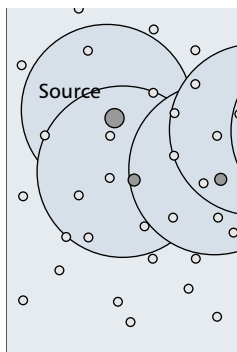


PROLONGING THE LIFETIME OF WIRELESS SENSOR NETWORKS BY CROSS-LAYER INTERACTION

LODEWIJK VAN HOESSEL, TIM NIEBERG, JIAN WU, AND PAUL J. M. HAVINGA
UNIVERSITY OF TWENTE



WSNs differ greatly from traditional ad hoc wireless networks and therefore require the usage of new types of network protocols, which are energy-efficient to ensure a node lifetime of several years on a single battery and which can operate without assistance of central managers in a dynamic network topology.

ABSTRACT

This article presents a cross-layered approach for networking in wireless sensor networks. WSNs differ greatly from traditional ad hoc wireless networks and therefore require the use of new types of network protocols that are energy-efficient to ensure a node lifetime of several years on a single battery and can operate without assistance of central managers in a dynamic network topology. We show that a tightly integrated set of networking protocols is a good solution to reach the target of highly energy-efficient WSNs. Our approach combines medium access organization with routing.

INTRODUCTION

The technology to let tiny smart devices create their own network, allowing them to transport sensor data while requiring little power and transmission range, is potentially “the next big thing” to happen [1]. Recent advances in sensor technology, low-power analog and digital electronics, and low-power radio frequency designs have enabled the development of these cheap, small, low-power sensor nodes, integrating sensing, processing, and wireless communication capabilities.

Sensor nodes collaborate to be able to cope with the environment: they operate completely wirelessly, and are able to spontaneously create an ad hoc network, assemble the network themselves, dynamically adapt to device failure and degradation, manage movement of sensor nodes, and react to changes in task and network requirements.

There are many challenges in wireless sensor networks (WSNs). In our work, we address in particular energy efficiency and the dynamics of a WSN. Where traditional communication protocol stacks assume an excess of resources and can spare the energy and memory to send many messages, sensor nodes need to save on every bit transmitted to ensure an acceptable network lifetime.

Some nodes in a WSN can be mobile, while others are fixed in walls or other immobile objects. In order to conserve energy, sensor nodes are in a low-power or off state for significant amounts of time. Communication during

those periods is not possible. From the network point of view, this means that the network topology changes over time. Hence, the networking protocols must be able to cope with mobility and changes of network density.

Sensor nodes must assist each other in forwarding their sensor readings to a data sink in the network. A routing protocol has the task of establishing an efficient route for messages to travel in a multihop sensor network. Nodes along the route can suddenly fail or simply move away, in which case the routing protocol has to defer messages to a new route. The highly unpredictable environment makes this a challenging task.

This article presents a cross-layered approach for networking in WSNs, as part of the ongoing European research project EYES (IST 2001-34734, <http://eyes.eu.org>). The approach addresses a self-organizing medium access control (MAC) protocol that uses an algorithm to decide the grade of participation of a sensor node in creating a connected network based on local information only, and a tightly integrated, efficient routing protocol.

Our lessons learned in developing network protocols for WSNs in the last couple of years show that using the traditional layered networking approach has several drawbacks in the resulting performance and efficiency of the system. Quite often, significant improvements are possible for network protocols, but they require a significant amount of information to be passed along the layers of the system. Although this approach in principle allows independence between the various protocols, it incurs significant overhead in parameter transfer. Moreover, improvements performed in a specific layer can cause impairments and even be counterproductive for other layers.

Optimization can be more effective when taking into account the overall system and using all available knowledge. When this information has to be distributed to other sensor nodes, this effect is even larger. A solution in which such information is piggybacked to other messages can limit this extra message exchange. During the development of various protocols and services (e.g., localization protocols), the lowest layers of our system (e.g., the MAC layer) were

increasingly being used to pass information to these higher layers. The overall result of these developments has led to the cross-layered approach described in this article.

OVERVIEW

The MAC protocol consists of a fully distributed and self-organizing time-division multiple access (TDMA) scheme, in which each active node periodically listens to the channel and broadcasts a short control message. This control message is needed for medium access operation and is also used to piggyback various types of information at low energy costs.

Information in the control message is used to create a *maximal independent set* of nodes. This set of nodes creates a connected network, and nodes in the set are *active*, while other nodes are *passive* and save energy by exploiting the infrastructure created by the connected network.

The control message is also used by the routing protocol to establish and maintain efficient routes in a dynamic topology. The routing protocol uses local topology information gathered by the medium access protocol and is therefore efficient in re-establishing routes when they become disconnected.

The presented approach is compared to the Dynamic Source Routing (DSR) protocol on top of the Sensor MAC (SMAC) protocol. Our approach to networking protocols for WSNs clearly benefits from the cross-layer interaction we are able to use. In a dynamic network topology, a network lifetime at least three times the lifetime of a DSR and SMAC network could be reached in simulation.

We give an overview of related work. We discuss the design of the EYES MAC protocol, especially designed for WSNs, that allows the benefits of the cross-layer approach discussed in this article to be exploited. We pay special attention to the decision mechanism sensor nodes use to either actively take part in the network or save energy by using resources of the backbone nodes in the network. The designed routing protocol is presented, and we then discuss simulation results.

RELATED WORK

MEDIUM ACCESS CONTROL

Although the research field of WSNs is relatively new, some interesting studies of MAC protocols can be found in the literature. One of those protocols is SMAC [2], which we use later to compare results.

