MANET and WSN: Are they alike?

Javier Gomez
Electrical Engineering Department
National University of Mexico
Edificio Valdez Vallejo, 3er piso
Circuito interior, Ciudad Universitaria,
Coyoacan, C.P. 04510, D. F. Mexico
Tel: +52 55 56223058
E-mail: javierg@fi-b.unam.mx

J. Antonio Garcia-Macias
CICESE
Km. 107 Carretera Tijuana - Ensenada
Código Postal 22860
Ensenada, B.C. Mexico
Tel: +52 646 1750500
E-mail: jagm@cicese.mx

Introduction

A mobile ad-hoc network (MANET) is a self-configuring network where nodes, connected by wireless links, can move freely and thus the topology of the network changes constantly. A great amount of resources has been devoted to research in the MANET field in the past three decades; many conferences have been held, many projects have been funded, many articles have been written; however very few MANET-type applications have emerged from all this hard work. A great body of knowledge about MANETs has been produced and many researchers in the field are now trying to apply this knowledge to the field of wireless sensor networks (WSN). The reasoning is that both MANETs and WSNs are auto-configurable networks of nodes connected by wireless links, where resources are scarce, and where traditional protocols and networking algorithms are inadequate. However, as we discuss in this chapter, great care should be taken before applying algorithms, protocols, and techniques to WSNs, if they were originally developed for MANETs. Although both types of networks indeed have many similarities, the differences are also such that WSN can arguably be considered a whole different research field.

Similarities

Probably the main reason why WSNs immediately resemble an ad hoc network is because both are distributed wireless networks (i.e., there is not a significant network infrastructure in place) and the fact that routing between two nodes may involve the use of intermediate relay nodes (also known as multihop routing). Besides, there is also the fact that both ad hoc and sensor nodes are usually battery-powered and therefore there is a big concern on minimizing power consumption. Both networks use a wireless channel placed in an unlicensed spectrum that is prone to interference by other radio technologies operating in the same frequency. Finally, self-management is necessary because of the distributed nature of both networks.

Wireless ad hoc networks were developed in the early 70´s with the US military as the main customer. Three decades later when commercial applications based on ad hoc technology are finally emerging, one wonder if there is any more work to do in this field or it is enough to simply leverage all these previous research. The answer to this question is that these commercial applications are quite different from traditional military applications and therefore they require a new fresh look to the problem [5]. Assumptions such as single purpose-application, cost unaware, large scale, and unique hardware/radio commonly given in military ad hoc networks can not be exported to emerging ad hoc nets such as disaster recovery, long-lived applications, peer-to-peer, WSNs, human context interaction, P2P etc.

Recently there is a re-emergence of ad hoc networks pushed by two confluent forces, on one hand there is a technology push resulting in smaller more powerful mobile devices, and on the other hand new types of ad hoc applications are emerging. Higher chip integration and hardware architectures optimized for low power operation tighten with new UWB and MIMO radios taking advance of wider unlicensed spectrum are creating new type of mobile devices with unseen capabilities. The key Internet paradigm that the network core should be kept simple (only care for the delivery of data packets) while the intelligence is at the edges, does not fit well in commercial ad hoc networks, which several people argue are quite different from traditional Internet or WLAN. New applications are changing the face of traditional ad hoc networks (i.e., pure routing) to networks where there is need for networking, processing and storage everywhere in the network.

What makes WSN different

Although there are important similarities between WSNs and MANETs, there are also fundamental differences. Some of these differences derive from the nature of both types of networks: MANETs are usually “close” to humans, in the sense that most nodes in the network are devices that are meant to be used by human beings (e.g., laptop computers, PDAs, mobile radio terminals, etc.); conversely, sensor networks do not focus on human interaction but instead focus on interaction with the environment. Indeed, nodes in a sensor network are usually embedded in the environment to sense some phenomenon and possibly actuate upon it; this is why some people say that WSNs can be considered as a “macroscope”. As a consequence, the number of nodes in sensor networks, as well as the density of deployment, can be orders of magnitude higher[1] than in ad hoc networks; this of course involves thinking about scalability issues.

If a network is going to be deployed in the outdoors, on the top of an active volcano, in the middle of the ocean, or in some other environment where sensor network applications typically take place, some nodes will eventually get damaged and fail. This means that the topology of the network may change dynamically, not due to node mobility like in ad hoc networks, but because some nodes will fail. In this case reconfiguration mechanisms will have to be used, so the network design should consider that nodes are prone to failure. It is worth saying that there are some applications where nodes are attached to

[1] The vision of seminal projects such as SmartDust [Kahn et al., 1999] contemplates networks with thousands or millions of nodes, although the largest deployment up to now had about 800 nodes.

animals, cars, or moving objects, but in the majority of applications nodes remain static, so some issues that are important in mobile networks may not be of great importance in wireless sensor networks. Besides failure, topology may also change due to the sleep-awake cycle observed in some protocols designed with sensor networks in mind. These protocols go through these cycles in order to achieve energy savings, which is one of the biggest concerns and design requirements in resource-scarce sensor networks. This scarcity of resources, again, constitutes a differentiating feature in sensor networks: nodes are typically left unattended for extended periods of time (i.e., months, years) and they are expected to operate on batteries; the range of communications is typically within a few meters and at low rates (some kilobits per second); there is typically a few kilobytes of memory and the processor may operate at speeds of only some megahertz.

It should also be pointed out that the service offered by wireless sensor networks is not simply to move bits from one place to another, but instead to provide answers. These answers should respond to questions such as: what are the regions where the temperature is above the specified threshold? What is the path followed by the herd? Thus, responding to these types of questions implies taking into account geographic scopes, which is a requirement that is not needed in most other networks. Indeed, in some applications the ID (e.g., the address) of individual nodes is irrelevant and location becomes a more important attribute. In general, communication paradigms are affected by the application-specific nature of sensor networks, and we will discuss this point in more detail ahead.

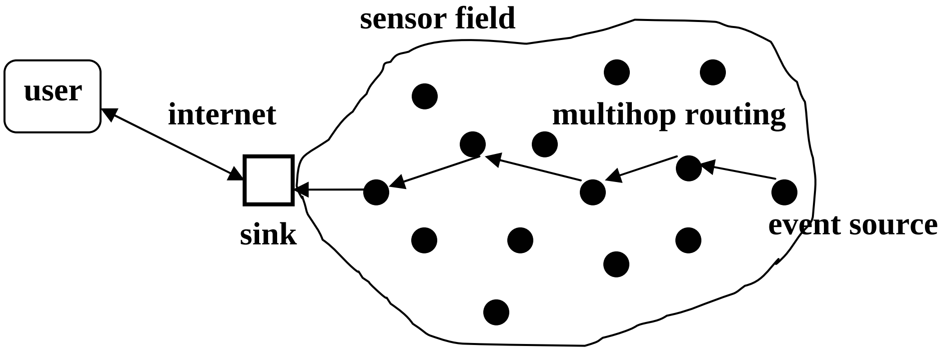


Figure 1, Typical functional architecture for a WSN

Protocol Stack

A protocol stack for WSN is shown in ***Figure 2*** for illustration purposes only. This stack is similar to the one used in MANET networks (which is also the same stack used in TCP/IP networks) except for the addition of a power, mobility and task management planes that operate across all layers. While this layered approach has remained accepted and mostly untouched in MANET networks for a quite long time[2], most researchers find serious difficulties to adhere to it in WSN. The main arguments for this opposition are that WSN is very application-specific and resource-constrained, so a layered architecture may not be the best way to approach the wide range of applications and optimize the limited resources. In fact, a cross-layer view of WSN is becoming more and more accepted in the research community.

