An ad-hoc sensor network for disaster relief operations

Nikos Pogkas, George Karastergios*, Christos Antonopoulos, Stavros Koubias, George Papadopoulos
Department of Electrical and Computer Engineering
University of Patras, Campus of Rio, Greece
(*) Industrial Systems Institute Rion, Patras Greece
E-mail: pogkas@isi.gr

Abstract

This paper presents an ad-hoc sensor network especially developed for a disaster relief application that provides the rescue teams with a quickly deployable, cost-effective and reliable tool to collect information about the presence of people in a collapsed building space and the state of the ruins. The hardware/software architecture of the wireless sensor nodes is developed for a low cost design implementation. Energy efficiency is another objective of this paper, achieved by the combination of a low power mode algorithm and a power aware routing strategy. A selected set of simulation studies indicate a reduction in energy consumption and a significant increase in node lifetime whereas network performance is not affected significantly. Finally, a lightweight management architecture is presented to facilitate autonomous management of ad-hoc sensor networks.

1. Introduction

Recently, there has been increased research interest in the area of ad-hoc networks. Such networks can operate without the presence of any preinstalled network infrastructure, while being able to handle node mobility and dynamic network topologies. Owing to these properties, ad-hoc networks have been employed in many critical applications, such as military, law enforcement or disaster relief operations.

The case study of ad-hoc networking, presented in this paper, focuses on handling disastrous events, such as buildings collapse caused by an earthquake. In order to decrease human fatalities, it is of main importance that the rescue teams have reliable information about the state of the ruins and the places where people have been trapped. The concept is to install, inside buildings, special, low cost, low power equipment (sensors), able to grab audio, visual information and transmit it on request to the terminals employed by the search teams[1]. The work presented here concerns the design and implementation of a low cost ad-hoc sensor network for the previously mentioned application [1] focusing on issues of extreme interest such as routing, network management, communication-related energy consumption, security and well as the equipment's implementation.

A sensor is a properly packaged device capable of sustaining a building collapse that can record image and audio data for a short time period close to the commencement of a disastrous event and transmit them to the rescue teams. Although sensors seem to be inactive during normal operation, they continuously take snapshots and analyze them so as to distinguish serious deformations on the structural elements of the building. Such an event is regarded as a building collapse situation which activates the recording mechanism and the wireless communication. The activated sensors, which contain captured information during the collapse, enter a power saving state where only the wireless communication receiver is enabled, waiting for the rescue teams.

The search and rescue operations start with a network discovery, which is a network management operation initiated by a rescue team's terminal, requesting the identities (IP addresses) of all the activated sensors. The network discovery operation provides, to the rescue team's terminal, the topology of the network, the sensor identities, their remaining power resources and the possible data routing paths. Then an automated process begins, controlled from specialized software running on the rescue team terminals, which transfers captured information from the sensor nodes for visualization and analysis. This information accompanied by image and audio snapshots taken during the rescue operation is really valuable and very helpfull for the rescue teams to take the right decisions and complete successfully their operations.

2. Sensor Architecture

It is apparent that a sensor is a rather complex device which implements many computational demanding tasks such as image capturing and compression, building deformation detection or wireless communication with other sensors forming an ad-hoc sensor network. Even though sensors are complex devices should also be low cost since the application requires decades of them to be installed in order to cover a building entirely. Furthermore, they should not consume much power in order to be able to operate on batteries for many hours while waiting buried in the ruins for the rescue teams. In addition their size is also important because as smaller is

[1] This work was supported by the European Commission under Project LOCCATEC (IST-2000-29401) while the sensor depicted in figure 2 was the result of a strong collaborative work between the Industrial Systems Institute Greece and Commisariat à l'énergie atomique France.

a sensor as easier it can be packaged adequately in order to survive from a building collapse. All the previously mentioned sensor attributes limit the alternative design options. The design constraints could only be fulfilled if the sensor implemented utilizes a small number of chips, low power technologies like CMOS, a powerful but power efficient processor and low cost commercial products. Taking these into consideration, a sensor prototype was designed utilizing system on chip (SoC) techniques and compatible technologies so as it can be easily implemented in a single chip.

