Data Management in Wireless Sensor Networks

Iacopo Carreras, Francesco De Pellegrini, Csaba Kiraly, and Imrich Chlamtac [1]
CREATE-NET, *Via Solteri 38/A, 38100 Trento, Italy*

Abstract. Wireless Sensor Networks are emerging as one of the most promising research directions, due to the possibility of sensing the physical world with a granularity unimaginable before. In this chapter, we address some of the major challenges related to the collection and elaboration of data sourcing from such networks of distributed devices. In particular, we describe the concepts of data storage, data retrieval and data processing. We discuss how such data management techniques will be able to sustain a novel class of data-intensive applications, which use the network as an interface to the physical world. Then, we identify some threats to the deployment of such networks on a large scale basis. In particular, we argue that even if appealing, the underlying composition of very large and heterogeneous wireless sensor networks poses enormous engineering challenges, calling for innovative design paradigms. Finally, we discuss an alternative solution, and the related data management mechanism, for the provisioning of sensor-based services in future pervasive environments.

Keywords. Wireless Sensor Network, scalability, data aggregation, distributed data storage

1. Introduction

Sensors have been applied in systems for more than a century, with the aim of providing feedback to human operators and implementing control feedback loops. The common notion of a sensor network, however, arose quite recently, when conveying and processing the information delivered by spatially distributed sensors became feasible. A sensor network is generally composed by an ensemble of network nodes with computation, sensing and communication capabilities. These nodes are distributed over a region to be monitored and are capable of sensing the environmental conditions and conveying information to the final end-user.

One of the earlier applications of sensor networks, where monitoring was obviously of strategical importance, was location detection and tracking. In fact, early sensor networks were developed for military reasons and one of the first example of a sensor network was the SOSUS (Sound Surveillance System) [1]. The SOSUS, in particular, was developed during the early 50's in order to detect and localize enemy submarines.

Later on, in the 70's, under the Distributed Sensor Networks (DSN) DARPA project, the research on sensor networks started focusing on a network composed of a large num-

[1]This work was partially funded by the EU under the BIONETS project EU-IST-FETSAC-FP6-027748

ber of low-cost sensor nodes. At that time, most studies were concerned with sensing technology, communication issues, processing techniques and how to compose a distributed software architecture.

Anyway, it was certainly during the last decades that major advances were achieved in all these areas. Sensing (and actuating) technology benefited largely from the advent of MEMS (Micro Electro-Mechanical Systems) technologies. In particular, low-cost and low-energy wireless communication protocols have been developed, and Wireless Sensors Networks (WSNs) [2,3] are emerging as one of the most promising research directions for the next decades. As a consequence, recent protocols have been standardized in the IEEE family in order to match the requirements of WSNs. Examples are the IEEE 802.15.1, which embodies mostly the Bluetooth standard, and the IEEE 802.15.4, which implements part of the communication stack of the popular ZigBee sensor technology. RFID and active RFID are also technologies discussed in research papers and widely deployed in industrial applications.

It is worth mentioning, though, that besides RF technology, research has been carried out on alternate propagation techniques, suitable for specific media and application scenarios. It is the case of acoustic waves, used for underwater communications [4], and infrared, employed for short range communications [5], or even magnetic induction or capacitive coupling for proximity communications.

Meanwhile, several hardware/software solutions appeared in mass production, the most popular being the Mote open-source hardware and software platform developed at U.C. Berkeley. Nowadays, several enhanced versions of the Mote exist, the de-facto worldwide standard being the Crossbow MICA Mote [6], which can be found in almost any WSN lab. Other HW platforms try to compete with the MICAs in size, price and performance. The processing capabilities can range from simple 8-bit microcontrollers to general purpose 400 MHz processors. Platform-independent operating systems and middleware softwares are also being developed to support energy-aware sensing, communication and processing capabilities.