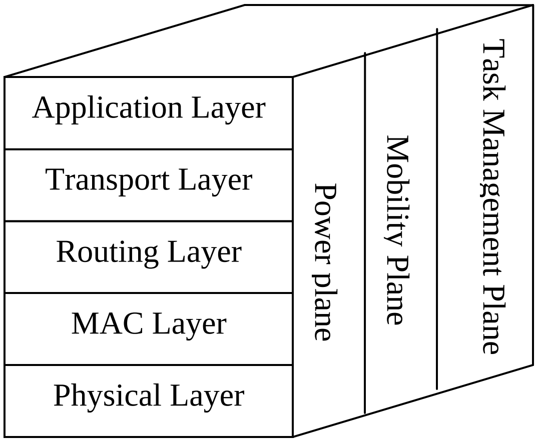


Figure 2, Protocol stack for WSN

The protocol stack shown in Figure 2 consists of the *physical layer, MAC layer, routing layer, transport layer and application layer*. The physical layer addresses the lower-level operations of the radio interface for a robust transmission and reception of packets in a harsh wireless environment. These operations include the frequency selection; transmit power, modulation, signal detection and coding. The MAC layer is responsible for proper channel access among competing transmitters. The MAC should avoid collisions as much as possible and turn-off the radio whenever a sensor is not actively transmitting or receiving packets in order to save energy. The routing layer is responsible for node addressing and routing in a network that is commonly multihop. Terms such as unicast and multicast common in MANETs, are hardly applicable in WSN where we find other forms of routing such as one-to-many, many-to-one, many-to-many, etc. Proper delivery of packets is addressed by the transport layer. A provision for congestion control either within the transport layer or as a separate module should be included in order to reduce the probability of network overflow. Finally, the application layer creates an abstraction of the main classes of applications found in WSN. General-purpose software associated with a given class can be reused in various applications, thus reducing prototyping time.

Across all layers in Figure 2 we find the power, mobility and task management planes. The power plane emphasizes the power-awareness that should be included in each layer and across all layers in WSN. For example, a sensor may keep its radio on after sensing

[2] Few authors have proposed a cross-layered optimization in MANET; however, these initiatives appear to be isolated proposals.

some activity in the channel, or it may turn it off if it is not generating any data and it does not belong to any active route. A sensor that is running low in energy may turn-off its radio and save its energy for sensing activities only. The mobility plane is responsible for maintaining the full operation of the sensor network even in the event of sensor mobility. While most sensing applications we can think of are static, we can not discard that sooner or later mobile sensing applications will emerge. This could be the case when sensors are mounted on mobile platforms such as robots, persons, animals, cars, etc. Routes used to carry information across the network have a limited lifetime and need to be periodically repaired because of node mobility. Even without mobility, routes may change due to the fact that nodes run out of power or follow an awake/sleep duty cycle, so a route that is valid at some point in time may no longer be valid a little later [12]. In both cases, the routing layer is mainly responsible for route maintenance. The task management plane should be capable of coordinating all nodes toward a common objective in a power-aware manner. Some sensors in a given region, for example, may be temporarily turn-off if there is enough sensing-redundancy from other sensors in that region.

Below we present a detailed description of each layer. It should be noted that this layered-protocol approach is just a reference model commonly used in the literature, so we are using it for presentation purposes in this chapter only. However, very active attention has been paid in the research community to cross-layer approaches where layers and their functions are not as strictly defined.

In a traditional layering approach, layers in a protocol stack provide services and interact only with contiguous layers through well-defined interfaces. Thus, there is a clear separation of functions and strict boundaries are imposed between layers. Diverging from this traditional layering approach, the cross-layer approach is more flexible and allows a more intensive feedback between layers [35]. For instance, using a cross-layer design, adaptive modulation and coding at the physical layer can be designed considering the radio link level error control technique (e.g., ARQ) to maximize network capacity under constrained QoS requirements. Cross-layer techniques can also be developed at the application layer for wireless multimedia services which can exploit physical and radio link layer information, thus performing adaptations according to varying conditions in the lower layers. These tight interactions between different layers are very beneficial in wireless networks, but the benefits are exacerbated in MANETs and in WSNs; indeed, in these types of wireless networks resources are more scarce (power, bandwidth, etc.) and should be managed very efficiently.

Physical layer

The physical layer, as seen in traditional layered-network architectures, is responsible for the lower-level operations of the radio interface including frequency selection, transmit power, modulation, signal detection and coding. Signal detection and coding are strongly related to hardware capabilities like processor speed or memory size. For this reason we focus on the frequency selection, transmit power and modulation issues below.

Frequencies used in today's sensor networks include the 915 MHz and recently the higher 2.4 GHz of the industrial, scientific and medical (ISM) band, although the 310 MHz and 433 MHz bands can also be found. Lower frequencies should be preferred because of the higher signal attenuation experienced by higher frequencies. Unfortunately, the limited bandwidth spectrum available in the lower frequency bands is pushing sensor networks to higher frequencies where more bandwidth is available, allowing for higher transmission rates. Signal attenuation in wireless channels is also affected by terrain conditions. The attenuation experienced by a transmitted signal over a distance *d* is proportional to $d^{n,}$, with $2<n<4$. Ground-lying sensor networks are likely to observe attenuations with the exponent *n* closer to 4 (*n* is equal to 2 in free-space conditions only). Higher attenuation means a higher transmit power is required in order to guarantee a proper packet reception. This results in higher energy consumption in a power-scarce sensor network. All techniques available to reduce transmission power over the wireless channel should be used in order to save energy in sensor networks, this include spatial, frequency and time diversities.