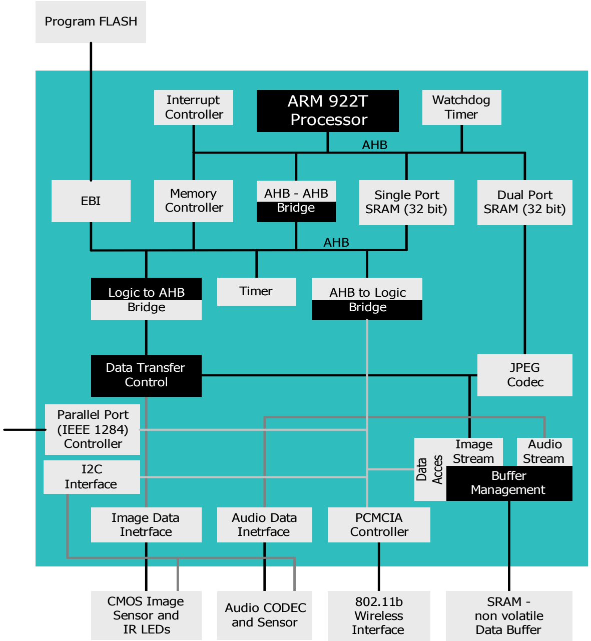


Fig. 1 Sensor architecture

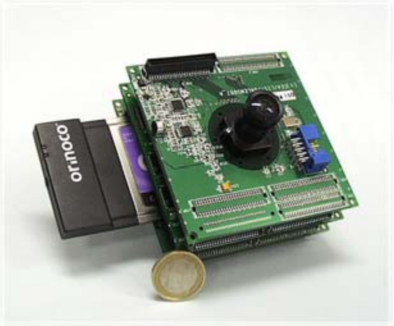


Fig. 2 Sensor prototype

The sensor was based on the powerful ARM922T SoC processor and comprises an image CMOS monochrome sensor with resolution 640x480 pixels, an audio PCM audio codec and a PCMCIA 802.11b wireless LAN modem. The sensor architecture is presented in Fig. 1, where the ARM has access to every peripheral through the high performance AHB AMBA bus. The peripheral devices (802.11b wireless interface etc) are connected to AHB bus through special designed interface controllers. The prototype sensors were implemented as a stackable set of boards utilizing commercial products for the wireless interface, the image sensor, the audio codec and an Altera Excalibur EPAX10 chip. The EPAX10 chip, which incorporates an ARM922T core CPU and a 1 million equivalent gates FPGA, it was used to lay the road for a SoC implementation.

3. Software Architecture

Although the mobility requirements in the network are rather low, yet the high attenuation, the multi-path propagation environment, the mobility of central stations and the movement of obstacles and machinery in the area can result in significant topological changes in the network at unpredictable times and rate. Therefore, an ad-hoc routing protocol was developed in order to maintain strong connectivity to support communications among the nodes. Our routing protocol implementation is based on Dynamic Source Routing (DSR) which has better performance [8], [9] and lower power consumption [7] compared to other ad-hoc protocols like AODV, DSDV and TORA. Another important property of DSR is that it does not transmit any periodic routing control packets and as a result there is no energy consumption imposed from the routing protocol during idle network periods. The routing protocol has been modified in order to incorporate a power aware routing algorithm (PAR) which makes routing decisions based on some energy efficient metric instead of the traditional shortest path criterion.

At the link layer a low power mode algorithm (LPM) has been implemented. This algorithm controls the operation of the wireless card in order to turn off the radio to reduce power consumption while maintaining the connectivity of the network. In order to meet the requirements of high rate connection oriented traffic, among the nodes of the network, a lightweight version of TCP Reno with fast retransmit and fast recovery has been used at the transport layer. At the application layer a simple client/server protocol has been implemented using a request response scheme. This protocol uses TCP sockets in order to support seamless communication among the sensor nodes and the central units. Upon a request from a central unit, audio/video data at the remote sensor node are encrypted and transferred. In order to facilitate the deployment of wireless-aware applications and network management operations, a protocol (Ad hoc Network Management Protocol - ANMP) has been implemented at the network layer which performs sufficiently in dynamic network topologies and has low memory and CPU performance requirements. The management entities use the ANMS subsystem (Ad hoc Network Management Service) which supports the necessary service primitives and provides a common service interface for the underlying management mechanisms.

For the wireless network interface, IEEE 802.11b DCF was chosen because it supports high transmission rates, there is availability of low cost products and it is suitable for multi-hop ad hoc networks with reasonable power consumption. The implementation of the protocol stack, depicted in Fig 3, is based in a complete state driven design with no busy-locks and a simple task synchronization scheme. The code memory size of the protocol stack is less than 128KBytes which permits its execution from internal SRAM, as a result performance is enhanced and the CPU energy consumption is reduced.

