus A_j is larger when f is nearer to the annulus A_{j-1}. A transmission of f that is located at the outer edge of annulus A_{j-1} yields the largest portion of I_{j-2} $\forall j = 1, \ldots, N-1$. This corresponds to a maximum number of potential forwarding nodes and thus packet transmissions in annulus A_{j-1}. If we draw a circle C_1 of radius r centred at s and then draw all possible circles of radius r with centres on the circumference of C_1, then the (exterior) envelope of all these circles will itself be a circle C_2 of radius $2r$ centred at s, see Fig. 7. The intersection $I^{\max}_{N-1} = C_2 \cap A_{N-1}$ constitutes the largest possible region that can be covered in the annulus A_{N-1} by a given source s. The same process can be repeated for all the annuli and generalised as follows:

$$I^{\max}_{\gamma} = C_{\delta} \cap A_{\gamma} \quad (2)$$

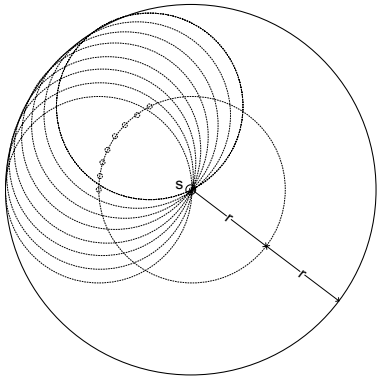

Fig. 7. The exterior envelope of all circles centred at a circumference of a circle is itself a circle.

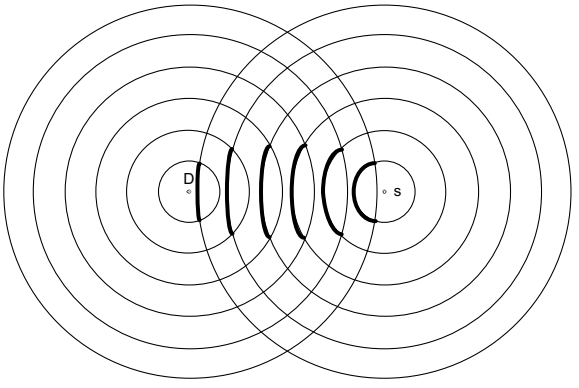

Fig. 8. The edges of the forwarding regions I. Note that the longest edges are obtained when the source is located at the border with the lower annulus.

where $\gamma = 1, 2, \ldots, N-1$ and $\delta = N-1, N-2, \ldots, 1$. See Fig. 9.

Each $I_\gamma^{\max}$ represents the largest region in A_γ covered by the process of forwarding from s to D. Furthermore, the location of s within the annulus A_N has a large impact on the successive set of regions $I_\gamma^{\max}$. Clearly, *the greatest set* I_γ^{MAX}, $\gamma = 1, 2, \ldots, N-1$ *is obtained when s is located at the edge with* A_{j-1}. The second step demands the calculation of the number of non-overlapping packet retransmissions for each I_γ^{MAX}. In general, a couple of non-overlapping transmissions occur when the forwarding nodes are at a distance greater than r between each other. Recall that the region $I_{j-1}^{\max}$ is obtained by taking into account forwarding nodes at the edge of the region $I_j^{\max}$. Thus, we can utilize the length $l_j^{\max}$ of each arc to calculate the maximum number the number of non-overlapping packet retransmissions as follows:

$$T^{\max} = 1 + \sum_{j=1}^{N-1} \lceil (r * l_j^{\max}) / (r * \pi/3) \rceil = 1 + \sum_{j=1}^{N-1} \lceil 3 * l_j^{\max} / \pi \rceil \quad (3)$$

where (1) $\lceil a \rceil$ denotes the smallest integer $\geqslant a$; (2) $l_j^{\max}$ is the length of the arc in Fig. 8 corresponding to each region $I_j^{\max}$; (3) the denominator $r * \pi/3$ represents the maximum possible arc length in radians between two nodes that can overhear each other, as shown in Fig. 11; (4) the number 1 added represents the initial transmission of s.

In particular for a source s located at the edge with A_{N-1}, which is the worst case location, the upper bound of packet retransmissions from s in annulus A_{N-1} is

$$T^{\mathrm{MAX}} = 1 + \sum_{j=1}^{N-1} \lceil 3 * l_j^{\mathrm{MAX}} / \pi \rceil \quad (4)$$

Clearly, *the minimum number of packet retransmissions needed to forward a packet from s in annulus* A_N *to D is equal to N*. This represents the lower bound of a packet retransmissions.

4.3. Analytical results

This section presents the analytical results, obtained with Matlab, when computing the upper bound and the average number of packet transmissions in MERLIN on the basis of the preceding discussion. In order to compute the absolute upper bound on the number of packet transmissions, the analysis assumes that transmissions at each annulus

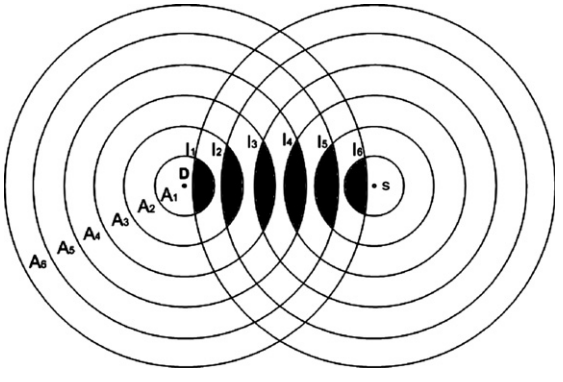

Fig. 9. The I regions are obtained intersecting the annuli and the envelopes relative to each set of forwarding nodes.