The choice of a good modulation scheme is a key factor for the correct delivery of information among sensor nodes. Different modulation schemes may differ in various aspects including the number of bits per symbol, bit error rate (BER), power efficiency and spectrum efficiency among others. Complex modulation schemes such as M-ary are capable of transmitting several bits per symbol, but this is at the expense of a higher transmitted power and increased BER. Simpler binary modulation schemes like PSK or QPSK transmit fewer bits per symbol but require less power and are more robust against channel errors. In [22] it is shown that, considering the transmit power as the dominant factor in a sensor node, binary modulation schemes are more energy-efficient. In near future it is expected that sensors could implement some sort of adaptive modulation; allowing the radio to dynamically change the modulation that better match current channel conditions. Ultra wideband (UWB) has also been proposed for future sensor networks requiring high transmission rates. Only the baseband signal is transmitted in UWB (i.e., no carrier frequency is used) which makes it simpler to built and more resilient to attenuation and multipath effects.

Opposite to ad hoc networks where the IEEE 802.11 radio interface has become a de-facto standard for communications (and its underlying physical-layer settings), the choice of a physical layer in sensor networks can vary significantly among the different radio-hardware choices available in the market today. The new IEEE 802.15.4 standard is an effort to set a radio standard for general-purpose sensor deployments. This radio is aimed at low-power low–range communications devices that may allow for years of battery-life without replacement. This standard provides support for one-hop reliability and some basic QoS support. There are several good reasons to built sensor applications above a common radio-interface, the most important one being the possibility to recycle functionality (e.g., code, algorithms, etc.) among different applications, thus reducing deployment time and costs. A sensor radio interface produced in large quantities is also likely to be cheaper and more robust than a prototype radio. Recycling functionality is one of the main reasons behind the layered design of IP networks. Having a single common radio interface in sensor networks, although desirable, is not feasible in practice.

Different sensing applications may simply need a different type of radio: consider the extreme case of passive RFID tags operating without batteries. It is likely that in the future, as it is case today, there will be several radio interfaces available to fit the wide-range of sensing applications with the underlying differences found in the physical layer of each radio.

MAC

The MAC layer is responsible for ordered channel access among competing transmitters. As previously mentioned, the IEEE 802.11 has become a de-facto standard in most MANET network deployments. The IEEE 802.11 uses Channel Sense Multiple Access (CSMA) with collision avoidance (CA). In IEEE 802.11, the CSMA/CA protocol is also known as the Distributed Coordination Function (DCF). The main goals of DCF are achieving strong connectivity among nodes and transmission fairness.

The main components of CSMA/CA are listening, backoff and collision avoidance. The listening component let potential transmitters know if the channel is occupied by an ongoing transmission in order to avoid unwanted collisions. Upon detecting the channel occupied or after a collision, a node triggers an exponential backoff algorithm to re-schedule its transmission. This mechanism has the effect of time-spreading competing transmitters, thus reducing the probability of future collisions. The collision avoidance component reduces the impact of collisions created by hidden terminals.

There are several issues why it will be inappropriate to use DCF in WSNs. First DCF follows the always-ready paradigm of the Internet. To achieve this goal MANET nodes remain in a continuous awake mode in order to be ready for either transmission or reception of packets. This always-ready operation pays a high price on power consumption, inappropriate for a power-scarce sensor node. Second, the DCF function operates better when packet births are stochastically distributed in time. This assumption is opposite to the high data correlation found in WSN where periodic streams of sensed data may be common.

Given the good knowledge (and why not popularity also) of CSMA/CA in multihop (MANET) networks, a good deal of research has focused on modifying this protocol to suit WSN requirements, in particular the power consumption issue. In this line of thinking we find the different versions of SMAC [28]. The Energy Efficient MAC protocol for WSNs (SMAC) is based on a listen/sleep cycle specifically designed for WSN. In SMAC, a sensor node transmits SYNC packets carrying the node’s listen/sleep schedule so that other nodes know exactly when they can communicate with it. SMAC schedules communications without the need for a local or global synchronization entity. Because nodes in SMAC operate with a low duty cycle (i.e., sleeping periods are much longer than listening periods), energy consumption is reduced significantly. A node wanting to transmit a packet but knowing the intended receiver is currently in sleep mode must queue its data and wait for the next receiver’s listening period, resulting in a delay. This delay is particularly onerous in multihop networks such as WSN where a packet may travel through several intermediate sensors before reaching the intended receiver or sink node.

The adaptive MAC [29] allows a node to briefly wake up in the middle of a sleep period if future activity in the MAC is predicted to occur (e.g., after a NAV timeout). Cleary adaptive S-MAC reduces the delay compared with SMAC at the expense of a slight increase in energy consumption. DSMAC [30] doubles the duty cycle for faster data transmission based on the presence of queued data and the average one-hop latency. DMAC [31] adjusts listening periods according to the traffic load and performs optimization based on a data-gathering tree-structure. DMAC reduces the long delays observed in multihop routes compared with fixed listen/sleep based protocols. The authors in [32] use routing information to predict future activity in the channel in order to turn on/off the radio.

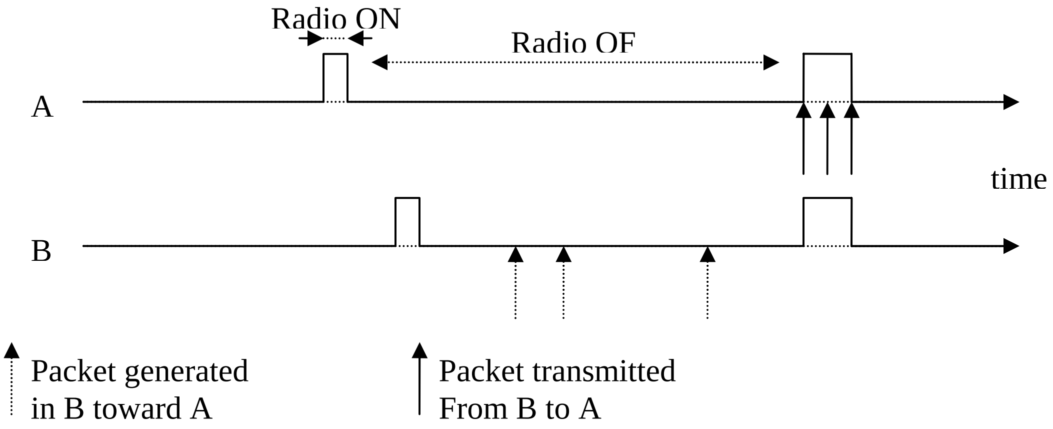


Figure 3, Example of the operation of SMAC. Node B generates three packets that are destined to node A. Node B, however, must wail until the next active interval of node A before transmitting those packets.

Although the previously discussed MAC protocols successfully address the power consumption issue of CSMA, they do not properly address the problem of the highly-correlated data found in WSN. A hybrid TDMA/FDMA protocol for WSN proposed in [33] is an attempt to address both issues simultaneously. This MAC assumes power-constrained sensor nodes can communicate directly with a nearby-located high-powered base station. TDMA is used to accommodate a single sensor in order to minimize delays whereas FDMA is used to guarantee a minimum bandwidth to each sensor. TDMA is not always preferred in this scheme because of the costs associated with synchronization.