Besides all these technical efforts and all the papers showing the widespread diffusion of sensor networks, expected to change the way we interact with our environment with major advantages on our everyday life, such forecasts are far from reality. In fact, the cost of sensor nodes is still far from the estimated price of some cents per unit which would permit massive deployment of sensors. Available solutions, in practice, are mostly custom made sensor devices, built with multiple sensing elements, long lasting battery and proper casing. In the end, including deployment into the figure, their cost is in the order of tenths of dollars. In the last part of this chapter we will present recent results in the WSN research field aimed at solve the cost issue and pave the way for massive sensor deployment.

1.1. Typology of sensor networks

Many types of sensor networks exist. In some cases wired energy supply and/or communication is not feasible, and this forces the adoption of wireless sensor networks (WSNs). Other settings require nodes to be small, cheap or densely deployed. In general, we find convenient to define what sensor networks are by characterizing them based on the type of nodes, the network, the communication and processing techniques applied [7,8]:

a) *Node constraints*. Nodes of a sensor network differ in size, energy and cost constraints. The *size* of sensor nodes ranges from huge radar stations to coin sized sensor nodes and cubic millimeter sensor nodes are under development [9,10]. Size, as well as *cost* and *deployment*, pose constraints to the sensor processing capability, onboard memory size and available energy sources. Energy sources, for example, can range from batteries through solar panels to mains supply.
b) *Network architecture*. The *number of nodes* in a sensor network varies from dozens to thousands, or theoretically even to millions [9] of devices. Sensor networks differ also in their *spatial coverage* and *node density* since some of them should be aimed at monitoring very large areas (e.g. woods or oceans), whereas others could be spread on the surface of a mug [10]. A sensor network can be *homogeneous*, consisting of identical nodes, or not homogeneous, where sensors differ in their physical sensing devices or in their computation/communication capabilities.
Sensor networks need also to be *deployed* and organized into a network: self-organizing structures for randomly deployed nodes or a planned deployment are two possible options; *mobility* of devices has also to be accounted for.
c) *Communications*. The range of foreseen applications pose different requirements on the way the communication stack is implemented. Some applications require simple event notifications with very low *bandwidth*, while others transmit continuous streams of measurement data (e.g. video streams). *QoS* constraints are also different, ranging from high delay best-effort to delay-constrained services (e.g. high speed control for industrial plants). Different physical *media*, including RF, optical/infrared, and acoustic waves are possible.
d) *Processing architecture*. Sensor networks can implement a centralized or a distributed *processing architecture*, depending on node capabilities, processing, communications and energy constraints and the targeted applications. *Processing capabilities* of nodes can range from dumb ADCs through image processing DSPs to general purpose processors.

Due to the broad variability of sensor networks and the various application domains, it is not surprising that sensor networks are custom made, specialized solutions, even if some standard components are already available. In what follows, we will focus on how data are managed in one of the most promising field, namely Wireless Sensors Networks (WSNs). In particular, in the next section we describe existing data management solutions for WSNs. In a further section, we investigate on a WSN scenario where sensor nodes can be deployed on a very large scale and we describe what the challenges are for data management in such an environment. In the end, we describe possible solutions and research directions.

2. Wireless Sensor Networks

As we mentioned before, the advances in embedded systems, sensors and miniaturized radios, made wireless sensors networks (WSNs) emerge as one of the most promising research directions for the next decades [2,3]. With a WSN we mean a large number of cheap, self-organizing network nodes able to autonomously perform sensing operations, to communicate and/or store data. A WSN is typically *immerse* in the phenomenon to be observed and, after a start-up phase, it is able to convey the information generated from sensors to a remote end-user. For several applications, moreover, we expect that WSNs are deployed in hostile environments. Other types of applications will require that sensors are embedded into structures where maintenance is simply not feasible. The general assumption is then that, someday, the cost of nodes will be low enough that they can be simply discarded rather than recharged. Clearly, prolonging network lifetime for these nodes is a critical issue, and it has to be taken into account in the design of each part of a WSN.