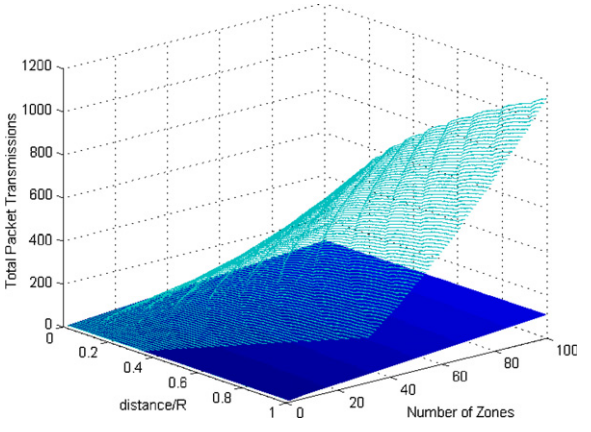

Fig. 10. Total packet transmissions in the worst case and an average case, plotted against the number of zones and d.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

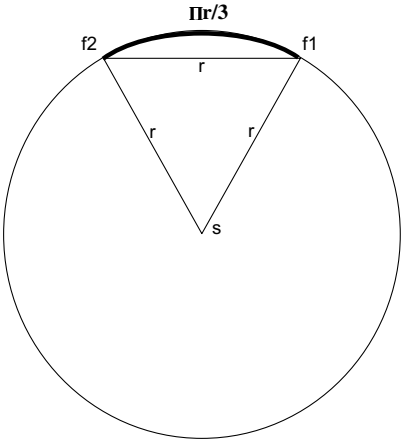

Fig. 11. The maximum length of the arc that allows the overhearing mechanism working for $f1$ and $f2$ is $r * \pi/3$.

j (A_j) cover the largest possible I_{j-1}^{MAX} of annulus $j-1$. In other words, packet forwarders in each region j (I_j) are always located on the edge of I_j. Having determined I_j^{MAX}, the algorithm then computes T_j^{max} for region I_j^{MAX}. Repeating the process for all annuli yields the number of transmissions for each zone.

Since the probability of the occurrence of the worst case scenario is rather low, we also compute the case when the forwarders in I_j are located in the middle of I_j. In order to determine the average case at each annulus, the algorithm first obtains the point G at the *centre of gravity of region* I_j. Projecting a circle centred at the transmitter with a radius equal to the distance between G and the transmitter in annulus $j+1$ yields I_j^{AVG} that represents the coverage region in annulus j in this average case. The process is repeated for subsequent zones by always taking the point at the centre of gravity of zone I_j^{AVG} to define the transmission coverage area in zone j.

The analysis investigates the effect of the location of the source node within its zone on the number of packet transmissions. For this purpose, we define d as *the distance of the source node in annulus j from the outer boundary of annulus j, normalized to r* that corresponds to the distance between the inner and outer of an annulus. The worst case scenario for the source location is when d is almost equal to r, placing the source at the edge of annulus $j-1$. The analytical simulations vary d between 0.001 (best location of s) and 0.99 (worst location of s), together with the number of zones ranging between 2 and 100 zones, yielding 98010 possible scenarios.

Fig. 10 plots the total packet transmissions for the worst case and average case versus d and the number of zones. We recall that the average case is obtained by taking forwarders that are at the centre of gravity of each intersecting regions. The transparent surface in Fig. 10 represents the worst case T^{MAX} and the darker solid surface represents the average case. Initial observation of these results clearly shows a large disparity between the worst case and the average case. In the average case the total number of transmissions for 100 zones and $d = 0.99$ (worst location of s within the annulus) remains at 104 transmissions, while the worst retransmission case is almost 11 times higher. The total transmissions in the average case almost achieves the lower bound of transmissions, which is 100 transmissions in the case of 100 zones. Values of d larger than 0.4 cause a significant increase in the upper bound on transmissions for networks with few zones. The upper bound on transmissions for larger networks is noticeably higher than the average case, although the difference is more pronounced for larger values of d.

Fig. 13 illustrates the number of packet transmissions at each annulus for the worst and average cases in four scenarios with 10, 20, 60, and 100 zones, taking $d = 0.99$. The number of transmissions in the worst case follows a bell-shaped curve with the maximum number of transmissions at the annuli in the middle, before converging for zones closer to the gateway. In the average case, the number of transmissions always converges to 1 after the packet has traversed five hops from the source. By definition, the average case always defines the

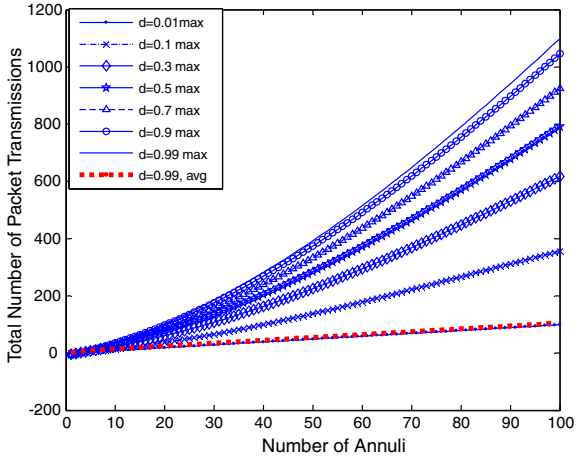

Fig. 12. The impact of d on the total number of transmissions.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