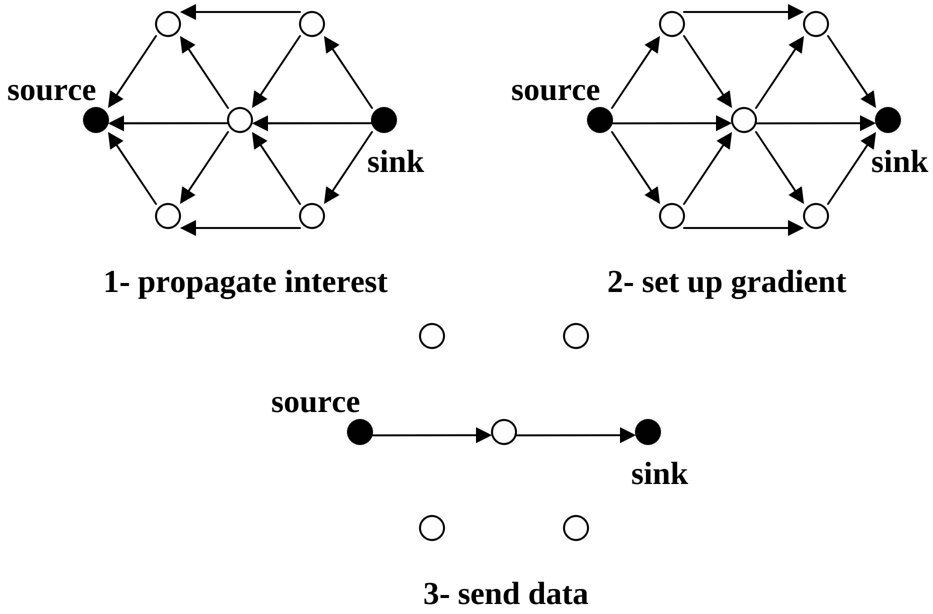
The very-low-power limited-range IEEE 802.15.4 MAC deserves a special note in this category. This standard was designed for several applications including home networking, automotive networks, industrial networks, interactive toys and remote sensing. Network topologies include star and peer-to-peer using the well-known CSMA/CA channel access protocol described before and operating in the 2.4 GHz ISM band. The standard allows for two types of devices. Full function devices (FFD) can become network coordinators and can talk to any other device. Reduced function devices (RFD) are limited to star topologies and cannot become a network coordinator. RFD devices have a very simple implementation and therefore can become extremely low cost. An optional super-frame structure with contention-access and contention-free periods allows for nodes requiring guaranteed bandwidth.

Routing

Any textbook on computer networking will tell you that the address of a node is a fundamental concept to understand routing. After all, since the beginning of computer networking history, routing mechanisms have relied on knowledge of the addresses of nodes in order to establish routes between them. For instance, the Internet routing mechanism relies on IP addresses and a hierarchical structure to establish routes and to route data between nodes. Under this view, the same name is used to identify individual nodes and also to identify communication endpoints; this coupling has resulted in problems when trying to achieve mobility of nodes in IP networks [24].

Mobile ad-hoc networks follow a traditional node-centric approach for routing, *i.e.*, routing relies on individual nodes and their corresponding addresses. Routing in ad-hoc networks has been classified as proactive, reactive, and hybrid, based on how the network reacts to route invalidation. With proactive routing, the network is under constant survey in order to know all possible routes between nodes at any given time; this means that routes are constantly being discovered, even if routes have not been invalidated. In contrast, reactive routing attempts to establish routes between nodes only when they are needed or when routes are no longer valid. The hybrid approach, as the name suggests, uses a mix of both proactive and reactive routing.

Due to their data-centric and application-specific nature, node-centric approaches do not constitute the best communication paradigm for sensor networks. Instead, data-centric communications are preferred [12], since it is more adequate for applications where the data read from sensors is important, and not the address of specific nodes. Indeed, a typical application may be interested in knowing the regions of a field where temperature is beyond a certain threshold; here what is important are the values of temperatures read by the sensor nodes, and communications are established according to this criteria, without communicating with specific nodes by their addresses. This way, a network-wide request is issued and only those nodes whose read values satisfy the criteria respond; then a data aggregation process takes place at various points along the path from the data sources to a data-gathering node commonly referred to as a sink. Along this process, the identity (*e.g.*, address) of the involved nodes is not important, as data is forwarded and aggregated from node to node according to its value. This type of many-to-one communications, where data is sent from different sources to a sink, is sometimes called gathercast. Directed diffusion [23] is a typical data-centric routing scheme, where following a data request a reverse tree rooted at the sink and with leaves at the data sources is set up. The tree is called a gradient tree, where the routing entries are the gradients and data matching these gradients is forwarded from sources. A gradient reinforcement process is then performed, where the best paths are kept while others simply time out and are removed.



1- propagate interest

2- set up gradient

3- send data

Figure 4, Basic operation of the direct diffusion data-centric routing protocol.

As previously stated, node mobility presents serious challenges in node-centric networks because communication endpoints and paths are tightly coupled to the names or identifiers of nodes. Being node-centric and also presenting high mobility of nodes, MANETs inherit these mobility problems. For instance, when one or more nodes move, paths involving these nodes are affected and a new path should be rediscovered by the routing mechanism (either proactively or reactively) generating important overhead. Mobility is not as important in WSNs, as typical applications do not involve moving nodes[3]. However, routes may change due to the fact that nodes follow an awake/sleep duty cycle, so a route that is valid at some point in time may no longer be valid a little later. The same consequence will be true due to the high rate of node failure that is expected in sensor networks, where not only hardware is cheap, but also is exposed to adverse conditions. Data-centric routing in sensor networks is not seriously affected by the on-off nature of individual nodes, as data is forwarded from the sources to the sink through any available nodes that match specific criteria.

It is also worth noting that, in both MANETs and WSNs, new routing schemes that take advantage of knowledge of the physical location of nodes are being used. Although many proposals have been made for position-based routing in MANETs [25][26] not all these are adequate for sensor networks, mainly due to restrictions in power and size of sensor

[3] Although it is true that there are applications where sensor nodes change their position over time, currently the most usual application involves sensing (and probably actuating) with fixed networks.

nodes. Ganesan et al. [27] identify at least a couple of benefits for position-based routing in wireless sensor networks:

- Sensor data is likely to be geographically correlated. Data reduction or aggregation schemes would need to route geographically to exploit such correlations.
- Queries that are geographically scoped are likely in many applications where users would prefer to query a small geographical region rather than the entire network. For instance, in a tracking application, the query is efficiently answered by querying only nodes on the trajectory of the target rather than all nodes in the network. Similarly, weather monitoring that is targeted at understanding local characteristics of data rather than global ones can be handled efficiently using geographically scoped queries.

