Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 30 (2007) 1655–1695
computer communications
www.elsevier.com/locate/comcom

Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards

Paolo Baronti [b,c], Prashant Pillai [a], Vince W.C. Chook [a], Stefano Chessa [b,c,*], Alberto Gotta [b], Y. Fun Hu [a]

[a] Mobile and Satellite Communication Research Centre, School of Engineering, Design and Technology, University of Bradford, United Kingdom
[b] Wireless Networks Laboratory, Istituto di Scienza e Tecnologie dell'Informazione, Pisa, Italy
[c] Department of Computer Science, University of Pisa, Pisa, Italy

Received 8 May 2006; received in revised form 11 December 2006; accepted 14 December 2006
Available online 29 December 2006

Abstract

Wireless sensor networks are an emerging technology for low-cost, unattended monitoring of a wide range of environments. Their importance has been enforced by the recent delivery of the IEEE 802.15.4 standard for the physical and MAC layers and the forthcoming ZigBee standard for the network and application layers. The fast progress of research on energy efficiency, networking, data management and security in wireless sensor networks, and the need to compare with the solutions adopted in the standards motivates the need for a survey on this field.

Keywords: Wireless sensor networks; ZigBee; IEEE 802.15.4

Contents

* Corresponding author.

0140-3664/$ - see front matter
doi:10.1016/j.comcom.2006.12.020

1. Introduction

Recent advances in Microelectromechanical Systems, tiny microprocessors and low power radio technologies have created low-cost, low-power, multi-functional miniature sensor devices, which can observe and react to changes in physical phenomena of their surrounding environments. When networked together over a wireless medium, these devices can provide an overall result of their sensing functionality.

Wireless sensors are equipped with a radio transceiver and a set of transducers through which they acquire information about the surrounding environment. When deployed in large quantities in a sensor field, these sensors can automatically organize themselves to form an ad hoc multihop network to communicate with each other and with one or more *sink* nodes. A remote user can inject commands into the sensor network via the sink to assign data collection, processing and transfer tasks to the sensors, and it can later receive the data sensed by the network through the sink.

Use of this technology appears to be limited only by our imagination and ingenuity. A diverse set of applications for sensor networks encompassing different fields have already emerged including medicine, agriculture, environment, military, inventory monitoring, intrusion detection, motion tracking, machine malfunction, toys and many others.

In the medical field sensor networks can be used to remotely and unobtrusively monitor physiological parameters of patients such as heartbeat or blood pressure, and report to the hospital when some parameters are altered [81,37,5,82].

In agriculture, they can be used to monitor climatic conditions of different zones of a large cultivated area and calculate different water or chemicals needs.

Pollution detection systems can also benefit from sensor networks. Sensors can monitor the current levels of polluting substances in a town or a river and identify the source of anomalous situations, if any. Similar detection systems can be employed to monitor rain and water levels and prevent flooding, fire or other natural disasters [119].

Another possible application that was recently experimented [120,26,124] is the monitoring of animal species and collection of data concerning their habits, population, or position. Sensors can be deployed to continuously report environmental data for long periods of time. This is a very important improvement with respect to previous operating conditions where humans had to operate in the fields and periodically take manual measurements resulting in fewer data, higher errors, higher costs and non negligible interference with life conditions of the observed species.

In structure health monitoring applications, sensor networks are deployed on structures such as bridges, buildings, aircrafts, rockets or other military equipment requiring continuous monitoring to ensure reliability and safety [75]. Sensor networks can be used to detect and locate damages as well as predict remaining life more effectively and economically with respect to traditional monitoring systems.

The military can take advantage of sensor network technology too. They can deploy such networks behind enemy lines and observe movements/presence of troops and/or collect geographical information on the deployment area.

Other possible fields include home/office automation, education [118], inventory monitoring, intrusion detection, motion tracking, machine malfunctions, toys and many others.

Several surveys [133,59,61,2], and [3] discussed various aspects on wireless sensor networks. In this survey, we give a comprehensive review on most recent developments and challenging issues that wireless sensor networks need to overcome and discuss solutions proposed in the literatures. In particular, this survey also deals with the increasing importance of the ZigBee/IEEE 802.15.4 [139,50] standards, giving a review of these standards and comparing their solutions with the ideas emerged in the recent literature.

The rest of the paper is organized as follows. Section 2 reviews the sensor networks hardware, and Section 3 presents the ZigBee/IEEE 802.15.4 standards. Energy efficiency, routing and localization issues are discussed in Sections 4–6, respectively. Section 7 presents data management techniques while reliability issues are discussed in Section 8 and security is covered in Section 9. Section 10 draws the conclusions.

2. Sensor network hardware

A wireless sensor is characterised by its small size, its ability to sense environmental phenomena through a set of transducers and a radio transceiver with autonomous power supply. Current low-end sensors employ low cost Reduced Instruction Set Computer (RISC) microcontrollers with a small program and data memory size (about 100 kb). An external flash memory with large access times may be added to provide secondary storage and to alleviate the application size constraints imposed by the on-chip memory. Common on-board I/O buses and devices include serial lines such as the Universal Asynchronous Receiver-Transmitter (UART), analog to digital converters and timers.

Two approaches have been adopted for the design of transducer equipment. The most general and expandable approach, as pioneered by Crossbow [28], consists in developing transducer boards that can be attached (and possibly stacked one on top of the other) to the main microcontroller board through an expansion bus. A typical transducer board from Crossbow provides light, temperature, microphone, sounder, tone detector, 2 axis accelerometer and 2 axis magnetometer devices. Alternatives include low cost versions that provide a reduced set of transducers or more expensive versions that boast GPS, for instance. Special boards are also available that carry no transducers but provide I/O connectors that custom developers can use to connect their own devices to the Crossbow sensors.

The other approach (followed by Moteiv [86]) is to put transducers directly on the microcontroller board. Transducers are soldered or can be mounted if needed but the available options are very limited and generality and expandability is affected. On the other hand, these on-board transducers can reduce production costs and are more robust than transducer boards which may detach from the microcontroller board in harsh environments.

By means of the transceiver circuitry a sensor unit communicates with nearby units. Although early projects considered using optical transmissions [117,54], current sensor hardware relies on RF communication. Optical communication is cheaper, easier to construct and consumes less power than RF but requires visibility and directionality, which are extremely hard to provide in a sensor network. RF communication suffers from a high path loss and requires complex hardware but is a more flexible and understood technology.

Currently available sensors employ one of two types of radios. The simplest (and cheaper) alternative offers a basic

Carrier Sense Multiple Access (CSMA) Medium Access Control (MAC) protocol, operates in a license free band (315/433/868/916 MHz) and has a bandwidth in the range 20–50 kbps. Such radios usually offer a simple byte oriented interface that permits software implementations of arbitrary (energy efficient) MAC protocols (see Section 4). Newer models support an 802.15.4 radio operating in the 2.4 GHz band and offering a 250 kbps bandwidth. The latter offers the possibility of using an internal (i.e., on-board) antenna which makes sensors more manageable and self-contained with respect to an external whip antenna. The radio range varies with a maximum of about 300 m (outdoor) for the first radio type and 125 m for the 802.15.4 radios.

Sensors are powered by batteries, usually a couple of standard AA standard batteries that can be replaced upon expiration (this is important since the day of cheap, disposable sensors is yet to come). Battery size usually determines the size of the sensor, so existing hardware is roughly a few cubic centimetres in size. An exception is represented by the Crossbow mica2dot mote [28] which uses a coin cell about the size of a quarter dollar but is also more resource constrained than larger sensors. Studies are currently under way to replace/integrate battery sources with some power scavenging methods such as solar cells but there are some reservations about the actual effectiveness of such methods. Solar cells, for instance, do not produce much energy indoor or when covered by tree foliage.

A final matter is the operating system i.e., the basic system software that application programmers can use to interact with the sensor hardware. TinyOs [122,44] is a widely used simple lightweight event-based operating system written in nesC [39] (it is used on Crossbow motes, Moteiv motes and similar devices). It supports the task concept: an execution entity that runs to completion without being preempted by other tasks and can post other tasks. Only interrupt service routines can interrupt a running task. Lengthy operations like reading from a transducer or sending a radio message are split-phase: the requesting task invokes a command that starts the operation and immediately returns. When the operation completes code from interrupt or TinyOs routines posts a notification task. Such task calls (signals) an event routine that collects results and does other chores in user space.

The command/event nature of TinyOs renders application programming rather complex and error prone. An interesting alternative comes from the Nut/OS operating system [95] that runs on Btnodes [20]. It offers non preemptive multithreading where a scheduled thread maintains processor control until it voluntarily relinquishes it, terminates or blocks on a lengthy I/O operation. Table 1 compares some existing sensor node architectures.

3. ZigBee and 802.15.4 overview

The ZigBee Alliance [139] is an association of companies working together to develop standards (and products) for reliable, cost-effective, low-power wireless networking. ZigBee technology will probably be embedded in a wide range of products and applications across consumer, commercial, industrial and government markets worldwide. ZigBee builds upon the IEEE 802.15.4 standard [50] which defines the physical and MAC layers for low cost, low rate

Table 1
Comparison for various sensor architectures

	Btnode 3	mica2	mica2dot	micaz	telos A	tmote sky	EYES
Manufacturer	Art of Technology	Crossbow	Crossbow	Crossbow	Imote iv	Imote iv	Univ. of Twente
Microcontroller	Atmel Atmega 128L	Atmel Atmega 128L	Atmel Atmega 128L	Atmel Atmega 128L	Texas Instruments MSP430	Texas Instruments MSP430	Texas Instruments MSP430
Clock	7.37 MHz	7.37 MHz	4 MHz	7.37 MHz	8 MHz	7.37 MHz	5 MHz
RAM (KB)	64 + 180	4	4	4	2	10	2
ROM (KB)	128	128	128	128	60	48	60
Storage (KB)	4	512	512	512	256	1024	4
Radio	Chipcon CC1000 315/433/868/916 MHz 38.4 Kbauds	Chipcon CC1000 315/433/868/916 MHz 38.4 Kbauds	Chipcon CC1000 315/433/868/916 MHz 38.4 Kbauds	Chipcon CC2420 2.4 GHz 250 Kbps IEEE 802.15.4	Chipcon CC2420 2.4 GHz 250 Kbps IEEE 802.15.4	Chipcon CC2420 2.4 GHz 250 Kbps IEEE 802.15.4	RFM TR1001868 MHz 57.6 Kbps
Max Range	150–300 m	150–300 m	150–300 m	75–100 m	75–100 m	75–100 m	75–100 m
Power	2 AA batteries	2 AA batteries	Coin cell	2 AA batteries	2 AA batteries	2 AA batteries	2 AA batteries
PC connector	PC-connected programming board	PC-connected programming board	PC-connected programming board	PC-connected programming board	USB	USB	Serial Port
OS	Nut/OS	TinyOS	TinyOS	TinyOS	TinyOS	TinyOS	PEEROS
Transducers	On acquisition board	On acquisition board	On acquisition board	On acquisition board	On board	On board	On acquisition board
Extras	+ Bluetooth						

personal area networks. ZigBee defines the network layer specifications for star, tree and peer-to-peer network topologies and provides a framework for application programming in the application layer. The following subsections give more details on the IEEE and ZigBee standards.

3.1. IEEE 802.15.4 standard

The IEEE 802.15.4 standard [50] defines the characteristics of the physical and MAC layers for Low-Rate Wireless Personal Area Networks (LR-WPAN). The advantages of an LR-WPAN are ease of installation, reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol stack.

3.1.1. The physical layer

The physical layer supports three frequency bands: a 2450 MHz band (with 16 channels), a 915 MHz band (with 10 channels) and a 868 MHz band (1 channel), all using the Direct Sequence Spread Spectrum (DSSS) access mode. The 2450 MHz band employs Offset Quadrature Phase Shift Keying (O-QPSK) for modulation while the 868/915 MHz bands rely on Binary Phase Shift Keying (BPSK). Table 2 summarizes the main features of the three bands. Besides radio on/off operation, the physical layer supports functionalities for channel selection, link quality estimation, energy detection measurement and clear channel assessment.

3.1.2. The MAC layer

The MAC layer defines two types of nodes: Reduced Function Devices (RFDs) and Full Function Devices (FFDs). FFDs are equipped with a full set of MAC layer functions, which enables them to act as a network coordinator or a network end-device. When acting as a network coordinator, FFDs send beacons that provide synchronisation, communication and network join services. RFDs can only act as end-devices and are equipped with sensors/actuators like transducers, light switches, lamps, etc. They may only interact with a single FFD. Two main types of network topology are considered in IEEE 802.15.4, namely, the star topology and the peer-to-peer topology. In the star topology, a master–slave network model is adopted. A FFD takes up the role of PAN coordinator; the other nodes can be RFDs or FFDs and will only communicate with the PAN coordinator. In the peer-to-peer topology, a FFD can talk to other FFDs within its radio range and can relay messages to other FFDs outside of its radio coverage through an intermediate FFD, forming a multihop network. A PAN coordinator is selected to administer network operation.

Table 2
Radio front-end and physical layer specification

	2450 MHz	915 MHz	868 MHz
Gross data rate	250 kbps	40 kbps	20 kbps
No. of Channel	16	10	1
Modulation	O-QPSK	BPSK	BPSK
Chip pseudo-noise sequence	32	15	15
Bit per symbol	4	1	1
Symbol period	16 μs	24 μs	49 μs

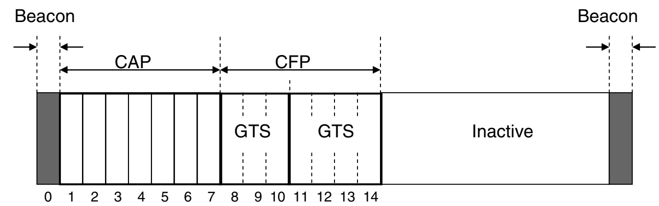


Fig. 1. MAC super-frame.

The PAN coordinator may operate its PAN with a superframe or without it. In the first case it starts the superframe with a beacon serving for synchronization purposes as well as to describe the superframe structure and send control information to the PAN. The superframe (see Fig. 1) is divided into an active and an inactive portion (where the PAN coordinator may go to sleep and save energy). The active portion is divided into fixed size slots and contains a Contention Access Period (CAP), where nodes compete for channel access using a slotted CSMA-CA protocol, and a Contention Free Period (CFP), where nodes transmit without contending for the channel in Guaranteed Time Slots (GTS) assigned and administered by the PAN coordinator. When an end-device needs to send data to a coordinator (non GTS) it must wait for the beacon to synchronize and later contend for channel access. On the other hand, communication from a coordinator to an end-device is indirect. The coordinator stores the message and announces pending delivery in the beacon. End-devices usually sleep most of the time and wake up periodically to see if they have to receive same messages from the coordinator by waiting for the beacon. When they notice that a message is available, they request it explicitly during the CAP. When a coordinator wishes to talk to another coordinator it must synchronize with its beacon and act as an end-device.

The other option for PAN communication is to do without a superframe. The PAN coordinator never sends beacons and communication happens on the basis of unslotted CSMA-CA. The coordinator is always on and ready to receive data from an end-device while data transfer in the opposite direction is poll-based: the end device periodically wakes up and polls the coordinator for pending messages. The coordinator then sends these messages or signals that none is available. Coordinator to coordinator communication poses no problems since both nodes are active all the time.

In addition to data transfer, the MAC layer offers channel scan and association/disassociation functionalities. The scan procedure involves scanning several logical channels by sending a beacon request message and listening (active scan, for FFDs) or just listening (passive scan, for RFDs) for beacons in order to locate existing PANs and coordinators. Higher layers decide which PAN to join and later ask the MAC layer to start an association procedure for the selected PAN. This involves sending a request to a coordinator and waiting the corresponding acceptance message.

If accepted in the PAN, the node receives a 16-bit "short" address that it may use later in place of the 64-bit "extended" IEEE address.

3.2. The ZigBee standard

ZigBee [139] standardizes the higher layers of the protocol stack. The network layer (NWK) is in charge of organizing and providing routing over a multihop network (built on top of the IEEE 802.15.4 functionalities), while the Application Layer (APL) intends to provide a framework for distributed application development and communication. The APL comprises the Application Framework, the ZigBee Device Objects (ZDO), and the Application Sub Layer (APS). The Application Framework can have up to 240 Application Objects, that is, user defined application modules which are part of a ZigBee application. The ZDO provides services that allow the APOs to discover each other and to organize into a distributed application. The APS offers an interface to data and security services to the APOs and ZDO. An overview of the ZigBee protocol stack is shown in Fig. 2.

3.2.1. The network layer

ZigBee identifies three device types. A ZigBee end-device corresponds to an IEEE RFD or FFD acting as a simple device. A ZigBee router is an FFD with routing capabilities. The ZigBee coordinator (one in the network) is an FFD managing the whole network. Besides the star topology (that naturally maps to the corresponding topology in IEEE 802.15.4), the ZigBee network layer also supports more complex topologies like the tree and the mesh. Fig. 3 shows examples of these topologies. Among the functionalities provided by the network layer are multihop routing, route discovery and maintenance, security and joining/leaving a network, with consequent short (16-bit) address assignment to newly joined devices.

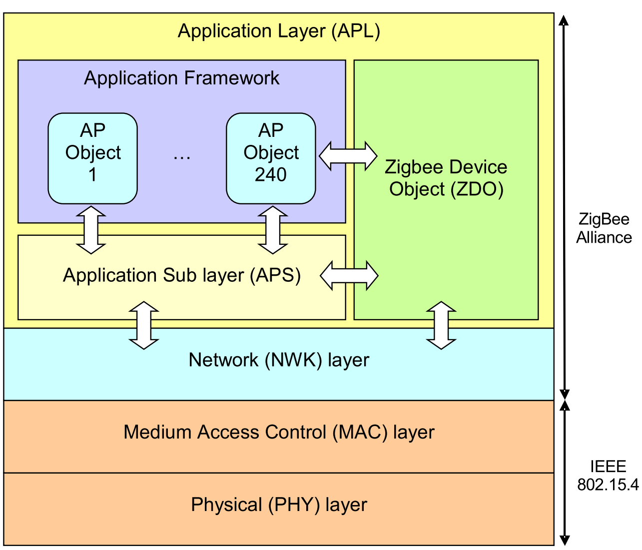


Fig. 2. ZigBee functional layer architecture and protocol stack.

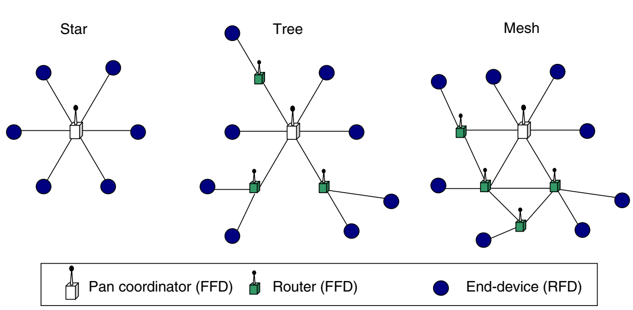


Fig. 3. Network topologies in ZigBee.

3.2.1.1. Network formation and address assignment. A Multihop network is established by means of the join procedure. When a device c wishes to join an existing network, the network layer is requested to start a network discovery procedure. With support from the MAC layer scan procedure (see Section 3.1.2), it learns about neighbouring routers that announce their networks. After the upper layer has decided which network to join (several ZigBee networks may overlap spatially, using different channels), the network layer selects a "parent" node p (in the desired network) from his neighbourhood, and asks the MAC layer to start an association procedure. Upon receiving an indication of the association request from the MAC layer, p's network layer assigns c a 16-bit short address and lets the MAC layer successfully reply to the association request. Node c will use the short address for any further network communication.

Parent-child relationships established as a result of joins, shape the whole network in the form of a tree with the ZigBee coordinator as the root, the ZigBee routers as internal nodes and ZigBee end-devices as leaves. This tree structure is also at the basis of the distributed algorithm for network address assignment. The ZigBee coordinator fixes the maximum number of routers (R_m) and end-devices (D_m) that each router may have as children and also fixes the maximum depth of the tree (L_m). On the basis of its depth in the tree, a newly joined router is assigned a range of consecutive addresses (16-bit integers). The first integer in the range becomes the node address while the rest will be available for assignment to its children (routers and end-devices). The size $A(d)$ of the range of addresses assigned to a router node at depth $d < L_m$ is defined by the following recurrence:

$$A(d) = \begin{cases} 1 + D_m + R_m & \text{if } d = L_m - 1 \\ 1 + D_m + R_m A(d+1) & \text{if } 0 \leqslant d < L_m - 1 \end{cases}$$

Nodes at depth L_m and end-devices are obviously assigned a single address. The recurrence is easily solved and used by each router to assign addresses to its children. Assume that a router at depth d receives the range of addresses $[x, x + A(d))$. It will have address x and it will assign range $[x + (i-1)A(d+1) + 1, x + i + A(d+1)]$ to its i-th router child ($1 \leqslant i \leqslant R_m$) and address $x + R_m A(d+1) + j$ to its j-th end-device child ($1 \leqslant j \leqslant D_m$). Fig. 4 depicts an example network with $R_m = 2$, $D_m = 2$ and $L_m = 3$ where all addresses have been assigned to routers (white nodes) and

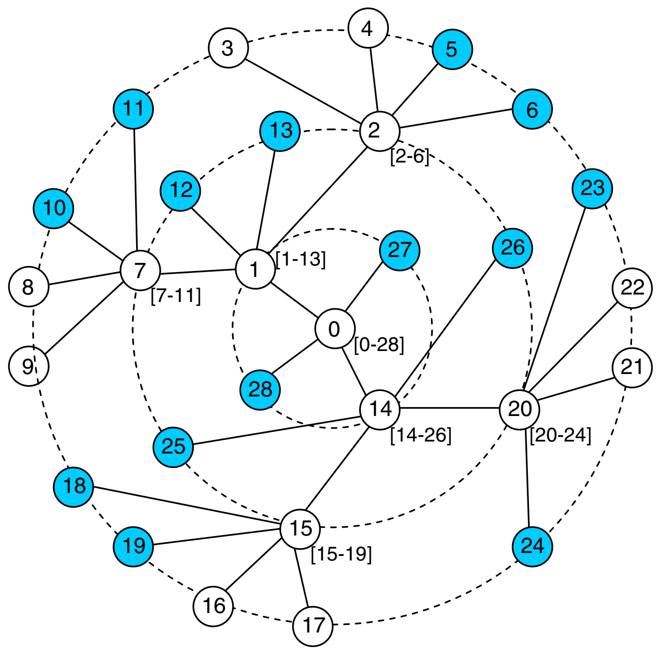


Fig. 4. Address allocations for $R_m = 2$, $D_m = 2$ and $L_m = 3$.

end-devices (gray nodes). The address appears inside the circle representing each node, while the assigned address ranges are displayed in brackets next to each router.

3.2.1.2. Routing. The routing algorithm depends on the topology used in the sensor network. In a tree topology routing can only happen along the parent-child links established as a result of join operations (this is called "tree-based routing"). Routers maintain only their address and the address information associated with their children and parent. Given the way addresses are assigned, a router that needs to forward a message can easily determine whether the destination belongs to a tree rooted at one of its router children or is one of its end-device children. If so, it routes the packet to the appropriate child; otherwise it routes the packet to its parent. This kind of routing algorithm is not necessarily the most energy-efficient but is very simple to implement and allows routers to operate in a beacon-enabled network. In other words, all ZigBee routers (and the ZigBee coordinator) send beacons, communicate via a slotted CSMA-CA protocol (as described in Section 3.1.2) and sleep in the inactive portion of their superframe. The trick is to have short active portions as compared to the beacon interval and have neighbouring routers start their superframe suitably offset with respect to one other to avoid overlapping. Communication from a child to a parent happens in the CAP (Contention Access Period) of the parent while communication from a parent to a child is indirect. In any case a node has to synchronize with the parent's beacon to exchange data with it, while it drives communication with its children according to its superframe.