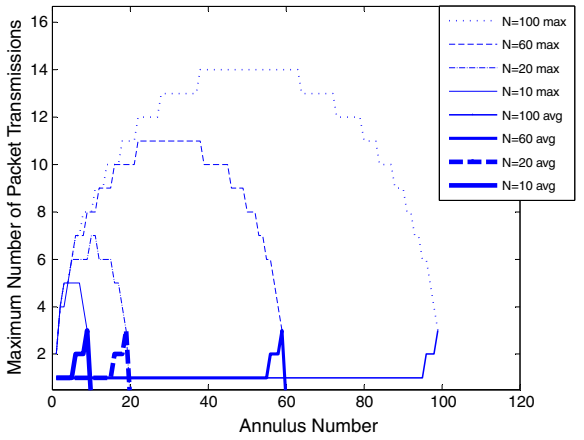

Fig. 13. Worst and average cases of total packet transmissions for $d = 0.99$.

coverage region I_j^{AVG} in annulus j by first determining the point G that splits the area of I_j in half. At every step in the retransmission process, the size of area I_j^{AVG} and the possible number of forwarders shrink until only one packet forwarder in I_j^{AVG} is possible. The results in Fig. 13 for the average case further explain the proximity between the total number of transmissions for the average case and the best case in Fig. 10. An interesting aspect is that the route convergence causes a decrease of number of retransmissions therefore reducing the load on nodes in proximity of D.

Finally, Fig. 12 investigates the impact of d on the total number of packet transmissions by taking a cross-section of the results in Fig. 10. Fig. 12 plots the upper bound on the total number of transmissions versus the number of annuli, for seven values of d: 0.01, 0.1, 0.3, 0.5, 0.7, 0.9 and 0.99. It also plots the average case starting with a source having d equal to 0.99, which is the worst case location for the source. In the case of the upper bound, the effect of d is most evident for networks with a large number of zones, as small increases in d, corresponding to a source that is slightly closer to the lower zone, cause large increases in the total number of packet transmissions. A source located at the inner boundary of an annulus has almost the same total number of transmissions in the average case as a source located at the outer border of the annulus in the worst case. The small number of packet transmissions for all d in the average is once again attributable to the shrinking of I_j^{AVG} as j decreases, eventually result in a single forwarder in each subsequent annulus.

The described analytical model of MERLIN shows that the protocol can effectively reduce the number of packet retransmissions in ideal conditions without the need of addressing the receiving node. The benefits are obtained from the MERLIN timezone formation together with an appropriate scheduling policy and the overhearing mechanisms. However, we stress the fact that these results can be generalized to any transmission mechanism that employs a similar directed broadcast and a message overhearing mechanism. Potential benefits of MERLIN are thereafter assessed in a more realistic network simulation environment against related architectures.

5. Simulation

All simulations of MERLIN have been conducted within the OmNet++ [17] discrete event simulator. The sensor network framework of OmNet++ is based upon earlier work on the codesign template [23], used in the EU EYES project [17], and slightly modified to obtain performance metric values for this assessment. This section describes comprehensively the performance metrics, scenarios and parameters used for the assessment in order that other researchers may replicate the experiments if they so choose.

5.1. Scenarios and performance metrics

The metrics for the assessment of MERLIN against both SMAC and SMAC + ESR are the following:

Energy consumption per message received: the amount of energy required to relay a message from source to destination.

Network lifetime: the operative network life expectancy under network parameters described in Section 5.2. We define network lifetime as the time at which 30% of nodes deplete their batteries, which is likely to cause network partitions with a consequent disruption of normal network activity.

End-to-end latency: the time that elapses between the source node transmission and the destination node reception, also referred to as end-to-end (E2E) latency.

Total message overhead: the total number of messages generated by the network activity including control messages and duplication of data messages due to multiple path generation.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

Percentage of sleeping time: the percentage of node sleeping time calculated relative to the total node activity time.

Node depletion is closely related to node duty cycle, i.e. the ratio between the listen and sleeping activities in a single time period. Hence, the assessment shows both network lifetime and E2E latency of packets relative to the node duty cycle. In MERLIN, the duty cycle is tuned by variating the length of the frametime. In particular, the duty cycle depends on the number and length of CCAs in a frametime. For the V-table this is equal to $[4T_{cca} + (1/4)T_{cca}]/\text{Frametime} = 4.25/\text{Frametime}$. Node energy consumption has been calculated by computing the duration of sleeping, transmitting, receiving and switching between states. As demonstrated in [1], the switching energy becomes a significant source of energy consumption for low duty cycle typically 10% or less.

The performance evaluation of MERLIN considers two network scenarios: a five node two-hop scenario and a random multihop scenario. For consistency and fairness in the evaluation of MERLIN against SMAC, scenarios and setup parameters are the same as those used in the assessment of SMAC in [25]. For multihop scenarios in particular, SMAC uses the ESR protocol to relay data to the gateway. Furthermore, protocol scalability has been evaluated under both high and low conditions of data traffic and network density.