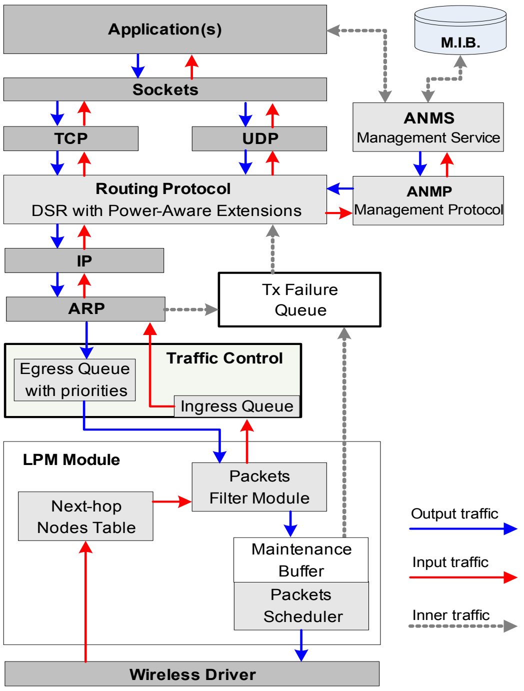


Fig. 3 Protocol stack architecture

3.1. Low Power Mode Algorithm

Since the sensor nodes of the network operate using batteries, it is important to minimize their communication-related power consumption. Our energy efficient solution is based on the combination of a low power mode algorithm and a power aware routing strategy, presented in the next sections. The basic idea of the proposed Low Power Mode algorithm (LPM) is to reduce the idle power consumption by turning off the radios of nodes that are idle. The algorithm's operation is driven entirely by the communication in the network. The main design considerations of our LPM implementation are the following: distributed robust operation in ad-hoc networks, the use of the algorithm should not affect significantly network performance, no need for synchronization between the nodes (the synchronization of nodes proposed in [2] is difficult to implement in dynamic topologies and induces a significant network overhead), no periodic exchange of packets or beacons (periodic transmission of packets or beacons proposed in [3] increases the energy consumption unnecessarily when the network is idle, also channel bandwidth and network capacity is decreased while the beacons or control packets content for channel access with normal data packets), low implementation complexity and no need for modifications at the MAC layer in order to be easily deployed with current commercial and future wireless products.

The algorithm has been implemented as an intermediate network driver at the LLC OSI layer for commercial IEEE 802.11 wireless cards. In Fig 4 we present the state diagram of the algorithm. When a node is idle it periodically turns off its radio for duration *Ts* entering the *Sleep* state. After T_S it enters the *Idle* state for a period of T_L where it listens for incoming packets. When it receives a broadcast or unicast packet addressed to that node it enters the *Active* state where it remains for duration of T_A. The same state transition occurs when there is a packet available at the egress queue ready for transmission. When a node is in the *Sleep* state it cannot receive any packets from any other node. As a result, a mechanism must be implemented in order to guarantee the successful reception of packets for a node that periodically enters the *Sleep* sate, this mechanism is presented in Fig 5. Every node that transmits a packet remains in the *Active* state for T_A. If the destination is a node that is possibly in the power saving state (namely transits periodically between the *Idle* and the *Sleep* state), the source node must retransmit the packet *R* times with an interval T_O in order to overlap the packet transmission with the destination's *Idle* state.

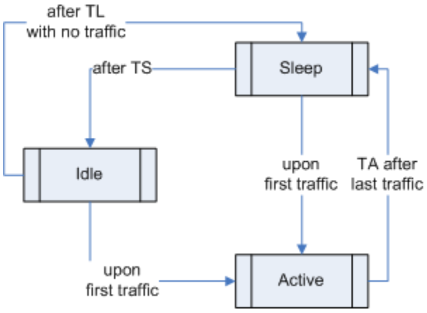


Fig 4 State diagram of the LPM algorithm

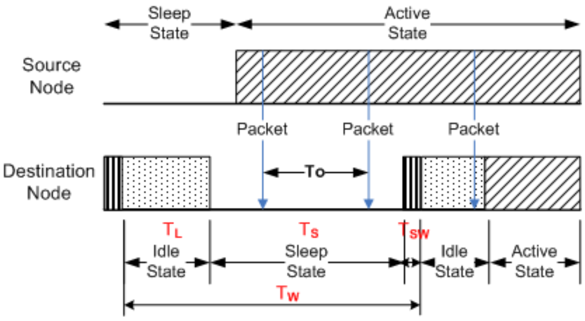


Fig 5 Packet retransmission mechanism

In order to reduce possible local congestion or overhead in the network, the following mechanism has been implemented. Each node maintains a table (*Next-hop*

Nodes Table) with the possible states of its next-hop neighbours in order to decide if it has to make the *R* re-transmissions or not. If the node transmits or receives a packet from a neighbour node, it marks this node as active in its table. After T_A seconds it marks the same node as possible-inactive (power-saving state) until it transmits successfully or receives another packet from that node. If the wireless interface is working in promiscuous mode, the nodes can trace any packet transmission in the network in order to update the status of next-hop nodes, improving the algorithm's performance.

The proposed LPM algorithm is implemented at the link layer and its system architecture is depicted in Fig 3. During a unicast transmission the outgoing packet is transferred from the egress-queue to the *maintenance buffer* at the LPM subsystem. A *packet filter module* searches the next-hop nodes table for the state of the destination node. If the destination node is possibly in the power saving state, a *packet scheduler* (at the source node) coordinates the retransmission of the packet. If the packet scheduler receives a MAC layer acknowledgment it cancels any pending retransmissions while the destination node enters the *Active* state. If the packet scheduler fails to transmit a unicast packet after *R* retransmissions, the packet is stored in a special queue (*Tx Failure Queue*) and the routing protocol is triggered with a link failure event.

However, during a broadcast transmission there is no MAC-ACK, so the source node has to transmit a broadcast packet every T_O seconds for *R* times. In order to reduce this overhead, a variable is maintained (*T_last_broadcast*) which contains the last time that a broadcast packet has been transmitted. Broadcast packets will cause every next-hop node to enter the active state for at least a period of T_A. Thus, when a sending node transmits a broadcast packet at time $t < T_A + T\_last\_broadcast$ it assumes that all its next-hop nodes are still in the active state and the packet is transmitted normally without any retransmissions. This mechanism reduces significantly the number of retransmissions in the network which are caused by the use of the algorithm.