The mesh network topology is more complex to handle and beaconing is not allowed but is more robust and resilient to faults. Routers maintain a routing table (RT) and employ a route discovery algorithm to construct/update these data structures on the path nodes. A routing table entry is described in Table 3.

Table 3
Routing table in ZigBee

Field Name	Description
Destination Address	16-bit network address of the destination
Next-hop Address	16-bit network address of next hop towards destination
Entry Status	One of Active, Discovery or Inactive

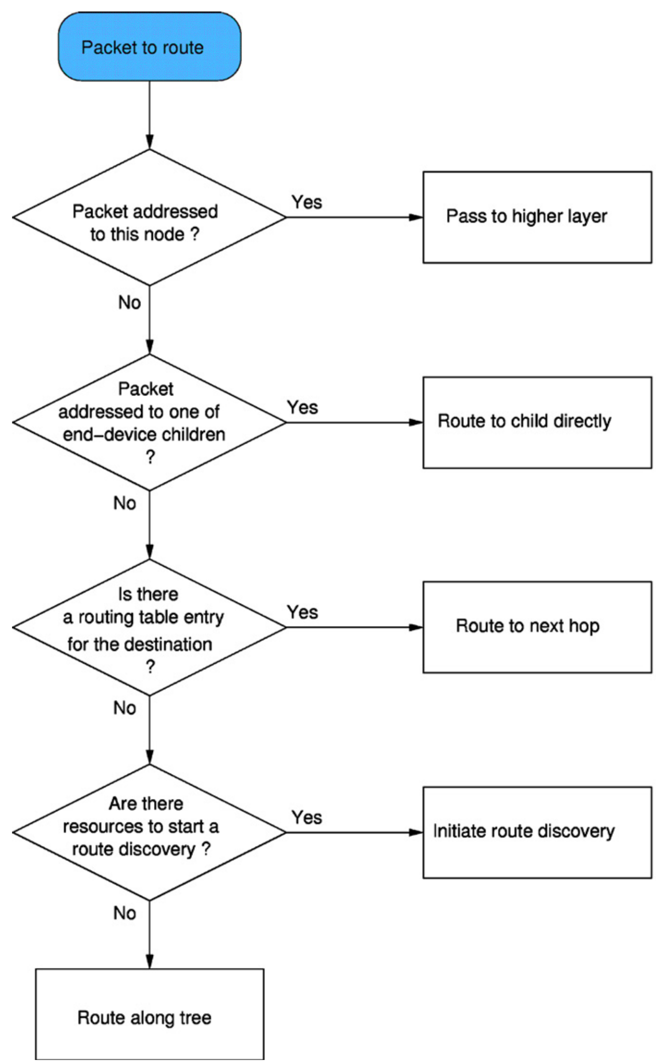


Fig. 5. A sketch of the routing protocol.

Fig. 5 illustrates a simplified version of the algorithm used to route a packet. As can be seen, when trivial routing is not possible, the routing table is consulted for the next hop to the destination. If no entry addresses the given destination, the network layer attempts to start the route discovery procedure and in case sufficient resources are not available it falls back to tree-based routing.

3.2.1.3. Route discovery. Route discovery is a process required to establish routing table entries in the nodes

Table 4
Content of the Route Discovery Table

Field Name	Description
RREQID	Unique ID (sequence number) given to every RREQ message being broadcasted
Source Address	Network address of the initiator of the route request
Sender Address	Network address of the device that sent the most recent lowest cost RREQ
Forward Cost	The accumulated path cost from the RREQ originator to the current device
Residual Cost	The accumulated path cost from the current device to the RREQ destination

along the path between two nodes wishing to communicate. A Route Discovery Table (RDT) is maintained by routers and the coordinator to implement route discovery. Table 4 illustrates the contents of one of its entries.

Route discovery in ZigBee is based on the well-known Ad hoc On Demand Distance Vector routing algorithm (AODV) [100]. When a node needs a route to a certain destination, it broadcasts a route request (RREQ) message that propagates through the network until it reaches the destination. As it travels in the network, a RREQ message accumulates (in one of its fields) a forward cost value that is the sum of the costs of all the links it traversed. The cost of a link can be set to a constant value or be dynamically calculated based on a link quality estimation provided by the IEEE 802.15.4 interface. Each RREQ message carries a RREQ ID which the originator increments every time it sends a new RREQ message. This way the RREQ ID and source address can be used as a unique reference for a route discovery process. Reception of a RREQ triggers a search within the RDT for an entry matching the route discovery. If no match is found, a new RDT entry is created for the discovery process and a route request timer is started (upon timer expiration the RDT entry will be removed). Conversely if an entry is found in the RDT, the node compares the path cost for the RREQ message and the corresponding value in the RDT entry. If the former is higher it drops the RREQ message, otherwise it updates the RDT entry. Finally, if the node is not the route discovery destination, it allocates an RT entry for the destination, with status Discovery, and rebroadcasts the RREQ after updating its path cost field. If the node is the final destination, is replies to the originator with a route reply (RREP) message that travels back along the path. Fig. 6 shows a block diagram illustrating RREQ processing.

The RREP message is addressed to the route discovery originator and carries with it a residual cost value field that each node increments as it forwards the message. Upon receipt of a route reply (RREP) message, a node retrieves the RDT and RT entries for the associated route discovery. If the node is the RREQ originator and this is the first RREP it received, it sets the RT entry to Active and records the residual cost and next hop in the RDT entry. In all other cases it compares the residual cost from the RREP with the one from the RDT entry. If the former is higher the node discards the RREP message; otherwise it updates the RDT

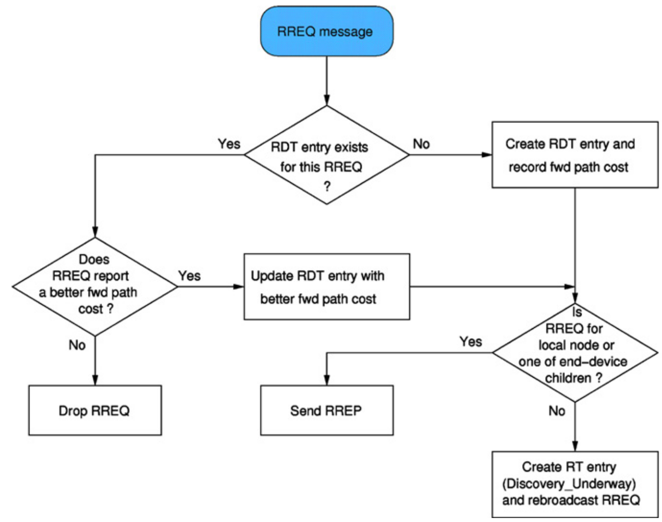


Fig. 6. The RREQ processing.

entry (residual cost) and the RT entry (next hop). A node that is not the RREP originator must also forward the RREP towards the originator. Note that intermediate nodes never change the RT entry status to Active as a result of receiving a RREP message. They will only change the entry status upon reception (and routing) of a data message for the given destination. Fig. 7 illustrates the RREP message processing.

3.2.2. The application layer

A ZigBee application consists of a set of Application Objects (APOs) spread over several nodes in the network. An APO is a piece of software (from an application developer) that controls a hardware unit (transducer, switch, lamp) available on the device. Each APO is assigned a locally unique endpoint number that other APOs can use as an extension to the network device address to interact with it. The ZigBee Device Object (ZDO) is a special object which offers services to the APOs: it allows them to discover devices in the network and the service they implement. It also provides communication, network and security management services. The Application Sublayer (APS) provides data transfer services for the APOs and the ZDO. Fig. 2 illustrates the various components in the Application Layer.

A ZigBee application must conform to an existing (ZigBee Alliance-accepted) application profile. An application profile defines message formats and protocols for interactions between APOs that collectively form a distributed application. The application profile framework allows different developers to independently build and sell ZigBee devices that can interoperate with each other in a given application profile. Each APO encapsulates a set of attributes (data entities representing internal state, etc.) and provides functionalities (services) for setting/retrieving values of these attributes or being notified when an attribute value changes. In the context of a profile, a group of related attributes is termed a "cluster" and identified with a numeric id. Typically a cluster represents a sort of interface (or part of it) of the APO to the other APOs.

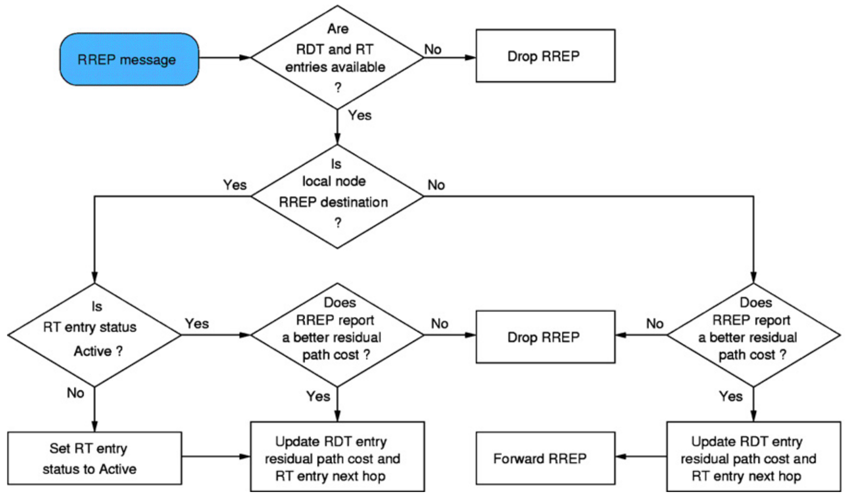


Fig. 7. The RREP processing.

The application profile must specify one of two possible communication service types. For the Key Value Pair (KVP) service type the ZigBee standard has predefined message layouts which must be suitably filled by APOs to request a given operation on attributes residing on a remote APO. The interactions between APOs is limited by the operations supported on attributes. The Generic Message service type is suitable for applications that do not fit in the KVP service type and leaves responsibility to the application profile for specifying message types and their contents.

A special application profile, named the Device Profile, must be implemented by all nodes in a ZigBee network. The object responsible for this profile is the ZDO. The Device Profile requires its implementing objects (ZDOs) to support device/service discovery procedures wherein a node attempts to discover existing nodes in the network, active endpoints on some node and/or the services they implement (available cluster ids).

Discovery procedures are crucial to APO addressing. In direct addressing mode a message is addressed to a specific destination address (16-bit network address) and endpoint number and the sending node is responsible for discovering both via the ZDO discovery services. Indirect addressing mode only requires the sender to supply a cluster id but needs support from a neighbouring (or local) ZigBee router (or coordinator) to locate the destination node(s) for the message. This is possible thanks to the APS of the ZigBee router that maintains a binding table associating (source address, source endpoint, cluster id) tuples to a list of (destination address, destination endpoint) tuples, one for each device the message must reach. A message sent by an end-device with indirect addressing reaches the parent node. Here the APS consults its binding table in order to determine the actual destinations and send them appropriate messages with direct addressing. Adding and removing entries in the binding table is commanded by the ZDO in response to local/remote binding requests, as defined in the Device Profile.

3.3. Security in ZigBee

Security services provided for ZigBee include methods for key establishment, key transport, frame protection, and device management [139]. The ZigBee Alliance describe the security functionalities based on an open trust model for a device whereby the different layers of the communication stack and all applications running on a single device trust each.

The ZigBee specifications provide different means to achieve the following security requirements:

- *Freshness:* ZigBee devices maintain incoming and outgoing freshness counters to maintain data freshness.These counters are reset every time a new key is created. Devices that communicate once per second will not overflow their freshness counters for 136 years.
- *Message Integrity:* ZigBee specifications provide options of providing 0-, 32-, 64- or 128-bit data integrity for the transmitted messages. The default is 64-bit integrity.
- *Authentication:* Network level authentication is achieved by using a common network key. This prevents outsider attacks while adding very little in memory cost. Device level authentication is achieved by usingunique link keys between pairs of devices. This prevents insider and outsider attacks but has higher memory cost.
- *Encryption:* ZigBee uses 128-bit AES encryption. Encryption protection is possible at network level or device level. Network level encryption is achieved by using a common network key. Device level encryption is achieved by using unique link keys between pairs of devices. Encryption can be turned off without impacting freshness, integrity, or authentication as some applications may not need any encryption.

The ZigBee architecture includes security mechanisms at the MAC, NWK and APS Layers of the protocol stack. Furthermore, the APS sub-layer provides services for the establishment, and maintenance of security relationships.

The ZigBee Device Object (ZDO) manages the security policies and the security configuration of a device [139]. The following architectural design choices for security are made in ZigBee specifications:

- The layer that originates a frame is responsible for initially securing it. For example, the MAC layer frames and NWK command frames are secured by MAC layer security and Network Layer security respectively.
- NWK layer security shall be used for all frames except those communicated between a router and a newly-joined device until this newly joined device receives the Network key. A device can only send messages over multiple hops after the sender device successfully joins the network and receives the Network key.
- The open trust model allows the re-use of the same keying material among the different layers on the same device thereby providing end-to-end security on a device-to-device basis rather than between pairs of particular layers on two communicating devices. Reuse of keys helps reduce storage costs.
- End-to-end security is provided where secret key is shared only between the source and destination devices. Additionally, this ensures that routing of messages between devices can be realized independent of trust considerations.
- The security level used by all devices in a given network and by all layers of a device shall be the same. If anapplication needs more security than is provided by a given network, it shall form its own separate network with a higher security level.

3.3.1. Security keys

ZigBee devices use 'link keys' and 'network keys' to secure data communication in the network. A 128-bit link key shared between two ZigBee enabled devices is used to secure all unicast communications between peer entities. On the other hand, all broadcast communications in the network are secured using a 128-bit Network Key which is shared among all devices in the network.

The security between devices hence depends on the secure initialization and installation of these keys. A master key is used for the generation of the link keys. The master key may be pre-installed in the factory, sent out-of-band or even sent from the trust centre. The Link and the Network keys may also be pre-installed in the factory, but this would not provide high security for the network, as if any device is attacked by an adversary, the link/network key would be released and the adversary would be able to easily attack the whole network. A possible method of obtaining the link key suggested by the ZigBee specification is to use Symmetric-key Key Establishment (SKKE) protocol handshaking between the two devices. Both the link key and the network key have an option of being able to be transported from the trust centre.

In a secured network there are a variety of security services like re-keying of keys available to avoid any re-use of keys across different security services. The Network key may be used by the MAC Layer, NWK Layer, and APL layer. The same Network key and associated outgoing and incoming frame counters shall be available to all of these layers. On the other hand, the link and master keys may be used only by the APS sub-layer.

3.3.2. Security trust centre

The ZigBee specification defines the role of a trust centre as a device that would be trusted by all other devices on the network. The trust centre would distribute keys for the purpose of network and end-to-end application configuration management [139]. Each network shall have no more than a single trust centre. Each device on any given network shall be associated to no more than one trust centre. The trust centre application can be configured to operate in either commercial or residential mode of operation. The commercial mode of the trust centre provides high-security for commercial applications. On the other hand, the residential mode is designed for low-security residential applications.

In the commercial mode, the trust centre shall maintain a list of all devices, link keys, master keys, and Network keys that it needs to control. The trust centre establishes and maintains keys and freshness counters with every device in the network. This allows centralized control and update of the security keys. It would also enforce the policies required for Network key updates and network access control. Larger the number of devices and keys for the network that the trust centre need to keep track of, larger is the memory required in the trust centre to save this information.

In the residential mode, the trust centre may maintain a list of devices and the master/link keys with all the devices in the network. The trust centre shall also maintain the Network key and the controls policies for network access control. In contrast to the commercial mode for the residential mode, the memory required for the trust centre does not scale with the number of device in the network.

For the commercial mode devices are usually preloaded with the address of the trust centre and the initial master key. On the other hand for residential mode, the communication between the any device and the trust centre is based on the Network key which can be either preconfigured or sent via an in-band unsecured key transport. The trust centre provides the following three functions:

- *Trust Manager:* The trust manager is responsible to identify and authenticate the device that request to join the network.
- *Network Manager:* The network manager is responsible to maintain and distribute the network keys to the devices that it manages.
- *Configuration manager:* A configuration manager is responsible for binding two peer applications and enabling end-to-end security between devices it manages.

3.3.3. MAC layer security

To provide security for the MAC Layer frames, ZigBee would use MAC Layer security specified in the 802.15.4 specifications [50]. This will be used to secure the MAC Layer command, beacon, and acknowledgement frames. Securing MAC Layer data frames only provides security for messages transmitted over a single hop. But to provide security for multi-hop messages, ZigBee would rely on higher layer security, e.g. NWK Layer security. The MAC layer uses the Advanced Encryption Standard (AES) as its core cryptographic algorithm and describes a variety of security suites that use the AES algorithm. The MAC layer does the security processing, but the upper layers, which set up the keys determine the security levels to use. Fig. 8 shows ZigBee outgoing frame structure with the security fields used to provide MAC Layer security. As can be seen from the Figure, the MAC Layer adds an auxiliary header along with the MAC Layer header for carrying security information. The message integrity code (MIC) may take the values 0, 32, 64 or 128 and determines the level of data integrity.

When the MAC layer transmits (receives) a frame with security enabled, it looks at the destination (source) of the frame, retrieves the key associated with that destination (source), and then uses this key to process the frame according to the security suite designated for the key being used. Each key is associated with a single security suite and the MAC Layer frame header has a bit that specifies whether security for a frame is enabled or disabled. The security processing of the outgoing and incoming MAC Layer frames with MAC Layer security is explained in [139].

3.3.4. NWK layer security

Like the MAC layer, the NWK layer's frame protection mechanism shall make use of the Advanced Encryption Standard (AES). The NWK layer will broadcast route request messages and process received route reply messages to provide support for multi-hop routing of messages. Route request messages are simultaneously broadcast to nearby devices and route reply messages originate from nearby devices. If the appropriate link key is available, the NWK layer shall use the link key to secure outgoing NWK frames [139].

Fig. 9 shows the security fields that are present when NWK Layer security is used to secure a NWK frame. As can be seen from the Figure, the NWK Layer adds an auxiliary header along with the NWK header for carrying security information. The MIC determines the level of data integrity provided.

Another case may arise when the appropriate link key is not available. In this case the NWK layer shall use its active Network key to secure outgoing NWK frames in order to secure the messages while for the incoming NWK frames, either the active or the alternate Network key is used to secure incoming NWK frames. The security processing of the outgoing and incoming NWK frames with NWK Layer security is explained in [139].

3.3.5. APS layer security

The APS sublayer performs the security functions to provide security for the frames originating at the APL Layer. The APS layer frame security is based on link keys or the Network key. Fig. 10 shows the APL Layer frame with the security fields present when APL Layer security is applied. It can be seen in the Figure that the APS sublayer adds an auxiliary header along with the APS header for carrying security information. Here also the MIC is used which determines the level of data integrity provided.

The APS layer has to also provide applications and the ZDO with key establishment, key transport, and device management services. The security processing of the outgoing and incoming APS frames with APS Layer security is explained in [139].

Some of the security services provided by the APS Sublayer are briefly explained below:

- *Key establishment*: The secret key called the Link Key shared between two ZigBee devices is derived using the mechanism specified by the APS sublayer's key establishment service [139]. Key Establishment involves two

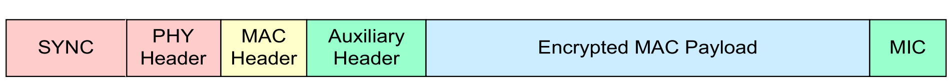


Fig. 8. ZigBee frame with MAC layer security.

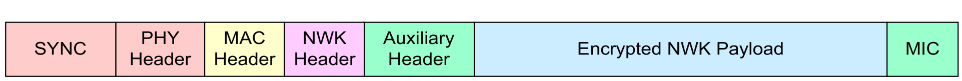


Fig. 9. ZigBee frame with NWK layer security.

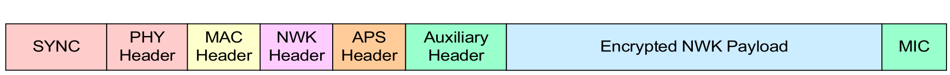


Fig. 10. ZigBee frame with APS layer security.

entities, an initiator device and a responder device. The two devices share a master key that would be later used to generate the link key. This master key may be pre-installed during manufacturing, may also be installed by a trust centre, or may be based on user-entered data (e.g., PIN, password, or key). The key-establishment protocol involves three conceptual steps: the exchange of temporary data, the use of this temporary data and the master key to derive the link key, and the confirmation that this link key was correctly computed.

- *Transport key*: The transport-key service provides the means to transport a key between devices securely or insecurely. The secured transport-key command provides a means to transport a master, link or Network key from a key source (e.g., trust center) to other devices. The unsecured transport-key command provides a means for loading a device with an initial key. In this case, the security of the transported key can be realized by non-cryptographic means, e.g., by communicating the command via an out-of-band channel [139].
- *Update device*: The update-device service provides a secure means for a given device to inform another device that a third device has had a change of status that must be updated. A change of status may refer to the device joining or leaving the network. This is the mechanism by which the trust center maintains an accurate list of currently active network devices.
- *Remove device*: The remove device service provides a secure means by which a device (e.g., a trust center) may inform another device (e.g., a router) to remove a connected device from the network that has not satisfied the trust center's security requirements for network devices.
- *Switch key*: The switch-key service provides a secure means for a device to inform another device that it should switch to a different active Network.
- *Request key*: The request-key service provides a secure means for a device to request the current Network key, or an end-to-end application master key, from another device.

4. Energy efficiency

Energy efficiency is probably the most important issue in Wireless Sensor Networks (WSNs). Since future sensor nodes are envisioned as disposable, it is extremely important to develop techniques that prolong battery lifetime as much as possible. Unnecessary energy consumption must be avoided by attentive hardware/component design as well as low level (i.e., operating system and support middleware) and high level (i.e., application) software programming. Recent commercial sensor devices provide a high level of flexibility allowing programmers to selectively turn on/off the various hardware components, including transducers, ADC and the radio. While transducer activation can potentially be handled locally to a node, radio operation requires coordination among neighbouring nodes and more generally network-wide since it is needed for communication.

The radio transceiver is also the most power-hungry of all the devices available on a typical sensor node with transmit and receive mode operation having similar power consumption (Table 5 reports radio current consumption for some motes from Crossbow). As a consequence the "radio always on" solution is unacceptable and different approaches to radio resource management have been investigated in the literature. The major reason for energy waste is idle listening, where a node is listening to the radio channel, waiting for something. Other reasons include packet collisions, overhearing a packet destined to another node and control packet overhead [132].

Several proposals attempt to reduce energy waste at the MAC layer but recent research recognizes the importance of coordination with the higher layers of the protocol stack (Network and Application layers) to adapt radio usage to application communication patterns and achieve higher energy savings. In the following we review and categorize different approaches to radio energy conservation. Section 7.2 discusses database applications for sensor networks where radio as well as transducer consumption are accounted for in order to schedule operations on sensor nodes.

4.1. Connected dominating set approaches

One of the first proposals follows from the observation that in dense networks many close-by nodes are equivalent from a routing point if view. The idea of Connected Dominating Set (CDS) approaches is to select some of the nodes to constitute a network backbone and be active all the time providing network connectivity and temporarily storing messages for neighbouring non-backbone nodes. Non-backbone nodes sleep most of the time (saving energy) and periodically wake up to exchange messages with their backbone node neighbour. Since backbone nodes consume more energy than the other nodes, CDS protocols require nodes to alternate between backbone and non-backbone status.