5.2. Simulation setup

The simulation setup was configured using parameters from the Transceiver Tr1001 data sheet in Table 1, as well as with measurements of the switching energy [1] of the EYES node [11]. All experiments, except the transmission overhead experiments, used a data message of 16 bytes, a typical size of sensor data payload that carries, for example, sensed temperature data. However, for the total packet overhead, the simulations show the behaviour for a wider range of packet size for a number of simulation of 10 min each. In addition, the data generation rate was varied between a low data traffic scenario (12 messages per minutes) and a high traffic scenario (60 messages per minute). In line with the initial SMAC assessment in [25], the SMAC duty cycle is obtained by dividing the listen period by the sleep period. The SMAC periodic listen time has been kept constant at 80 ms in order that node duty cycle is modified by changing the SMAC sleeping period length. In MERLIN, a similar duty cycle can be achieved by dividing the CCA length by the frametime with the observation that a node wakes up twice in a single frametime to transmit upstream and downstream, and once every four frames for local broadcast. Therefore, similar duty cycle configurations could be achieved by changing the frametime length. Both protocols schedule a synchronization period every 13 s, by means of local broadcasts for SMAC and gateway flooding for MERLIN, so as to compensate for node clock skew and offset. With respect to the ESR, we kept the same setup used in the earlier evaluation of the protocol against DSR [9], as described in [24].

Table 1
Data sheet of the EYES prototype transceiver and direct measurements of switching parameters

Current (mA)			Power (mW)		
Sl	Rx	Tx	Sl	Rx	Tx
0.005	4.8	12	0.015	14.4	21
Switching time (μs)					
Sl to Rx	Sl to Tx	Rx to Sl	Tx to Sl	Rx to Tx	Tx to Rx
700	700	10	10	700	700
Switching energy (μJ)					
8.82	25.2	0.116	2.83	25.2	8.85

The five nodes two-hop scenario in Fig. 15 is the same topology utilised within the SMAC assessment [25], and it consists of two source nodes, one forwarding node and two gateways. The chosen signal strength causes contention for both source nodes to transmit as well as for both gateways to send an acknowledgment. The forwarder is within the transmitting range of both sources and gateways. Measurements have been conducted of the energy consumption per message forwarded to the gateway on the node forwarder as this constitutes the node most subjected to node activity and collisions. Message transmission starts after an initialisation time of 20 s and ends when 100 packets are successfully forwarded to one of the gateways.

The multi-hop scenario consists of 70 nodes randomly placed in the network. Random scenarios have been obtained by using 10 independent seeds for each run. To randomly select the source nodes for generating data, 10 seeds were used with different values than the ones for scenario generation. Periodically, five randomly selected nodes report a data packet to the gateway. The displayed results represent the average of all the simulation runs. Ini-

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

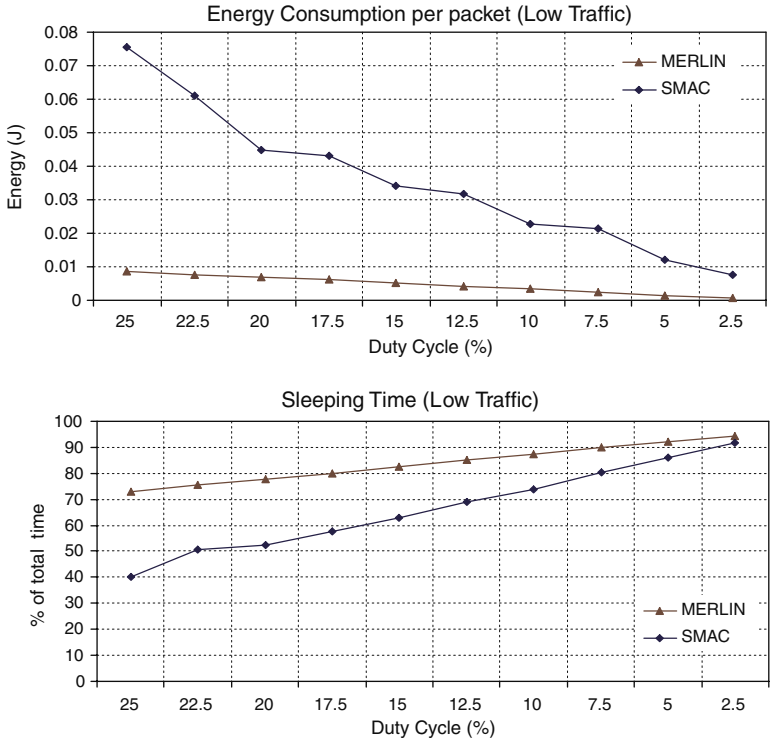

Fig. 14. Energy consumption and sleeping time in low data traffic condition for two-hop scenario.