In the presence of unstable links or high mobility, the above mechanism performs sufficiently well and in some cases improves network performance as a result of the additional retransmissions imposed in unicast and broadcast packets [4]. The major disadvantages of the proposed LPM algorithm are the increased latency during a route discovery (a basic problem found in most of the asynchronous low power mode algorithms) and the additional overhead induced by broadcast packets that are transmitted in the network (and especially for packets that are flooded in the entire network).

In the next paragraphs we present a systematic approach to address the LPM algorithm's design parameters and we calculate the maximum achieved energy efficiency of such an algorithm. An idle node not implementing the LPM algorithm in a time period $T_W=T_L+T_S+T_{SW}$ will consume energy

$$E_1 = P_L*T_L+P_L*T_S+P_L*T_{SW} \qquad (1)$$

During the same period an idle node that implements the algorithm will consume energy

$$E_2 = P_L*T_L+P_S*T_S+P_{SW}*T_{SW} \qquad (2)$$

where P_L is the power consumed at the idle state, P_S is the power consumed when the radio is off, Ts is the time a node spends in the sleeping state and T_L in the listening state. Tsw is the time needed to switch the radio from off to on plus the time needed to switch from on to off and, during this period, the power consumption is approximately equal to P_L ($P_{SW}\approx P_L$) [6]. Using equations (1) and (2) we define the power efficiency factor P_{EFF} of the algorithm for idle nodes by:

$$P_{EFF} = \frac{E_1 - E_2}{E_1} = \frac{(P_L - P_S)*T_S}{P_L*T_W} = K_{DEV}*\frac{T_S}{T_W} \qquad (3)$$

The constant value K_{DEV} depends on the hardware specifications of the wireless card and bounds the maximum value of the power efficiency factor. The maximum number of retransmissions required for a sleeping node to receive a packet is defined by:

$$R = T_W / T_O \text{, where } T_L > T_O \qquad (4)$$

Also we define the ratio $D = T_S / T_O$ (5)

Using equations (3), (4) and (5) the power efficiency is found to be

$$P_{EFF} = K_{DEV}*\frac{D}{R} \text{, where } R \in N^+ \text{ and } R > 1 \qquad (6)$$

A guideline to select adequate parameters for the algorithm is the following; T_S must have a small value to reduce latency. A positive integer is selected for the number of retransmissions *R* in order to achieve the desirable energy efficiency. The minimum value of T_L depends on the MAC layer specifications. Also T_O must be lower than T_L by an amount that reflects possible transmission delays caused by the driver, operating system, packet transmission delay or the MAC congestion avoidance mechanism. A set of values that has been used in the implementation and in the simulations of Section 4 is the following: T_L=69ms, T_{SW}=1ms, T_S=290ms, T_O=60ms and *R*=6. This set in case of the Orinoco wireless card (P_{TX}=1408mW, P_{RX}=914mW P_{IDLE}=785mW P_{SLEEP}=65mW) can achieve *PEFF*=0.73.

3.2. Power Aware Routing

There have been several research efforts regarding power aware routing algorithms. These algorithms must select the best path to minimize the total power needed to route packets on the network and maximize the lifetime of all nodes. Minimum cost battery routing MCBR [7] proposes the remaining battery capacity of the nodes as a metric. Min-max battery routing MMBR [8] defines as a cost metric of a route the maximum battery cost value of the nodes that constitute the path. Although the previous algorithms reduce network consumption and increase node lifetime, or both, in networks where the wireless channel is characterized by multi-path propagation it is observed that some links may experience an increased packet error rate. These unstable links can decrease the network performance significantly [5] due to packet losses and the initiation of the route discovery mechanism.

Efforts towards this direction have been also made to add criteria based on a combination of shortest-path, link quality, or least congested paths, that is, network load. Link quality estimations can be based on either the signal-to-noise ratio or the expected transmission count (ETX) metric [10]. The results in [10],[11] show that with stationary nodes the ETX metric significantly outperforms shortest path routing; also network-load and packet delay metrics perform poorly because they are load-sensitive and hence suffer from self-interference. However, Draves et al. in [11] concludes that in a mobile scenario shortest-path routing performs better, compared to ETX, because it reacts more quickly to fast topology change. Therefore the path length is also considered in the metric for our route selection.