Of course, as in the case of data-centric routing, the mobility of nodes is not a problem when using position-based routing; here it is only necessary to know the position of the endpoints and of any intermediate nodes, without the need to construct routing tables or perform routing updates.

Transport and congestion control

Transport protocols are yet another important area where MANET and WSN diverge significantly. MANET traditionally implement the full TCP/IP protocol stack, meaning MANET nodes will have IP addresses or something similar, support broadcast, unicast and multicast routing, and more important, be fully compatible with UDP/TCP transport protocols. Some researchers argue there is a need for native TCP support in WSN also. This way the WSN can be directly connected to an outside network without the need for special proxy servers or protocol converters. Bringing TCP/IP to wireless sensor networks is a difficult task, however. First, because of their limited physical size and low cost, sensors are severely constrained in terms of memory and processing power. Traditionally, these constraints have been considered too limiting for a sensor to be able to use the TCP/IP protocols. Second, the harsh communication conditions make TCP/IP perform poorly in terms of both throughput and energy efficiency. Sensor networks may exhibit higher packets losses (2% to 30%) compared to ad hoc networks. A good deal of research has been devoted to improving TCP performance in MANET in the past decade. Although the main perceived trend in the research community is not to consider the use of TCP in WSN due to the issues presented before, there are few researchers who argue there should be some TCP support in WSN.

The common trend in the research community is to consider WSN a totally different world from traditional TCP/IP networks, and therefore transport protocols should refrain from copying TCP ideas. Most sensor applications are event-driven and therefore do not need a reliable transport protocol such as TCP. They will be likely optimized for a particular task/operation and may tune their transport protocol to suit specific requirements.

Some researchers argue, however, that even if not related to TCP, there is some need for reliable transport in WSNs in the near term. They argue future WSN may become general purpose sensor platforms to some extend, requiring the ability to reprogram the functionality of the sensor network periodically. Reprogramming the sensor network necessarily requires a reliable transport protocol. An example of this way of thinking is the PSFQ protocol [8]. PFSQ (Pump Slowly and Fetch Quickly) recovers from losses locally and avoids using end-to-end ack messages. This results in minimum signaling involved for loss detection and recovery. When a packet is lost in PSFQ, the packet is retransmitted locally while copies of received packets with higher sequence number are buffered and transmitted only until successful retransmission of the lost packet occurs. Another protocol in this category is the Reliable Data Transport in Sensor Networks (RSMT) [10]. This protocol operated as a filter within the directed diffusion stack [10]. Reliability in RSMT refers to the delivery of all fragments of a large packet (called entity) to all the subscribing sinks in a WSN.

The ability of controlling the rate of transmitted packets (i.e., congestion control) to match the available bandwidth in the network has always been a primary source of concern in packet networks. Congestion control can be implemented end-to-end as a part of the transport protocol (e.g., TCP) or as a separate protocol. TCP implements congestion control by means of a sliding window that grows slowly when no packet losses are detected, and decrease fast when packet losses do occur. In this way TCP attempts to transmit information between end points as soon as possible (e.g., files, web pages, etc.) but without overloading the network. Opposite to TCP, UDP does not have any congestion control provisioning. Without congestion control, UDP packets can easily overload the MANET network, possibly disrupting other connections including TCP sessions. An example of a congestion control mechanism for MANET with both TCP and UDP traffic is presented in SWAN [1]. SWAN uses rate control for UDP and TCP packets and sender-based admission control for UDP real-time traffic. SWAN uses explicit congestion notification (ECN) to dynamically regulate admitted real-time traffic in the face of network congestion.

Event-driven WSN suffer from a different source of congestion, here and idle or lightly loaded sensor network may suddenly become active in response to a detected or monitored event. Transport of these events to the sink points may result in sudden congestion in the network depending on the sensing application. It is during this periods of activity in the network that the probability of congestion is greatest and the importance of the monitored information most significant.

An example of congestion control algorithm for WSN is CODA [9]. This protocol uses two complementary congestion control techniques. First there is an open loop hop-by-hop backpressure mechanism that signals nodes upstream (from the congested node toward the source) to reduce their pace of forwarded packets (e.g., drop packets). Second, there is a closed-loop multi source regulation to specifically tell sources to slow down their transmit rate. The ESRT [11] is an event-to-sink reliable transport protocol which also implements congestion control. In ESRT, any forwarding node experiencing buffer overflow sets the congestion flag on in each forwarded packet. Upon reception of packets with the congestion flag set, the sink node signals all sources to slow down their transmit rate by using a high power transmission.

QoS Issues

In the beginning of wireless packet networks in the 70s nobody care much about QoS as long as nodes in the network were connected to each other some how. Transmission of data packets was then the dominant type of traffic, and a best-effort delivery by the network was considered good enough. In the multimedia world we are immersed now, connectivity is not enough to guarantee that the different types of media will be properly delivered by the network. Applications such as VoIP, real-time video, etc. require tight bandwidth, jitter, delay and packet losses guarantees to work properly.

Providing QoS in ad hoc networks is quite complex and has become one big obstacle in the deployment of commercial ad hoc networks. This situation is mainly due to the poor end-to-end channel utilization found in current ad hoc networks based on IEEE 802.11 technology. For example, considering mobility and hidden terminals only, measured end-to-end channel utilization can get below 18% even for routes with few hops [6]. Forwarding over a common channel and packet header overhead can bring this utilization down to 1% even with RTS-CTS in place. The picture is even worst if we consider that this 1% is shared by data and control packets indistinctly. For routes with several hops and high node mobility the end-to-end channel utilization approaches zero.

The poor QoS performance shown by current ad hoc networks is making several researchers to rethink how ad hoc networks should be built. J. J Garcia-Luna-Aceves [6] identified some key factors that may need a fresh look from the community in order to improve QoS:

- ***Traditional packet switching.*** Current ad hoc networks do not make any distinction about how different types of packets are handled by the network. In order to guarantee bandwidth and delay constrains for real-time applications, it is necessary to distinguish the way packets are queued and forwarded by each node. Soft-state approaches and switching flows of packets rather than packets in isolation are promising approaches to be tested.

- ***End to end connectivity.*** The famous end-to-end Internet paradigm assumes that any pair of nodes can be connected by the network transparently. The presence of obstacles and network partitions make impossible to guarantee end-to-end connectivity in ad hoc networks. New directions in this area call for use of storage, processing and communication resources opportunistically in order to live with network disruption.

- ***Resource allocation.*** Current ad hoc networks use a common channel that is shared by all nodes. This approach results in high levels of interference and low channel utilization. New trends to improve performance in this area consider the use of several channels. This strategy has several advantages over the common-channel approach beyond having an increased network capacity. Separating which applications are allowed to use a given channel reduces interference and can provide some coarse QoS control. Similarly, a communication that failed in one channel can be attempted in a different channel providing a richer connectivity.