SMAC recognizes two phases in transceiver usage of nodes: listen and sleep periods. In a sleep period, nodes turn off their power-consuming transceiver. After a sleep period, nodes wake up and listen for communication addressed to them or initiate communication themselves. This implies that the sleep and listen periods should be (locally) synchronized between neighboring nodes. The protocol uses carrier sense multiple access with collision detection (CSMA/CD) in the listen period.

When a node is turned on for the first time in a network, it has to determine its schedule for sleeping and listening. First, it will try to synchro-

nize with the schedules of its neighbors. This is done by listening for a defined amount of time to surrounding communication. If a node receives a synchronization message during this initial phase, it adjusts its schedule to the given information and follows the sleep/listen pattern in the network. When a node does not receive a schedule from a neighbor, it randomly chooses a time to enter the sleep phase and broadcasts this information in a synchronization message. This node now defines the schedule in the (local) network.

A node that has already chosen a schedule and becomes aware that one of its neighbors is following a different schedule keeps its own schedule and additionally wakes up according to the schedule of the other node. All nodes maintain a table with the schedule of their neighbors.

Communication takes place in the listen period, which is divided into two sections. The first part is reserved for synchronization, and the other for request to send (RTS) messages. SMAC also offers omnicast messages, which are not acknowledged by the receiving parties.

The SMAC protocol requires fine-tuning of the schedules (e.g., length of the sleeping intervals for different data traffic patterns and network densities). A dynamic topology results in many overlapping schedules, reducing the amount of possible sleeping periods of each node. Our approach tries to reduce this fine-tuning to a minimum. Especially the connected active set incorporated into the MAC protocol eliminates the need to adjust the lengths of sleeping intervals in order to obtain, say, a connected subnetwork at all times or reduce the latency induced by routing through (partially) sleeping nodes.

ACTIVE AND INACTIVE NODES

For the decision about which nodes have to remain active to ensure an operational and connected network, we use ideas coming from clustering techniques. In the context of WSNs, clustering is mostly used to group the nodes for routing protocols. Such a cluster is usually controlled by a designated node.

Several authors [3–5] focus on clustering schemes where the controlling nodes form an independent set in the wireless network. A set of local protocols that create and maintain a set of independent control nodes in the face of dynamic environments (i.e., mobility) is given in [4].

Some ideas from these clustering algorithms are applied directly to the MAC layer in order to create a backbone of the network consisting of the active nodes. In order to obtain an overall connected structure, so-called bridges are introduced and used to create connections between controlling instances of the clusters. Our mechanism provides nodes with the ability to be idle and in a low-power mode for long periods of time, while nodes retain the possibility to quickly use the connected communication infrastructure, and creates an efficient and connected backbone.

MULTIHOP ROUTING

In a WSN, data generated by one or more sources usually has to be routed through several intermediate nodes to reach a destination due to the limited range of each node's radio. There

For the decision about which nodes have to remain active to ensure an operational and connected network, we use ideas coming from clustering techniques. In the context of WSNs, clustering is mostly used to group the nodes for routing protocols.

The main goal in designing a MAC protocol for WSNs is to minimize energy consumption, while limiting latency and loss of data throughput. Therefore, we have three modes of operation in our MAC protocol: active, passive, and dormant.

has been significant work and research on routing mechanisms that deal with this problem, also in the face of frequent topology changes. Related work for WSNs include Directed Diffusion [6], Gradient Broadcast [7], DSR [8], and Ad Hoc On-Demand Distance Vector Routing (AODV) [9]. There are also a broad range of routing protocols that use positional information of the nodes to create routes. Often, Global Positioning System (GPS) receivers are proposed to obtain the geographical locations of nodes. However, obtaining a node's position will be rather costly in terms of energy consumption and pricing of sensor nodes.

Directed Diffusion is a data-centric routing scheme that relies on local interactions between nodes to create efficient paths for data flow. Directed Diffusion does not scale well when the nodes become mobile due to an end-to-end four-way handshake protocol that has to be repeated every time the destination (sink) moves.

DSR and AODV are routing protocols designed for dynamic networks. In this article, we compare our routing protocol with DSR (on top of SMAC). In brief, DSR works as follows. Each node stores known routes in its *route cache*. A new route is needed if information to the destination of a message is missing in this cache. The route discovery process is initiated to create the new route.

The network is flooded with *route request* messages. Each node adds itself to the route list in the message in order to build up possible routes to the destination, and transmits the modified message to its neighboring nodes. When a request reaches the destination node, a route reply message is sent back once to the source. This is also done when an intermediate node has routing information to the destination stored in its cache. Nodes on the return path to the source will update their cache with the new route information.

MEDIUM ACCESS PROTOCOL

Sensors equipped with transceiver, processor, and memory will be deployed by the millions. Hence, the costs of a single smart sensor must be minimal. This translates not only to scarce resources like energy and memory in sensors, but also to complexity of hardware. During the design of the MAC protocol, we assumed a single-channel transceiver that has three operational states: transmit, receive, and standby. Typically, transmitting consumes more power than receiving, and standby lies beneath the power consumption of receiving by a factor of 1000 or more.

In our research on energy-efficient WSNs, we explore a MAC protocol whose operation is entirely distributed and localized. The main goal in designing a MAC protocol for WSNs is to minimize energy consumption, while limiting latency and loss of data throughput. Therefore, we have three modes of operation in our MAC protocol: active, passive and dormant. When a node is in *active mode*, it will contribute to the network by taking part in forwarding messages to a destination and accepting data from passive nodes. *Passive nodes*, on the other hand, conserve energy by only keeping track of active nodes, which can forward their data and inform

them of network-wide messages. The nodes in *dormant mode* put themselves in a low-power state for an agreed amount of time or, for example, when their power source runs out of energy and has to be charged again using ambient energy like light. In our view, dormant mode has to be initiated from the application side.