The route selection algorithm, deployed in the presented ad-hoc sensor network, combines a link-stability metric with the battery level of the sensor-nodes and the route length. When the nodes have battery levels above some threshold and the links are stable, shortest path routing is performed. In case of link failures stable links are preferred and as the battery level of nodes is decreased a combination of the above metrics is encountered in route selection. Following the methodology used in [7] we define the path cost $C(n_0, n_k)$ of a path from a source node n_0 to a destination node n_k by:

$$C(n_0, n_k) = \sum_{i=1}^{k} z(n_{i-1}, n_i) \qquad (7)$$

where $z(n_{i-1}, n_i)$ is the cost of the link from node n_{i-1} to node n_i . The path cost C depends on the path length of the route and the cost of the links that compose the route. The link cost is defined as a function of the energy cost Zen and the stability cost Zst of the link from node n_{i-1} to node n_i by:

$$z(n_{i-1},n_i) = f(Zen(n_i),Zst(n_i)) = Zen(n_i)*Zst(n_i) \quad (8)$$

The multiplication of the above metrics increases the cost of nodes that have unstable links and low energy levels, as a result longer hop paths are preferred and packets are routed over these nodes. The energy cost metric of a node is defined as:

$$Zen(ni) = 1 + EF * g(ni) \qquad (9)$$

where $g(n_i)$ is the normalized energy consumed by the node,
$g(n_i) = (E_{INITIAL}-E(n_i))/E_{INITIAL}$, EF is the energy cost factor which bounds the maximum value of the energy cost and $E_{INITIAL}$ is the initial and $E(n_i)$ the current residual energy level of the node. The stability cost metric is defined as:

$$Zst(n_i) = 1 + \frac{1 + SF}{stability\ (n_i)} \qquad (10)$$

where $stability(n_i)$ is the stability metric of that node and SF a stability cost factor which bounds the maximum value of the stability cost. An optimal assignment of the EF and SF factors is required for meeting the network communication constraints. Increasing EF a priority is given in power aware routing whereas increasing SF more stable links are preferred. The stability metric was first proposed in [9] in order to estimate a dynamic link timeout cache policy, the Link-Max-Life (LMF) link cache. When a node is added in the cache it has an initial link stability value (*SINITIAL*). When a link from the route cache is used in routing a packet originated by that node, the stability metric of the two end point nodes is additively increased by a stability increase factor *SINCF*, $stability(n_i) = stability(n_i) + SEDCF$. Upon a link error, the source node will receive a route error packet containing the broken-link. In this case the stability metric of the two end point nodes is multiplicatively decreased by a stability decrease factor *1/SDECF* (where $SDECF \geq 2$), $stability(n_i)=stability(n_i)/SEDCF$. In any case the stability metric is bounded in a set [2, MAX_STAB]. Using equations (8), (9) and (10) the path cost is equal to:

$$C(n_0, n_k) = \sum_{i=1}^{k} (1 + EF * g(n_i)) * \left(1 + \frac{1 + SF}{stability(n_i)}\right) \qquad (11)$$

In the presence on unstable links in the network, nodes that route packets capture the received route error messages and update the stability metric of the nodes in the network. Each node maintains in the route cache a local view of the stability metric of links in the network. As a result, nodes that experience a high link loss ratio have decreased values of stability. Using equation (11) source nodes in the network route packets avoiding the use of unstable links and nodes with low energy, increasing the network lifetime and communication robustness [4]. A set of values that has been experimentally evaluated in the deployed wireless ad-hoc network is the following: *EF*=1, *SF*=2, *SDECF*=2, *SINCF*=2, *SINITIAL*=25, *MAX_STAB*=300.

In order to incorporate the proposed power aware routing strategy in the DSR protocol, the nodes must have information about the topology and the node energies in the network (energy map). To accomplish this target satisfactorily more accurate network topology information is required compared to shortest path routing because the source nodes must have alternative paths to route packets energy-efficiently.

The following modifications have been implemented in the DSR routing protocol in order to support power aware routing. The battery levels are included in the route request (RREQ) and route reply (RREP) messages. During the route discovery the initiator node broadcasts a RREQ message with a unique sequence number and its battery level. Intermediate nodes rebroadcast the first RREQ and any other with the same sequence number and a *Zen* cost (which is calculated from the message's header) lower from any other received so far. In a similar way, the target node responds to the first RREQ and any other with a lower *Zen* by sending a RREP to the initiator. In this way the initiator of the route discovery locates the shortest path and any other disjoint path that has a lower *Zen* cost and thus, is more energy efficient.

Besides the above modifications, the route maintenance mechanism has been extended as follows. When an intermediate node forwards packets towards a destination, it searches its route cache and if there is a more energy efficient path it sends a gratuitous cached reply message to the source node. The dynamic link timeout cache policy invalidates old links and the cache is updated with topology and new energy information (battery levels of nodes) during the route discovery operations.

3.3. Network Management

Network management is an important issue in the deployment and operation of ad-hoc networks; protocols and services designed for wired networks are not adequate or perform poorly in ad-hoc networks due to the following challenges and constraints. First, the network topology may vary rapidly and unpredictably and as a result packet losses are a usual phenomenon rather than an exception. Second, all the hosts in the network must cooperate in order to establish management operations under severe constraints as limited battery, varying link quality and limited storage capacity. The management protocol must balance between performance and message overhead and should adapt well in periods of high packet loss rate or communication malfunction.