- ***One-to-one competitive communications***. Probably the main culprit of the poor QoS performance shown by ad hoc networks is the use of WLAN technology. The use of a common channel and CSMA channel access work well in a WLAN setting, but has several drawbacks when it is used in multihop ad hoc networks. New directions in this area call for new radios and new communications models. New types of radios that could exploit multi-user detection and equipped with directional antennas could improve performance in case only one channel is available [36]. Current communication models are based on competition-driven approaches that try to fight interference. Because in most cases a common channel is used, any transmission creates interference almost everywhere in the network, leading to scaling problems [38]. A better communication model could be, for example, the one proposed by Grossglauger [38] where information is delivered taking advantage of node mobility.

Although there has been some research on different aspects of QoS (mainly QoS routing) for MANETs, little has been done in the field of WSN. This may be due to the fact that sensor networks are very resource-constrained, thus providing not only any kind of service, but service with quality guarantees, poses an extremely complex problem.

But, the question arises: is there a need for QoS in WSNs? Clearly, traditional monitoring and control applications (*e.g.*, greenhouse temperature control) do not require strict observation of common QoS parameters such as bandwidth, delay and jitter. Real-time applications may not require bandwidth guarantees but certainly will need temporal guarantees, *i.e.*, delay and jitter. Some recent applications involve audio and/or video traffic, so bandwidth along with temporal guarantees may be needed; take for instance a recently presented application [39] where sensor nodes (called cyclops) equipped with a tiny camera provide a network for image sensing and interpretation.

Younis et al. [40] identify important design considerations for handling QoS traffic in wireless sensor networks:

- ***Bandwidth limitation***. Applications may generate both real-time and non real-time traffic, so using the limited available bandwidth to accommodate both may result difficult to say the least. The traditional approach of reserving bandwidth for QoS traffic is simply unacceptable in WSNs.

- ***Removal of redundancy***. Data fusion, data aggregation, and many other data-handling techniques are common in sensor networks; these take advantage of the fact that many applications generate considerable amounts of redundant data. However, these techniques can not be readily applied to QoS traffic (*e.g.*, audio, video) that require more complex manipulations, which in turn generate more processing overhead that would deplete energy supplies.

- ***Energy and delay trade-off***. Multihop transmission is one aspect that helps WSNs reduce energy consumption, at the cost of delaying the delivery of packets. An important element in this cumulative delay will be the time for queuing and classifying packets that handling QoS traffic will require. Thus, the energy and delay trade-off commonly present in WSNs is only exacerbated when QoS traffic is introduced.

- ***Buffer size limitation***. The buffers required for routing QoS traffic may suffer the same fate of other resources: scarcity. Not having adequate buffer sizes would complicate classification, introduce delays, and generally would reduce the possibility of granting QoS guarantees. The introduced delays would also have a negative impact on medium access scheduling.

- ***Support of multiple traffic types***. Currently emerging sensor network applications are increasingly complex as they involve not only monitoring temperature, light, and other similar parameters, but also transmitting audio/video, tracking objects, etc. Consequently, managing such diversity of traffic implies handling different data rates, different QoS constraints, and multiple data delivery models. This heterogeneity raises the challenges for providing QoS, as routing becomes more complex and more exhaustive processing is needed.

Although some proposals for QoS routing [41][42] and providing adequate MAC support for QoS [43] have been made, the issue of providing QoS in wireless sensor networks remains largely an open issue.

Application Issues

An oft-cited distinctive feature of wireless sensor networks is that they are very application-specific. The applications of these networks are becoming increasingly sophisticated, but some common classes of applications can be identified. The most common class involves communicating sensed data from the sensor field to a sink node; some other applications do some level of in-network processing, and more complex ones involve multiple kinds of distributed interaction and communication. Of course, all this diversity of applications poses a diverse set of requirements (*e.g.*, sensor field to sink vs. sensor to sensor communications, long-lived vs. ephemeral data streams, etc.) that have a great impact on the architecture, algorithms, and protocols of the network. For instance, a routing algorithm (or MAC or transport protocol, for that matter) that is suited for an environmental-monitoring application, where data is read at specific time intervals, may be almost useless for an intelligent road application where automobiles should be constantly informed of road conditions, presence of other vehicles, etc.

The address of a node is a fundamental element for communications in “traditional” networks. Indeed, at the core of IP-based network is the concept of the network address used to identify nodes and endpoint communication entities (MANETs are usually IP-based). Contrary to these node-centric networks, sensor networks are data-centric [12] since the identifier of individual nodes is not really as important as the data gathered by sets of nodes. As nodes in the network will frequently and randomly turn on and off, due to reduced duty cycles, it would be inefficient to base communications on constantly changing endpoint identifiers; instead, communications should be oriented toward the actual data gathered in certain regions of the network. The ultimate goal of most sensor networks is to answer to requests of the type "obtain the data that satisfies this (or these) condition(s)". In order to answer these queries, the identities of the nodes satisfying the given conditions is not known, and it does not really matter, so network wide discovery should be used in order to find the nodes that have the needed data. As an analogy, responses to queries in a database do not need to include the addresses of the records satisfying the queries, as only the actual data therein will suffice. In fact, the similarity with databases has generated much research efforts in the sensor network community.

Given that sensor networks are best designed in a data-centric manner, several research groups have explored a novel view of the sensor network as a database [13]. In most projects adopting this view, an SQL-like language is used for querying the network and sensor data is considered as a single table with one column per sensor type. In the TinyDB project [15], each sensor node has its own query processor, while other projects such as Cougar [14] perform query processing in a database front-end, leaving only some basic functions to the sensor nodes. Another point of concern is how to efficiently handle the flow of data from sources to sinks, taking into account that communication activities take a heavy toll on available energy. Observing that when a given phenomena occurs several sensors in a region will likely have similar or redundant data, techniques involving in-network filtering and processing have been proposed; data aggregation is one of the most widely used techniques, and the main idea is to combine the data coming from different sources enroute, thus eliminating redundancy, minimizing the number of transmissions and saving energy [16].

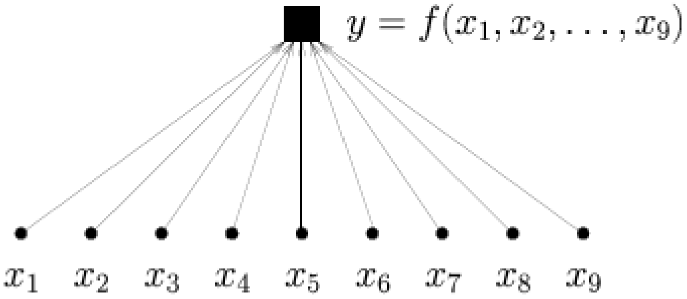


Figure 5, With data aggregation, the result (y) is determined by an aggregator node and a function (f) taking as inputs data sensed by a set of nodes (x_1,…,x_9).