MAC protocols for WSNs must be able to cope with mobility and changes of network density. We assume that the change of network topology is low compared to network events; thus, mobility is assumed to be limited.

FRAMES AND TIME SLOTS

The medium access protocol is based on TDMA. Time is divided into *time slots*, which nodes can use to transfer data without having to contest for the medium or deal with energy wasting collisions of transmissions. We assign *only one* time slot to each node and give this node control over this time slot. After the frame length, which consists of several time slots, the node again has a period of time reserved for it. To limit the number of time slots necessary in the network, we allow time slots to be reused at a non-interfering distance. But unlike traditional TDMA-based systems, the time slots in our protocol are *not* divided among the networking nodes by a central manager. Later, we explain how wireless sensors can autonomously pick time slots with local network knowledge only.

A time slot is further divided into three sections: communication request (CR), traffic control (TC), and the *data* section. In the CR section other nodes can make requests to the node controlling the current time slot. Nodes that have a request to the time slot owner will pick a random start time in the short CR section to make their request. These messages are comparable to RTS messages in SMAC. Communication in this section is not guaranteed to be collision-free. Nodes that do not have a request for the current slot owner will keep their transceiver in a low-power state during the entire CR section.

The owner of a time slot will always transmit a TC message in the time slot, regardless of whether or not a request was filed. All nodes within one-hop distance of the controller of the current time slot will put effort into receiving this message, since this message is used for synchronization purposes and control information. When a time slot is not controlled by any node, all nodes remain in sleep state during that time slot.

The time slot owner also indicates in its TC message what communication will take place in the data section. If a node is not addressed in the TC section or its request was not approved, the node will resume standby state during the entire data section. The TC message can also indicate that the controlling node is about to send an omnicast message. In case the controlling node announced that it is going to send data, the data will be glued directly after the TC section and hence saving additional energy of transmitting a preamble and preventing wastage of valuable data throughput.

A passive node in general will neither control nor claim a time slot. It is still able to communicate to the network by sending its requests to an active node. This allows significant energy con-

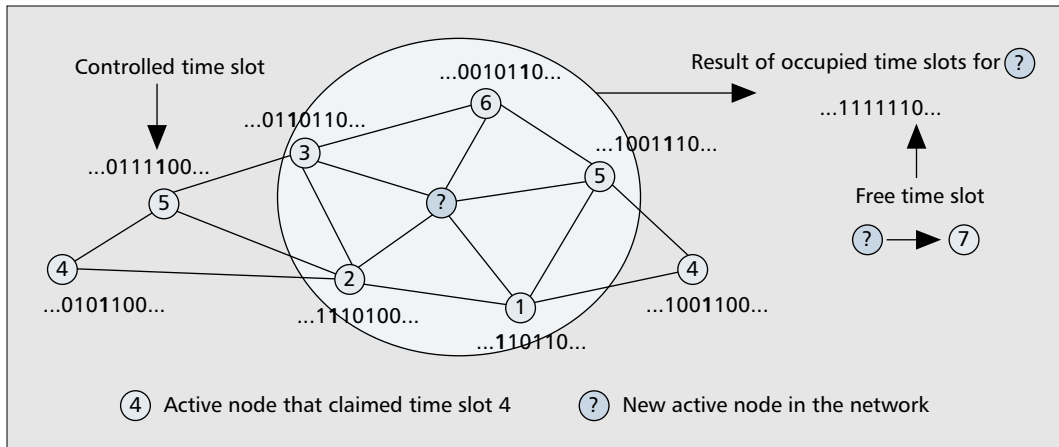


Figure 1. A new active node in the network can pick a time slot when it has discovered all its neighbor nodes.

servation in passive nodes and the lifetime of the network is largely extended, certainly if the role of active and passive nodes is changed over time.

Routing protocols that allow messages to be routed over an ad hoc network typically require knowledge of the actual topology in order to efficiently route packets over the network and deliver them at their destinations. By listening to TC sections of neighboring nodes, nodes have knowledge of *local topology*. This assists routing and reduces the number of routing messages in the network. A special portion of the TC section is reserved to efficiently transmit short omnicast messages generated by the routing protocol.

CHOOSING A TIME SLOT

For spatial reuse of time slots, the nodes use an algorithm based on local information only. Active nodes transmit a small table in the TC message that contains those time slots the node considers to be occupied by itself and its one-hop neighbor nodes. This information can be efficiently encoded by a number of bits equal to the number of time slots in a frame. Nodes can start controlling a time slot when the slot is considered free by all its neighbors. To reduce the chance that two nodes start controlling the same time slot, nodes pick a random time slot from those not yet controlled.

This method ensures that a time slot is only reused after at least two hops and, if no errors are made in choosing a time slot, no collisions occur. In practice the clear to send (CTS) message in SMAC takes care of a similar distance between two transmissions at the same time. Figure 1 gives an example of how a new active node in a network can pick a time slot after it has discovered all its neighbors. Note that active nodes only use their own time slot to transmit data.

From time to time (depending on the expected mobility in the network), active nodes give up their time slot and re-execute the algorithm to pick a time slot. This prevents collisions when an active node travels through the network and meets another active node that has claimed the same time slot. When a sensor node is mobile, it should be given a preference to be a passive node in order to minimize degradation of network performance.

MAINTAINING SYNCHRONIZATION

The TDMA-based scheme requires nodes to synchronize, but this synchronization need not be very strict. Small cheap sensor devices will operate on low-frequency clocks that suffer from inaccuracies. Since TC sections, which are also used for synchronization, will be transmitted often, clock drift will be smaller than the clock period. By switching the transceiver to receive a little bit early, deviations can be tolerated. In our simulations, we do not assume perfect clocks in the nodes. The nodes suffer from clock drifts, but synchronization can be maintained.