The intimate connection with its immediate physical environment allows each sensor to provide localized measurements and detailed information that is hard to obtain through traditional instrumentation, thus paving the way for new classes of applications, which are currently raising the interest of the research community.

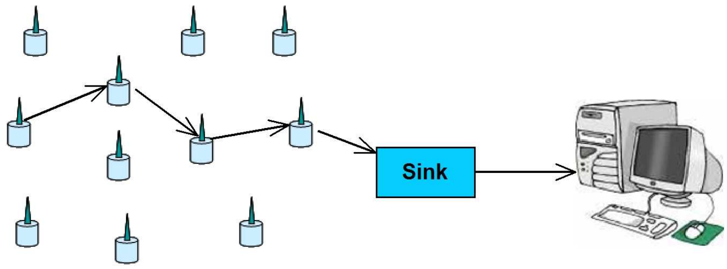

Figure 1. Wireless Sensor Network architecture.

2.1. Data Management in Wireless Sensor Networks

A typical WSN architecture is the one shown in Fig. 1, where densely deployed sensor nodes monitor some phenomenon trough their sensing capabilities and, once useful information is available, data is sent back to the sink node, which acts as the gateway between the WSN and a standard communication network such as the Internet. The *usefulness* of the information can not be determined a-priori, since it depends on the kind of application (e.g., tracking, monitoring, alerting, etc.), the physical phenomenon to be observed (e.g., the redundancy of the phenomenon) and the expected lifetime of the sensor networks. As two extreme cases, we can think of an application reporting the temperature in a remote region, and a fire alarm application, which detects the fire in a public office. In both application scenarios, the temperature is the phenomenon to be measured. However, in the first case the end-user is mostly interested in collecting a large number of sensor readings for a long period of time. This first case is characterized by large volumes of data collected for an extremely long period of time, but without the need for an extreme accuracy on the collected information (missing data can be eventually extrapolated). Conversely, the fire alarm application will be in a stand-by state for most of the time, but it needs to be extremely reliable and accurate when fire is detected. All these considerations are reflected in the way a WSN is programmed and the data circulating in the sensor network are processed and managed. Data management in WSNs deals with the challenging task of defining how sensor-originated data are efficiently managed, stored and conveyed outside a WSN.

Existing sensor networks assume that the sensors are preprogrammed and send data to a central front end where the data is aggregated and stored for offline querying and analysis. This approach has two major drawbacks. First, the user cannot change the behavior of the system dynamically. Second, communication in today's networks is orders of magnitude more expensive than local computation; thus in-network storage and processing can vastly reduce resource usage and extend the lifetime of a sensor network.

2.2. *WSN as a Distributed Database*

Typically, a WSN is *immerse* around the phenomenon to be monitored. The expected mode of usage of a WSN is that users program the sensors through *queries* [11,12,13], using a query-like declarative specification (such as SQL) of the information to be collected. This is an efficient way of abstracting the system details, and relies on a user-friendly interface for programming the sensors. As an example, in Alg. 1 the remote user is querying the WSN for light and temperature information from those sensors measuring a temperature value above a given threshold (10 in this specific case). The queried information is collected with a sampling period of 2 seconds.
A *query layer* [14] is in charge of translating the declarative language of the query into execution plans, taking into account the different constraints deriving from the specific application and sensor networks deployment.

Algorithm 1 Example of a query injected in a Wireless Sensor Network

```
SELECT nodes id, light, temp
FROM sensors
WHERE temp > 10
SAMPLE PERIOD 2 s
```

The main differences between a WSN and a traditional database system can be listed as follows:

- delivers a stream of data: the sensor networks, when receiving the query, answers sending data at constant predefined time intervals;
- communication errors: the data generated by the sensor nodes are delivered back to the sink through a multi-hop communication, where the communication links may be extremely unreliable and affected by errors. This means that data is reaching the sink with an extremely variable delay and reliability;
- real-time processing: since the energy spent in processing operations is several orders of magnitude lower then the one spent for communicating, it is usually preferable to process the information in real-time, in order to avoid unnecessary transmissions. This will be further explained in the following.