GAF [128] and Span [27] are two examples of CDS protocols. In the former, nodes rely on GPS to identify their location and to partition the network into a grid. A distributed algorithm takes care of electing a leader in each grid area and nodes alternate in states active (or grid leader), sleeping (or non grid leader) and discovery where they evaluate the possibility of taking over grid leadership. Grid

Table 5
Radio current consumption in mA for Crossbow motes

Mode	mica2	mica2dot	micaz
Rx	9 mA	9 mA	18.8 mA
Tx (0dBm)	15 mA	15 mA	17.4 mA
Power Down	10^{-3} mA	10^{-3} mA	10^{-2} mA

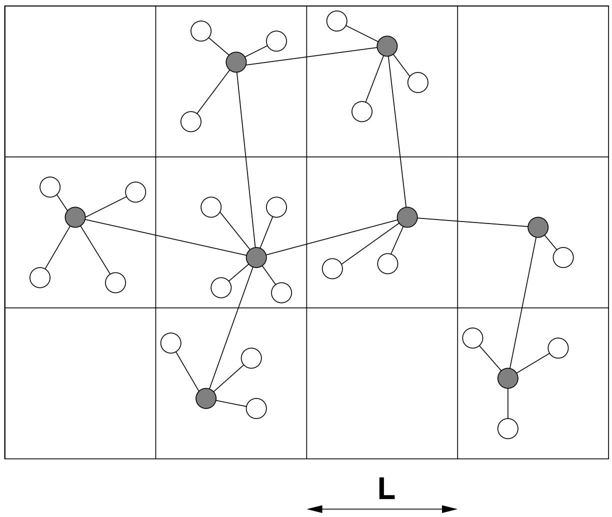


Fig. 11. In GAF the sensor field is divided into square grids with side $L = r/\sqrt{5}$ where r is the radio range, so that arbitrarily located grid leaders (gray nodes) of adjacent grids can talk to each other. Non grid leaders (white nodes) send to/receive from their grid leader while inter-grid routing is handled by grid leaders.

leaders use a standard ad hoc routing protocol like AODV [100] or DSR [52]. Fig. 11 outlines GAF grid partitioning.

Span does not use GPS or location information to build and maintain the backbone. There is no fixed network partition as in GAF but nodes periodically determine whether to join, leave or stay in the backbone on the basis of neighbourhood connectivity information. Specifically each node evaluates a utility measure related to the number of pairs of its neighbours that would become connected if it were a backbone node. A randomized method that takes into account utility as well as remaining energy and the number of neighbouring backbone nodes is used to decide on transitioning to backbone state.

Backbone nodes may also employ some other energy efficient protocol (see below) to avoid running their radios all the time, provided they are able to maintain network connectivity and exchange data with neighbouring non-backbone nodes.

4.2. MAC layer approaches

MAC layer solutions attempt to achieve energy savings by exclusive use of medium access control facilities, so that higher layers in the protocol stack are unaffected and unaware of this. Such solutions are generally inflexible to different, path specific, data rates (this information is only available to higher layers) and suffer from fixed minimum overhead. Potentially large latencies over multihop paths are generally unavoidable since MAC activity coordination can only happen locally between two neighbours.

4.2.1. Slot-based protocols

In slot-based protocols [134] time is divided into periods each containing a certain number of fixed size lots. Nodes stay active in a certain predefined subset of the slots where they send beacons announcing their schedule (in relative time units) and listen for communication requests from neighbours. Activation schedules can be found such that any two neighbouring nodes eventually can hear each other's beacons. Fig. 12, adapted from [134], illustrates that with a period of 7 slots and activation schedule of the form 1101000 (where 1s represent active slots and 0s represent inactive slots) for all nodes in the network, any two neighbours can hear each other (they have at least one overlapping active slot) in the hypothesis that clocks are not synchronized but slots fully overlap. The previous assumption is not really needed and neighbours can actually hear each other even if slots do not fully overlap in time. Nodes hearing each other's beacons can keep track of their respective activation slots and wait for one of them to send data to neighbours. A suitable activation schedule does not necessarily exist for any values of the number of slots t, number of active slots k and minimum number of overlapping active slots m. However, it has been proved that if the number of active slots is $k = q + 1$ where q is a power of a prime number, then an activation schedule exists for $t = q^2 + q + 1$ slots and $m = 1$ overlapping active slots. For a given number of slots t in the period, the larger the value of m, the lower the latency for hop-to-hop (and multihop) communication but energy consumption will be higher. Also slot activation is irrespective of the number of neighbours and actual data rates.

4.2.2. TDMA protocols

The obvious solution to idle listening and MAC layer contention issues is to schedule transmissions a priori so that any node exactly knows when it must turn on its radio and no collisions can ever result. In classic TDMA protocols all nodes can see each other and a master starts a superframe providing synchronization timing for network operation. The superframe contains a sequence of slots that may be statically or dynamically allocated. Small portions of the superframe are used for master to slaves control communication

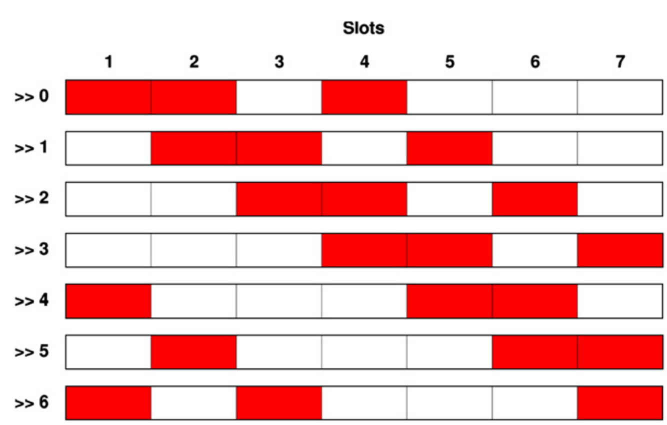


Fig. 12. Any two neighbours can eventually hear each other if they use a 1101000 activation schedule since they always have at least one overlapping slot.

(such as slot assignment) and for slave to master communication including slot requests or for slot reservation competition. This setting is suitable for small single-hop networks but its multi-hop extension poses serious problems, including achieving the required strict time synchronization and allowing multiple concurrent transmissions if sufficiently far away from each other (Spatial TDMA).

IEEE 802.15.4 MAC supports a superframe-based operating mode where a slotted CSMA-CA contention access period is followed by a contention free period where time slots are assigned by a network coordinator and by an inactive portion where nodes con sleep. ZigBee actually employs this superframe mode for star topologies, where nodes can only talk to the coordinator, and tree topologies, where internal nodes coordinate their own star networks and, at the same time, act as a slave in their parent's star network. Internal nodes must start their superframe suitably offset from their parent's in order to avoid overlapping of their active portions. IEEE 802.15.4 and ZigBee are discussed in detail in Section 3.

TRAMA [104] is another TDMA-based protocol that elaborates on NAMA [11] to turn it into an energy efficient protocol for WSNs. Each node has knowledge of its 2-hop neighbourhood and uses a (hash) priority function (based on slot number and node id) to compute its *winning slots*. A winning slot is a slot for which it has the highest priority within its 2-hop neighbourhood and where it can freely transmit without experiencing interferences. TRAMA assumes low data rates compared to slot size to compensate for clock drifts, resulting from a low resolution network clock synchronization algorithm. Random access periods consisting of small contention access slots alternate with scheduled periods of several larger time slots. Contention slots are used to propagate 1-hop neighbourhood information while scheduled slots are used for actual data transfer. After having determined the winning slots over a certain schedule interval, a node uses the last winning slot from the current schedule interval to announce which winning slots it plans to use in the next schedule interval and the intended receivers. If a node is not indicated as a receiver for a packet during a certain scheduled slot it can sleep, saving energy. Similarly, a node can sleep during all its winning slots it decides to give up. Nodes that need more slots may compete for reuse of slots unused by legitimate winners on the basis of their priorities. Apart from the required network-wide time synchronization, the complexity of this algorithm (both computational and in terms of control message overhead) and the requirement to listen during random access periods and schedule announcement slots are deterrents for its widespread adoption.

4.2.3. S-MAC, T-MAC and DS-MAC

Another common approach is to divide time into periods of fixed duration T consisting of a radio-on active window and a radio-off sleep window. Neighbouring nodes must organize someway to exchange information about their relative active windows. In S-MAC [132] active (and sleep) windows have a fixed network-unique duration A and are divided into two parts. The first part is reserved for reception of SYNC messages from neighbours. A node informs neighbours of its schedule (the time to the next activation window) by means of periodic SYNC messages. Relative times avoid global time synchronization while periodic messages alleviate clock drift effects.

At startup a node listens for some time to receive schedules from neighbouring nodes. It adopts the schedule from a neighbour if it receives one. Otherwise it chooses one on its own and starts to advertise it in SYNC messages. If a node receives a new schedule after choosing its own, it discards the latter if it is not yet sharing it with any neighbours or starts to follow both otherwise. The above procedure attempts to coordinate nodes so that they use the same schedule but it is distributed in nature and some nodes may have to adopt multiple schedules. SYNC messages are preceded by a carrier sense contention while actual data transmission follows a RTS/CTS/DATA/ACK protocol [13] taking place during the second part of the active window of the receiving node.

Large sleep windows result in low energy consumption but multi-hop communications introduce latencies of n times the period duration, for a path of length n. The authors propose *adaptive sleeping* to halve the latencies. A node hearing a RTS or CTS from a neighbour learns the duration of the data packet and wakes up for a short time after its transmission to see if it happens to be the next hop. This way two hops can be travelled in every period.

The two main deficiencies of S-MAC are the high latency and the insensitivity to varying traffic loads, given its fixed duty cycle. Timeout MAC (T-MAC) [29] builds on S-MAC and attempts to mitigate these problems. Nodes select their schedule as in S-MAC but active windows are not fixed in duration: they may extend, adapting to different traffic rates. Every node turns its radio on at the beginning of its active window and turns if off if no activation event occurs for a certain period. Reception of messages is an activation event that prolongs the active window. Fig. 13 compares S-MAC and T-MAC.

DS-MAC [76], also based on S-MAC, starts with a system defined period length but allows a node to double or halve it dynamically depending on traffic load conditions. If the average packet reception delay is too high a node will halve its current period duration. If packet reception delay is low the node will double the period duration (Fig. 14 illustrates). In either case the active window is kept constant. As a node receives an updated schedule in SYNC messages it will adopt the new schedule itself (and advertise it in SYNC messages) if it has queued packets for the SYNC originator and the new schedule has just halved the period. This way latency will be quickly reduced over a congested path.

4.2.4. Data and signaling channel

Some researchers investigated the energy savings that can be achieved augmenting the data channel with a sepa-

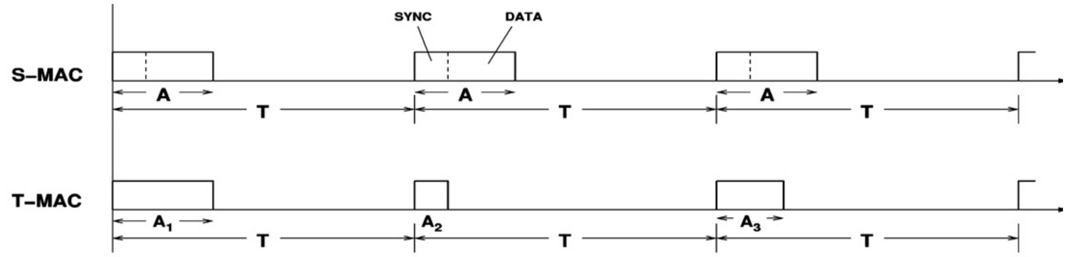


Fig. 13. S-MAC has fixed active windows while T-MAC has variable active windows that extend as long as messages are received or other activation events occur.

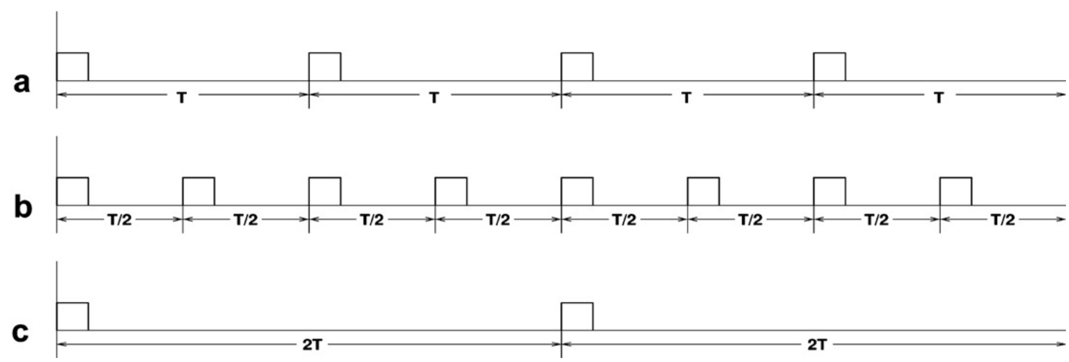


Fig. 14. DS-MAC dynamically varies the period duration depending on reception delay: if average reception delay is high a node may halve its current period duration from T to $T/2$ (a) $\rightarrow$ (b), while if average delay is low it may double it from T to $2T$(a) $\rightarrow$ (c).

rate signalling channel. The data channel is used for data and some control messages and is only turned on when required, with the signalling channel providing wakeup notifications. The signalling radio is characterized by a fixed, low duty cycle but sleeps by different nodes are unsynchronized.

In [111] two alternatives are described. In STEM-B a node announces its intention to transmit by broadcasting beacons containing the address of the destination until the latter replies with an ack. The data radio can then be used to send the message. In STEM-T a continuous busy tone is sent on the signalling channel for long enough to allow all neighbours to wake up and start listening on the data channel for the actual message. All but one of the neighbours will turn off their radio again, which is rather inefficient, especially in dense networks. On the other hand a busy tone radio is cheaper and consumes less power than the full fledged radio required for beacons in STEM-B.

Miller and Vaidya [83] improve on STEM-T suggesting that a sender may buffer outgoing messages and only start a wakeup procedure when the queue for a given destination grows beyond a certain threshold. Furthermore a sender attempts to transmit data at regular intervals (when possible) so that the receiver automatically wakes up and listens to the data radio for a short time at the expected transmission time and the wake up signal can be saved.

Apart from the unnecessary wake up overhead, equipping nodes with two radios may add up substantially to the final cost. B-MAC [101] is an extremely simple protocol that actually performs a busy tone-like signalling on the data channel using a very long message preamble, avoiding the signalling channel altogether. The preamble must be large enough to allow the receiver to wake up (according to its very low duty cycle), hear it and decide that it must stay on to receive the message. Communication overhead is shifted onto the sender: the receiver stays up only to receive messages and for very short activation intervals to detect if some neighbour is trying to reach it. B-MAC may suffer from unnecessary wakeups but its simplicity and energy efficiency make it a viable choice.

WiseMAC [31] reduces the overhead resulting from the fixed preamble length of B-MAC by requiring that nodes include the time to their next wakeup in acks to previous data messages. This way a node learns about its neighbours' wake up schedules and can prepend smaller preambles to data packets by starting its transmission a short time before the destination wakes up to sample the channel.

4.2.5. IEEE 802.15.4 energy efficiency

The IEEE 802.15.4 standard [50] supports many features that combined together result in significant power savings. However, achieving a desired data rate and maximizing the lifetime of individual sensors are often conflicting goals and are subject of ongoing research. The first dichotomy is between the beacon-enabled and the beacon-less modes. In [121] the authors study the effective path capacity of the beacon-less mode of IEEE 802.15.4 networks. They remark that the CSMA-CA scheme employed in IEEE 802.15.4 does not involve RTS/CTS exchanges as IEEE

802.11 does. As a result, unslotted CSMA-CA (used in beacon-less mode) is able to achieve higher channel utilization than slotted CSMA-CA (used in beacon-enabled mode), it allows scalability and self-organization, but it suffers from the well-known hidden terminal problem in multi-hop environments.

The hidden terminal problem [137] occurs when a receiver is within radio range of two transmitters that cannot see each other directly. In such a case if the two transmitters send packets simultaneously a collision may occur. This problem may be serious in large scale wireless network. In [49] the authors proposed a group strategy mechanism to solve the hidden terminal problem in 802.15.4 LR-WPANs. Nodes are classified into groups according to their hidden node relationship such that all nodes in each group are not hidden to each other. Guaranteed time slots are then allocated to each of these groups to access the channel using CSMA-CA. This problem was also addressed in [9], where the authors propose to use a sufficiently large contention window to guarantee an acceptably low collision probability due to hidden terminals.

The IEEE 802.15.4 unslotted CSMA-CA mode has no power saving mechanisms and it does not provide any time delivery guarantee. On the other hand the slotted mode, that adopts coordinated periodic sleeping, achieves higher energy efficiency and better copes with time delivery constrains. The performance of a beacon-enabled cluster is studied in [84,85,90] where the authors take into account service level agreements in term of reliability, device utilization and throughput.

One of the most important energy efficiency features is the possibility of turning the transceivers off most of the time and activating them only when required. Hence low overall system duty cycle would result in low average power consumption. [65] studies how such *sleeping mode* affects transmission delay. In slotted CSMA-CA, a packet might be delayed by several sleep periods when a node fails to access the channel due to contention (especially near data convergence points like the sink node) although overall traffic in the network is relatively low. This sleep delay is called *contention-inherited sleep delay* (CSD). To mitigate the CSD the authors propose a priority-based scheme that temporally separates medium access by different groups of nodes, according to packet priorities. This results in a sort of pseudo TDMA channel where each group of nodes experiences less contention compared to basic CSMA-CA, thus reducing the probability of CSD occurrence and providing bounded delays for high priority packets.

It should be mentioned that short and frequent duty cycles result in frequent and fast warm-up times [136] while long duty cycles with short active intervals require longer transceiver warm-up times and produce significant power loss due to the settling of transients in the signal path. Direct Sequence Spread Spectrum (DSSS) techniques inherently have short settling times for their wide channel filters and greater channel spacing. In these low-power systems receiver active power is often greater than transmitter active power due to the large number of active signal processing circuits in the receivers. This needs to be considered for any power consumption strategies. In a star network, the beacon-enabled mode allows the transceiver to be completely switched off up to 15/16 of the time when nothing is transmitted/received while still allowing the transceiver to be associated to the network and able to transmit or receive a packet at any time [136].

The beacon-enabled mode contention procedure starts immediately after the end of beacon transmission and introduces a significant overhead in energy consumption. However, for low data rate applications, contention access is a reasonable choice since the probability of finding the channel busy is quite low and energy savings are significant. After the beacon, a transmitter waits for a random backoff period (that depends on the symbol rate) in order to avoid collisions. The default back-off period for the slotted CSMA-CA is too short and leads to frequent collisions [137].

When a collision occurs, the backoff period is exponentially increased. When the duty cycle is very low, such a procedure could be power consuming due to the monitoring periods required by the receiver in order to support such operations during transmission data periods. For this reason, IEEE 802.15.4 supports a Battery Life Extension (BLE) mode, in which the back-off exponent is limited to the range 0-2. This greatly reduces the receiver duty cycle in low traffic rate applications. However, in dense networks, this mode results into excessive collision rates. In [18], the energy efficiency in dense IEEE 802.15.4 wireless sensor network was evaluated. The authors considered a scenario with 1600 nodes uniformly distributed in a circular area around a sink and a traffic rate of 1 byte every 8 ms per node over a single hop and evaluated which packet size leads to minimum energy per bit. On one hand, small packets require the same MAC overhead as large packets but require more transmissions, increasing the energy per useful bit. On the other hand, large packets are more subject to transmission errors, and frequently require retransmissions. In addition, when network load is high, large packets will increase channel access failure probability. Intuitively a trade-off is expected. The energy per bit decreases monotonically up to a packet payload size of 123 bytes, which is the maximum possible in IEEE 802.15.4. Allowing larger packets would allow further energy efficiency improvement, at the cost of increased latency. It has been shown that in the considered scenario, less than 50% of the energy is used for actual data transmission. A significant percentage of energy is consumed during the contention procedure (25%) and waiting for an acknowledgement (15%). This is due to the multiplicative effect of the CSMA-CA. The overhead of the contention is mainly due to the receiver start-up energy when doing clear channel assessment. The acknowledgement overhead results from the receiver power consumption when waiting for an acknowledgment.

For applications with timing constraints, timely delivery may be more crucial than energy saving. The Guaranteed

Time Slot (GTS) protocol mode is one potential candidate to achieve predictable real-time performance for Low-Rate Wireless Personal Area Networks. This mode offers the possibility of allocating/deallocating time slots in a superframe and provides predictable minimum service guarantees. From an allocation point of view, the concept of a GTS allocation is similar to a Time Division Multiplex Access (TDMA) time slot allocation. A reserved amount of bandwidth is periodically granted for a given data flow. The amount of bandwidth is determined by the duration of the time slot and its periodicity. The IEEE 802.15.4 GTS mechanism is more flexible than classic TDMA since the GTS duration may be dynamically adjusted through some parameters. The analysis in [66] gives a full understanding of the behavior of the GTS mechanism with regards to delay and throughput metrics, modelling and dimensioning an IEEE 802.15.4 cluster.

4.3. Cross layer approaches

Information form higher layers of the protocol stack can be combined with MAC layer approaches to achieve higher energy savings. The Network and the Application layer in particular have much better information on actual communication patterns, multihop data paths and associated data rates and this information can be used to obtain better radio activation schedules.

4.3.1. Network support

Zheng and Kravets [135] propose a protocol that uses Network layer information to drive a MAC layer supporting active and power-save modes, as shown in Fig. 15. In the former mode the radio is always on and operational. In the latter it operates in a low duty cycle mode and communication is possible only after the node is woken up and it transitions in the active mode. Arrival of Network layer messages (i.e., route reply messages in on demand routing protocols or path set up messages in connection oriented communication) fires a transition to active mode and starts a keep alive timer. As long as actual data messages arrive the timer is refreshed and the node remains in active mode. Timer expiration indicates that no more traffic is expected and the node may transition back to power-save mode. A drawback is that the keep alive timer is oblivious of the actual data rate that flows through the node, when in active mode (this requires Application layer support).

4.3.2. Tree-based stream scheduling

More can be done if both routing (Network) and data rate (Application) information is available for a given data flow. In trivial data gathering applications, nodes sample data from the environment and send them to the sink. In this leaf-to-root tree communication pattern, child to parent communication can be optimized by a sort of slot scheduling.

In [45] time is divided into periods each one consisting of fixed-size slots (coarse grain clock synchronization is required). A node wishing to send or forward data to the sink must reserve a slot in the parent's schedule, as depicted in Fig. 16. Once reserved, a slot data transmission suffers no collisions (rare collisions are possible due to reservation attempts by two nodes that cannot hear each other). Apart from random allocation of some idle slots for reservation purposes, nodes only need to operate their radios during used slots. A similar approach is discussed in [77]. The principal limitation of tree-based stream scheduling is that it is not suitable for arbitrary peer to peer communication which is required in more sophisticated in-network processing applications.

4.3.3. Flexible stream scheduling

Sichitiu [116] defines a more flexible dynamic scheduling approach that easily extends to peer-to-peer communication and is not limited to fixed size slots. Protocol operation contemplates two phases for each data stream: a Setup/Reconfiguration phase and a Steady State phase (Fig. 17).

In the first, a data path is established with the help of the Network layer and a RTS/CTS/RouteSetup/ACK exchange takes place to define schedules on each of the path edges $u_i \rightarrow u_j$. The interval between RTS send (t_1) and ACK reception (t_2) must not be smaller than the size of data packets in the Steady State phase. If the previous route setup message exchange terminates successfully, both link endpoints agree to reserve the time slot $[t_1, t_2]$ within each period for u_i to u_j data transfer. If u_i receives no ACK, the route setup is re-attempted later.