We also assume that some nodes (e.g., the sink nodes) have more precise timing than other nodes in the network. The grading of timing accuracy is transmitted in the TC section, which allows nodes to choose the schedule to which they want to synchronize. This results in more precise timing overall. In many applications measurements need to be *timestamped*; therefore, accurate knowledge of time is needed in the nodes anyway.

ACTIVE AND INACTIVE NODES

In this section we present an algorithm used to identify the nodes that actively participate in networking tasks such as routing. The decision is made locally according to information from neighboring active nodes only. We present a local distributed algorithm whose control information easily fits into the TC section of the MAC scheme presented in the previous section.

Since inactive nodes do not actively participate in the routing process of the network, the set of active nodes is required to form a connected set. This way, each node of the network can eventually be reached by an ad hoc routing process.

The set of active nodes is from here on referred to as the *connected active set* of nodes. Nodes that need to be active to ensure the above properties are contained in this set. For the remainder of this section, we call only these nodes active. Nodes that are not in this set are passive. Note that passive nodes may well use a time slot and participate in the network, but in general redeem this right to save energy.

By switching the transceiver to receiving a little bit early, deviations can be tolerated. In our simulations, we do not assume perfect clocks in the nodes. The nodes suffer from clock drifts, yet synchronization could be maintained.

Since inactive nodes do not actively participate in the routing process of the network, the set of active nodes is required to form a connected set. This way, each node of the network can eventually be reached by an ad hoc routing process.

Description	Encoding
Anchor node	AID = ID
Bridge	AID = (Anchor1 XOR Anchor2)
Undecided active	AID = 0
Nonmember	AID = Lowest ID anchor

■ **Table 1.** Roles of a node ID.

ROLES AND THEIR ENCODING

In order to decide which nodes are active and passive, several roles are given to the nodes participating in the network. Nodes that own a time slot periodically transmit a TC section; thus, all surrounding nodes are informed about their neighbors and their *AID*. This *AID* is an indicator of the role the node is performing with respect to the connected active set.

These roles are given in Table 1, together with their encoding in the *AID* field of the TC section. The *anchor* nodes are locally created to cover the network so that no two anchor nodes are direct neighbors. If an anchor node can reach (via other active nodes) all anchor nodes that are at most three hops away, the entire set of active nodes is connected. To achieve this, *bridge* nodes are introduced. There are two types of bridge nodes. A node that receives the TC sections of two or more anchor nodes is called a *direct bridge*. If two intermediate nodes are needed, these two nodes form a *distributed bridge*. For the created structure, not much difference exists between these types of bridge nodes; they are encoded and used in the same way.

For the *AID* field, the first bit when using node IDs is always set to 0. This is done to identify bridges, which have a leading 1 in the *AID* field. Also, the value given there is not mistaken for a possibly nonexistent node ID. Nodes that are not part of the connected active set (passive nodes), but participate in the network by owning a TC section, are identified by having an *AID* corresponding to the neighboring anchor node with the lowest ID. This encoding also helps in identifying distributed bridges.

A special role is *undecided active*, which is mainly used when a node enters the network, say, by waking up, and has not found a neighboring anchor.

Generally speaking, the anchor nodes form the main part of the connected active set and are spread out and maintained over the sensor network; the bridge nodes are then formed to connect the adjacent anchor nodes.

THE LOCAL DECISION ALGORITHM

Each node that enters the network, e.g. by waking up or being deployed, has to decide whether it is needed as part of the connected active set. This is achieved by the following algorithm. Additionally, this decision process is performed when a change in the local topology given by the active nodes occurs. This is witnessed by a change in a frame.

A schematic overview of the decision algorithm run in each node upon deciding is present-

ed in Fig. 2. Next, we present the individual steps and decisions in more detail.

Neighboring anchor: If there are neighboring anchors, the node cannot become an anchor itself. However, if there is no anchor identified, the lowest ID criterion is used to elect an anchor. For this, a node checks whether it is the undecided active node with the lowest ID in its neighborhood and becomes the anchor node if this is the case. Otherwise, it waits for undecided nodes with lower IDs to decide first. This follows the idea behind *lowest ID* clustering [5]. Other decision parameters can be used instead of ID of the node.

Bridging decision: If there are two or more anchor nodes in the neighborhood, a node checks whether there is already a direct bridge in the neighborhood connecting pairs of anchor nodes for which the XOR is also locally computed.

Distributed bridging decision: For a node to become a distributed bridge, one of the anchor nodes is not in its neighborhood. This can be detected if there is a neighboring nonmember node whose *AID* is not in the neighborhood. In that case, these two nodes can form a distributed bridge. Each node locally stores the ID of the node with which it forms a distributed bridge. Of course, this is only done after checking that there are not already nodes bridging these anchor nodes.

Become passive: A node that has come to the decision that it is not needed in the connected active set, does not drop out of the process immediately. For the next frame, it transmits its neighboring anchor with the lowest ID for distributed bridging detection. If after that no change in the neighborhood is detected, it can become inactive.