There are several implementations of query processors, which translate the SQL-like syntax into system operations. TinyDB [15] is a query processor implementation running on top the TinyOS [16] operating system. By taking advantage of a user-friendly interface, remote users are able to easily query the WSN using the appropriate SQL syntax. The COUGAR sensor database [13] is another implementation of a query processing layer. There COUGAR platform implementations for Mica Mote and Sensorial demos.

2.3. Distributed storage in WSN

Storage is an extremely limited resource in WSNs, since data generated from sensors can easily saturate the memory available on sensor nodes. In general, the data generated from a sensor network can be stored in a centralized or distributed fashion. When applying a centralized paradigm, all the information generated from sensors are collected from the remote end-user and stored in a centralized repository with unlimited power and storage resources [11]. This highly facilitates the querying of information, but strongly impacts the energy consumption related to conveying every single sensor reading outside the WSN. As a consequence, a centralized storage approach is considered to be appropriate only for low-data rate application scenarios.
An alternative approach consists in storing the information inside the network, according to a distributed paradigm [17,18]. This differs from traditional distributed storage systems because of the stringent energy and storage limitations imposed by the sensor nodes, and because of the spatial/temporal dependence of the stored data that allows the use of compression techniques. even in this case, the application scenario defines the constraints of the mechanism to be applied. There is a vast class of applications where data is first collected and then analyzed offline. Hence, data is generated continuously, but read only once. As an example, we can think of a military application where sensors are scattered in the battlefield and read from airplanes flying over the sensor networks [19]. Another example can be found in [20], where a sensor networks for wildlife tracking was deployed in order to monitor the migration of zebras. In this application, sensors were attached to animals and scientists collected information as zebras were coming in range with them. Information was therefore stored on zebras's sensor node until a new data collection point (sink) was reached.
In these application scenarios, distributed data storage defines how data is collaboratively stored, queried and managed in order to meet the sensor nodes memory limitations and the application requirements. The fundamental questions to be answered are in this case are: how information is *queried/searched* and how information is *stored/managed*.
There are different approaches to the search problem. One possibility is to store all the information locally on each sensor node. The injected query is then propagated until the interested nodes are reached. The nodes start to send back the requested information. The most popular protocol implementing the described paradigm is Directed Diffusion [21], where a sensing task is disseminated in the sensor networks from the sink in the form of an interest for *named data*. Data is named using attribute-value pairs. A path is then set up to the sensor nodes matching the interest, and the detected events matching the interest start to flow toward the sink following the established path. The drawback of this approach is that it could be extremely expensive to forward the interest to all he nodes of the network in the case of large deployments. Moreover, the limited capacity of the memory of sensor nodes is not explicitly addressed.
Distributed indexing addresses the problems highlighted before utilizing structured replication and distributed search techniques. The fundamental idea is to group the events together and map them to a precise spatial location. A node detecting an event stores this event to the mirror nearest to its location. This allows the construction of hierarchical search trees, thus reducing the cost for reaching the nodes with useful information. Data-centric storage (DCS) [22] is an example of the described indexing schemes, where a hash function is defined for mapping a detected event to a geographical location. As op-

posed to traditional database systems where latency is the optimization criteria adopted in most of the indexing schemes, in DCS the indexing of data is targeting the minimization of the communication cost required for extracting data from the sensor network.
Through the described techniques information is *searched* more efficiently, but problems still subsist in the case of long-term storage. In this case, in order not to exceed the storage capabilities of sensor nodes, a mechanism is needed that defines the lifetime of the gathered information. A typical approach consists in varying the resolution of the stored information, depending on its age. The most recent information is stored with the maximum resolution, while of older information only summaries are kept. In [17], the use of wavelets is explored for creating information summaries with a different resolution. The fidelity of the information is sacrificed for guaranteeing a long-term storage.