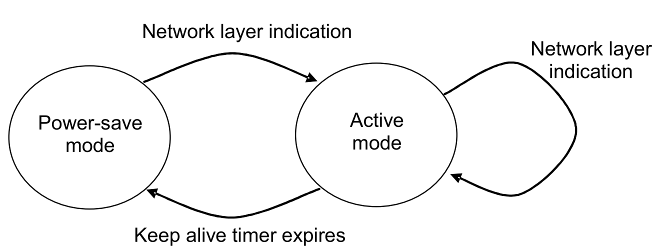


Fig. 15. Arrival of network layer messages triggers a transition from power-save to active mode and starts a keep alive timer that new network messages may refresh. Timer expiration forces a transition back to power-save mode.

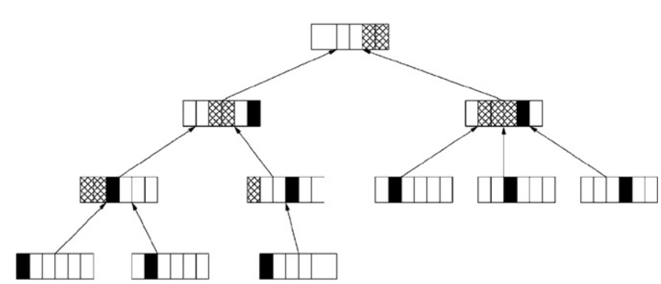


Fig. 16. Children reserve time slots in their parent schedule to guarantee the absence of collisions. Black slots indicate transmissions while hatched slots indicate reception.

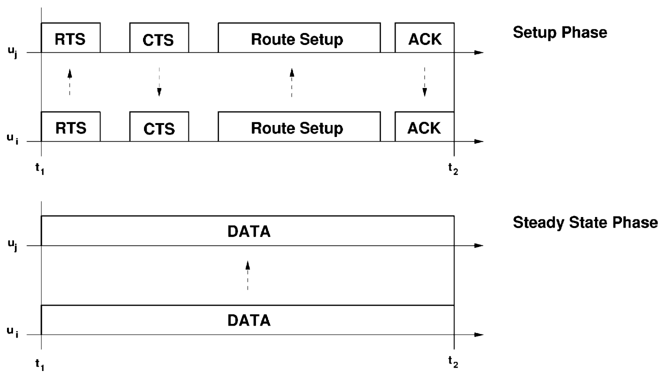


Fig. 17. Flexible stream scheduling phases: In the Setup Phase node u_i, reserves time interval $[t_1, t_2]$ with a RTS/CTS/RouteSetup/ACK protocol exchange for communication to u_j; In the Steady State Phase u_i, uses intervals $[t_1, t_2]$ to send data packets to u_j.

During the Steady State phase u_i sends its data message to u_j within the $[t_1, t_2]$ interval of every period avoiding the RTS/CTS overhead. This introduces the possibility of collision with a route setup message from another stream but the event is highly unlikely. The protocol accommodates different data rate streams by reserving multiple time slots within the same period and avoids collisions of Steady State streams providing energy efficient radio operation.

4.4. Topology control

In wireless sensor networks, the use of topology control [60] mainly focuses on two aspects: extending network lifetime by reducing node energy consumption and increasing network capacity. [109] and [110] provide a comprehensive survey on existing techniques.

In most cases individual nodes are battery powered; therefore, in order to prolong the life-time of the network, it is essential to minimize the power consumption in performing the data transactions between nodes without compromising network connectivity. This has implications on the routing protocols as, in the case where multiple routing paths are available between the source and the sink nodes, the shortest path may not always be the most energy efficient. In this case topology control can be used to remove energy-inefficient links between nodes.

The second aspect of topology control is related to network capacity. In wireless communication, the same physical medium is shared by all nodes, and channel interferences can be regarded as unwanted transmissions from other nodes in the same area. Boosting transmission power means increasing the range of interference with other communications. Topology control can be used in this case to optimize signal strength in order to reduce the interferences and thus improve network capacity.

In deploying sensor networks, characteristics such as number of nodes or individual node transmission range can be dynamic. Increasing the number of nodes in the network or their transmission range may affect routing since it may provide an increased number of alternative routes in the path discovery process. Although the routing protocols can be designed to be energy efficient, more energy can be saved if the underlying network topology is energy efficient by itself. This can be achieved by the topology control mechanisms that tune the transmission power of individual nodes in order to provide an energy efficient topology that preserves some important network features (such as connectivity). A side effect of this tuning process is that nodes do not need to use their maximum power in transmission, thus reducing contention on the wireless channel.

In [46], it is stated that a topology control algorithm takes a graph $G = (V, E)$ representing the network, where V is the set of all nodes in the network and E is the set of edges $(v_i, v_j) \in E \subseteq V^2$, and transforms it into a graph $T = (V_T, E_T)$ such that $V_T \subseteq V$ and $E_T \subseteq E$. Metrics used to evaluate the efficiency and quality of a topology control algorithm include connectivity, stretch factors, graph metrics, throughput, robustness to mobility and the algorithm overhead. In particular, *connectivity* implies that the algorithm should not disconnect G. In other words if there is a path between nodes u and v, there should also be a path connecting these two nodes in graph T.

The *stretch factors* consist of the *hop stretch factor* and the *energy stretch factor*. The hop stretch factor is defined as the maximum increase in path length for any pair of nodes u and v between the topology controlled path in T and that of the original graph G. Similarly, the energy stretch factor is defined as the maximum increase in the energy consumed along the most energy-efficient path in graph T and that in graph G.

4.4.1. A model for topology control

Given two sensors i and j in a free space environment, the power p_{ij} required by i to correctly transmit a message toy should satisfy [106]:

$$p_{ij} \geqslant \beta \cdot \delta_{ij}^{\alpha}$$

where δ_{ij}, is the Euclidean distance between i and j, $\beta > 1$ is a parameter expressing the transmission quality, and α is the *distance–power gradient*. In the ideal case $\alpha = 2$, but in real settings it is generally close to 4. In any case, it is commonly accepted that it should be included in the range [2–6]. Although the previous equation only holds for perfect free space environments, it is widely accepted due to its simplicity.

A power assignment consists in assigning to each node i a transmission power p_i. For a given power assignment, the network topology can be expressed by the graph $G = (V, E)$ where V is the set of all nodes in the network and E is a set of directed edges between node pairs. Edge $(i,j) \in E$ iff node j is within the transmission range of node i, that is, iff $p_i \geqslant p_{ij}$.

4.4.2. Taxonomy of topology control approaches

Topology control acts on the network topology by selecting an appropriate power assignment. The power assignment is chosen to satisfy constraints on the network topology such as strong connectivity or strong connectivity of a sufficiently large fraction of the network, to ensure that only a negligible fraction of nodes result unreachable by the rest of the network.

Two main approaches for topology control are identified:

- Homogeneous power assignment.
- Non-homogenous power assignment.

Fig. 18 shows the classification of topology control approaches.

4.4.2.1. Homogeneous transmission range assignment. In this approach, transmission range is the same for all nodes despite the fact that the radio transmission is also dependent on the propagation environment. Hence, the implementation of the topology control mechanism can be simplified to calculating the Critical Transmission Range (CTR) of the network.

The CTR problem consist in finding the minimum transmission range r such that the graph obtained by removing all the edges longer than r is connected. Solving the CTR problem is easy if node positions are known. In this case, the CTR is the longest edge of the Euclidean Minimum Spanning Tree (EMST) formed by the sensor nodes. Fig. 19 shows an example of the EMST, where e is the longest edge of the tree. If the transmission range value r_{CTR} is set to be smaller than e, node a will get disconnected from the network. Therefore, in order to maintain network connectivity, the minimum value of r_{CTR} cannot be less than the longest edge of an EMST [88].

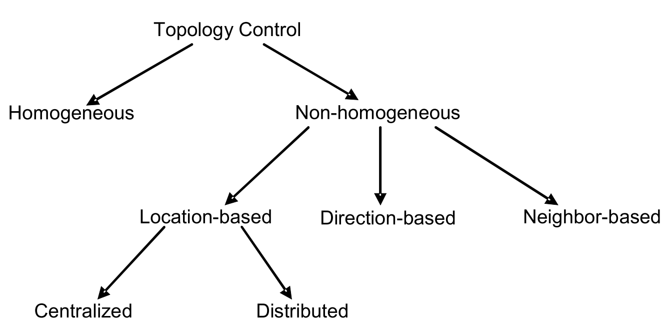


Fig. 18. A taxonomy of the topology control approaches.

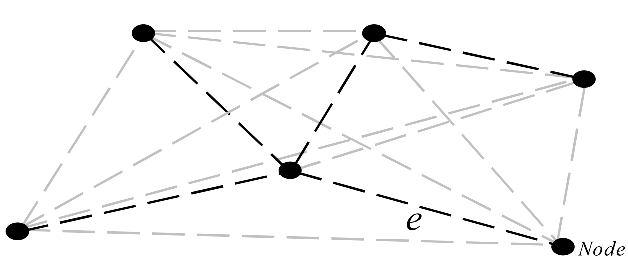


Fig. 19. Euclidean minimum spanning tree.

In real situations, node positions are unknown and probabilistic approaches have to be adopted to find a value of r that guarantees connectivity with high probability. The Geometric Random Graph (GRG) theory is commonly applied to solve the CTR problem. In [96], it has been found that if nodes are distributed uniformly at random in $[0,1]^2$, the CTR for connectivity with high probability is given by

$$r = c\sqrt{\frac{\log n}{n}}$$

where c is a constant greater than zero. In [41], by applying also the GRG theory to nodes uniformly distributed on a disk area, the CTR is found to be as follows:

$$r = \sqrt{\frac{\log n + c(n)}{\pi n}}$$

where $c(n)$ is an arbitrary function such that $c(n) \to \infty$ for $n \to \infty$. The previous equations apply to dense networks. For sparse networks, the CTR is expressed as follows [12]:

$$r = l\sqrt{c\frac{\log l}{n}}$$

for a square region $[0,l]^2$ with $l \to \infty$, c a constant.

The COMPOW protocol [88] determines the CTR in a distributed manner by maintaining a routing table for each power level and set the minimum level as the CTR. In doing so, the routing table contains all the nodes in the network. This protocol can maximise network capacity and reduce contention in accessing the wireless network, thus extending the network lifetime. A drawback is that there is a significant overhead in maintaining the table and the protocol requires a global knowledge of individual power levels.

4.4.2.2. Non-homogeneous transmission range assignment. In non-homogeneous transmission range assignment, different nodes are assigned different transmission powers and consequently different transmission ranges. In this case, the nodes adjust their transmission power based on locally available information. Many topology control algorithms exist for non-homogeneous transmission range assignment. [109,110] give a good overview of these algorithms. In this paper, only a few representative ones are described. Non-homogeneous transmission range assignment can be further subdivided into location-based, direction-based and neighbour-based topology control.

In *location-based topology* control, the nodes are aware of their physical location. In the centralized approaches this information is collected by a single node which uses an optimization algorithm to select the transmission power of each node. On the other hand, in the distributed approaches this information is exchanged between nodes to compute an almost optimal power assignment. A representative of this type of topology control algorithm is the one proposed by [108], which considers the notion of a relay region. The goal is to form a network topology that minimises power consumption by choosing suitable relay nodes in a region to forward the packets to the destination. The drawback of this protocol is its reliance on the location information of individual nodes.

In *direction-based* topology control, it is assumed that the nodes do not know their position. Instead, their directions are made available using angle-of-arrival techniques. The Cone Based Topology Control protocol (CBTC) [125] is one such protocol. The CBTC algorithm consists of two phases. In the first phase, a node u discovers its neighbours by sending broadcast messages initially with very small transmission power. Any discovered neighbours are added to u's neighbour list. Node u continues this process until there is a neighbour v in every cone of angle ρ centered at u. A connected communication graph can then be produced if $\rho \leqslant 2\pi/3$. The second phase involves pruning energy-inefficient edges without impairing connectivity to achieve optimal performance. Several variants of CBTC have been introduced such as those proposed in [10,48].

In *neighbour-based* topology control, nodes will be connected to its k closest neighbours. A typical protocol in this type of topology control is the K-NEIGH protocol [14]. The basic idea is to keep the number of neighbours per node around an optimal value k. The K-NEIGH protocol is distributed and generates a connected graph with high probability. Nodes announce their ID at high transmission power to discover potential neighbours. Neighbours will then be sorted by their separation distance. The k nearest neighbour that can mutually reach each other use the smallest transmission power that is sufficient to reach all of them. The value of k can be determined by using the formulas in [129]. It has been found that on average, the K-NEIGH protocol is 20% more energy-efficient than the CBTC.

5. Routing

Traditional IP-based routing protocols impose a hierarchical addressing structure on the network and base routing decisions (i.e., packet forwarding) on the destination address and a set of tables indicating the next hop to reach that address. In a WSN environment, where nodes can be deployed at random and in large quantities and the network topology may vary due to sensor failures or energy efficiency decisions, assigning and maintaining hierarchical structures is impractical. The message overhead to maintain the routing tables and the memory space required to store them is not affordable for the energy and resource constrained WSNs.

Reactive protocols such as AODV [100] and DSR [52] alleviate some of these problems (ZigBee actually uses an AODV-based protocol) but questionably scale to very large networks since they depend on flooding for route discovery. Furthermore, DSR requires the management of large route caches and large packet headers to store the path.

Routing protocols for WSNs should be lightweight in both processing power and memory footprint and should require minimal message overhead. Ideally they should be able to route packets based on information exchanged with its neighbourhood and should be resilient to node failures and frequent topology changes. For these reasons most of the research on routing in sensor networks has focused on localized protocols which are tree-based or geography-based.

5.1. Routing trees

Simple data gathering applications where readings collected by sensors are sent to the sink, possibly with some aggregation along the path, need trivial routing. As the query propagates through network, each node just remembers its parent toward the sink and later forwards it any messages it receives/originates (see Fig. 20) [78,79]. Directed Diffusion [51] is a variant that routes packets along the edges of a DAG rooted at the sink and allows for multipath data delivery. Routing trees are very easy to construct and maintain but this approach is not suitable for more complex applications that require end-to-end communication.

5.2. Geographic "greedy" routing

Geographic (or greedy) routing naturally supports end-to-end communication. All nodes are assigned a location according to some flat (i.e., network-wide) coordinate system and a distance is defined for any two locations. Each node periodically broadcasts its location to neighbours. On the basis of the destination location (carried in each data packet) a node forwards the packet to the neighbour that minimizes remaining distance (compass routing [67] is a similar algorithm that chooses the next hop as the neighbour with smaller angular distance to the destination). Fig. 21 illustrates greedy routing for the Euclidean distance function. Although greedy routing is extremely simple, some problems have to be solved:

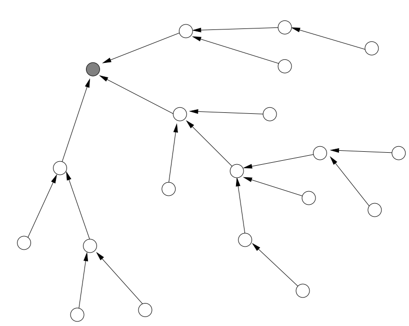


Fig. 20. Routing tree for trivial applications with stream data paths owing towards the sink (the gray node).

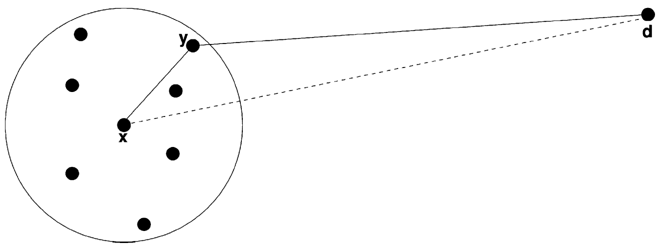


Fig. 21. In greedy routing node x chooses node y as the next hop for a message with destination d when Euclidean distance is used.

1. how a node learns about its coordinates;
2. what happens when greedy routing fails.

The first is a localization problem and consists in assigning a tuple of coordinates to each node. An obvious possibility is to use a physical (geographical) coordinate system with nodes equipped with GPS (or manually configured) or let nodes approximate their physical position from connectivity information with only a few GPS-equipped anchor nodes. An alternative to real coordinates is to run a protocol that assigns virtual coordinates to all nodes. Virtual coordinates are not bound to the physical position but only depend on relative position (i.e., node connectivity). Node localization will be discussed in Section 6.

5.2.1. Greedy routing failure

Greedy routing alone cannot guarantee delivery in every possible network topology. Fig. 22 shows a situation where a node cannot forward the packet since it is closer to the destination than any of its neighbours. Dropping the packet reduces routing efficiency and may preclude communication between a pair of nodes. Resorting to flooding solves the problem but at a high cost. The solution is to integrate greedy mode with a special fallback mode that is entered when greedy mode fails.

When coordinates (either physical or virtual) are two-dimensional, it is possible to apply a perimeter mode procedure that traverses faces in planar graphs ([17] and GPSR [56] introduced this concept). Perimeter mode can be exited when greedy routing can safely take over again (e.g., the current node is closer to the destination than the node where greedy routing failed). Fig. 23 illustrates greedy and perimeter modes combined.

Two-dimensionality is a must since face traversal is applicable only to a planar graph (i.e., one with no intersecting edges) and distributed algorithms for graph planarization are only known for two-dimensional graphs. GOAFR [68] and GOAFR+ [69] are a refinement to GPSR that provides a worst case optimal and average case efficient algorithm by restricting face traversal to an adaptively resized area. Face traversal is an expensive operation that may lead to extremely long paths compared to optimal solutions. [70] introduces GPVFR and suggests storing at nodes (limited) edge information for adjacent faces and choosing the best face and face direction when face traversal is required. GPSR, GOAFR+, GPVFR and similar face traversal algorithms based on graph planarization are not perfect. Inaccuracies in position estimates and irregular radio ranges (possibly due to obstacles) may result in errors in the planarization procedure and produce graphs with unidirectional links, disconnected components and cross links. The effect is routing failures and infinite loops in face traversals [64,113].

Fang et al. [33] observe that face traversal and similar recovery procedures require calculating and maintaining planar graph information at every node in the network which is clearly inefficient given that such information is rarely used and only needed in proximity of network holes. The authors present a planarization-free algorithm for discovering hole boundaries and building routes around them and suggest caching hole boundary information locally to the hole regions and possibly starting the discovery

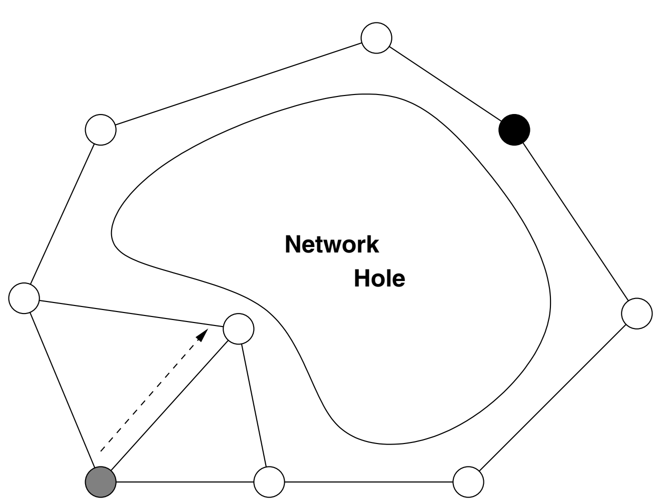


Fig. 22. The packet originating from the gray node and destined to the black node gets stuck.

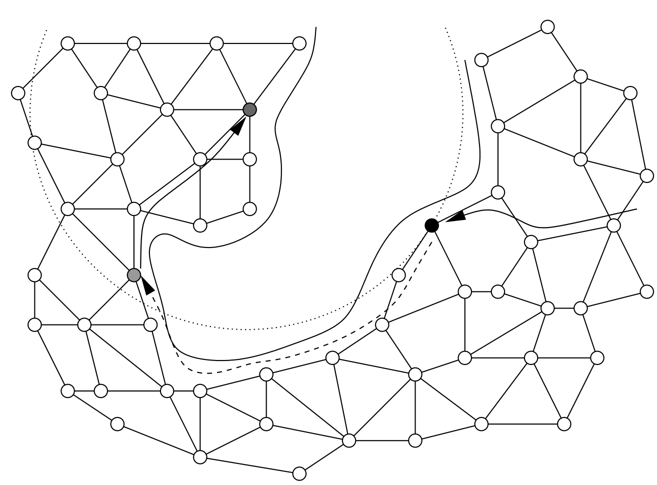


Fig. 23. Greedy mode (solid line) gets stuck at the black node. Perimeter mode takes over up to the light gray node (dashed line) where greedy mode can resume (solid line) and reach the destination dark gray node.

algorithm only when needed (i.e., when a routing failure happens).

Multi-dimensional (>2) coordinates can be mapped to a two-dimensional space and the previous algorithms can be applied. However such mappings usually lose some connectivity information and result in suboptimal behaviour. The alternative is to use a different recovery mode. BVR [35] uses a set of randomly chosen anchor nodes and defines coordinates as the hop distances to such anchors. Its metric function embodies the preference of moving toward an anchor if it is closer to the destination than to the forwarding node but also takes into account that moving away from an anchor is not always good when the destination is farther from it than the current node (the anchor might lie in between the two nodes and moving away might mean going in the wrong direction, see Fig. 24). When greedy routing fails BVR routes the packet along the path to the anchor that is closest to the destination. Each node on the path will first try greedy forwarding and send the packet to its parent in case of failure. If the packet reaches the anchor, this node reverts to a scoped flooding, the scope range being the destination's hop distance to the anchor.

CLDP [63] achieves 100% delivery in the face of wrong location information, obstacles, etc. Probes, sent by each node on its outgoing links, travel the network according to the right hand rule, recording any intersection with the first link traversed, and finally return to the originator. The latter can now take a decision as to whether keep or remove the link. The possible situations can be reduced to the four cases shown in Fig. 25, where node A is probing edge (A,B). In case (a) edge (A,B) is crossed by (C,D) and each edge is traversed only once so node A can safely remove (A,B) without disconnecting the graph. In case (b), the crossing edge (C,D) is traversed in both directions which means removing it would disconnect D. However, (A,B) can safely be removed. If edge (A,B) is traversed in both directions but (C,D) is only traversed once (c) then removal of (A,B) would disconnect the network but (C,D) can be removed and node A signals C and D to remove (C,D). The final case (d) occurs when both (A,B) and (C,D) are traversed twice: there is nothing to do here since removing either edge would disconnect the network. CLDP maintains the minimum number of cross links to ensure connectivity and completely eliminates unidirectional links. It has a large message overhead since multiple probes must be sent on each link before the algorithm attains a stable network topology and each probe potentially travels many hops before returning to the originator. Furthermore, concurrency issues must be taken into account since nodes send their probes and modify the network topology at the same time: a locking protocol must be used in order to guarantee consistency.