As sensor networks grow, it becomes increasingly important to raise the level of abstraction for programmers. The sensor network as a database paradigm, although providing a good abstraction for some applications, has already shown important limitations; as some have pointed out, real-world data issues such as probabilistic availability of data, various levels of confidence in data, and missing or late data, can make this paradigm insufficient. Another approach in providing appropriate abstractions is to use a middleware, which is the software that resides between the applications and the underlying operating systems and networks. Middleware systems should provide reusable services that can be composed, configured, and deployed for the rapid creation of networked robust applications. Although distributed middleware (*e.g.*, CORBA, DCOM, etc.) have been in use for a long time, they are not suitable for WSNs, due to the fact that they demand a lot of memory, computational power, and other resources. Middlewares for sensor networks should be simple, easy implementable, and lightweight, and they also have to take into account the unique operating modes that make WSN different from traditional networks, including ad-hoc deployments, untethered operation and dynamic operating environments [17][1].

Besides providing higher-level abstractions, it is important to also provide programming mechanisms that scale to the foreseen size of future sensor networks. Current sensor networks are programmed node by node ("manually"), using low-level programming languages, interfacing directly to the network and the hardware via primitive operating system constructs; this is of course a cumbersome and error-prone method that can not scale to networks of hundreds, thousands, or even million of nodes. Over-the-air programming techniques, in which programs are sent to the nodes, have been proposed in order to solve the problem of programming large networks. Some of these techniques involve novel operating systems [18] where modules can be inserted or extracted dynamically, while others take an approach of having a virtual machine in every node to interpret code sent to them [19] [20].

Network Design

Traditional network design requires careful engineering to determine the right topology, conduct appropriate network dimensioning, test typical network performance, etc. When mobile ad-hoc networks appeared, network design requirements had to be revisited, as fixed topologies could not be assumed, network dimensioning could not be precisely performed due to the dynamic nature of the network itself, and additionally, new requirements had to be introduced including energy consumption considerations. This situation was exacerbated with the introduction of wireless sensor networks. Indeed, as Kömer and Mattern point out [34], there are several dimensions through the design space of WSNs that should be closely examined; some of them include:

- ***Deployment***. Typical applications for WSN mentioned in the literature include dropping sensor nodes off an airplane for military purposes and installing sensor nodes in fields for agricultural monitoring. Thus, the diversity of applications implies that some networks will have a pre-designed topology while others will have nodes randomly placed. Also, some networks will remain fixed once their nodes are in place, while others will change as nodes are added, removed, or replaced. All these factors have implications on the density of the network, the degree of network dynamics, the available links and routing hops, etc.

- ***Mobility***. Once sensor nodes are dropped off a plane, or deliberately placed in selected locations, the most common situation is that they remain in their place for the rest of their lifetime. However, some applications require placing these nodes on buoys at the sea, inside automobiles, or attached to some other moving entities; nodes may even have their own mobility means. Either way, mobility is a factor that should be considered in these cases, as it will affect the design of communication protocols and distributed algorithms.

- ***Node features***. In order for sensor networks to be practical, they have to be economical, operate unattended for long periods of time, and be sufficiently powerful. Achieving these goals involves sometimes-conflicting requirements, as making a node more powerful normally has an impact on the size, energy consumption and cost. There are currently a great variety of wireless sensor nodes, ranging from the millimetre-scale ones of the Smart Dust project to brick-sized nodes found in some environmental monitoring applications. Although the traditional approach is to have very homogeneous sensor networks, with the increasing diversity of node types these networks are also becoming increasingly more heterogeneous; as a result the networks are becoming more complex, including the software executed on them and the management of whole systems.

- ***Network size and coverage***. As sensor networks grow from current prototypes of tens or hundreds of nodes, to the envisioned ubiquitous networks of millions of nodes, so grows the scalability requirements of their algorithms and protocols. The geographic coverage, combined with the number of nodes determine the density of the network. High-density networks will obviously be more expensive than sparse ones, but may result in more accurate sensing and involve more sophisticated data-processing algorithms.

With these and other designs considerations [34], it should be clear that WSNs have very distinctive characteristics that imply particular considerations not commonly present in other types of networks. MANETs, although requiring similar design considerations (*e.g.*, energy-saving, mobility, etc), normally present less stringent requirements, as they are by definition spontaneous, short-lived networks with more powerful nodes.

Summary

In this chapter we have analyzed the similitudes, as well as the differences, between mobile ad-hoc networks (MANETs) and wireless sensor networks (WSNs). Even though we discuss many characteristics of both types of networks, this chapter by no means should be considered a comprehensive survey of neither of them, as the main focus of the discussion is on highlighting how they are (or not) alike. We have seen that although from certain points of view a WSN can be seen as a special type of MANET, the protocols, algorithms and design issues of the former cannot be applied to the latter unchanged. For the discussion, we have followed the well-known model of the protocol stack, analyzing each layer to see how aspects in each of them compare in both MANETs and WSNs; nonetheless, we have warned the reader that the layered protocol approach is considered by many researchers inappropriate for designing and implementing real-life wireless networks with limited resources.

Although wireless ad-hoc networks have been studied for nearly three decades, there are still opportunities for research. Some of these opportunities may derive from the fact that most previous research revolved around military applications and that the basic assumptions for these applications may not hold for non-military commercial applications. The research field of wireless sensor networks is full of research opportunities; some argue that it has just barely reached the end of its exploratory stage, where the foundations have been laid out. In practice, WSNs have been used for applications with limited scope or just as prototypes to provide proof of concepts; the vision of ubiquitous networks with thousands or millions or nodes has yet to be realized.

References

[1] G. S. Ahn, A. T. Campbell, A. Veres and L. H. Sun, "*SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks*", In Proceedings of IEEE INFOCOM 2002.

[2] Mario Gerla, “From battlefields to urban grids: New Research Challenges in ad hoc wireless networks”, Pervasive and Mobile Computing 1, (2005) 77-93.

[3] J.J. Garcia-Luna-Aceves, “Another Look at Ad hoc Wireless Networks”, keynote presentation, 4th International Conference on Ad hoc Networks and Wireless (AdhocNOW), Cancun, Mexico, October 6-8, 2005.

[4] J.M. Kahn, R.H. Katz, and K.S.J. Pister. “Next century challenges: Mobile support for Smart Dust”. Proc. ACM MOBICOM, Seattle, WA, August 1999.

[5] Mario Gerla, “From battlefields to urban grids: New Research Challenges in ad hoc wireless networks”, Pervasive and Mobile Computing 1, (2005) 77-93.

[6] J.J. Garcia-Luna-Aceves, “Another Look at Ad hoc Wireless Networks”, keynote presentation ADHOCNOW 2005, Cancun, Mexico.

[7] Chieh-Yi Wan, Andrew T. Campbell, Lashman Krishnamurthy, “PSFQ: a reliable transport protocol for wireless sensor networks”, in proceeding of 1st ACM international workshop on Wireless sensor networks and applications**,** Atlanta, Georgia, USA, 2002.