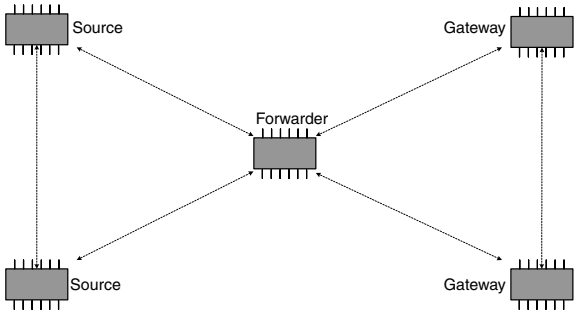

Fig. 15. Local scenario of five nodes, two sources, one forwarder, two destinations.

tial node battery energy for the performance evaluation, for the purposes of computing network lifetime, is 2 J and the battery model assumes linear energy depletion. In order to evaluate the scalability of MERLIN, the multi-hop scenario comprises high and low node densities, obtained by keeping the transmitting radius constant and selecting two different network sizes of $400 * 300\,m^2$ and $600 * 500\,m^2$ respectively. Recall that experiments have been conducted under four combinations of the two high and low data traffic scenarios and the two high and low network density scenarios.

5.3. Simulation results

Two-hop scenario: The graphs in Fig. 14 show both the energy consumption and sleeping percentage per message forwarded to the gateway for node two under low data traffic conditions with respect to the node duty cycle. Graphs show a quasi-linear behaviour of energy consumption that decreases for lower duty cycles. The results show a better performance of MERLIN for the entire range of duty cycles studied. As duty cycle percentage increases, so the performance difference between the two protocols diverges. The node in SMAC wakes up more often to listen to the whole SYNC and RTS periods, which causes a large increase in idle listening time at the receiver. The result is confirmed by the graph of the node sleeping time percentage in Figs. 14 and 18.

The comparison of energy consumption in SMAC for the low traffic graphs in Fig. 14 and the high data traffic graphs in Fig. 18 shows that the low data traffic conditions lead to an even higher energy consumption, when contrasted with high traffic conditions. The reason behind this is that energy consumption is calculated over the timespan that is required to receive 100 message correctly. Therefore total simulation time is much longer for

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

low data traffic than for high data traffic. One of the limitations of the SMAC protocol is that node idle listening dominates energy consumption. In contrast, the combination of scheduling and a short CCA effectively minimises node idle listening in MERLIN. Both traffic settings present large energy savings in MERLIN relative to SMAC. Energy savings increase for higher duty cycles, particularly in low data traffic scenarios. The primary reason for the larger energy savings is the high idle listening time of SMAC that is more pronounced when a node wakes up more frequently and when data traffic is light. The graphs of node sleeping percentage in Figs. 14 and 18 confirm this characteristic.

Multi-hop scenario: Fig. 16 shows the behaviour of SMAC and MERLIN in terms of network lifetime as the duty cycle varies between 2 and 20, for the scenario described in Section 5.1. Initial observation of the graphs reveals a crossover point of the two protocols studied at a certain value of duty cycle. In particular, MERLIN is more energy-efficient than SMAC beyond the crossover point for smaller values of node duty cycles. However, such graphs have a limited relevance when considered independently of E2E latency. The energy consumption relates closely to the message delay from source to destination. For example, applications that sustain a delay of n seconds present a very different network lifetime profile to applications that sustain a delay of $2n$ seconds even if the same protocol is used. To this end, graphs in Fig. 17 illustrate the behaviour of both maximum and average E2E latency of message for MERLIN, SMAC and SMAC + ESR. In particular, the maximum E2E latency is due to communication from those nodes most distant from the gateway, typically 11 or 12 hops. It should be noted that the highest threshold of latency of SMAC + ESR was set at 40 s for the simulations of the maximum latency. Therefore, in the case of 2% duty cycle, the maximum latency could not be depicted.

Observation of the graphs also indicates the following:

(1) In terms of maximum E2E latency, MERLIN outperforms SMAC between 1.5 and 3 times and SMAC + ESR between 3.7 and 5 times.
(2) In terms of average E2E latency, MERLIN outperforms SMAC by about 20% but performance tends to converge for lower duty cycles; MERLIN outperforms SMAC + ESR by approximately a factor of 4 to 5.
(3) The simulation results demonstrate that reducing the duty cycle causes an approximate exponential increase of E2E latency for SMAC + ESR.

Joint analysis of the energy consumption and E2E latency results leads to an alternative interpretation. If an application sustains a 10 s or higher