GDSTR [71] is a recent routing algorithm that avoids planarization altogether. It replaces face traversal with a visit in an overlay tree spanning all nodes in the network. Each node stores a geometric representation of the convex hull of its descendants thereby aggregating location information (Fig. 26(a)). When greedy routing fails an algorithm searches the descendants of the stuck node and, in case of failure passes the packet to the parent toward the root, in order to explore the rest of the network. The algorithm only explores a subtree of a node if the convex hull happens to include the location of the destination. Fig. 26(b) depicts a scenario where the convex hulls of

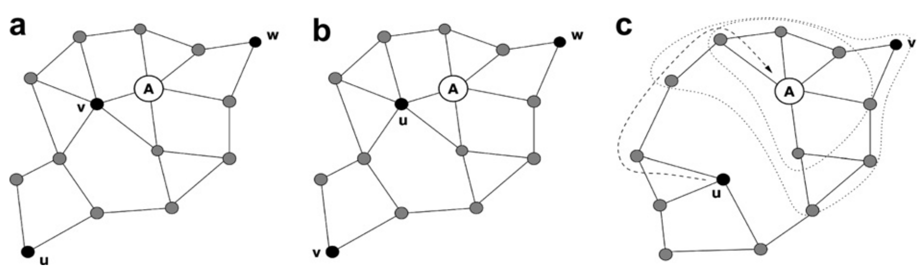


Fig. 24. BVR operation: (a) Since v and w are closer to anchor A than forwarding node u, moving the packet toward A is a good decision; (b) Both v and w are farther away from A than forwarding node u but while moving the packet away from A is a good decision for routing toward v, it is not for routing toward w; (c) Upon greedy routing failure at u the packet eventually reaches A, the anchor closest to v, which performs a scoped broadcast on 2 hops (its distance to v) indicated by the dotted areas.

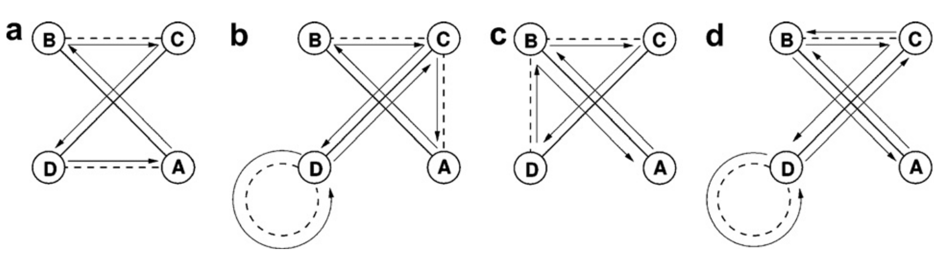


Fig. 25. The four cases where CLDP must decide whether to remove or maintain a link. Solid lines indicate edges while dashed lines indicate possibly multihop paths. Arrows represent the path travelled by a probe originating from A. (a) Edge (A,B) can be removed; (b) Edge (A,B) can be removed but edge (C,D) must be maintained; (c) Edge (C,D) can be removed but (A,B) must be maintained; (d) No edge can be removed.

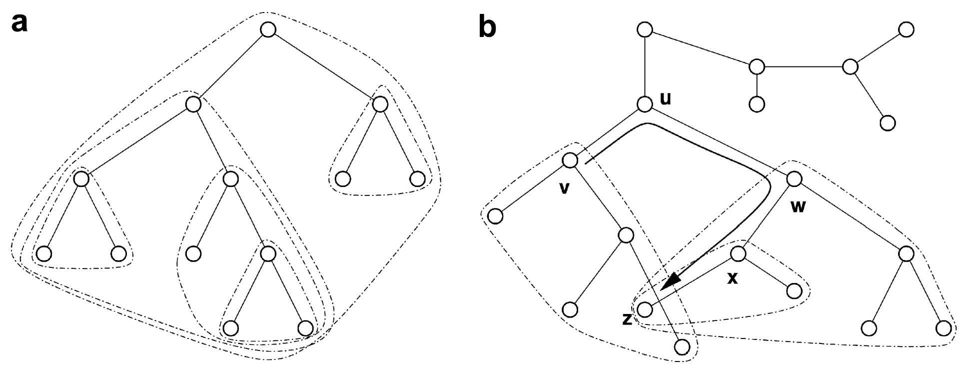


Fig. 26. GDSTR operation: (a) The convex hull stored at different nodes in the overlay tree; (b) The search for node z explores subtrees based on the convex hulls stored at the various nodes in the overlay tree.

two nodes overlap and illustrates the recovery algorithm assuming that greedy routing for a packet bound to z fails at v. Since z lies within v's convex hull, the tree rooted at v is (unsuccessfully) explored before v passes the packet to its parent u. The convex hull of u's child w also contains z so the search proceeds in w's subtree. When entering x's subtree z is finally found. The algorithm is immune from location errors since convex hulls (or rather their approximations) are constructed on (possibly wrong) location information provided by the nodes themselves.

5.3. Hierarchical routing

Greedy routing is efficient in areas densely and regularly populated with nodes. It fails in the presence of voids or obstacles that introduce discontinuities in the topological connectivity structure. Recently developed alternatives to greedy routing consider taking a compact representation of the global sensor network topology structure and storing such representation at all nodes. The representation identifies and divides the network into a set of topologically regular regions. A local coordinate system is defined within each region and a greedy-like routing algorithm suffices to perform intra region packet forwarding. The role of the representation is to glue the regions together and drive long range routing across the network. Routing decisions within a given node consist of identifying an inter region path from the current node to the destination, and using local (greedy-like) routing to reach the next region in the path or the final destination (if it is in the current region).

One of the disadvantages of these approaches might lie in the complexity of deriving the high level topological structure of the whole network. Also the size of this representation must be small enough to be stored at each node, which precludes very articulated networks (e.g., sparse networks). Finally, local coordinate systems within regions tend to be a little more complex than integer tuples (as in flat greedy routing) and so are the corresponding greedy-like routing functions.

MAP [19] and GLIDER [34] are two hierarchical routing algorithms. MAP uses the medial axis concept to represent the high level topology of the sensor network. The medial axis is defined as the set of points with at least two closest points on the network boundary and is a sort of skeleton for the sensor network. Adjacent points (nodes) with two closest boundary points constitute segments of the medial axis. Segments terminate at medial vertices: points with more than two closest boundary points. Segments, chords connecting medial vertices with their closest boundary points and the network boundary define regions. See Fig. 27 for an illustration of the MAP algorithm. Each point v in a region is named on the basis of the closest medial axis point w and the normalized distance vw/zw where z is the boundary point lying on the chord through v and w. Routing amounts to finding the shortest path on the medial axis between the closest medial axis points for source and destination, routing in parallel to this medial axis path across adjacent regions and finally moving along the chord connecting the destination with its closest medial axis point.

GLIDER first selects a set of landmarks and for each landmark u it defines a tile as the region of points that are closer to u than to any other landmark. The high level network topology information consists of the tiles (as graph nodes) and information on tile adjacency (as graph edges): this is enough to plan inter tile routing. Within a tile, each node is assigned a set of coordinates based on the id of the closest landmark and the distance to the latter and the neighbouring landmarks. Routing from node a to node b (see Fig. 28) consists in a two step process. At each

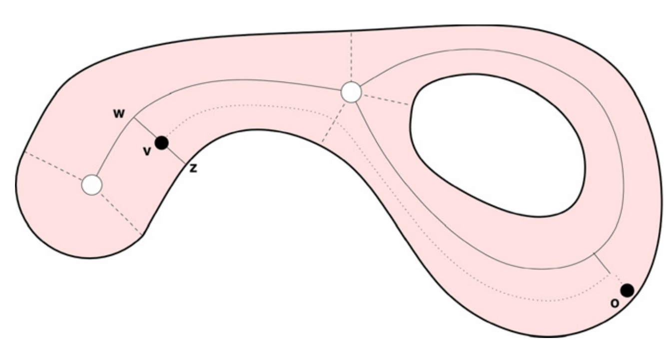


Fig. 27. A simple network in MAP with two medial vertexes (white nodes) and medial edges (internal solid lines): routing (dotted line) from v to o proceeds in parallel (region wide) to the medial edges up to a point lying on the chord through o and moves along this chord to finally reach the destination.

hop the high level topology graph is consulted to determine the next tile. Intra tile routing then chooses the next hop as the neighbour that is closest to the landmark of the next tile (local node coordinates include distances to landmarks of all adjacent regions). When the packet finally reaches the destination tile, intra tile routing directs the packet toward the destination node. Intra tile routing falls back to tile flooding when it reaches a local minimum. Landmark selection can be handmade or automatic (following automatic detection of hole boundaries).

6. Localization

The purpose of localization is to provide some kind of location information for nodes in a sensor network. It can be used to support routing algorithms (see Section 5) and/or to identify a data source location for application requirements.

6.1. Physical coordinates

Assigning physical (i.e., real geographical coordinates) to all nodes in the network directly and effectively addresses both localization functions. The most immediate solution is to use a physical coordinate system by equipping all nodes with a GPS receiver. However such solution is often not applicable given GPS receivers cost, power consumption and size requirements. It may also fail to work if some nodes cannot receive GPS signals (e.g., they are located indoor or obstacles prevent reception).

A cheaper alternative is to approximate real (physical) coordinates according to some localization algorithm where only a few anchor nodes have GPS receivers (or are manually given correct coordinates) and all the others use radio-based communication protocols and connectivity information to derive their approximate position. Localization algorithms can be classified according their usage of ranging techniques to measure relative distance/position between neighbours. Ranging techniques include

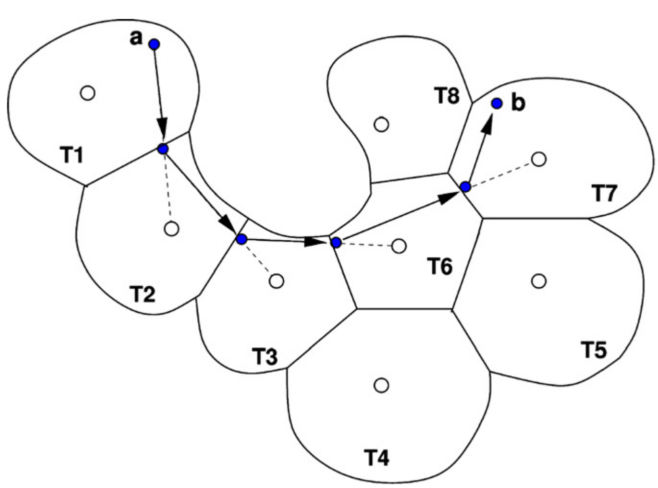


Fig. 28. GLIDER routes a packet from node *a* to node *b* by first selecting the next tile and then routing toward that tile's landmark.

Received Signal Strength Indicator (RSSI)
On the basis of measured received power, known transmit power and a propagation power loss model, a node estimates distance from the sender. Error sources for this method include environmental variability of power loss models and poor calibration of cheap (sensor) radio components.

Time Difference of Arrival (TDoA)
As illustrated in Fig. 29, a node measures the difference of arrival times of two simultaneously sent messages. The two messages use different communication mediums so they have different propagation times (radio and ultrasound are commonly used [40,102]). It may suffer from non line of sight effects and requires special hardware.

Angle of Arrival (AoA)
Nodes use antenna arrays to measure the angle of arrival of received messages (Fig. 30). This method only provides bearing information but can nevertheless be used to help in localizing nodes. Costly, large and power demanding antenna arrays can be replaced with a TDoA technique applied to two on-board acoustic receivers [94].

Range-free algorithms localize nodes relying solely on connectivity information i.e., knowledge of neighbouring nodes.

6.1.1. Range-based methods

One range-based approach is to propagate location information from anchor nodes so that non anchor nodes become aware of the position of at least three anchors

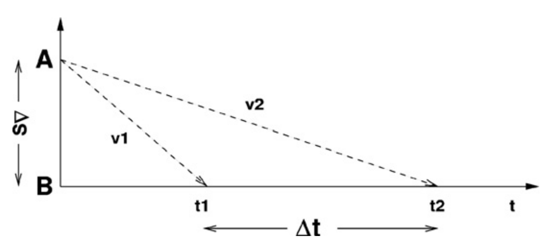


Fig. 29. TDoA principles: A sends B two messages propagating with speeds v_1 and v_2 and arriving at times t_1 and t_2. Knowing v_1 and v_2 and measuring $\Delta t = t_2 - t_1$, B can estimate the distance to A as $\Delta s = \frac{v_1 \cdot v_2}{v_1 - v_2} \cdot \Delta t$.

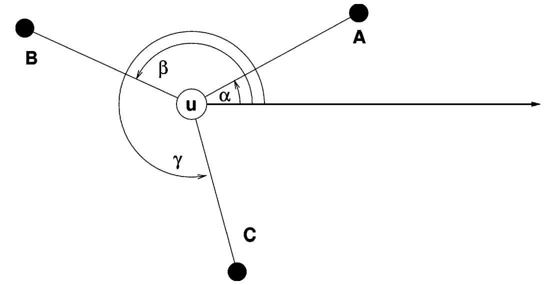


Fig. 30. Node *u* measures the angle of arrival of messages received from nodes A, B and C as α, β and γ according to a local angular system.

and can compute their position via a multilateration procedure. The computation minimizes square distances between ascertained (i.e., measured) and estimated (i.e., based on the yet to be determined node position and anchor positions) distance to the anchors.

A possibility is that each node directly acquires the coordinates of some anchors, as depicted in Fig. 31(a). In the algorithm described in [112] a node starts multilateration if it receives location advertisements from at least three neighbours (or some equivalent support is available from two-hop neighbours). These can either be anchors or nodes that were able to run multilateration previously. [89] uses anchors equipped with high power transmitters emitting beacon signals on a narrow directional beam rotating with a constant angular velocity. Sensors note the difference in arrival times of the beacon signals and determine angular bearing to the anchors and their location via triangulation. A disadvantage of these approaches is that many anchors must be deployed to let each node locate itself and/or that anchors must be equipped with special hardware.

Fig. 31(b) suggests a simpler solution where nodes determine their position only on the basis of anchor locations that they receive via multihop paths, at the expense of less localization accuracy. In [93], after receiving distances to an anchor from two neighbours each node measures distances to these neighbours, computes the real Euclidean distance to the anchor via trigonometric relations and propagates the latter to its neighbours (Fig. 32). When the node has computed distance to several anchors it runs multilateration. A conceptually similar algorithm [94] uses AoA instead of distance measurements to propagate bearing information to anchors. If a node knows bearings to at least three anchors and their positions, it can locate itself.

Another option involves a first stage where each node builds a local virtual coordinate system and a second stage that uses GPS-equipped nodes to translate local virtual coordinates into global physical coordinates. After measuring distances to neighbours and locally exchanging such information, each node defines a local coordinate system where it localizes many of its one and two-hop neighbours via trigonometric calculations. At this point each node computes transformation matrices translating coordinates

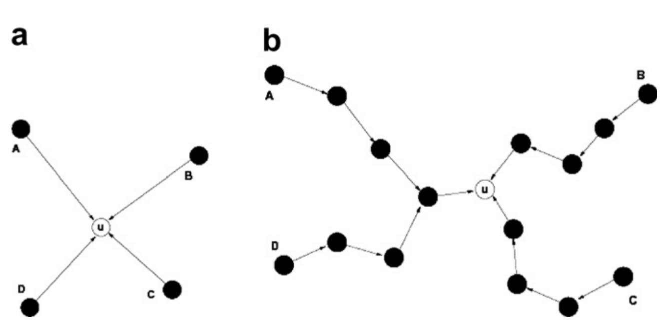


Fig. 31. Part (a): Node u directly receives coordinates from high-powered anchors or neighbours that previously located themselves A,B,C and D. Part (b): Node u receives coordinates of anchors A,B,C and D via a multihop path.

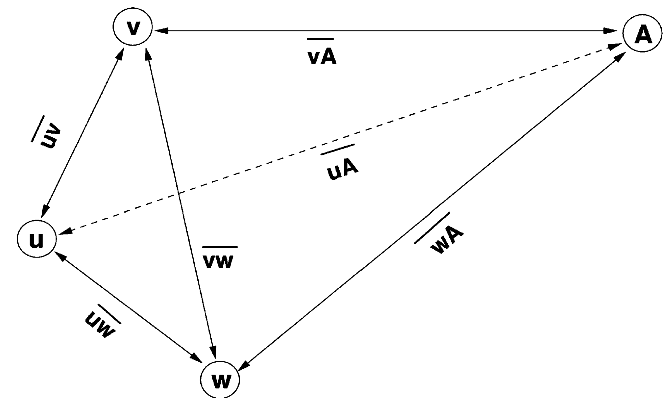


Fig. 32. Upon receiving distances $\overline{vA}$ and $\overline{vw}$ from v and $\overline{wA}$ from w, node u estimates its distances to v ($\overline{uv}$) and w ($\overline{uw}$) and uses trigonometry to estimate its distance to A ($\overline{uA}$).

from the local coordinate system of neighbours. The physical coordinates of anchors now propagate through the network. Each node translates them to its local system (and forwards translated coordinates to neighbours), trivially computes distances to the anchors and runs multilateration on the global (real) coordinate system to determine its location [93].

A similar algorithm [24] defines local coordinate systems for each node as described above but manages to create a network-wide virtual coordinate system by choosing a special network origin reference node (or a virtual origin point) and adjusting the coordinate systems of all nodes with respect to it by appropriate translations, rotations and mirroring. The resulting virtual coordinate system is isomorphic to the real coordinate system and can support greedy routing. An important difference with other virtual coordinate systems (Section 6.2), is that it is not based on connectivity but only on geographic proximity and it makes use of ranging techniques.

6.1.2. Range-free methods

Range free methods offer a cost-effective but possibly coarser alternative to range-based methods. In [21] nodes compute their position as the centroid of the coordinates of the anchors they can hear from (at least three anchors are needed for each node). Obviously such an algorithm requires a high percentage of anchors to achieve reasonable location accuracies.

In a more effective algorithm each anchor floods the network with a message containing its location so that each node can record hop count distance to the anchors and their coordinates. Hop count distance is related to radio range and actual network topology (i.e., the path to the anchor) but average distance covered per communication hop can be estimated and used to translate hop distance to the anchor into physical distance [87,93]. Another approach [43] requires anchors with high powered transmitters to periodically broadcast their location. For any different triplet of anchors a node hears from, it tests

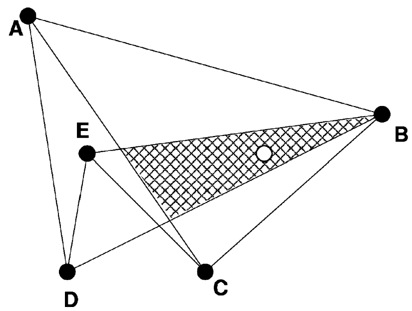


Fig. 33. The white node lays in the intersection of triangles ABC, ABD, BCE and BDE. It locates itself as the centre of gravity of the hatched area.

whether it lies inside the triangle having the anchors as vertexes and finally locates itself as the centre of gravity of the intersection of all the triangles it is in (Fig. 33). Neighbouring nodes must be able to compare their proximity to a given anchor. Upon exchanging this information nodes can approximate triangle tests. Anchor proximity comparison can be achieved by means of RSSI ranging techniques. Even if RSSI is not used to assess distance to the anchor, poor calibration can still contribute to errors in test outcomes. The number of tests, and hence the number of anchors, compensates for inaccuracies/errors in triangle tests.

6.1.3. Analytical methods

Recently algorithms that rely on mathematics formulations of the coordinate assignment problem have gained attention. One approach is based on multidimensional scaling (MDS): a technique based on matrix theory that maps a matrix containing proximity information regarding objects (nodes) to positions of those objects in a low (2 or 3) dimensional space such that Euclidean distance is related to the original proximity information. Each node establishes a local map of its neighbourhood (typically 2 hops), computes the shortest path matrix of all these nodes and applies MDS to obtain a local two-dimensional relative map. Maps of neighbouring nodes can later be merged on the basis of common nodes and transformed into absolute (physical) maps with the help of anchor nodes. [114] and [53] describe MDS-based localization. Proximity information used to compute the shortest path matrix can be simple neighbourhood or measured distance so these algorithms can either be range-based or range-free.

MDS-based approaches have the advantage of effectively modelling anisotropy in sensor networks. On the other hand, their distributed implementation imposes restricting the size of local maps, which may ultimately reduce its accuracy. An alternative approach [74] tries to reduce complexity and at the same time maintain global information. By means of a series of floods each anchor acquires proximity information to the other anchors (either coarse hop counts or more accurate distance estimates). The anchors calculate a Proximity-Distance Map with Singular Value Decomposition (SVD) applied to the inter-anchor proximity matrix they collected. Each node retrieves the map (a matrix) from the closest anchor and uses it to translate its vector of anchor proximities into a vector of anchor geographical distance estimates. It finally uses multilateration to compute its coordinates.

6.2. Virtual coordinates

Physical coordinates are very effective at locating data sources but require expensive/complex hardware and protocols, and may suffer from non-negligible measurement and approximation errors. Also geographic proximity doesn't necessarily mean topological proximity and greedy routing applied on physical coordinates may lead to stuck nodes (Fig. 22) and heavy use of expensive recovery procedures like flooding or face traversal algorithms. The aim of virtual coordinate assignment protocols is to support greedy routing with a coordinate system that is based on network connectivity.

A distributed algorithm where nodes compute their virtual coordinates from essentially no initial information is proposed in [105]. As the first step, nodes on the network boundary learn they are on the boundary on the basis of hop distances from a special bootstrap node. Each boundary node floods the network with a Hello message so that all boundary nodes discover their distance to all other boundary nodes and each can later flood the network with a message containing such distances. On the basis of these distances, each boundary node finally computes its virtual coordinate via a triangulation procedure. Non boundary nodes finally run an iterative relaxation algorithm to derive their virtual coordinates. Several drawbacks are apparent: computational complexity, message (floods) overhead and per-node memory space requirements that is linear in the number of nodes.

Other, less demanding, solutions are based on identifying a set of anchor nodes and defining coordinates as the tuple of hop distances to these. [23] proposes such an approach for a configurable number of anchors and uses the Euclidean metric to drive greedy routing. Randomly choosing anchor positions may lead to many widely separated nodes sharing the same coordinates. Under the hypothesis of uniform distributions of the nodes, [25] shows that the size of the areas of nodes sharing the same coordinates is minimized when anchors are located on the network boundary and as far as possible from each other and, in this case, the maximum width of each area is at most three hops. The authors propose a distributed algorithm that elects three anchors that satisfy this property (Fig. 34) and show that simple greedy routing combined with proactive routing within the areas achieves similar performance as with physical coordinates. In BVR [35] anchors are chosen randomly in the network but the metric function is not Euclidean. It embodies the preference of moving toward an anchor when it is closer to the destination than to the forwarding node. It also takes into account that moving away from an anchor when the destination is

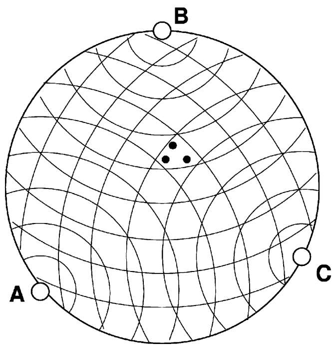


Fig. 34. The number of nodes sharing the same hop-based coordinates is minimized when anchors are positioned on the network boundary.

farther from it than the current node is not always good: the anchor might lie in between the two nodes and moving away would mean going in the wrong direction (Fig. 24). Fig. 35 illustrates the use of virtual coordinates for a set of 3 anchors.

6.3. Location service

The purpose of a location service is to map high level node names to low level names suitable for reaching the given node. In the context of sensor networks high level names may be string mnemonics for nodes with a particular function (e.g., "light detector" or "data collector" or "Group 1 leader") or spatial location (e.g., "Main Street sensor" or "South-East quadrant"). The low level name returned by the location service is a coordinate tuple representing the location of a node in the coordinate system used for routing (either physical or virtual). The characteristics of a sensor network impose that location services be distributed, scale to large network sizes and have low memory requirements.