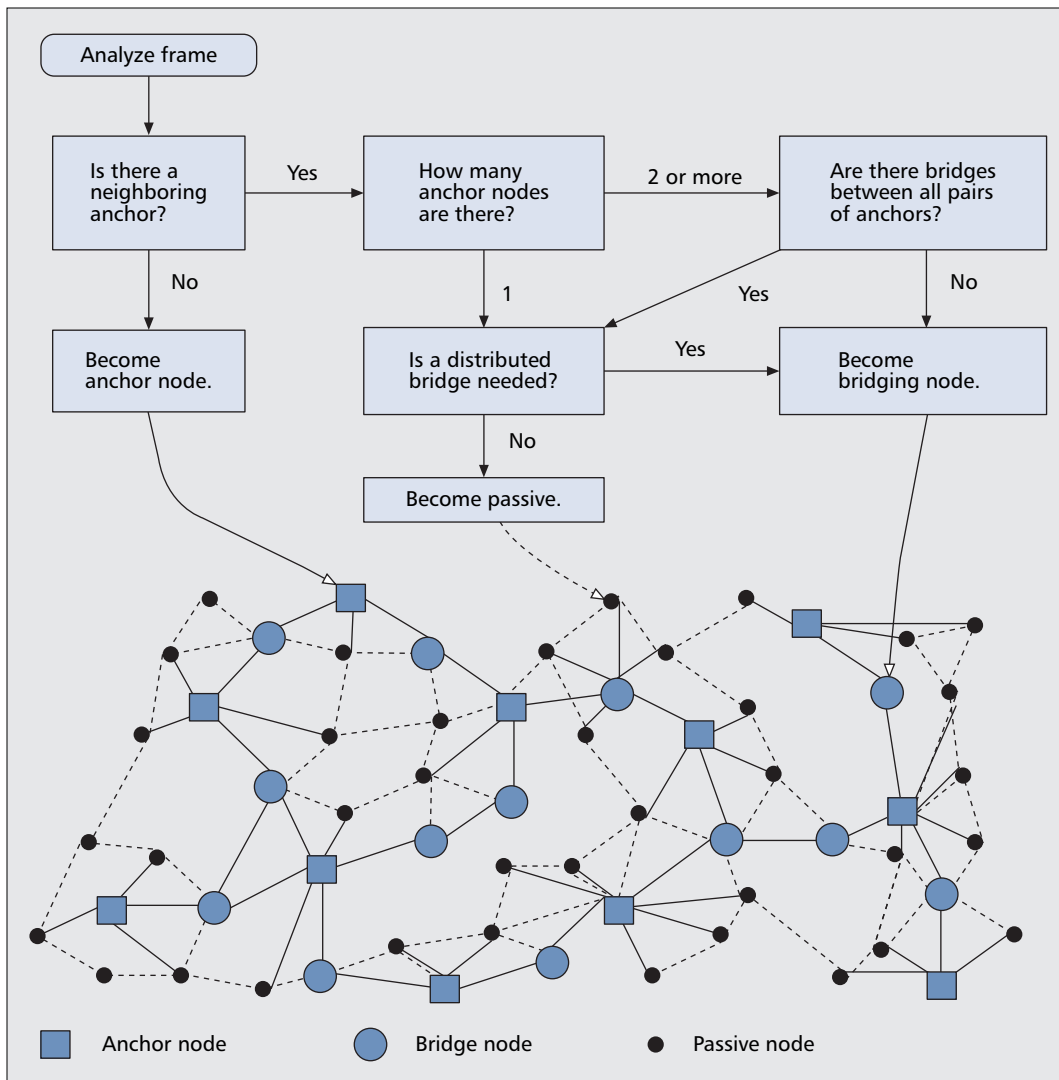
Note that if there are undecided nodes, the undecided node with the lowest ID in the neighborhood is always able to decide on its role other than undecided. The undecided role is thus only a temporary one.

Obviously, a node that participates in the network as part of the connected active set consumes more energy than a passive node. Therefore, the principle of role rotation is supported in our scheme. An active node can drop its status and become inactive. Surrounding nodes will detect this and adapt by creating a new anchor or bridge if needed for connectivity.

Especially in a dense network, many nodes are capable of performing the connecting duties of bridges. In our approach, only a few bridges actually have to remain active, as other nodes in the area realize their redundancy by the *AID* field. Thus, overall we obtain a connected dominating set given by the active nodes that uses only few nodes.

ROUTING

The EYES Source Routing (ESR) algorithm is an on-demand algorithm that enables dynamic, self-starting, multihop routing to be established when a source sensor node wishes to send a data packet. The reason we use an on-demand routing protocol is that it is very well suited to WSNs with high mobility and has the ability to address individual sensor nodes. All routing messages in ESR are small fixed-length packets. The ESR



■ Figure 2. The internal decision algorithm of a node to become active.

Especially in a dense network, many nodes are capable of performing the connecting duties of bridges. In our approach, only a few bridges actually have to remain active, as other nodes in the area realize their redundancy by the AID field.

algorithm has three phases: *route setup*, *route maintenance*, and *route re-establishment*, each of which is explained in the following sections. The routing protocol is essentially applied on the connected active set only, which implies that passive nodes should forward data to one of the nodes of this set first. The protocol utilizes the topology information already provided by the MAC protocol to efficiently manage topology changes due to mobility, node and communication failures, and power duty cycling.

ROUTE SETUP

Initially, when a node wants to send a data packet to another node, no prior knowledge of the location of the destination is available. In this stage, the source has to flood the network with route requests in order to notify the destination that it has a packet for it. The length of the request is small and constant to minimize the energy required in flooding.