The proposed network management architecture is based in a service (ANMS) and a protocol (ANMP) that cooperate to execute transparently management operations across the nodes of the network. The ANMP is a lightweight connectionless network-layer protocol that uses routing protocol services and conforms to a request/response scheme. A node, with the role of a manager, sends requests to specific nodes, or all the nodes of the network, which have the role of agents. These management requests are processed from the ANMS service which executes the requested management operations across the protocol stack. The ANMS uses a modified ISO/IEC 9595: 1991 abstract syntax which supports the basic management functions, the ANMS service primitives are binary encoded and encapsulated in ANMP PDUs. Query data in agent reply messages are processed from manager nodes and stored in the management information base (MIB) which is a conceptual repository for management information.

Network management is essential in our application scenario for two main reasons: first the central units do not know in advance the number of sensor nodes and their IP addresses in the network, as a result, a topology discovery service must be executed before the establishment of communication between central stations and wireless sensor nodes, and second network management can support security services with low overhead, such as authentication and encryption between the central units and the wireless nodes. Network management is also important for administration purposes such as network installation and monitoring, node failure detection and network recovery.

The management scheme supports three types of communication:

1) Unicast where a manager requests the execution of an operation from a specific agent node, in this case unicast ANMP requests and replies are encapsulated in IP datagrams and routed at the network layer.
2) Broadcast where all the nodes of the network, or the nodes in a zone of the network, participate in management operations, this type of communication is used in topology discovery and network security services.
3) Anycast where a manager-request is flooded in a specific zone or the whole network and a subset of nodes, which support the specific operation or expose a requested service, participate in the management operation. This type of communication is used in service discovery operations.

The implementation of the ANMP protocol is based in source routing and has two major mechanisms, the request phase and the reply phase. During the request phase, initiated by the ANMS service, a manager-request packet is generated which encapsulates the requested service primitive, a sequence number, a source route and a target address (which is stored in the IP destination address). If the management operation is broadcast or anycast, a number of agents must participate in the specific operation. In this case, the packet is flooded in the network in a similar way that route-request packets are flooded in the network during the route discovery process of the DSR protocol. Each node that receives a request examines the sequence number and the manager's address and if it hasn't received another request with the same sequence number from that manager executes the requested operation, inserts its IP address in the source route and rebroadcasts the packet, so that each node forwards exactly one manager request.

During the agent reply phase, for a broadcast/anycast management request that needs confirmation, intermediate nodes buffer agent replies for short periods and send a unique packet that aggregates management information in order to reduce message overhead. This technique involves the use of a timer-based mechanism; each node that receives a broadcast or anycast manager-request packet will buffer any agent-replies that receives, for the specific operation, during a period *D* which is calculated from Equ. 12,

$$D = H * (Z - h + r) \qquad (12)$$

where *h* is the route length (number of hops) from the manager to the agent, *r* a random floating point number (with uniform distribution) between 0 and 1 and *H* a constant delay (at least twice the wireless propagation delay) to be introduced per hop, this constant delay prevents an agent reply storm. The variable *Z* expresses the zone length of the management operation and may be used to limit the number of intermediate nodes allowed to forward the request. This hop limit is stored in the IP header's TTL field of the packet carrying the manager request.

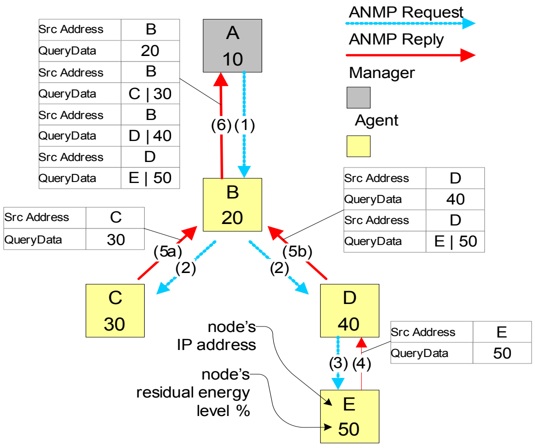


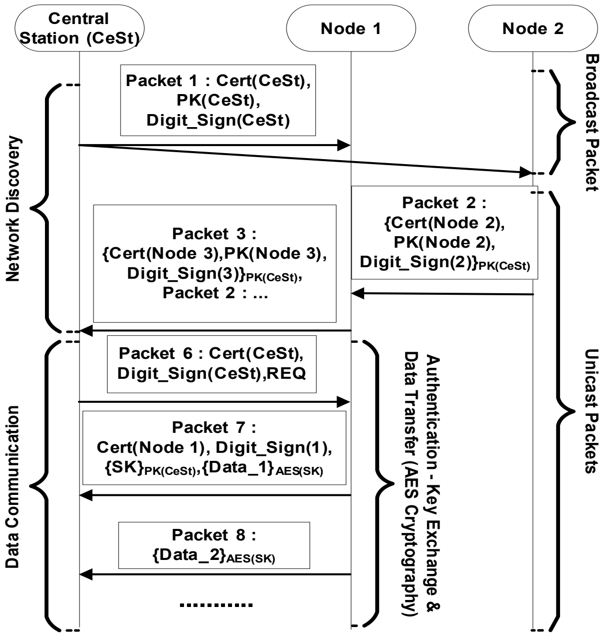
Figure 6. The aggregate agent-reply scheme during a network discovery

In Fig 6 the aggregate response scheme is depicted for a combined sensor node energy and topology discovery operation (network discovery). A manager-request packet is flooded in the entire network, initiated from manager node *A* (central station), agent-reply messages are aggregated in intermediate nodes and, in this example, a unique packet is received from node *A* which carries ANMS query reply data from all the agents (sensor nodes).