The idea behind Grid Location Service (GLS) [72] is that each node has an associated group of location servers that know its location. Location servers are selected on the basis of a numeric hash of the node name and a hierarchical decomposition of the network field such that an initial square area is recursively partitioned into 4 equal subsquares up to the point that nodes in the same square are within communication range of each other. GLS also assumes that all nodes know about the network subdivision

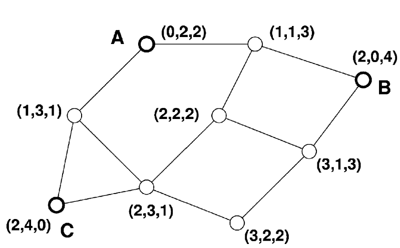


Fig. 35. Virtual coordinates for a node are defined as the tuple of hop distances to a set of anchors. Next to each node is the triplet of hop distances to anchors A, B and C, in this order.

hierarchies, which is easily achieved for coordinate based partitioning.

Every node chooses a location server in each sibling of the area where it resides at each level of the partition hierarchy. As a consequence, location servers get sparser as we move away from the node. The algorithm to select a location server in a given area only depends on the node name and attempts to spread selections uniformly so that each node in the network acts as a location server for a small number of other nodes, workload is evenly distributed and the algorithm scales to large network sizes.

Looking up the coordinates of a node reduces to the search of a location server for the node. The search proceeds in *query steps* with each query step forwarding the query to a node in the higher partitioning level area. It terminates either when a location server is found or, in the worst case, when the current partitioning level area contains both the searching and the searched nodes. A node forwards a query to another node by means of greedy routing (Section 5.2) and retrieves the address of the latter via the data structures it maintains as a location server for other nodes.

Data Centric Storage schemes (Section 8.1) like GHT [107] can also be used for location service.

7. Data management

The ultimate goal of a sensor network is to provide users with relevant data from the sensor field. Of course, the user must have a way to indicate what is relevant i.e., he must interact with a PC program (usually a GUI) that interfaces to the sensor network. The program injects commands into the network and displays data returned by the network.

Two classes of applications can be distinguished. One is involved in event detection whereby each sensor periodically checks if some environmental conditions are locally satisfied or match a predefined pattern (e.g., animal sightings). In such applications neighbouring nodes may cooperate to achieve a higher confidence on the event characteristics and pattern matching degree but the event data is stored in the network (for later retrieval) or directly sent to the sink.

The other class is engaged in long running environmental observations that continuously perform sampling and result in data streams. This extremely large amount of data cannot be stored in the network, given the limited memory resources of nodes and must ultimately flow to the sink or be discarded. The need to collect data from many highly distributed nodes must be balanced with the high cost of communication. A simple way to reduce messages is to act at the network layer and combine several messages bound for the sink into one big message. This solution only alleviates problems since messages can only grow up to a maximum (usually small) size in a sensor network. Data aggregation and in-network data processing is a more promising approach that consists in moving computing activities from the PC into the network [78,79]. Instead of just forwarding data toward the sink, nodes perform computation and data-management tasks so that user requested data is not extracted from raw data on the PC but is directly furnished by the network. Nodes can do some processing on a data stream (like taking temporal averages or computing functions) or combine it with other data streams (like joining or taking spatial averages) and ultimately produce another data stream which they forward to another node. This Section concentrates on this class of applications and describes approaches to data collection. In network data storage schemes are described in Section 8.1.

7.1. Directed diffusion

Directed Diffusion [51] is an early attempt to define a data management paradigm in sensor networks. A user request for specific data is translated into an interest for some kind of data with a certain data rate.

Interest dissemination begins with the sink broadcasting the interest message to its neighbours. Before forwarding the message each node records the interest and data rate in its cache and sets up a gradient toward the source of the message. This way the interest propagates throughout the network.

Nodes that detect or receive data matching one of their cached interests forward such data along gradients with the associated data rates. Via neighbour-to-neighbour propagation, data finally reaches the sink. The sink can reinforce paths by sending a new interest message with a higher data rate through selected paths. Nodes on the path that are not reinforced ultimately clear their cached interest upon timer expiration. Nodes choose to reinforce a neighbour on the basis of higher quality/rate of received data. Reinforcement can also be triggered by non sink nodes when they detect reduced quality data from existing paths.

Chief advantage of Directed Diffusion is that data exchange is exclusively based on locally exchanged interests. There are no explicit end-to-end multihop paths and no need for routing and network-wide addresses. Multipath data delivery (via reinforcing multiple paths) and local data path repair (via node-triggered reinforcing) are also available. A disadvantage is load unbalance since nodes close to the sink have to manage a large part of control and data traffic. Another problem is limited possibility for in-network data processing and aggregation since different data can be combined only if they are routed through a common node.

7.2. The database approach

An interesting approach that recently gained in popularity and offers powerful, application-independent, data abstraction and manipulation functionalities is to view the sensor network as a distributed database system. The user formulates data requests via an SQL-like query language that includes syntax to specify sampling rates as well as query duration [78,79]. The high level query is translated into a set of data acquisition (sampling), data processing and data transfer operations that must be carried out by the nodes in the network. Query optimization then evaluates several task allocation alternatives (query execution plans) and chooses one that minimizes energy consumption (Fig. 36).

The selected query execution plan is then injected into the network as a series of commands. A node can be instructed to join two data streams, implement filtering operations selecting records on the basis of some predicate or compute functions depending on record contents. Other forms of in-network aggregation include taking temporal and spatial averages of transducer readings. While the former can take place on the sensing node, the latter requires collecting readings from several nodes using a tree built over the area where the average must be taken and can be done on-the-fly as data moves along the tree edges. A similar technique can be applied to other aggregate operators like Min, Max, Count and Sum. Reducing message exchange also demands that data aggregation be applied as close as possible to data sources (transducers). As a result of distributed in network query execution only the query outcome reaches the sink.

Query execution should also be tolerant to node failures: task assignment should not be rigid and immutable but mechanisms should guarantee automatic recovery. Yao and Gehrke [131] suggest that constructing a query execution plan should amount to linking together several flow blocks. Each flow block has a certain data collection task involving a set of geographically close nodes (e.g., taking a spatial average). A leader is elected among these nodes, and data is collected and routed towards the leader with aggregation and computation performed along the path and possibly at the leader itself. The leader periodically notifies the other nodes that it is still alive to prevent automatic reconfiguration of the flow block internal organization. Query optimization should consider flow blocks as basic, locally autonomous building blocks.

7.2.1. TinyDB

TinyDB [80] is a sensor network database implementation developed at UC Berkeley. An SQL-like language with extensions for query duration and sample rates is used to express queries over a single sensors table that represents all sampled data in the network (with one row for each sensor being continuously updated). TinyDB supports spatial aggregation operators as described in [79], filtering based

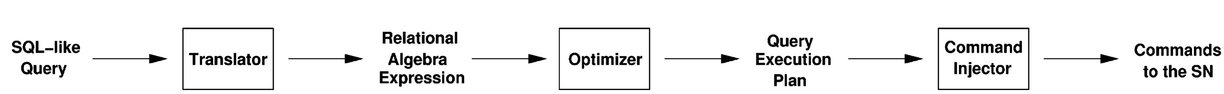


Fig. 36. An SQL-like query is translated into a relational algebra expression that is later optimized to produce a query execution plan which is finally converted into commands to inject in the sensor network.

on predicates and special joins taken over the sensors relation and a storage point or two storage points (a storage point is a bounded subset of a stream i.e., a limited number of records).

Power-aware optimization and query execution plan generation is performed on the basis of meta data concerning transducers and operator parameters and it results in a suitable ordering of sampling activities and predicate-based selection. Query dissemination is achieved via Semantic Routing Trees (SRTs): routing trees (Section 5.1) built from the sink. During the tree construction process each node gathers range information regarding the values of some attribute covered by each of its subtrees. A query later propagates down the various paths in the SRT as long as there are interested nodes.

A major limitation of TinyDB is that data streams flow towards the sink along the edges of the routing tree: queries involving more complex data communication patterns are not allowed.

7.2.2. Cougar

Cougar [15,16,130] is a sensor network database developed at Cornell University and shares many similarities with TinyDB. The user expresses a query in a high level declarative language that extends SQL. Nodes are modelled as Abstract Data Types (ADTs) with interface functions providing access to encapsulated data. The FROM clause of a Cougar query may refer to a sensor network relation, say R, including attributes identifying a node position as well as the node ADT, say s, while SELECT and WHERE clauses may refer to actual node specific data invoking access methods on node ADTs like R:s:getTemp(). A query optimizer running on a PC generates a query execution plan that specifies data flow and computation activities to carry out at each node, including organization of aggregation trees. From an implementation point of view a *virtual relation* is associated with each method available for the node ADT. The virtual relation for a method includes attributes for the node id, input arguments, output value(s) and timestamp. A virtual relation is fragmented over all nodes that produce records for it (i.e., implement the associated method) and is stored distributively in the network.

7.2.3. MaD-WiSe

Mad-WiSe [6–8] implements a distributed database system that supports in-network query processing. Similarly to the previous approaches, it parses an SQL-like query and selects one of several query plans for execution. Query optimization is carried out by applying several transformation rules based on heuristics to considered query execution plans. These rules take into account transducer sampling costs, predicate selectivity and transmission costs. Query processing is based on streams that abstract data channels between operators of a query algebra and drive their pipelined behaviour (computation and aggregation is carried out on flowing records with almost no need of storage). Operators include selections, projections, spatial and temporal aggregates as well as unions and joins. The ability to perform joins between streams is unique to MaD-WiSe and permits comparison of data from different sources to be carried out in the network.

8. Reliability

The problem of reliability is central to Wireless Sensor Networks. Nodes are battery-powered and communications are radio-based which means nodes can fail and temporary/permanent disconnections may occur. The measurements collected by individual nodes are rarely indispensable. Rather, information collected by several nodes is usually aggregated to provide better accuracy and significance. As a consequence reliable communication in sensor networks is not focused on each single end-to-end delivery but is of a more general nature, encompassing network-wide relevance. The Network and higher layers must address this issue in order to improve on the low reliability and limited scope of the Physical and MAC layers.

Greedy routing strategies (discussed in Section 5.2) are inherently tolerant to node and link failures given their statelessness and their dependence on local information (which is periodically refreshed). Planarization (Section 5.2.1) and Localization (Section 6) algorithms, that assist routing and applications, are affected but have graceful degradation properties. Failure of a single node or link may prevent correct routing to some nodes but does not usually compromise the whole network. Periodic refreshing through algorithm reruns helps maintain acceptable levels for the associated supporting functions.

More attention must be dedicated to the Application layer where data from the sensor network is actually managed and collected. As hinted at in our previous discussion, the focus is usually on *highly informative* data that is the result of aggregating several measurements or is anyway worth remembering and collecting. Such data packets can either be stored in the network for later retrieval or sent to a collection point (e.g., a sink node). In the following we describe the two alternatives in more detail and how reliability can be achieved.

8.1. Data centric storage

Data Centric Storage (DCS) [107] is an in-network data storage technique that selects locations for data storage based on data names. It applies to event detection applications that store data in the network for later user retrieval. Retrieval requests can be formulated via the data name (which is enough to identify the data location) and efficiently performed via a unicast request message.

Scalability, topology changes (node failures), energy efficiency and persistency concerns are addressed by GHT [107] a Data Centric Storage implementation that uses a hash function to map a data name (also called *key*) to a geographic position. GHT uses a variation of GPSR [56] (actually its perimeter mode) to select a *home node* as the closest node to this geographic position and stores a (key, value) pair at the home node and the nodes in the *home perimeter* (that is, the nodes in the perimeter surrounding the geographic position chosen for the key) to guarantee data persistency. GPSR can later be used to locate the home node given the geographic position of data. Perimeter mode routing automatically walks around the home perimeter and guarantees data retrieval.

DIM [73] and KDDCS [4] are alternatives to GHT that recursively divide the network region into zones and map events (defined as attribute values) to binary codes that correspond to zones. They cope with node failures by replicating information stored at a node (in a zone) at a nearby node (in a backup zone) that can easily be reached in the search process. As a totally different alternative the back up zone can be the one whose binary code is the complement of the code for the primary zone. With this alternative two searches can be started in parallel or sequentially. Higher resilience to regional node failures is offset by higher cost for the parallel search or longer latency. This technique is easily generalized with a hash function that outputs several different (possibly far away) storage points for a given event that can be searched in parallel or sequentially.

GEM [91] is another interesting DCS algorithm based on virtual coordinates. It distributively defines a Virtual Polar Coordinate System (VPCS) and uses a Virtual Polar Coordinate Routing (VPCR) algorithm to route over the virtual coordinates. VPCS is obtained by assigning each node a level in a sink-rooted tree (as the hop count distance) and a virtual angular range from a fixed size interval (e.g., $[0, 2^{16}-1]$). The root gets the full size interval while its children are assigned a subrange proportional to the size of their respective subtrees. The subranges are assigned consecutively to children according to increasing angular position with respect to the root. The process is repeated recursively by each non leaf node (see Fig. 37 for an example). Nodes at the same level that are contiguous (according to the assigned subrange) and that can hear each other are connected by a cross link edge. VPCR routing takes place over the tree level and cross link edges defined above. When a node must forward a packet it selects the neighbour which has an angular range that is closer to the final destination than its own range. If such a neighbour does not exists (e.g., a cross link is missing in the topology) the node simply forwards the packet to its parent in the tree. Eventually the packet either reaches the destination or an ancestor of the destination, in which case it can be routed down the tree. DCS can be supported by using a function mapping a data name to a virtual coordinate in VPCS so that VPCR can be used to reach the node storing the data item.

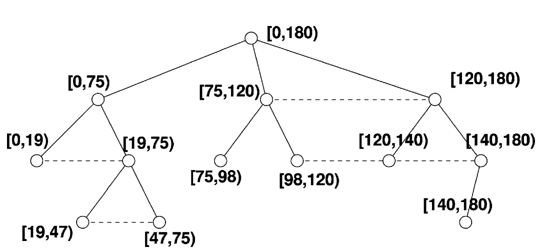


Fig. 37. A simple example of VPCS where the root range is [0, 180] and cross link edges are shown with dashed lines.

GEM can also conceivably be used as a routing mechanism over a virtual coordinate space assuming that a location service (Section 6.3) is available to map a node name into its virtual coordinates.

8.2. Transport reliability

The simplest solution to transport reliability is to use acknowledgments for important data packets. Providing such a service at the MAC layer is expensive, unable to cope with link failure and cannot selectively be applied to specific transmissions. Application layer acknowledgments may be a better solution but issues like retransmission timeouts, that heavily depend on the number of hops in the path, can be hard to handle.

Fault tolerance and reliability were also addressed in the data management paradigm of Directed Diffusion [51] (see Section 7.1) where multiple paths to the sink ensure better data availability. A more in-depth discussion is provided in [36] where the authors present and compare local algorithms to construct *disjoint* paths (non intersecting) as well as *braided* paths (i.e., with overlapping parts). Disjoint paths are inherently more resilient to failures since several nodes/links may fail on a path without affecting alternate paths. However disjoint paths tend to be longer with respect to the optimal shortest path and consequently rather energy inefficient, except in dense network scenarios. Braided paths have better energy efficiency properties, especially in sparse networks but a single failure may compromise all paths.

[62] evaluates the application of erasure codes to WSNs to achieve end-to-end reliability. Instead of sending the same data packet over multiple paths, erasure codes encode a set of m messages into $n > m$ packets in such a way that receiving any subset of m packets is enough to reconstruct the original messages. The authors also suggest that support from the Network layer coupled with early detection of link failures may be used to rebuild broken paths.

ESRT [1] is a protocol that takes care of providing global transport reliability as well as conserve energy resources by dynamically modifying the rate at which nodes send their packets to the sink. Nodes are capable of locally assessing network congestion state and report such state to the sink. A measure of reliability is evaluated at the sink as a function of the number of received and expected packets for each interval in which time is divided. On the basis of current network congestion state and reliability, the sink determines (and broadcasts to the network) the reporting rate for the next interval in order to achieve the desired reliability and eliminate network congestion, if any. If there is no congestion and reliability is below the acceptable level, the report rate is increased. In case of network congestion and low reliability the report rate is always decreased. Indeed, a low reliability in such a state is necessarily the result of packets being dropped. Reducing the report rate should reduce congestion and increase reliability. The report rate is also decreased in case of congestion with high reliability because that means reducing dropped packets and saving energy resources.

9. Security

Sensor nodes in a wireless sensor network are limited in computational power and communication resources. Due to these strict resource constraints existing network security mechanisms are inappropriate for this area. Efficient encryption of measured data can be achieved at the cost of increased overheads in the length of the message. But as radio communications is the most energy consuming function performed by these nodes, hence the communications overheads have to be minimised to achieve long life [42].

9.1. Security issues in wireless sensor networks

9.1.1. Security requirements

This section identifies the security requirements of wireless sensor networks.

Data confidentiality: Data confidentiality means keeping important transmitted information secret from unauthorised people. This is particularly important in the case of wireless networks where data is transmitted using a radio frequency and anybody with a radio receiver can intercept the data. Data confidentiality is usually achieved by encrypting the information before transmission so that only authorised people can decrypt the transmitted information. Hence an adversary should not be able to recover the important information even if it got hold of the transmitted data. Encryption is classified into two categories: symmetric encryption and asymmetric encryption. In symmetric encryption, a secret key is shared between the authorised parties, while in asymmetric encryption, the sender encrypts the data with a public key and the receiver decrypts it using a private key.

A strong encryption mechanism not only prevents message recovery but also prevents adversaries from decoding even partial information about the message. This property is called semantic security, which implies that the encryption of the same plaintext two different times should give two different cipher texts [99].

Data authenticity: Data authenticity provides a means to detect messages from unauthorised nodes thereby preventing unauthorised nodes to participate in the network. In other words, data authentication allows a receiver to verify that the data is sent by the claimed sender. This is particularly important in sensor networks where an adversary node can easily inject a large number of messages into the network [42] causing other nodes to process these messages thereby using up their power resources. Hence the receiver of these messages needs to be able to ensure that the message is from an authorised source.

Data authentication can be achieved by calculating a Message Authentication Code[1] (MAC) using a shared secret key for the transmitted data. This MAC is also sent along with the data. The receiver would also calculate the MAC for the received data using the shared key, and then compare this computed MAC value to the one sent by the sender along with the data. If the two matches, then the receiver know that the data had to be sent from the correct sender [99]. Hence the message is authenticated.

Data integrity: Communications in wireless sensor networks are based on broadcasts; hence messages can be easily eavesdropped and/or tampered by an adversary hearing on wireless medium. Data integrity provides a way for the receiver of the message to know if the data has been tampered while in transit by an attacker [99].

Data integrity is closely related to data authentication since the MAC used for data authentication also provides data integrity. The receiver of the data calculates the MAC and compares it to the one transmitted by the sender. If the two MAC's match then it ensures that the data was not tampered with. In other words, if an adversary has tampered with the message then the MAC calculated by the receiver cannot be equal to the MAC that was initially calculated by the sender at the time of sending the message.

Data freshness: Data freshness ensures that the received data is recent and that an adversary has not replayed old messages at a later time. Data freshness can be divided into two categories: weak freshness and strong freshness [99]. Weak freshness provides partial data ordering preventing data from being replayed, but carries no delay information [42]. On the other hand, strong freshness uses a request-response model to provide complete ordering of messages and delay estimation to prevent the data to be held by an attacker. Weak freshness is required for sensor measurements, while strong freshness is required for time synchronisation within the network.

One of the most common methods to provide data freshness is to use a monotonically increasing counter with every message and reject any messages with old counter values. However, every recipient would need to maintain a table of the last counter value from every sender. This method may result unfeasible in wireless sensor networks where the sensor nodes are memory constrained, and would not be able to store such a table for even a moderately sized network.

9.1.2. Security threats

Wireless sensor networks like any wireless technology are susceptible to several security attacks due to the broad-

[1] In general networking MAC usually stands for the Medium Access control Layer of the OSI protocol stack but in this Section MAC would be used for Message Authentication Code unless otherwise stated.

cast nature of transmission medium. Moreover, a wireless sensor network is more vulnerable as the sensor nodes are usually placed in hostile or dangerous environments [42]. Some of the different types of attacks on wireless sensor networks are described briefly below:

Eavesdropping: Due to the broadcast nature of the transport medium in wireless sensor networks, any adversary with a good receiver could easily eavesdrop and intercept transmitted messages. The intruder would be able to retrieve information like location of node, Message IDs, Node IDs, timestamps, application specific information, etc. Strong encryption techniques should be used to counter eavesdropping.

Denial of service: A Denial-of-Service (DoS) attack refers to the attempt where an adversary disrupts, subverts or destroys a network [127]. A DoS attack diminishes or eliminates a network's capacity to perform its expected function.

Message tampering: Malicious nodes can tamper with the received messages thereby altering the information to be forwarded to the destination. When the destination receives this tampered message, it would compute the Cyclic Redundancy Code (CRC). And failing the redundancy check would result in dropping the packet. In case the CRC check was successful then the destination node would receive incorrect information.

Selective forwarding: Like any multi-hop network, wireless sensor networks are based on a neighbour trust model where each node would trust a neighbouring node to faithfully forward the messages it receives. In a selective forwarding attack [57], a malicious node may refuse to forward certain messages and simply drop them, ensuring that they are not propagated toward their destination. If this node drops all the packets it receives then the neighbouring nodes would think that this node is down and would look for an alternative route. The malicious node may also choose to selectively drop some messages but forward the remaining traffic.

Sinkhole attacks: In a sinkhole attack, the adversary manipulates the neighbouring nodes to lure nearly all the traffic from a particular area through a compromised node thereby creating a sink [55]. This malicious sink can now not only tamper with the transmitted data but can also drop some important messages thereby leading to other attacks like eavesdropping and selective forwarding. Sinkhole attacks typically work by making a compromised node look especially attractive to surrounding nodes with respect to the routing algorithm [57]. This could be done by spoofing or replaying an advertisement for an extremely high quality route to a sink. All the neighbouring node of the adversary will hence start forwarding packets destined for a sink through the adversary, and also propagate the attractiveness of the route to their neighbours.

Wormhole attacks: In the wormhole attack an adversary tunnels messages received in one part of the network over a low-latency link and replays them in a different part [57]. An adversary could convince nodes who would normally be multiple hops from a sink that they are only one or two hops away via the wormhole. This would not only confuse in the routing mechanisms but would also lead to creation of a sinkhole since the adversary on the other side of the wormhole can pretend to have a high quality route to the sink, potentially drawing all traffic in the surrounding area [47]. An adversary situated close to a sink may be able to completely disrupt routing by creating a well-placed wormhole [55].

Sybil attacks: In a Sybil attack [30], a single malicious node illegitimately presents multiple identities to other nodes in the network. The Sybil attack can significantly reduce the effectiveness of fault-tolerant schemes such as distributed storage, disparity and multipath routing, and topology maintenance [57]. Sybil attacks also pose a significant threat to geographic based routing protocols.

The Sybil attacks can take advantage of different layers to cause service disruption [115]. Sybil attack at the MAC layer would help the malicious node to claim a large fraction of the shared radio resource leaving limited resources for legitimate nodes to transmit. Sybil attack at the routing layer will help the malicious node to draw in large amounts of network traffic to go through the same entity [115]. This would result in a sinkhole being created and the attacker can hence do selective forwarding on received packets. Newsome et al. [92] proposes several defence mechanisms against Sybil attacks suited for sensor networks.

9.2. Approaches to security

To achieve the various security requirements discussed in Section 9.1.1 two main areas for security have to be considered:

- Firstly, the key management techniques that looks into the different ways to establish and distribute the security keys among the different nodes in the sensor network.
- Secondly, the cryptographic mechanisms used to encrypt the important data (to provide data confidentiality) and to calculate the MAC (to provide data authenticity and data integrity) using these security keys.