[8] Chieh-Yi Wan, Andrew T. Campbell, Lashman Krishnamurthy, “PSFQ: a reliable transport protocol for wireless sensor networks”, in proceeding of 1st ACM international workshop on Wireless sensor networks and applications**,** Atlanta, Georgia, USA, 2002.

[9] Chieh-Yih Wan, Shane B. Eisenman, Andrew T. Campbell, “CODA: Congestion Detection and Avoidance in Sensor Networks”, in proceedings of First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003), November 2003.

[10] Fred Stann and John Heidemann, “RMST: Reliable Transport in Sensor Networks”, in proceedings of [International Workshop on Sensor Net Protocols and Applications, Anchorage, Alaska, USA, May 2003.

[11] Y. Sankarasubraniam, O. Akan, and I. Akyildiz, “Event-to-sink reliable transport in wireless sensor networks”. In Proc. Of the 4th ACM Symposium on Mobile Ad hoc Netowrking & Computing (Mobihoc), Maryland, June, 2003.

[12] D. Niculescu. “Communications Paradigms for Sensor Networks”. IEEE Communications, march, 2005. Pp 116-122.

[13] R. Govindan, J. M. Hellerstein, W. Hong, S. Madden, M. Franklin, S. Shenker, “The Sensor Network as a Database”, USC Technical Report No. 02-771, September 2002.

[14] P. Bonnet, Jo. Gehrke, and P. Seshadri. “Towards sensor database systems”. In Proc. 2nd Int. Conf. on Mobile Data Management, Hong Kong, 2001. pp. 3-14.

[15] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. “The design of an acquisitional query processor for sensor networks”. In ACM SIGMOD 2003, June 2003. pp. 491-502.

[16] B. Krishnamachari, D. Estrin, and S.B.Wicker. "The Impact of Data Aggregation in Wireless Sensor Networks". In Proc. 22nd Int. Conf. on Distributed Computing Systems, July, 2002, Washington, DC. pp. 575-578.

[17] Y. Yu, B. Krishnamachari, and V.K. Prasanna, "Issues in Designing Middleware for Wireless Sensor Networks," IEEE Network Magazine, January 2004.

[18] C. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava. “A Dynamic Operating System for Sensor Nodes”. 3rd Int. Conf. on Mobile Systems, Applications, and Services (Mobisys). Seattle, WA, USA, June, 2005.

[19] P. Levis, and D. Culler. 2002. "Maté: A Tiny Virtual Machine for Sensor Networks". Tenth International Conference on Architectural Support for Programming Languages and Operating Systems. San Jose CA, USA, October 2002, pp. 85– 95.

[20] P. Levis, D. Gay, and D. Culler. "Active Sensor Networks". Second USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2005). Boston MA, EUA, May 2005.

[21] J. J. Garcia-Luna-Aceves, “Another view at Ad hoc Wireless Networks”, Keynote Speech, Ad hoc NOW, Cancun, Mexico.

[22] E. Shin *et al.*, “Physical Layer Driven Protocol and Algorithms Design for Energy-Efficient Wireless Sensor Networks,” Proc. ACM MobiCom 2001, Rome, Italy, July 2001, pp 272-286.

[23] C. Intanagonwiwat, R. Govindan and D. Estrin. "Directed diffusion: A scalable and robust communication paradigm for sensor networks". Proc. MOBICOM '00, August 2000, Boston, USA.

[24] P. Bhagwat, C. Perkins, S.K. Tripathi. "Network layer mobility: an architecture and survey". IEEE Personal Communication, vol 3(3), June 1996.

[25] S. Giordano, I. Stojmenovic, L. Blazevic, "Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy," Ad Hoc Wireless Networking, Kluwer, 2003.

[26] M. Mauve, J. Widmer, and H. Hartenstein. "A survey on position-based routing in mobile ad hoc networks". IEEE Network Magazine, 15(6):30--39, November 2001.

[27] D. Ganesan, A. Cerpa, Y. Yu, W. Ye, J. Zhao and D. Estrin. “Networking Issues in Sensor Networks”. Journal of Parallel and Distributed Computing (JPDC). Elsevier Publishers, December 2003.

[28] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks”, in IEEE INFOCOM, June 2002.

[29] W. Ye, J. Heidemann, D. Estrin, “Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks”, in IEEE/ACM Transactions on Networking, June 2004.

[30] P. Lin, C. Qiao, and X. Wang, “Medium Access Control With A Dynamic Duty Cycle For Sensor Networks," in IEEE WCNC, Mar 2004.

[31] G. Lu, B. Krishnamachari, C.S. Raghavendra, “An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks", in IEEE IPDPS, April 2004.

[32] M. L. Sichitiu, “Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks," in IEEE INFOCOM, Mar. 2004.

[33] E. Shih *et al.*, “Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks,” Proc. ACM MobiCom, Rome, Italy, July 2001.

[34] K. Römer, F. Mattern. “The Design Space of Wireless Sensor Networks”. IEEE Wireless Communications, Vol. 11 No. 6, pp. 54-61, Dec 2004.

[35] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks," IEEE Commun. Mag., vol. 41, no. 10, pp. 74--80, October 2003.

[36] L. Bao and J.J.Garcia-Luna-Aceves. _”Transmission scheduling in ad hoc networks with directional antennas”. In ACM MobiCom'02, September 2002.

[37] Gupta and P. R. Kumar, "The Capacity of Wireless Networks" IEEE Trans. Info. Theory, vol. 46, no. 2, Mar. 2000, pp. 388--404.

[38] M. Glossglauser, D. Tse. “Mobility Increases the Capacity of Ad hoc Wireless Networks”, INFOCOM 2001. Proceedings. IEEE, Volume: 3, 2001: 1360 –1369.

[39] M. Rahimi, R. Baer, O. Iroezi, J. Garcia, J. Warrior, D. Estrin and M. Srivastava, “Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks,” ACM Conference on Embedded Networked Sensor Systems (SenSys), San Diego, USA, 2005.

[40] M. Younis, K. Akayya, M. Eltowiessy and A. Wadaa, “On Handling QoS Traffic in Wireless Sensor Networks” Proc. Intl. Conf. on System Sciences (HICSS-37), Big Island, Hawaii, January 2004.

[41] K. Akkaya and M. Younis, “Energy-aware Routing of Delay-constrained Data in Wireless Sensor Networks,” Journal of Communication Systems, Vol. 17, pp. 663-687, 2004.

[42] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A stateless protocol for real-time communication in sensor networks," in Proc. Intl. Conf. Distributed Computing Systems, Providence, RI, May 2003.

[43] C. Lu, B.M. Blum, T.F. Abdelzaher, J.A. Stankovic, and T. He, "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks" IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2002), San Jose, CA, September 2002.