The destination node replies to the first received request and discards duplicate ones. It sends back a route reply that confirms only the nodes on the fastest route to the source, which we call the on-route nodes. All other nodes in

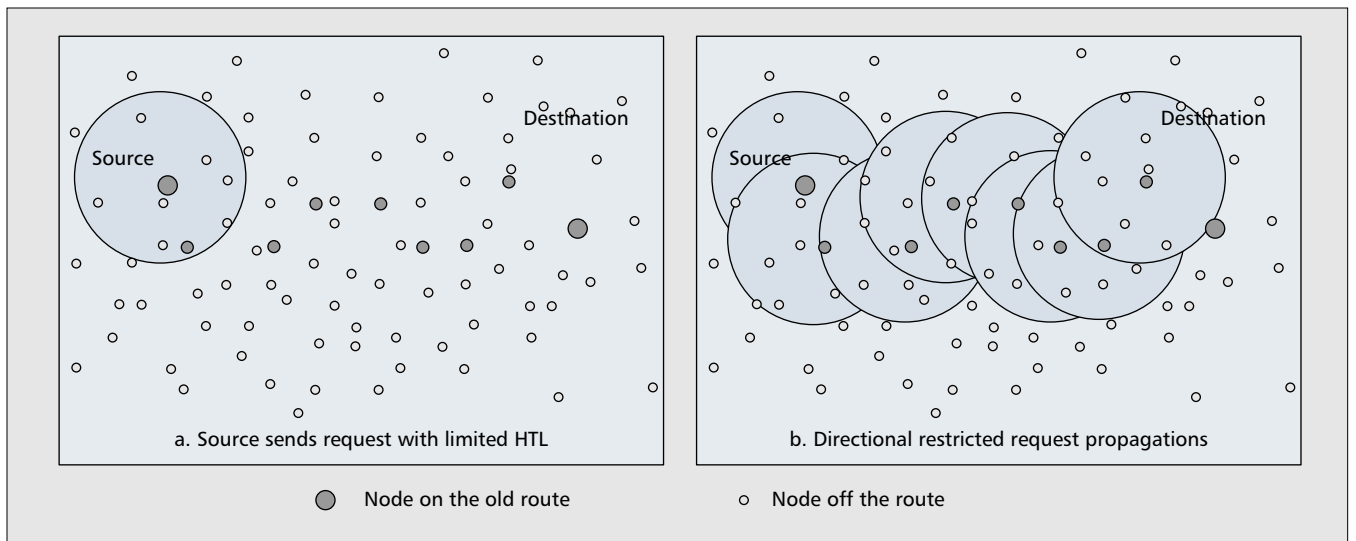
the network delete information of this source/destination pair if no reply comes back after a waiting time.

Any data packet between source and destination can be sent without knowledge of the complete route, and each intermediate node makes route decisions locally, according to its own *best neighbor pair*. This reduces the routing overhead during data packet transmission.

ROUTE MAINTENANCE

Due to the dynamics of WSNs, links have a high probability of breaking. Constantly re-establishing a lost link by reflooding the network with route requests is very energy consuming. The ESR protocol can recover lost links in a local and fast manner so that the frequency of network wide route re-establishments is significantly reduced.

A *route re-catch* message is sent when any on-route node finds out that its connection with the next-hop neighbor breaks. This happens when neighbors move in opposite directions or out of each other's radio range. Detecting such events is in general quite difficult and time consuming. In our approach we are able to detect this very efficiently and fast, because our MAC protocol



■ **Figure 3.** Directional flooding in a geographically limited area.

already has this kind of information present. Within one time frame the MAC protocol notifies the routing layer if any of the monitored neighbors is no longer available in the vicinity. This increases the possibility of a successful local and fast route recovery mechanism.

To restore its next-hop best neighbor, an upstream node sends a route re-catch messages locally, with its hops to live (HTL) set to a rather small value, depending on the average speed of the nodes and the density of the network. Any node that receives a route re-catch message checks whether it is on-route from the source to destination. If not, it records its best neighbor to the source and then forwards route re-catch messages when HTL is larger than 0. In the forwarded message, it decreases HTL by one. If it is on-route, the node sends *re-catch reply* back to the “catcher.” After the catcher receives the reply message, the broken link is restored successfully.

Re-catch messages are limited locally to a small-diameter set by the HTL. As the propagation speed of routing messages is much higher than the speed of the node, a properly selected HTL value results in a high probability of catching a lost link. Thus, the routing algorithm is able to locally restore the broken link in a fast and efficient way, which greatly reduces the frequency of network-wide flooding.

A *route cut* message is sent when an on-route node notices that its second order upstream best neighbor becomes its immediate neighbor. This happens when second order on-route neighbors move toward each other and come into each other’s radio range.

Then the node notices that its best neighbor to the source has changed. This could be the result of overhearing data transfer from the closing-in second order neighbor. To change its next-hop best neighbor, a node sends a route cut message.

When the second order neighbor receives a route cut message, it changes its best destination neighbor accordingly and forwards the following data packet to the respective node. The first order neighbor will change its state to on-route

when it receives the route cut message. This mechanism effectively shortens the redundant link in route maintenance.

ROUTE REESTABLISHMENT

Route re-establishment is necessary when route maintenance is not able to recover a broken link. In finding the destination node, the situation is different from the *route setup* stage, in which no information can be used to help locate the destination node. Temporary routes stored in on-route nodes can be explored. One important observation is that although nodes in the network are mobile, their maximum speed is limited. As a result, when a route is broken, the old on-route nodes are somehow still in the vicinity of the would-be shortest route to the destination that is going to be re-established by the source node, as shown in Fig. 3a. If the source could direct the request flooding along the old on-route nodes, it could reach the destination with dramatically fewer broadcast messages.