A notable example in which these two security areas have been addressed is the Security Protocol for Sensor Networks, (SPINS) [99]. It provides a simple and effective method to achieve the security requirements addressed in Section 9.1.1. SPINS consists of two security blocks SNEP and μTESLA. While Secure Network Encryption Protocol (SNEP) provides data confidentiality, data authentication and data freshness with low overheads, the micro version of Timed Efficient Streaming Loss-tolerant Authentication protocol (μTESLA) provides a key-chain distribution technique for authenticated streaming broadcasts [42]. Three types of communications are usually considered for using SPINS to provide security in sensor networks [99]:

- Node to sink communication, e.g. sensor measurements
- Sink to node communication, e.g. specific requests
- Sink to all nodes, e.g. routing messages, queries, or reprogramming of the nodes

SNEP is used for the first two types of communications while μTESLA is used for the third type of communication.

9.2.1. Key management and trust setup

This Section describes some of the key establishment and distribution mechanisms proposed to be used in a wireless sensor networks. A key management procedure is an essential constituent of network security. Several numbers of keys can be used in wireless sensor networks depending on the number of communicating nodes: a *pair-wise key* would be used to secure unicast communication between two nodes in the network, a *group-wise key* would be used to secure multicast communication among a group of nodes in the network and a *network-wise key* would be used to secure broadcast communication [22].

Key management techniques can be classified into the following categories depending on the trust required between the different entities and the amount of security information that is pre-installed in the nodes.

Single network-wide key: The most common way is to pre-load a single network-wide key onto all nodes before deployment. Any two neighbouring nodes that have this shared network key can now communicate with each other. This single key would be used for both generating the MAC and for encrypting the data. The major disadvantage of this approach is that the compromise of even a single node would reveal the secret key compromising the entire network. One variant on this idea is to use a single shared master key at pre-deployment and then use this master key to generate individual session keys for a pair of communicating nodes.

Using pairwise-shared keys: In this approach, every node in the sensor network shares a unique symmetric key with every other node in the network. Every node stores $n - 1$ keys, one for each of the other nodes in the network. The main problem for this approach is that it does not scale to large sensor networks as the number of keys that must be stored in each node is proportional to the total number of nodes in the network.

Hybrid-wise key approach: In this method, all the sensor nodes in the network are pre-installed with a combination of network-wise key, group-wise and pair-wise keys according to the security requirements of the given network. An example of such a mechanism is Localized Encryption and Authentication Protocol (LEAP) [138] which is a key management protocol for large scale sensor networks. It is designed to support in-network processing while reducing the security impact of a node compromise. LEAP supports the establishment of four types of keys for each sensor node, an individual key shared with the sink to provide secure communication between the sink and each node, a group key shared by all the nodes in the network used for securing broadcast messages from the sink, a pair-wise key shared between two adjacent nodes to secure communications between neighbours, and a cluster key shared by a node and its neighbours used for securing locally broadcast messages.

Trusted server approach: In this approach, a trusted server is used to establish the session keys shared between the various nodes. This approach is based on the fact that this server needs to be trusted by all the nodes in the network, but this provides a single point of failure prone to directed attacks [103]. Bootstrapping keys using a trusted sink is another option. Here, each node needs to share only a single key with the sink and set up keys with other nodes through the sink. This method is used in SPINS which proposes that the sink be the trusted by all the nodes in the network. At deployment each node is given a common master key which is shared with the sink. All the other session keys are derived from this key [99].

Asymmetric cryptography: This approach is based upon public key cryptographic protocols and algorithms. Public key cryptography is a popular method for key establishment in other wireless networks; however with the low memory, computational capabilities and energy constraints of sensor nodes, public-key algorithms common in asymmetric cryptography limit the practical use of this key distribution scheme [99,123]. Though some recent work has shown that public key cryptography may be possible to use in sensor networks [126,38,81].

μTESLA, a micro version of TESLA, has been proposed to be used in SPINS which overcomes the problems faced by asymmetric cryptography by introducing the asymmetry through a delayed disclosure of symmetric keys [42]. Several people have addressed the issues for authenticating broadcast messages, but most of the proposals use asymmetric digital signatures to provide efficient and strong authentication which have high computations, communication and storage overheads making them impractical for resource constrained sensor devices. To provide efficient authenticated broadcasts using μTESLA, the sink and nodes are loosely time synchronized, and each node knows an upper bound on the maximum synchronization error [42]. Before sending an authenticated packet, the sink would first compute the MAC on the packet with the secret key. This packet would then be transmitted along with the MAC, but the MAC key used for calculating MAC is not yet disclosed. When the sensor node receives this packet, it can verify that the corresponding MAC key was not yet disclosed by the sink. At this stage, the receiving node is assured that the MAC key is known only by the sink, and so an adversary could not have altered the packet in transit. But as the node does not yet have the MAC key it cannot process the packet. Hence the node stores the packet in a buffer. At the time of key disclosure (based on the time schedule for disclosing keys) the sink broadcasts the verification key to all receivers. When a node receives the disclosed key, it firsts verifies the correctness of the key and then uses it to authenticate the packet stored in its buffer.

The sender first chooses the last key K_n of the chain randomly and then repeatedly applies a one-way hash function F to generate a key chain where $K_i = F(K_i + 1)$. Fig. 38 shows the method of releasing the keys at different instances of time to achieve authenticated broadcast in μTESLA. All the nodes in the sensor network would be synchronised with respect to time and retrieve an authenticated initial key for the key chain in a secure and authenticated manner using SNEP [42], the other security block of SPINS which is described further in Section 9.2.2. Hence at time t_0, all the nodes know the initial key K_0. All the packets broadcasted by the sink in the time interval between t_1 and t_2 i.e. Packets P_1 and P_2 contain a MAC with key K_1. At this time the receiver nodes do not know the Key K_1 and so are unable to authenticate the packets P_1 and P_2. During the time interval t_2 and t_3 along with two packets P_3 and P_4 (which use Key K_2), the key K_1 is also broadcasted. Now the nodes would first authenticate the key broadcast by using $K_0 = F(K_1)$ and then use the key K_1 to authenticate the packets P_1 and P_2. The key broadcasts are not added to the data packets being broadcasted but instead the sender broadcast the current key periodically in a special packet.

Random key pre-distribution scheme: In the random key pre-distribution protocols a large pool of symmetric keys is chosen and each node is assigned with a random subset of the pool (key ring). The size of the key ring assigned to each node should be sufficiently large in order to ensure that each node shares at least a key with a sufficiently large number of neighbours (hence it can communicate directly with all of these neighbours), so that the network is fully connected and hence the nodes do not have to depend on a centralised trusted sink to distribute the keys.

Eschenauer and Gilgor [32] proposed a random key pre-distribution scheme for a distributed sensor network based on probabilistic key sharing and utilization of a simple shared-key discovery protocol for key distribution, key revocation, and node re-keying. Each sensor is installed with a key ring at pre-deployment. Upon deployment and network initialization, sensor nodes will be able to establish a secure and direct communication link provided that a shared key exists between one or more pairs of sensor nodes. If two nodes do not share a common key then an intermediary node with a common key between the two sensor nodes would be selected to establish a common session key. It was seen that to establish an almost certain shared-key connectivity for a network with 10,000 nodes, a key ring of only 250 keys randomly selected from a 100,000 pool has to be pre-distributed to every sensor node [32].

9.2.2. Cryptographic mechanisms

This Section describes the mechanisms by which the keys established and distributed to the nodes are used to provide data authenticity, integrity and data confidentiality.

Secure Network Encryption Protocol (SNEP): Strong encryption techniques are used in SNEP to provide data confidentiality while a MAC is used to provide data authentication in two-party communication. If the MAC verifies correctly, a receiver can be assured that the message originated from the claimed sender. To ensure freshness each node also maintains a counter which is synchronized with the one in the sink.

The transmitted messages are encrypted with a chaining encryption function i.e. Data Encryption Standard-Cipher Block Chaining (DES-CBC) to provide strong data confidentiality. SNEP also proved semantic security which ensures that an adversary would not be able to recover any information of the transmitted message even if it get holds of multiple encryption of the same message. To achieve this, randomisation for encryption is required where before encrypting the message, the sender precedes the bits with a random bit string. This is called the Initialisation Vector (IV). This would prevent the adversary from deducing the plaintext of the encrypted message if it knows the plain text-cipher text pairs encrypted with the same key [42,99]. However more energy would be required to transmit these extra bits over the radio channel and this may be crucial for sensor nodes due to their limited power resources. In order to avoid this extra overhead, SNEP proposes to use a shared counter between the sender and the receiver for the block cipher in counter mode [99]. As the two communicating devices increment the shared counter after each block, the same message is encrypted differently each time. As the counter state is kept at each end point it would not be required to be transmitted over the radio channel. The counter value is long enough that it never repeats within the lifetime of the node. This counter value in the MAC also prevents replaying old messages as any messages with the old counter values would be discarded by the device. This also provides weak freshness.

The following notation is used to describe the security protocol and cryptographic operations between two communicating nodes A and B in SPINS [99]:

K_{AB}: The secret symmetric key shared between A and B
$\{M\}_{<\mathrm{Kenc},C>}$: Message M encrypted with K_{enc} using C as the IV
$M_1|M_2$: Concatenation of two data messages M_1 and M_2
MAC(K_{mac}, $C|E$): The Message authentication code generated using the secret key K_{mac} and C as the IV on the encrypted data E.

The complete message sent from device A to device B consists of the encrypted message and the MAC generated for the encrypted data:

$$A \rightarrow B : \{D\}_{<\mathrm{Kenc},C>},\ \mathrm{MAC}\ (K_{\mathrm{mac}}, C|\{D\}_{<\mathrm{Kenc},C>})$$

Here K_{enc}, the key used for encrypting the data and K_{mac}, the key used to generate the MAC are both generated from the shared master secret key.

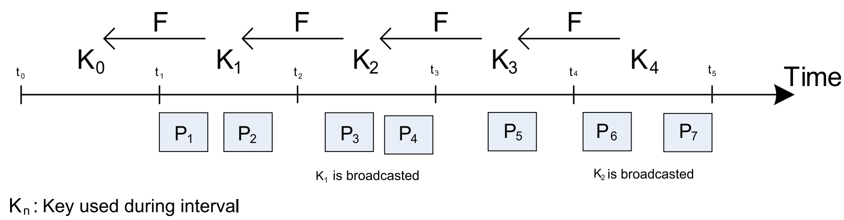


Fig. 38. Time release key chain in μTESLA.

TinySec: TinySec [58] is a link layer security protocol that provides data authentication, data integrity, data confidentiality and even semantic security. In-network processing is highly important in sensor networks in particular for data aggregation and duplicate message eliminations. Hence using an end-to-end security mechanism would create problems when intermediate nodes need to access, modify or discard messages. To counter this, the authors in [58] propose to use a link layer security mechanism to achieve the above mentioned basic security requirements and not hinder in-network processing. The TinySec packet format is based on the packet format for TinyOS shown in Fig. 39. The destination address (Dest), active message type (AM) and the length fields from the TinyOS are retained in TinySec also. To detect transmission errors, TinyOS computes a 16-bit Cyclic Redundancy Code (CRC) over the packet. To guarantee message integrity and authenticity TinySec replaces this CRC with a MAC. The MAC would detect any tampering of the transmitted data and would also detect transmission errors.

TinySec supports two different security options:

- *TinySec-Auth (authentication only):* In this mode, TinySec uses a 4 byte MAC to authenticate the entire packet but the data payload is not encrypted. TinySec uses cipher block chaining, CBC-MAC for computing and verifying the MAC. CBC-MAC is efficient and fast and it requires only a few cryptographic primitives as it relies on a block cipher [58]. The MAC is calculated over the packet header and the data payload thereby authenticating the whole packet. Fig. 40 shows the packet format for the TinySec-Auth mode.
- *TinySec-AE (authenticated encryption):* In this mode, the data payload is encrypted and then a MAC is used to authenticate the packet. The MAC is calculated over the encrypted data payload and the packet header. In this mode, two new fields, the source address and a 16-bit counter have been added to the packet header. This 8-byte packet header is used as the initialisation vector for encrypting the payload. The default block cipher used in TinySec is Skipjack [58]. Hence data confidentiality, data integrity and data authenticity is achieved by encrypting the payload and authenticating the whole packet (Fig. 41).

In TinySec two skipjack keys, one for encrypting the data and one for computing the MAC are used. Hence a keying mechanism that determines how the cryptographic keys are distributed and shared throughout the network is required. The simplest keying mechanism is to use a single network wide TinySec key. This method would be very simple to manage as the keys can be loaded onto the nodes at the

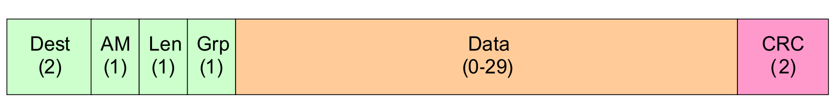


Fig. 39. Tiny OS packet format.

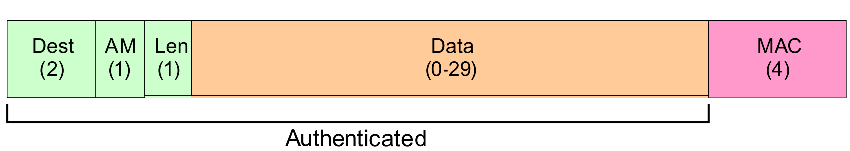


Fig. 40. TinySec-Auth packet format.

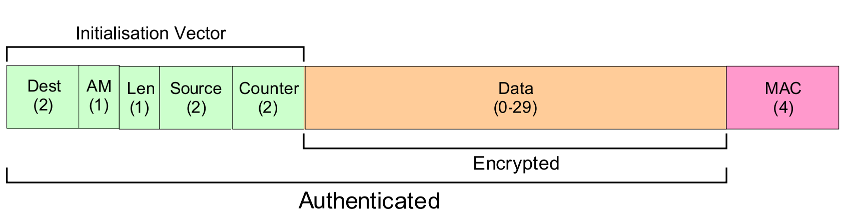


Fig. 41. TinySec-AE packet format.

time of deployment. However, a network-wide key would be highly insecure especially against node capture attacks. A more robust method is to have per-link keying where each communicating pairs of nodes would have unique key. Though this method provides highly secure transmission, it is highly challenging to manage the key distribution among large networks with hundreds of communicating sensor nodes. This method also poses problem for in-network processing as the intermediate nodes would not have the keys to decrypt the data as the packets were not directly addressed to them. It is also difficult to broadcast any messages through the whole network due to lack of common shared keys.

10. Discussion and conclusions

In this paper we have reviewed the ZigBee/IEEE 802.15.4 standards and the recent literature on wireless sensor networks. In particular this work presents an overview of the energy efficiency, communication, data management and security solutions adopted by the standard and proposed in the recent literature. In some case we observed a convergence of the standards and of the main research results (as it is the case of the security), while in others we observed significant differences.

10.1. Routing

As is apparent from the previous discussion, the ZigBee approach significantly differs from the ideas and concept emerging from recently proposed routing protocols. Specifically, while ZigBee adopts an AODV-based routing protocol, recent research has focused on geographic routing, either based on physical coordinates or on virtual coordinates. The geographic routing approach is motivated by the need of scalable routing protocols for very large sensor networks. It should be observed however, that physical coordinates require sensors embedding GPS devices, while virtual coordinate system are still in their early phase of research and they do not appear suitable yet to the purpose of standardization. From this point of view the ZigBee routing protocol is more stable and reliable. Furthermore current (and immediate future) sensor networks have small to moderate size and AODV-like protocols probably will be enough to handle routing.

10.2. Energy efficiency

The energy efficiency approach of the ZigBee standard is mainly at the physical and MAC layers. ZigBee supports two operating modes. One is based on a TDMA algorithm and it is very effective but limited in scope to star network configurations where fine grained clock synchronization and slot assignment can easily be provided by the coordinator. The other operating mode is based on CSMA and basically tries to reduce power consumption with very low duty cycles. Several MAC layer approaches have been proposed in the research community but these are generally inflexible with respect to different data rates. Research has recently proposed solutions based on cooperation from different layers in the protocol stack. Cross layer approaches can use network and/or application layer information to drive radio operation more efficiently and taking into account actual data rates. Cross layer approaches to energy efficiency could possibly be developed in the ZigBee standard. However, they are more complex to implement with respect to MAC layer only solutions. The latter also maintain independence of the stack protocols.

10.3. Security

The ZigBee standard standard specifies the requirements and mechanisms for providing sensor security. As discussed in Section 9 the ZigBee standard also acknowledges that the public key cryptographic mechanism may not be currently suitable for sensor networks and hence suggests using symmetric cryptography mechanism. Similar to the hybrid-wise key approaches discussed in Section 9 ZigBee also proposes to use two session keys, the link key for communication between two nodes, a network key used for broadcasting messages and also an initial master key that would be used to generate these session keys. Though using a common network key is required for broadcasting messages across the whole network, such a single network-wide key is can be easily used to attack the system like when an adversary may capture a node that has left the network but may still have the network key. Hence ZigBee proposes to periodically change the network level key so that when nodes leave or join the network, fresh network-wide keys would be used.

The ZigBee standard also proposes to have a centralised trust centre that is trusted by all the nodes in the network and is responsible for generation of session keys and admission control of nodes trying to connect to the sensor network. This mechanism is similar to the trusted server approach used by SPINS as described in Section 9. Having such a trust centre though has an advantage of providing a central control on security of the network, but it also leads to a single point of failure which could be prone to directed attacks. Also similar to the Tiny Sec and SPINS protocol, ZigBee also uses a counter to provide data freshness and Message Authentication Code to provide data integrity.

An important functionality of ZigBee that is different from the other proposed security solutions is that it provides mechanism to encrypt data at three different layers (MAC, NWK and APS layer). It also supports security in different layers together, example: An APS command may be secured by the APS layer security and when this packet is sent to the MAC Layer may be further secured using the MAC layer security.

References

[1] O.B. Akan, I.F. Akyildiz, Event-to-sink reliable transport in wireless sensor networks, IEEE/ACM Transactions on Networking 13 (5) (2005) 1003–1016.

[2] I.F. Akyildiz, W.J. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine (2002) 102–114.
[3] I.F. Akyildiz, W.J. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422.
[4] M. Aly, K. Pruhs, P.K. Chrysanthis, KDDCS: a load-balanced in-network data-centric storage scheme for sensor networks, in: Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM 2006), November 2006, pp. 317–326.
[5] G. Amato, S. Chessa, F. Conforti, A. Macerata, C. Marchesi, Health care monitoring of mobile patients, Ercim news (n. 60) (2005).
[6] G. Amato, P. Baronti, S. Chessa, MaD-WiSe: programming and accessing data in a wireless sensor network, in: International Conference on Computer as a Tool (EUROCON 2005), November 2005, pp. 1846–1849.
[7] G. Amato, P. Baronti, S. Chessa, V. Masi, The Stream System: a data collection and communication abstraction for sensor networks, in: 2006 IEEE International Conference on Systems, Man, and Cybernetics, October 2006.
[8] G. Amato, P. Baronti, S. Chessa, MaD-WiSe: a distributed query processor for wireless sensor networks, Technical Report ISTI-2006-TR-39.
[9] A. Bachir, D. Barthel, M. Heusse, A. Duda, Hidden node avoidance in wireless sensor networks, in: International Conference on Wireless Networks, Communications and Mobile Computing, Volume 1, 2005, pp. 612–617.
[10] M. Bahramgiri, M.T. Hajiaghayi, V.S. Mirrokni, Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks, in: Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN02), Miami, FL, 14–16 October 2002, pp. 392–397.
[11] L. Bao, J.J. Garcia-Luna-Aceves, A new approach to channel access scheduling for ad hoc networks, in: The Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom 2001), Rome, Italy, July 2001, pp. 210–221.
[12] C. Bettstetter, On the minimum node degree and connectivity of a wireless multihop network, in: Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, Lausanne, Switzerland, July 2001, 2002.
[13] V. Bharghavan, A. Demers, S. Shenker, L. Zhang, MACAW: a media access protocol for wireless LAN's, in: Proceedings of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, London ,UK, August–September 1994, pp. 212–225.
[14] D. Blough, M. Leoncini, G. Resta, P. Santi, The K-neigh protocol for symmetric topology control in ad hoc networks, in: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Annapolis, MD, 2003.
[15] P. Bonnet, J. Gehrke, P. Seshadri, Querying the physical world, IEEE Personal Communications 7 (5) (2000) 10–15.
[16] P. Bonnet, J. Gehrke, P. Seshadri, Towards sensor database systems, in: Proceedings of the 2nd International Conference on Mobile Data Management (MDM 2001), Hong Kong, China, January 2001, pp. 3–14.
[17] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Wireless Networks 7 (6) (2001) 609–616.
[18] B. Bougard, F. Catthoor, D.C. Daly, A. Chandrakasan, W. Dehaene, Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: modeling and improvement perspectives, in: Proceedings of Design, Automation, and Test in Europe (DATE), March 2005.
[19] J. Brack, J. Gao, A. (Andrew) Jiang, MAP: medial axis based geometric routing in sensor networks, in: Proceedings of the 11th International Conference on Mobile Computing and Networking (MobiCom 2005), Cologne, Germany, August–September 2005, pp. 88–102.
[20] BTnodes - A Distributed Environment for Prototyping Ad Hoc Networks. http://www.btnode.ethz.ch/.
[21] N. Bulusu, J. Heidemann, D. Estrin, GPS-less low cost outdoor localization for very small devices, IEEE Personal Communications Magazine 7 (5) (2000) 28–34.
[22] S.A. Camtepe, B. Yener, Key distribution mechanisms for wireless sensor networks: a survey, Technical Report TR-05-07, Rensselaer Polytechnic Institute, March 23, 2005.
[23] Q. Cao, T. Abdelzaher, Scalable logical coordinates framework for routing in wireless sensor networks, in: Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS 2004), Lisbon, Portugal, December 2004, pp. 349–358.
[24] S. Capkun, M. Hamdi, J.-P. Hubaux, GPS-free positioning in mobile ad-hoc networks, in: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS 2001), Maui, HI, USA, January 2001, pp. 3481–3490.
[25] A. Caruso, S. Chessa, S. De, A. Urpi, GPS free coordinate assignment and routing in wireless sensor networks, in: Proceedings of the 24th Joint Conference of the IEEE Computer and Communications Societies (Infocom 2005), Miami, FL, USA, March 2005, pp. 150–160.
[26] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, J. Zhao, Habitat monitoring: application driver for wireless communications technology, in: Proceedings of the 1st ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, San Jose, Costa Rica, April 2001, pp. 20–41.
[27] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Span: an energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks, ACM Wireless Networks Journal 8 (5) (2002) 481–494.
[28] Crossbow Technology Inc. http://www.xbow.com.
[29] T. Van Dam, K. Langendoen, An adaptive energy-efficient mac protocol for wireless sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003), Los Angeles, CA, USA, November 2003, pp. 171–180.
[30] J.R. Douceur, The Sybil attack, in: Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02), 2002.
[31] C.C. Enz, A. El-Hoiydi, J.-D. Decotignie, V. Peiri, WiseNET: an ultralow-power wireless sensor network solution, Computer 37 (8) (2004) 62–70.
[32] L. Eschenauer, V.D. Gligor, A key management scheme for distributed sensor networks, in: Proceedings of the 9th ACM conference on Computer and Communications Security (CCS02), Washington, November 2002, pp. 41–47.
[33] Q. Fang, J. Gao, L.J. Guibas, Locating and bypassing routing holes in sensor networks, in: Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2004), Hong Kong, March 2004, pp. 2458–2468.
[34] Q. Fang, J. Gao, L.J. Guibas, V. de Silva, L. Zhang, GLIDER: gradient landmark-based distributed routing for sensor networks, in: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2005), Miami, FL, USA, March 2005, pp. 339–350.
[35] R. Fonseca, S. Ratnasamy, J. Zhao, C. Tien Ee, D. Culler, S. Shenker, I. Stoica, Beacon vector routing: scalable point-to-point routing in wireless sensornets, in: Proceedings of the 2nd Symposium on Networked Systems Desing & Implementation (NSDI 2005), Boston, MA, USA, May 2005.
[36] D. Ganesan, R. Govindan, S. Shenker, D. Estrin, Highly-resilient, energy-efficient multipath routing in wireless sensor networks, ACM SIGMOBILE Mobile Computing and Communications Review 5 (4) (2001) 11–25.
[37] T. Gao, D. Greenspan, M. Welsh, Improving patient monitoring and tracking in emergency response, in: Proceedings of the International Conference on Information Communication Technologies in Health, July 2005.