The requirements for secure communication and node authentication have lead to the adoption of cryptographic techniques and key management schemes in order to prevent unauthorized access to sensitive data. Security has been incorporated in the network management architecture and the application layer protocol and is performed in two phases: during the network discovery management operation and after a TCP connection establishment for sensitive data transfer, as depicted in Fig 7.

During the network discovery phase authentication is performed between the central station (which has the role of a manager) and wireless nodes (which have the role of agents) and a key exchange scheme is used to support public key cryptography. The ANMP protocol is used for the exchange of certificates and digital signatures to guarantee node identification and data integrity. In order to facilitate secure data communication the client/server protocol, at the application layer, has been extended to support asymmetric and symmetric cryptography.

Notation
Cert(x) : The Certificate of node x
Digit_Sign(x) : The Digital Signature of node x
PK(x) : The public key of node x
$\{y\}_x$: y encryptred using public cryptography and x as key
$\{y\}_{AES(x)}$: y encrypted using AES symmetric algorithm and x as key



Central Station (CeSt)
Node 1
Node 2
Packet 1 : Cert(CeSt), PK(CeSt), Digit_Sign(CeSt)
Packet 2 : {Cert(Node 2), PK(Node 2), Digit_Sign(2)}PK(CeSt)
Packet 3 : {Cert(Node 3),PK(Node 3), Digit_Sign(3)}PK(CeSt), Packet 2 : ...
Packet 6 : Cert(CeSt), Digit_Sign(CeSt),REQ
Packet 7 : Cert(Node 1), Digit_Sign(1), {SK}PK(CeSt),{Data_1}AES(SK)
Packet 8 : {Data_2}AES(SK)
....
Network Discovery
Data Communication
Broadcast Packet
Unicast Packets
Authentication - Key Exchange & Data Transfer (AES Cryptography)

Figure 7. The authentication and cryptography phases

The asymmetric cryptography mechanism is used for node identification and symmetric key (SK) exchange. During the data transfer phase symmetric cryptography is used which has laxer demands of computational overhead, memory capacity and implementation complexity, compared to asymmetric cryptography [12]. After detailed evaluation of various security algorithms we have concluded that the Advanced Encryption Standard (AES) [13] is the most suitable symmetric algorithm for our application requirements. The creation of the AES key is assigned solely to the wireless node that participates in the communication session and is accomplished through the use of a hash function. In this way symmetric key generation does not add severe complexity to the system. Due to its obligatory frequent regeneration, security of a sole key is not considered critical since the hash function can not be determined by the key that produces.

4. Simulation Study - Conclusions

In this section we evaluate the performance of the proposed LPM and PAR algorithms and their effect in communication efficiency with simulations using the ns-2 network simulator. The network topology consists of 50 nodes randomly distributed in a 1000mx1000m square area. The results are average values for a number of simulations in which the nodes are either static or have a random way point movement with relatively low mean speeds varying from 0 to 5m/sec.

In Fig 8 we measure the power efficiency factor for different values of R and with variable number of 80Kbps CBR traffic flows. When the traffic in the network is relatively low (3 CBR flows) and R=6 the simulated power efficiency in Fig 4 is $P_{EFF} = 0.65$. Using the same parameters in a fully idle network, the maximum theoretical value of P_{EFF}, given by equation (3), is equal to 0.91*290/360=0.73, where T_S=290, T_W=360 and for an Orinoco wireless card $K_{DEV} = 0.91$. Comparing the above results, we conclude that, as the communication traffic in the network is decreased, the power efficiency of the proposed LPM algorithm approaches its theoretical maximum value, since the number of idle nodes in the network is increased.

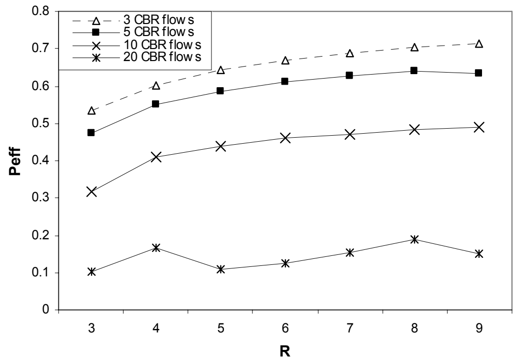


Figure 8. Power efficiency as a function of R

In Fig 9 we present the routing overhead (routing packets transmitted / total packets transmitted) induced by the use of the LPM as a function of R and a number of CBR traffic flows compared to the case where an LPM algorithm is not used. In the case where the number of simultaneous communication flows is relatively small, the use of LPM will increase the number of packets in the network by 1% to 3%. In the case where the number of flows is increased and the network becomes congested, the overhead will reach a value of 5%.