In the algorithm, directional and geographically limited flooding is achieved by controlling the HTL field in the route request. Figure 3b shows that whenever a request flood encounters an old on-route node, it is enlarged along the direction of the destination. The overall effect is a destination-aware directional request flood,

Parameter	Value
Energy consumption transmit	21 mW
Energy consumption receive	14.4 mW
Energy consumption standby	15 mW
Switch time transmit/receive	518 ms
Switch time receive/transmit	12 ms
Switch time standby/receive	518 ms
Switch time standby/transmit	16 ms

■ **Table 2.** Transceiver data (RFM TR 1001).

which soon dies out without the repeater effect of the old on-route nodes. It can be seen that directional flooding is efficient compared to network-wide flooding, and the effect can be even more advantageous if the network diameter grows.

SIMULATION RESULTS

For the simulation of our combined cross-layer optimized networking protocols, the OMNeT++ discrete event simulator, together with a framework for a mobile wireless network, is used. We compared the protocols presented in the previous sections to DSR and SMAC. The same network setup is used to compare the two implementations of medium access and routing protocols.

In the simulator, a physical layer with energy model is implemented to record the sending and receiving energy consumption of the transceiver. Additionally, switching between sending and receiving takes time and consumes energy, which is also considered in the simulation. The respective data for the transceiver are taken from an RFM TR 1001, which is also used in our prototype sensor nodes (Table 2). Although our prototype design can adjust its transmission range, we only consider the sending strength to be fixed to a high level, which yields an approximate coverage radius of 150 m.

In the simulation, 45 sensor nodes are randomly placed in a rectangle area of 5×5 times the radio range. Five of them are chosen to be source nodes, which actually produce sensing data. The length of a data packet is 5 bytes, and the data rate is varied in different simulation runs. One (active) node is designated as the data sink, which receives the data from these source nodes.

The nodes move in the area according to the random way-point model (RWP) with random speed (2–10 m/s) and waiting times (10–30 s). A node that has reached its destination does not immediately pick a new way-point, but waits for a given period of time before moving again. In this way, a mix of moving and static nodes is achieved.

We use *network lifetime* as the metric to evaluate the performance of our cross-layer optimized protocols. In WSNs, the metric of actual interest is not the transmission energy of individual packets, but the total operational lifetime of the network. Network lifetime measures the amount of time before a certain percentage of sensor nodes run out of battery power.

Both the data sources and sink have an infinite energy budget, so they will not affect network lifetime. During the simulation, when 30 percent of the normal sensor nodes are depleted of power, the whole network is considered to be down.

Figure 4 shows the network lifetime of our approach, and the reference DSR and SMAC under different network loads. Note that the graph is normalized to SMAC and DSR in the static scenario.

It is shown that our scheme prolongs the lifetime of the network significantly in the mobile scenario. A lifetime at least three times those of DSR and SMAC could be reached.

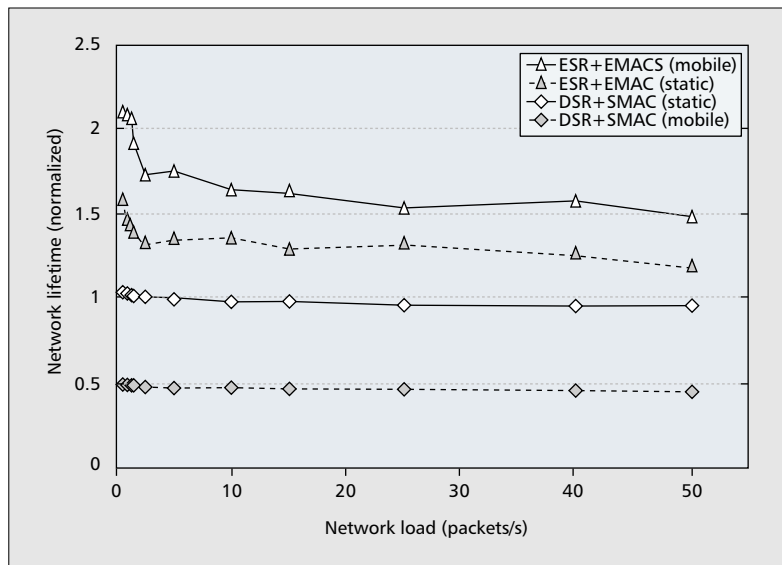


Figure 4. Network lifetimes of two different schemes.

Although our protocols were designed to be efficient in dynamic networks, we also compared the protocol performance for static networks. In that scenario our scheme extends the lifetime 25–50 percent.

DISCUSSION

It is interesting to see that the lifetime in SMAC and DSR is almost independent of message frequency. This can be explained by the fact that the nodes use their receiver anyhow in the time interval they are awake. The additional energy needed to exchange messages at relatively large intervals in our simulation is negligible compared to the energy used in the listen period. In fact, we would expect the lifetime of the network to get larger to some extent when message frequency is greater, because neighboring nodes of the transmitter and receiving nodes will switch their transceivers to standby to prevent energy waste in overhearing.

Our cross-layered approach performs better in scenarios where the nodes are mobile than in static cases. This can be explained by the fact that the roles *active* and *passive* are not changed in the latter case, while in the mobile case the dynamic changes in network topology force the nodes to reconsider their role. This leads to better and more even energy consumption between the nodes, which results in a longer network lifetime.

In contrast to our protocols, SMAC and DSR perform better in the static case than in the mobile one. This is clearly due to the overhead in routing; in the static scenario, routes need be established *only once*, while in the mobile scenario routes have to be updated regularly.

CONCLUSION

Our lessons learned in developing network protocols for wireless sensor networks in the last couple of years show that using the traditional layered networking approach has several drawbacks in the resulting performance and efficiency of the system. Quite often, significant improve-

In the near future we intend to implement the proposed protocol combination in a real-life testbed and prove its operation in the field. The MAC protocol has already been implemented and performs well.

ments are possible for network protocols. In this article we show that cross-layer optimization is indeed a useful approach for WSNs.