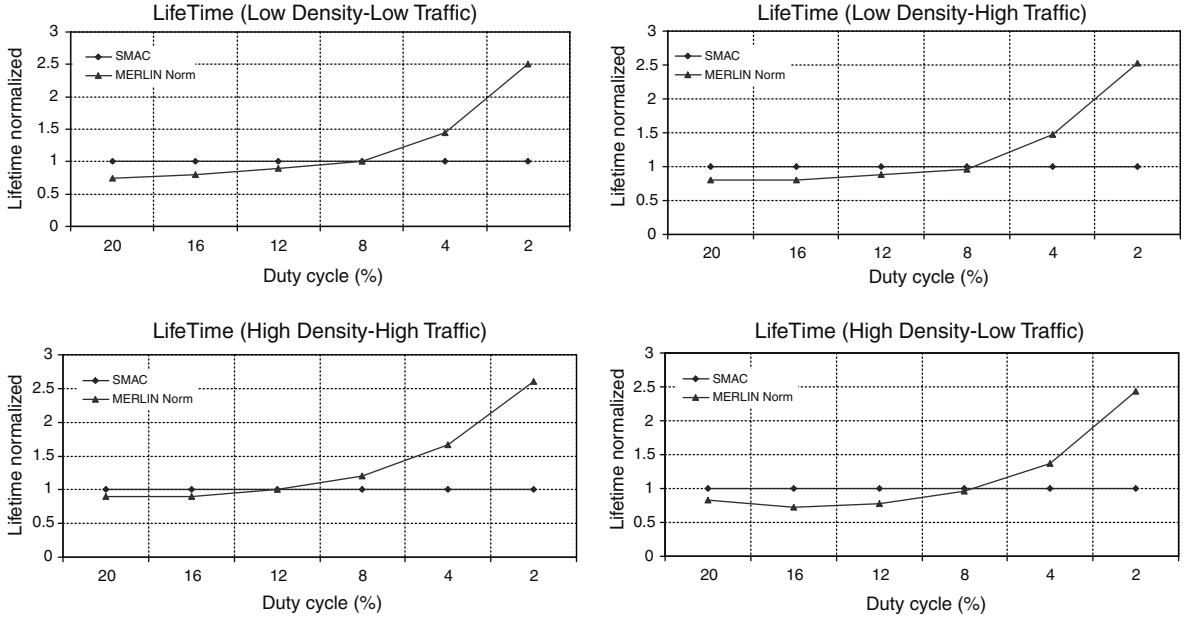

Fig. 16. Comparison of network lifetime for different combination of node density and data traffic.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

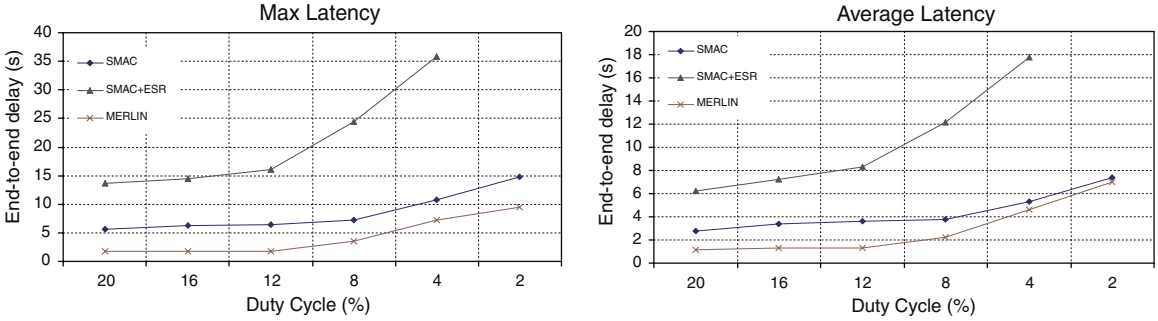

Fig. 17. Comparison of E2E latency for MERLIN, SMAC and SMAC + ESR in the multihop scenario.

threshold of maximum E2E latency, MERLIN can use a 2% duty cycle, resulting in an extension of the network longevity some 2.5 times higher than SMAC. In the case of a sustained maximum threshold of E2E latency less than 10 s, the SMAC + ESR protocol couple is unsuitable. In contrast, MERLIN can still be used up to an E2E latency of 1.8 s, by increasing the duty cycle. As a result, setting SMAC and MERLIN at their optimum according to certain maximum E2E application delay requirements result in a longer network lifetime for MERLIN.

We conclude our assessment by looking at the total overhead in transmission of MERLIN as compared to SMAC + ESR. In particular, we evaluated the total message overhead of the combination of SMAC for channel access and ESR for routing against the access to the channel and controlled multiple path generation of MERLIN. As described in Section 5.2, the total overhead has been evaluated

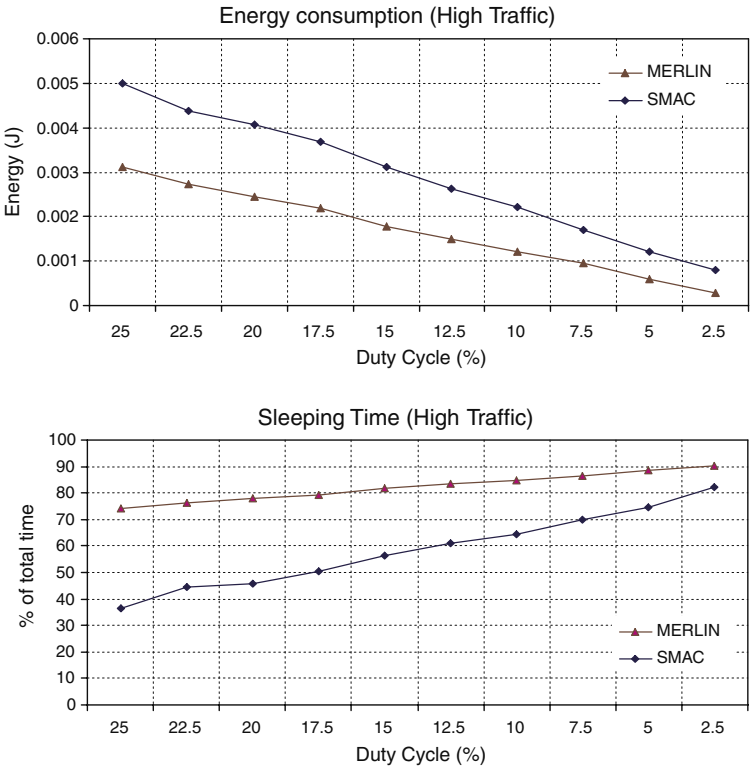

Fig. 18. Energy consumption and sleeping time in high data traffic condition for two-hop scenario.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