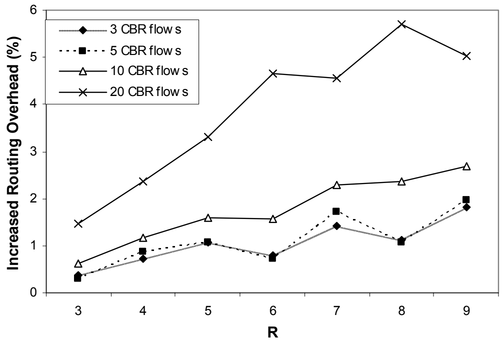


Figure 9. Routing overhead induced by the use of the LPM algorithm

In Fig 10 we present the expiration time of each node in the network (i.e., the time when a node exhausts its battery) for four different cases: when the LPM is not used and the routing strategy is MMBR, when LPM is used with MBCR and MMBR routing and when LPM is used with the proposed power aware routing (STABPAR). The presented simulation results derive from a network topology with 50 nodes and 5 simultaneous CBR traffic flows of 80Kbps. As we can observe, the combination of the proposed power aware routing and the LPM algorithm can achieve a significant increase in the lifetime of nodes. In this case energy efficient routes, that experience a lower link-loss ratio, are preferred for routing packets which significantly decrease routing errors and route discovery operations that waste energy.

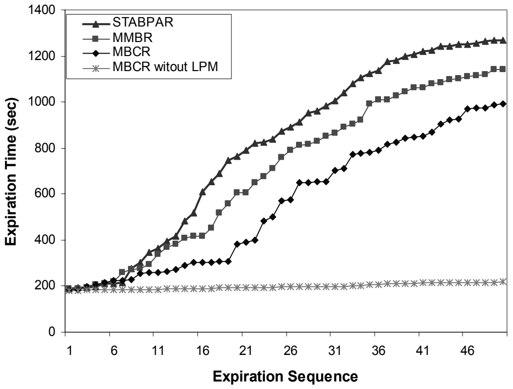


Figure 10. Expiration time vs. expiration sequence

In this paper we propose a low cost ad-hoc sensor network implementation for a specific disaster relief application. The hardware/software architecture of the sensor nodes was analyzed along with a low power mode algorithm, a power aware routing strategy and a lightweight network management architecture. Concluding from the above simulation results, the LPM algorithm can reduce the energy consumption of the wireless card up to 70%, while the network performance is not affected significantly. The PAR strategy proposes a new metric for route selection which combines the link stability and the node battery levels. As a result, energy efficient stable links are preferred for routing packets and, thus, node lifetime and network robustness is increased. During some preliminary experiments in real wireless networks, deployed with embedded sensor nodes and mobile PCs as central units, the network lifetime was increased significantly and the network performance was improved. As part of our on-going work we experimentally evaluate the PAR and LPM algorithms and several aspects of the protocol stack, such as TCP performance and tuning, in a wireless network with a large number of sensor nodes.

References

[1] G. Karastergios, N. Pogkas, et al. “System for acquiring and surveying data following catastrophic events, with scope of facilitating eventual aid or intervention,” European Patent No. 03425667.7.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R.Morris, “Span: An Energy- Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks,” Proc. of the International Conference on Mobile Computing and Networking, pp. 85–96, 2001.

[3] R. Zheng, J. C. Hou and L. Sha, “Asynchronous Wakeup for Ad Hoc Networks,” MobiHoc 2003.

[4] N. Pogkas and G. Papadopoulos, “Design and Implementation of a Low Cost Energy Efficient IEEE 802.11-based Ad Hoc Network,” International Workshop on Wireless Ad-hoc Networks IWWAN, May 2005.

[5] K. Benekos, N. Pogkas, G. Kalivas, G. Papadopoulos,”TCP Performance Measurements in IEEE 802.11-based Wireless LANs,” IEEE MELECON, May 2004.

[6] J. C Cano., P. Manzoni, “A Performance Comparison of Energy Consumption for Mobile Ad Hoc Networks Routing Protocols,” IEEE ACM MASCOTS, 2000.

[7] S. Singh, M. Woo, C. S. Raghavendra, “Power-Aware Routing in Mobile Ad Hoc Networks,” Proc. Mobicom, Oct. 1998.

[8] C. K. Toh “Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks,” IEEE Communications Magazine, June 2001, pp 138-147.

[9] Y. C. Hu and D. B. Johnson., “Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks,” Proc. ACM International Conference on Mobile Computing and Networking, August 2000.

[10] D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A High-Throughput Path Metric for Multi-Hop Wireless Routing,” MobiCom 2003, September 2003.

[11] R. Draves, J. Padhye, and B. Zill, “Comparison of Routing Metrics for Static Multi-Hop Wireless Networks,” ACM SIGCOMM, August 2004.

[12] Jerome Burke, John McDonald, Todd Austin, “Architectural Support for Fast Symmetric-Key Cryptography” ACM, November 2000.

[13] “Advanced Encryption Standard (AES)”, Federal Information Processing Standards Publication 197, November 26, 2001.