We discussed a TDMA-based medium access protocol whose operation is not dependent on a central manager or base station. The nodes in the network are capable of choosing their own time slot based on local information only. Nodes in the network can communicate with each other collision-free.

Not every node needs to actively participate in communication in the network for global connectivity. Hence, the MAC protocol allows some nodes to be passive. These passive nodes save energy by not controlling a time slot, but make use of a backbone in the network formed by active nodes. In this way, the MAC protocol overhead is greatly reduced for passive nodes. Passive nodes can communicate with active nodes, although this communication is not guaranteed to be collision-free.

This article presents a simple yet effective algorithm for nodes to make the decision between the active and passive medium access protocol states. Again, this decision is only based on local information. Simulations show that the number of active nodes is low.

The routing protocol benefits from local topology information already present in the MAC protocol. Only active nodes assist each other in forwarding messages to a destination that cannot be directly reached by the source node. An initial route is established by flooding of the active nodes of the network by the source. This is very energy consuming, yet inevitable. In this stage the benefit of the distinction between active and passive network participation of nodes already becomes clear. When a route gets disconnected (e.g., due to the highly dynamic topology of a WSN or energy depletion of nodes along the route), the protocol is able to efficiently reestablish a route between source and sink node.

We compared our cross-layer optimized networking protocols with traditional protocols for WSNs: SMAC and DSR. One of the key issues in WSNs is network lifetime. Simulations show that in equal network configurations, message frequency, and size assumptions, our cross-layered approach shows a longer network lifetime, especially when nodes are mobile. In the static case, the difference is smaller, which is mainly due to the fact that routing protocols have to establish a route only once. Our protocol has a small standard amount of data reserved for route updates; in the static case this space is wasted.

Our approach clearly benefits from active and passive modes in the MAC protocol. In the mobile case the roles are changed often, resulting in high network lifetimes. In the static case nodes keep their MAC mode until their energy gets depleted. Therefore, roles should be reconsidered every now and then.

In the near future we intend to implement the proposed protocol combination in a real-life testbed and prove its operation in the field. The MAC protocol has already been implemented and performs well.

A cross-layered optimization is in our eyes a good solution to reach the target of highly energy-efficient WSNs.

ACKNOWLEDGMENT

The authors would like to thank Pieter van Huizen for his great assistance in implementing the protocols for simulation and the MAC protocol for the prototype sensor nodes.

REFERENCES

- [1] R. D. Hof, "The Quest for the Next Big Thing," *Business Week*, Aug. 25 2003, pp. 91–94.
- [2] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *21st INFOCOM*, vol. 3, June 2002, pp. 1567–76.
- [3] A. D. Amis et al., "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," *Proc. INFOCOM 2000*, pp. 32–41.
- [4] S. Basagni, "Finding a Maximal Weighted Independent Set in Wireless Networks," *Telecommun. Sys., Special Issue on Mobile Computing and Wireless Networks*, vol. 18, no. 1/3, 2001, pp. 155–68.
- [5] M. Gerla and T.-J. Tsai, "Multicluster, mobile, multimedia radio network," *ACM/Baltzer J. Wireless Networks*, vol. 1, no. 3, 1995, pp. 255–65.
- [6] C. Intagagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Mobicom 2000*, Boston, MA, 2000, pp. 56–67.
- [7] F. Ye, S. Lu, and L. Zhang, "GRAdient Broadcast: A Robust, Long-Live Large Sensor Network," *UCLA Comp. Sci. Dept. tech. rep.*, 2001.
- [8] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *IETF Internet draft*, <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt>, Apr. 2003.
- [9] C. Perkins, E. Royer, and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing," *IETF Internet draft*, <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-13.txt>, Feb. 2003.

BIOGRAPHIES

LODEWIJK VAN HOESSEL (l.f.w.vanhoessel@utwente.nl) holds a Master's degree in electrical engineering from the University of Twente, the Netherlands. In the recent past he has developed simulation models for software-defined HiperLAN/2 radio and has been working on a bone densitometer in a large company in the United States. He is currently working within the European project EYES on energy-efficient wireless communication mechanisms. Research topics include the physical layer, medium access protocols, wake-up radio, and signal processing.

TIM NIEBERG (t.nieberg@utwente.nl) studied mathematics at the University of British Columbia, Vancouver, Canada, and the University of Osnabrück, Germany, where he received his Diplom. He is currently working toward a Ph.D. in applied mathematics at the University of Twente, where he is also with the European project EYES on energy-efficient sensor networks.

JIAN WU (j.wu@utwente.nl) is a full-time Ph.D. student in the Department of Mathematics, Electrical Engineering and Computer Science at the University of Twente. He graduated from Beijing University of Post and Telecommunications in 1999 with a B.S. degree. On a fellowship from IBM, he received a Master's degree in telematics from the University of Twente in 2002. His current research interests lie in the area of ad hoc wireless protocols, scalable wireless sensor networks, and embedded systems.

PAUL HAVINGA (p.j.m.havinga@utwente.nl) received his Ph.D. in mobile multimedia systems in 2000, and was awarded with the DOW Dissertation Energy Award for this work. Currently, he is within the Department of Computer Science of the University of Twente. His research interests are in the area of energy-efficient architectures and protocols, sensor networks, wireless communication, ubiquitous computing, personal communication systems, and (reconfigurable) hardware architectures. Currently, he is project leader of the Dutch project Smart Surroundings, on ambient intelligence, the Dutch project Featherlight on distributed operating system software, the European project EYES on energy efficient sensor networks, and the nationally funded project CONSENSUS on collaborative sensor networks. Besides these projects, he is also involved in several other projects, with a strong focus on sensor networks.