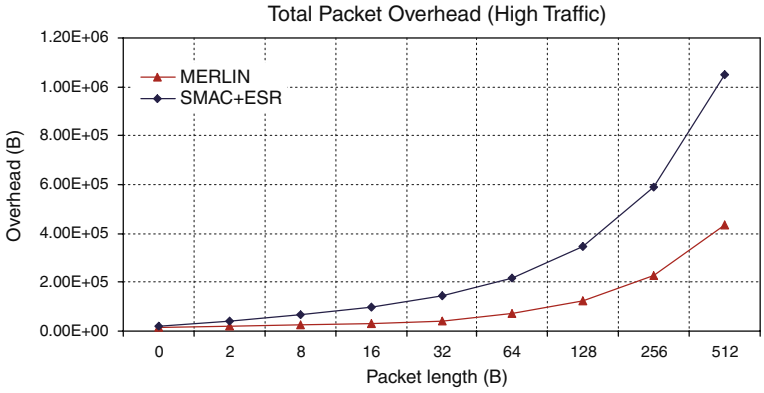

Fig. 19. Comparison of total transmission overhead for MERLIN and SMAC + ESR.

in 10 random multihop topologies for message lengths between 2 and 512 bytes, in conditions of high data traffic and high density that represents the worst case scenario.

The SMAC + ESR combination necessitates a large amount of control messages such as RTS, CTS, periodic SYNC and periodic route maintenance packets which deliver a significant impact on the total transmission overhead. The situation can be improved by working on their coordination, for example by merging the periodic SYNC and route maintenance into one packet. This is not an easy task due to the strict layering of traditional protocols, where the protocol at each layer have specific and often conflicting requirements. In MERLIN, the integration of MAC and routing specifically allows the reduction in the number of control packets. This in turn compensates for the duplicate packets caused by multiple paths that are generated by forwarding activity. Fig. 19 shows the large savings of packets transmitted by MERLIN relative to the SMAC + ESR couple for the whole range of considered packet sizes.

6. Conclusion

Traditionally, protocols for wireless sensor networks tolerate high packet latency in order to achieve energy savings. Such a delay is expected to increase when a combination of energy-efficient MAC and routing are jointly commissioned, rendering these protocols unsuitable for many WSN applications. In this paper we have presented a comprehensive description, analysis and evaluation of MERLIN as an integrated architecture for sensor networks. MERLIN is suitable for communication to and from gateways and nodes, rather than for communication between nodes as peers located several hops apart. MERLIN has been analytically modeled and bounds of packet transmission have been drawn. Analytical results showed an interesting route convergence behaviour which caused a decrease of number of retransmissions especially in proximity of the gateway. The protocol has been assessed against the combination of Sensor-MAC and ESR routing. The evaluation has shown how in multihop node/gateway communication the MERLIN integrated architecture delivers both significant energy savings and latency reduction relative to a wide range of node duty cycles. Finally, this paper demonstrated that a duty cycle of 10% or less for SMAC + ESR incurs an end-to-end delay of several tens of seconds. Furthermore, a node duty-cycle of 4% or lower prevented us from getting acceptable source-to-sink data delivery. In contrast, MERLIN remains stable under a 10-s delay up to a 2% duty cycle. The energy consumption profile of MERLIN results in an extension of network lifetime typically of a factor of 2.5 relative to SMAC.

Acknowledgements

Antonio Ruzzelli and Gregory O'Hare gratefully acknowledge the support of Science Foundation Ireland under Grant No. 03z/IN.3/1361. Raja Jurdak gratefully acknowledges the support of the Irish Research Council for Science, Engineering & Technology (IRCSET) through the Embark Initiative postdoctoral fellowship programme.

References

[1] A.G. Ruzzelli, G.M.P. O'Hare, R.Tynan, P. Cotan, P.J.M Havinga, Protocol assessment issues in low duty cycle sensor