[38] G. Gaubatz, J. Kaps, B. Sunar, Public Key Cryptography in sensor networks – revisited, in: Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004), 2004.

[39] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, D. Culler, The nesC language: a holistic approach to networked embedded systems, in: Proceedings of the ACM SIGPLAN conference on programming language design and implementation (PLDI 2003), San Diego, CA, USA, June 2003, pp. 1–11.

[40] L. Girod, D. Estrin, Robust range estimation using acoustic and multimodal sensing, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001), Maui, HI, USA, October–November 2001, pp. 1312–1320.

[41] P. Gupta, P.R. Kumar, Critical power for asymptotic connectivity in wireless networks, in: W.M. McEneany, G. Yin, Q. Zhang (Eds.), Stochastic Analysis, Control, Optimisation and Applications, Birkhauser, Boston, M.A, 1998, pp. 547–566.

[42] A. Hac, Wireless Sensor Network Designs, John Wiley, New York, 2003.

[43] T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: Proceedings of the 9th International Conference on Mobile Computing and Networking (MobiCom 2003), San Diego, CA, USA, September 2003, pp. 81–95.

[44] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.E. Culler, K.S.J. Pister, System architecture directions for networked sensors, in: Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), Cambridge, MA, USA, November 2000, pp. 93–104.

[45] B. Hohlt, L. Doherty, E. Brewer, Flexible power scheduling for sensor networks, in: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN 2004), Berkeley, CA, USA, April 2004, pp. 205–214.

[46] H. Karl, A. Willig, Protocols and Architectures for Wireless Sensor Networks, John-Wiley, New York, 2005.

[47] Y.-C. Hu, A. Perrig, D.B. Johnson, Packet leashes: a defense against wormhole attacks in wireless networks, in: Proceedings of the 22nd Joint Conference of the IEEE Computer and Communications Societies (Infocom), 2003.

[48] Z. Huang, C. Shen, C. Srisathapornphat, C. Jaikaeo, Topology control for ad hoc networks with directional antennas, in: Proceedings of the IEEE International Conference in Computer Communication and Networks (ICCCN02), Miami, FL, 14–16 October 2002, pp. 392–397.

[49] L.-J. Hwang, S.-T. Sheu, Y.-Y. Shih, Y.-C. Cheng, Grouping strategy for solving hidden node problem in IEEE 802.15.4 LR-WPAN, in: Proceedings of 1st International Conference on Wireless Internet, July 2005, pp. 26–32.

[50] Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.15.4-2003 "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), New York, IEEE Press. October 1, 2003.

[51] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, USA, August 2000, pp. 56–67.

[52] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, Mobile Computing, vol. 353, Kluwer Academic Publishers, Dordrecht, 1996, pp. 153–181.

[53] X. Ji, H. Zha, Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling, in: Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2004), Hong Kong, March 2004, pp. 2652–2661.

[54] J.M. Kahn, R.H. Katz, K.S.J. Pister, Next century challenges: mobile networking for "Smart Dust", in: Proceedings of the 5th International Conference on Mobile Computing and Networking (MobiCom 1999), Seattle, WA, USA, August 1999, pp. 271–278.

[55] R. El Kaissi, A. Kayssi, A. Chehab, Z. Dawy, DAWWSEN: a defence mechanism against wormhole attacks in wireless sensor networks, in: Proceedings of the 2nd International Conference on Innovations in Information Technology (IIT'05), Dubai, UAE, September 2005.

[56] B. Karp, H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, USA, August 2000, pp. 243–254.

[57] C. Karlof, D. Wagner, Secure routing in wireless sensor networks: attacks and countermeasures, Ad Hoc Networks 1 (2003) 293–315.

[58] C. Karlof, N. Sastry, D. Wagner, TinySec: a link layer security architecture for wireless sensor networks, ACM SenSys 2004, Maryland, USA, November 2004.

[59] J.N. Al-Karaki, A.E. Kamal, Routing techniques in wireless sensor networks: a survey, IEEE Wireless Communications (2004).

[60] H. Karl, A. Willig, Protocols and Architectures for Wireless Sensor Networks, Wiley, London, New York, 2005.

[61] K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks, Ad Hoc Networks 3 (3) (2005) 325–349.

[62] S. Kim, R. Fonseca, D. Culler, Reliable transfer on wireless sensor networks, in: 1st Conference on Sensor and Ad Hoc Communications and Networks, October 2004, pp. 449–459.

[63] Y.J. Kim, R. Govindan, B. Karp, S. Shenker, Geographic routing made practical, in: Proceedings of the 2nd Symposium on Networked Systems Designing and Implementation (NSDI 2005), Boston, MA, USA, May 2005, pp. 217–230.

[64] Y.-J. Kim, R. Govinidan, B. Karp, S. Shenker, On the pitfalls of geographic face routing, in: Proceedings of the Dial M for Mobility – principle of Mobile Computing (DIALM-POMC 2005), Cologne – Germany, September 2005, pp. 34–43.

[65] T.H. Kim, S. Choi, Priority-based delay mitigation for event-monitoring IEEE 802.15.4 LR-WPANs, IEEE Communications Letters 10 (3) (2006) 213–215.

[66] A. Koubaa, M. Alves, E. Tovar, GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks, Parallel and Distributed Processing Symposium (IPDPS 2006), April 25–29, 2006.

[67] E. Kranakis, H. Singh, J. Urrutia, Compass routing on geometric networks, in: Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG'99), Vancouver, BC, Canada, August 1999, pp. 51–54.

[68] F. Kuhn, R. Wattenhofer, A. Zollinger, Worst-case optimal and average-case efficient geometric ad-hoc routing, in: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), pp. 267–278.

[69] F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger, Geometric ad-hoc routing: of theory and practice, in: Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003), Boston, MA, USA, July 2003, pp. 63–72.

[70] B. Leong, S. Mitra, B. Liskov, Path vector face routing: geographic routing with local face information, in: Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP 2005), Boston, MA, USA, November 2005, pp. 147–158.

[71] B. Leong, B. Liskov, R. Morris, Geographic routing without planarization, in: Proceedings of the 3rd Symposium on Networked Systems Design and Implementation (NSDI 2006), San Jose, Costa Rica, May 2006, pp. 339–352.

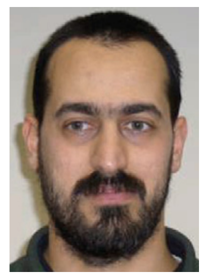
[72] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, R. Morris, A scalable location service for geographic ad hoc routing, in: Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, USA, August 2000, pp. 120–130.

[73] X. Li, Y.-J. Kim, R. Govindan, W. Hong, Multi-dimensional range queries in sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003), November 2003, pp. 63–75.

[74] H. Lim, J.C. Hou, Localization for anisotropic sensor networks, in: Proceedings of the 24th Annual Joint Conference of the IEEE

Computer and Communications Societies (Infocom 2005), Miami, FL, USA, March 2005, pp. 138–149.
[75] M. Lin, A. Kumar, X. Qing, S.J. Beard, S.S. Russell, J.L. Walker, T.K. Delay, Monitoring the integrity of filament wound structures using built-in sensor networks, in: Proceedings of the SPIE Volume 5054 – Smart Structures and Materials 2003: Industrial and Commercial Applications of Smart Structures Technologies, San Diego, CA, USA March 2003, pp. 222–229.
[76] P. Lin, C. Qiao, X. Wang, Medium access control with a dynamic duty cycle for sensor networks, in: Proceedings of the Wireless Communications and Networking Conference (WCNC 2004), Atlanta, GA, USA, March 2004, pp. 1534–1539.
[77] G. Lu, B. Krishnamachari, C.S. Raghavendra, An adaptive energy-efficient and low- latency MAC for data gathering in wireless sensor networks, in: Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS 2004), Santa Fe, NM, USA, April 2004, p. 26–30.
[78] S. Madden, R. Szewczyk, M.J. Franklin, D. Culler, Supporting aggregate queries over ad-hoc wireless sensor networks, in: Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002), Callicoon, NY, USA, June 2002, pp. 49–58.
[79] S. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong, TAG: a tiny aggregation service for ad-hoc sensor networks, in: Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI2002), Boston, MA, USA, December 2002.
[80] S. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong, The design of an acquisitional query processor for sensor networks, in: Proceedings of the SIGMOD 2003, San Diego, CA, USA, June 2003, pp. 491–502.
[81] D.J. Malan, M. Welsh, M.D. Smith, A public key infrastructure for key distribution in TinyOS based on elliptic curve cryptography, in: Proceedings of the 1st IEEE communications Society Conference on Sensor and Ad-Hoc Communications and Networks, 2004.
[82] D. Malan, T. Fulford-Jones, M. Welsh, S. Moulton, CodeBlue: an ad hoc sensor network infrastructure for emergency medical care, in: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks, April 2004.
[83] M.J. Miller, N.H. Vaidya, Minimizing energy consumption in sensor networks using a wakeup radio, in: Proceedings of the Wireless Communications and Networking Conference (WCNC 2004), Atlanta, GA, USA, March 2004, pp. 2335–2340.
[84] Misic, S. Shafi, V.B. Misic, Performance of a beacon enabled IEEE 802.15.4 cluster with downlink and uplink traffic, IEEE Transactions on Parallel and Distributed Systems 17 (4) (2006) 361–376.
[85] J. Misic, S. Shafi, V.B. Misic, Maintaining reliability through activity management in an 802.15.4 sensor cluster, IEEE Transactions on Vehicular Technology 55 (3) (2006) 779–788.
[86] Moteiv Corporation, http://www.moteiv.com.
[87] R. Nagpal, H. Shrobe, J. Bachrach, Organizing a global coordinate system from local information on an ad hoc sensor network, in: Proceedings of the 2nd International Symposium on Information Processing in Sensor Networks (IPSN 2003), Paolo Alto, CA, USA, April 2003, pp. 333–348.
[88] S. Narayanswamy, V. Kawadia, R.S. Sreenivas, P.R. Kumar, Power control in ad hoc networks: theory, architecture, algorithm and implementation of the COMPOW protocol, in: Proceedings of European Wireless 2002, Florence, Italy, February 2002.
[89] A. Nasipuri, K. Li, A directionality based location discovery scheme for wireless sensor networks, in: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA2002), Atlanta, GA, USA, September 2002, pp. 105–111.
[90] M. Neugebauer, J. Plonnigs, K. Kabitzsch, A new beacon order adaptation algorithm for IEEE 802.15.4 networks, in: Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN 2005), Jan 31–Feb 2, 2005.
[91] J. Newsome, D. Song, GEM: graph embedding for routing and data-centric storage in sensor networks without geographic information, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003), Los Angeles, CA, USA, November 2003, pp. 76–88.
[92] J. Newsome, E. Shi, D. Song, A. Perrig, The Sybil attack in sensor networks: analysis and defenses, in: Proceedings of the IEEE International Conference, Processing in Sensor Networks, April 2004.
[93] D.S. Niculescu, B. Nath, DV based positioning in ad hoc networks, Telecommunication Systems 22 (1–4) (2003) 267–280.
[94] D. Niculescu, B. Nath, Ad hoc positioning system (APS) using AOA, in: Proceedings of the 22nd Joint Conference of the IEEE Computer and Communications Societies (Infocom), San Francisco, CA, USA, March–April 2003, pp. 1734–1743.
[95] Nut/OS – The BTnode operating system core, http://www.ethernut.de.
[96] P. Panchapakesan, D. Manjunath, On the transmission range in dense ad hoc radio networks, in: Proceedings of the IEEE SPCOM '01, 2001.
[99] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J.D. Tygar, SPINS: security for sensor networks, in: Proceedings of the Mobile Computing and Networking, Italy, 2001.
[100] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999), New Orleans, LA, USA, February 1999, pp. 90–100.
[101] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD, USA, November 2004, pp. 95–107.
[102] N.B. Priyantha, A. Chakraborty, H. Balakrishnan, The Cricket location-support system, in: Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, USA, August 2000, pp. 32–43.
[103] A. Price, K. Kosaka, S. Chatterjee, A secure key management scheme for sensor networks, in: Proceedings of the 10th Americas Conference on Information Systems, New York, NY, August 2004.
[104] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, Energy efficient,collision-free medium access control for wireless sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003), Los Angeles, CA, USA, November 2003, pp. 181–192.
[105] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, I. Stoica, Geographic routing without location information, in: Proceedings of the 9th International Conference on Mobile Computing and Networking (MobiCom 2003), San Diego, CA, USA, September 2003, pp. 96–108.
[106] T. Rappaport, Wireless Communications: Principles and Practice, second ed., Prentice Hall, Upper Saddle River, NJ, 2002.
[107] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, F. Yu, Data-centric storage in sensornets with GHT, a geographic hash table, Mobile Networks and Applications 8 (4) (2003) 427–442.
[108] V. Rodoplu, T.H. Meng, Minimum energy mobile wireless networks, IEEE Journal of Selected Areas on Communication 17 (8) (1999) 1333–1344.
[109] P. Santi, Topology control in wireless ad hoc and sensor networks, ACM Computing Surveys 37 (2) (2005) 164–194.
[110] P. Santi, Topology control in wireless ad hoc and sensor networks, Wiley, London, New York, 2005.
[111] C. Schurgers, V. Tsiatsis, S. Ganeriwal, M.B. Srivastava, Optimizing sensor networks in the energy-latency-density design space, IEEE Transactions on Mobile Computing 1 (1) (2002) 70–80.
[112] A. Sawides, C.-C. Han, M.B. Strivastava, Dynamic fine-grained localization in adhoc networks of sensors, in: Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom 2001), Rome, Italy, July 2001, pp. 166–179.
[113] K. Seada, A. Helmy, R. Govindan, On the effect of localization errors on geographic face routing in sensor networks, in: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN 2004), Berkeley, CA, USA, April 2004, pp. 71–80.

[114] Y. Shang, W. Ruml, Improved MDS-based localization, in: Proceedings of the 23th Joint Conference of the IEEE Computer and Communications Societies (Infocom 2004), Hong Kong, March 2004, pp. 2640–2651.
[115] E. Shi, A. Perrig, Designing secure sensor networks, IEEE Wireless Communications (2004) 38–43.
[116] M.L. Sichitiu, Cross-layer scheduling for power efficiency in wireless sensor networks, in: Proceedings of the 23th Joint Conference of the IEEE Computer and Communications Societies (Infocom 2004), Hong Kong, March 2004, pp. 1740–1750.
[117] SmartDust: Autonomous sensing and communication in a cubic millimeter. http://robotics.eecs.berkeley.edu/pister/SmartDust/.
[118] M. Srivastava, R. Muntz, M. Potkonjak, Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments, in: Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom 2001), Rome, Italy, July 2001, pp. 132–138.
[119] D.C. Steere, A. Baptista, D. McNamee, C. Pu, J. Walpole, Research challenges in environmental observation and forecasting systems, in: Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, USA, August 2000, pp. 292–299.
[120] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, D. Culler, An analysis of a large scale habitat monitoring application, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD, USA, November 2004, pp. 214–226.
[121] T. Sun, L.-J. Chen, C.-C. Han, G. Yang, M. Gerla, Measuring effective capacity of IEEE 802.15.4 beaconless mode, in: Wireless Communications and Networking Conference, (WCNC 2006) April 2006, pp. 492–498.
[122] TinyOs Community Forum, http://www.tinyos.net.
[123] J.P. Walthers, Z. Liang, W. Shi, V. Chaudhary, Wireless sensor networks security: a survey, Technical Report MIST-TR-2005-007, July 2005.
[124] H. Wang, J. Elson, L. Girod, D. Estrin, K. Yao, Target classification and localization in habitat monitoring, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003), Hong Kong, April 2003, pp. 844–847.
[125] R. Wattenhofer, L. Li, P. Bahl, Y.M. Wang, Distributed topology control for power efficient operation in multihop wireless ad hoc networks, in: Proceedings of the IEEE Infocom, Anchorage, AK, April 2001.
[126] R. Watro, D. Kong, S.-F. Cuti, C. Gardiner, C. Lynn, P. Kruss, TinyPK, securing sensor networks with public key technology, in: Proceedings of the SASN'04, October 25, 2004.
[127] A. Wood, J.A. Stankovic, Denial of service in sensor networks, IEEE Computer 35 (10) (2002) 54–62.
[128] Y. Xu, J. Heidemann, D. Estrin, Geography informed energy conservation for ad hoc routing, in: Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom 2001), Rome, Italy, July 2001, pp. 70–84.
[129] F. Xue, P.R. Kumar, The number of neighbours needed for connectivity of wireless networks, Wireless Networks 10 (2) (2004) 169–181.
[130] Y. Yao, J. Gehrke, The cougar approach to in-network query processing in sensor networks, SIGMOD Record 31 (3) (2002) 9–18.
[131] Y. Yao, J. Gehrke, Query processing for sensor networks, in: Proceedings of the 1st Conference on Innovative Data Systems Research (CIDR 2003), Asilomar, CA, USA, January 2003.
[132] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking 12 (3) (2004) 493–506.
[133] F. Zhao, L. Guibas, Wireless Sensor Networks - An Information Processing Approach, Morgan Kaufman Publisher, S. Francisco, 2004.
[134] R. Zheng, J.C. Hou, L. Sha, Asynchronous wakeup for ad hoc networks, in: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), Annapolis, MD, USA, June 2003, pp. 35–45.
[135] R. Zheng, R. Kravets, On-demand power management for ad hoc networks, in: Proceedings of the 22nd Joint Conference of the IEEE Computer and Communications Societies (Infocom 2003), San Francisco, CA, USA, March–April 2003, pp. 481–491.
[136] J. Zheng, M.J. Lee, Will IEEE 802.15.4 make ubiquitous networking a reality? IEEE Communication Magazine (2004).
[137] J. Zheng, M.J. Lee, A comprehensive performance study of IEEE 802.15.4, in: Sensor Network Operations, IEEE Press, Wiley Interscience, New York, 2006, pp. 218–237, Chapter 4.
[138] S. Zhu, S. Setia, S. Jajodia, LEAP: efficient security mechanism for large scale distributed sensor networks, in: Proceedings of the 10th ACM conference on Computer and Communication Security (CCS03), October 2003, pp. 62–72.
[139] ZigBee Alliance, ZigBee Specifications, version 1.0, April 2005.



Paolo Baronti received the Master of Computer Science from the University of Pisa, Italy, in 2004. He was a researcher with the Istituto di Scienza e Tecnologie dell'Informazione of the Italian National Research Council (CNR) from 2004 to 2006. He is now a PhD student at the University of Pisa, Italy. His research interests include energy efficiency, routing, localization and data management in wireless sensor networks.

Prashant Pillai is currently working as a Research Assistant in the School of Engineering, Design and Technology at the University of Bradford. He has received his B. Sc in Electronics and M. Sc in Informatics from University of Delhi, India. He is also currently working towards his PhD in the field of network security at University of Bradford. He has worked before in the WirelessCabin Project funded under the EU FP5 IST framework, the SatNEx project funded under the Network of Excellence mechanism in the EU FP6 IST Framework. He is currently working on the Multicast BGAN Extension project funded by INMARSAT Ltd., and SatNEx-2. His main areas of work are in Mobile/wireless networks like 2G/3G, WLAN and Bluetooth and Satellite based networks, looking into the System architecture design, protocol development and design of the AAA/security architectures. His current research interests include AAA and security protocols for IP and Mobile Networks, DVB based systems and Wireless Sensor Networks. He also takes lectures at the University of Bradford in the areas of Mobile Communications and Wireless Sensor Networks.

Vince W.C Chook received the B.Eng. Honor degree in Electrical & Electronic Engineering and the MSc degree in Integrated Circuit Design from the Imperial College of London. Mr. Chook joined the University of Bradford as a researcher in 2003 and has been actively involved in various European Projects – the IST WirelessCabin, SatNEx and the Inmarsat BGAN Extension Project. During his involvement in the WirelessCabin project, Mr. Chook was appointed to participate in the demonstrator flight test where he was responsible for the system performance evaluation & analysis. In 2005, Mr. Chook was actively involved in the Inmarsat project where he was responsible for the system protocol validation and the Radio Resource Management design. Currently, Mr. Chook is completing his MPhil in Wireless Sensor Network.

Stefano Chessa received the M.S. degree ("cum laude") in Computer Science from the University of Pisa, Italy, in 1994, and the Ph.D. degree in Computer Science from the same university in 1999. In 1997 he was Visiting Student at the Iowa State University, USA; in 1999 he was Research Assistant with the Department of Mathematics of the University of Trento, Italy; and in 2000 he was Research Assistant with the Istituto di Elaborazione dell'Informazione, Pisa, Italy. In late 2000 he joined the Computer Science Department of the University of Pisa as Assistant Professor, and since 2001 he his also Research Associate with the Istituto di Scienza e Tecnologie dell'Informazione in Pisa. His current research interests are in the areas of wireless sensor networks and streaming authentication.

Alberto Gotta received the Master of Science in Telecommunication Networks and Systems Engineering from the University of Genoa, Italy, in 2002. From 2002 to 2004 he was a researcher for the Research Unit of the Italian National Consortium for Telecommunications (CNIT) at the University of Genoa. In 2004 he was a researcher with the ISTI Institute of the Italian National Research Council (CNR). Since 2005 he has been a Ph.D. student in Information Science and Communication Technologies at the University of Genoa. His current research interests are in bandwidth allocation, admission control, and fade countermeasure systems in terrestrial wireless and satellite multi-service networks.

Y. Fun Hu is Professor of Wireless Communications Engineering in the School of Engineering, Design and Technology (EDT) at the University of Bradford, where she leads the Mobile Networks and Applications Group of the Mobile and Satellite Communications Research Centre (MSCRC). Since 1992, Prof. Hu has been actively involved in the development of 3G mobile communications systems through participation in numerous high profile EU framework programme projects. Her research in network protocol design, mobility management, radio resource management and network security has attracted a continuous stream of funding support from the EC, ESA, EPSRC, DTI and industry since taking up her first academic post in 1994. She was a UK delegate of the EU COST 253 (1996–2000), COST 256 (1997–2001) and COST 272 Actions (2001–2005). She was an executive member of the IET Electronics and Communications Divisions Professional Network Group on Satellite Systems and Applications (2000-2002) and has been a member of the Technical Advisory Panel of the same Professional Network Group since 2002. She has co-authored one book and editor of two books. She is a member of the Editorial Board of International Journal of Services and Standards, InderScience. She is the co-ordinator of the MSc taught programme in 'Wireless Sensor Networks', which started in September 2005. This course is the first of its kind in the UK. She currently leads a small team to conduct research into the application of wireless sensor technologies for water level management with funding support from Yorkshire Water through the Knowledge Transfer Partnership. She also leads Bradford's contributions towards the Networking Group in the EU funded project SatNEx – Satellite Network of Excellence as well as leading a research team towards contributions to the Inmarsat BGAN system development.