networks: The switching energy, in: Proc. of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing,Taiwan, June 5–7, 2006.
[2] T.V. Dam, K. Langendoen, An adaptive energy efficient mac protocol for wireless sensor networks, ACM Sensys (2003).
[3] Amre El-Hoiydi, Spatial TDMA and CSMA with preamble sampling for low power ad hoc wireless sensor networks, in: Proceeding of Seventh International Symposium on Computers and Communications (ISCC 2002).
[4] D. Estrin, R. Govindan, J.S. Heidemann, S. Kumar, Next century challenges: scalable coordination in sensor networks, Mobile Computing and Networking (1999) 263–270.
[5] R. Jurdak, Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective, Springer-Verlag, 2007.
[6] R. Jurdak, P. Baldi, C.V. Lopes, Adaptive low power listening for wireless sensor networks, IEEE Transactions on Mobile Computing 6 (2007) 988–1004.
[7] L. Evers, W. Bach, D. Dam, M. Jonker, J. Scholten, P.J.M. Havinga, An iterative quality-based localization algorithm for ad hoc networks, in: 1st Int. Conf. on Pervasive Computing (Pervasive), Zurich, Switzerland, 2002, pp. 55–61.
[8] J. Hill, D. Culler, A wireless embedded sensor architecture for system-level optimization, 2001.
[9] D.B. Johnson, D.A. Maltz, Dynamic Source Routing in Ad hoc Wireless Networks, Mobile Computing, Kluwer, 1996.
[10] A. Ledeczi, A. Nadas, P. Volgyesi, G. Balogh, B. Kusy, J. Sallai, G. Pap, S. Dora, K. Molnar, M. Maroti, G. Simon, Countersniper system for urban warfare, ACM Transaction on Sensor Networks 2 (2005) 153–177.
[11] P.J.M Havinga, H.J. Kip, L.F.W. van Hoesel, S. Dulman, Design of a low-power testbed for wireless sensor networks and verification, Tech. report, University of Twente and Nedap N.V., The Netherlands, May 2003.
[12] G. Lu, B. Krishnamachari, C.S. Raghavendra, An adaptive energy-efficient and low-latency mac for data gathering in sensor networks, in: Proc. of International workshop on Algorithms for Wireless, Mobile, ad Hoc Sensor Networks, 2004.
[13] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, Wireless sensor networks for habitat monitoring, in: Proc. of the International Workshop on Wireless Sensor Networks and Applications, Atlanta, 2002.
[14] D.L. Mills, Z. Yang, T.A. Marsland, Internet Time Synchronization: The Network Time Protocol in Global States and Time in Distributed System, IEEE Computer Society Press, 1994.
[15] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, WMCSA'99: in: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, Washington, DC, USA, 1999, p. 90.
[16] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks (2004) 95–107.
[17] EYES Project, State of the art, European Union, <http://eyes.eu.org>, 2002–2005.
[18] I. Rhee, A. Warrier, M. Aia, J. Min, Z-mac: a hybrid mac for wireless sensor networks, in: Proc. of the third International Conference on Embedded networked sensor systems, Sensys05, ACM Press, New York, USA, 2005, pp. 90–101.
[19] A.G. Ruzzelli, L. Evers, S. Dulman, L.W. Van Hoesel, P.J.M. Havinga, On the design of an energy-efficient low-latency integrated protocol for distributed mobile sensor networks, in: Proc. of International Workshop on Wireless Ad hoc Networks, 2004.
[20] A.G. Ruzzelli, M.J. O'Grady, G.M.P. O'Hare, R. Tynan, An energy-efficient and low-latency routing protocol for wireless sensor networks, in: Proc. of the Advanced Industrial Conference on Wireless Technologies SENET 2005, IEEE Press, Montreal, Canada, 2005.
[21] A.G. Ruzzelli, R. Tynan, G.M.P. O'Hare, Adaptive scheduling in wireless sensor networks, in: Proc. of WAC2005, the Second Workshop on Autonomic Communication, LNCS press, Athens, Greece, 2005.
[22] D. Snoonian, Smart buildings, IEEE Spectrum 40 (8) (2003).
[23] EYES WSN, The codesign project, 2003, <http://www-home.cs.utwente.nl/mader/Codesign/>.
[24] J. Wu, P. Havinga, S. Dulman, T. Nieberg, Eyes source routing protocol for wireless sensor networks, in: Proc. of European Workshop on Wireless Sensor Networks EWSN, 2004.
[25] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking (2004).
[26] G. Zhou, T. He, S. Krishnamurthy, J.A. Stankovic, Impact of radio irregularity on wireless sensor networks, Proc. of the Second International Conference on Mobile Systems, Applications, and Services (MobiSys04), ACM Press, New York, USA, 2004.

Antonio G. Ruzzelli is currently in the final stage of Ph.D. writing at University College Dublin, (UCD), School of Computer Science and Informatics. For the last two summers he has been a visiting scientist of Philips research in The Netherlands. At UCD, he is involved in the Adaptive Information Cluster, AIC, which addresses the area of sensor networks and ambient intelligence. In particular, he is responsible for the implementation of MAC and routing protocols for energy-efficient and adaptive communication of sensor nodes. At Philips, his work focuses on 802.15.4 optimizations and networking capabilities 6LowPan complaint for sensor based-medical systems. He authored a patent on load balancing in sensor networks and about 17 papers on the area of wireless sensor networks in peer reviewed conferences and journals. He is a delegate of the Management Committee European COST Action 2100 for the Republic of Ireland, 2007–2011.

Gregory M.P. O'Hare is a senior lecturer at the School of Computer Science and Informatics at University College Dublin (UCD) and is Director of the School's Practice and Research in Intelligent Systems and Media (PRISM) Laboratory. His research focuses on multi-agent systems and on mobile and ubiquitous computing. He has a B.Sc. in Computer Science and an M.Sc in Information Technology from the University of Ulster. He is a Fellow of the British Computer Society and a Science Foundation Ireland (SFI) Investigator.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012

Raja Jurdak is a Research Fellow in the School of Computer Science and Informatics at University College Dublin. He received his Ph.D. in Information and Computer Sciences at the University of California, Irvine in 2005. From 2005 to 2006, he was a postdoctoral researcher at the University of California Irvine. He received the IRCSET Embark fellowship in 2006. He is the author of over 25 refereed journal and conference publications, a pending patent, as well as a book *Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective* by Springer in 2007. His research focuses on application-driven networks, modeling of ad hoc and sensor networks, emerging communication technologies, underwater acoustic networks, and cross-layer design.

Please cite this article in press as: A.G. Ruzzelli et al., MERLIN: Cross-layer integration of MAC and routing ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2007.11.012