2.4. In-network Processing

In-network processing [23,24], often referred as data aggregation, is one of the most common approaches to reduce the communication overhead. It is well understood that, for short-range communications, local computation is much cheaper than radio communications. The cost of 1 single bit over 100 m. costs as much as 3000 instructions [25]. Starting from this consideration, in-network aggregation tries to maximally exploit the correlations in data in order to minimize the size of data and, correspondingly the communication cost. Correlation can be expected along multiple axes: spatial, temporal, among different sensors. Besides the possibility of reducing the volume of bits transmitted, in a vast class of application scenarios the end-user is not interested in the complete historical data of the sensor network, but rather in detecting some specific events (e.g., the trigger of an alarm) or in a "condensed" view of the observed phenomenon (e.g, the maximum temperature in the monitored region).
In-network processing assumes store-and forward processing of messages, where a message is a meaningful unit of data that a node can process. On each node of the sensor network, an in-network processing layer is in charge of handling the incoming messages, process them and decide the next messages to be sent.
In [26], a generic aggregation service for a network of TinyOS sensor nodes is defined, namely the Tiny Aggregation Service (TAG). The service is based on a declarative interface, similar to the one described in Sec. 2.2, that allows the distribution of the operators among the nodes of the network in order to minimize the power consumption. TAG operates over the data as it flows through the sensor nodes, applying aggregation operators and combining different readings into compact ones, where possible.
Aggregation operators are typically supported by query layers, where a declarative approach is adopted for describing the data to be retrieved.

2.5. Discussion

The techniques introduced before have the potential to widen the services applicable on WSN networks. In fact data management over WSN will be critical to enable user-situated applications with direct interface to the surrounding environment. It is easily understood that such services and their related applications will be far from those of existing networks. For example, even simple applications as navigation systems might change compared to their current implementation. Currently, navigation systems use the Global Positioning System GPS to support navigation applications: the position of the GPS device is determined and then localized on a map. In the customary solution, the road-map data collection is fixed and does not change over time and route decisions are taken independent of the environmental conditions (time, date, presence of traffic jams, accidents, deviations etc.): using a WSN with the specific task of measuring traffic conditions on the surrounding area novel metrics such as delay or "crowdness" of a path could be introduced in the search for the optimal path.

Along this line, a novel class of applications and services are arising soon, with a major impact on the way we conceive today any technology-aided operation. But, we identify a fundamental catch: such applications will require a massive deployment of sensors, which is not a simple technical task. In fact, either deployed on large areas, or concentrated in a small area with a high density of sensors per square meter, such networks configure naturally as *large scale* networks. This is especially true for certain types of applications, such as location detection [27], where increasing the sensor density means finer resolution and also more robustness against devices failures. It is likely that the concept of large-scale network, which so far have been confined to a purely theoretical exercise, will become of paramount practical relevance. Thus, the common paradigm of a WSN would be a distributed communication/computing environment characterized by an extremely large number of devices [28,9][1]. Anyway, the more fine-grid data are required by applications to such WSN networks, the larger the number of data sources, i.e. sensors, and this means a potentially tremendous increase of the number of data flows. The injection of huge quantity of environmental data is then bound to raise strong scalability issues for the underlying networking infrastructure. Further, these scenarios will be populated by very different devices, ranging from small embedded sensors, TAGs and RFIDs to complex and powerful mobile phones and laptops. As described before, the information will be gathered from the surrounding ambient through sensing/identifying devices, but it might be consumed by user devices located very far from the source.

3. An Architecture for Large Scale WSNs

As we mentioned before, we can foresee that these upcoming pervasive communication/computing environments pose three main challenges to the conventional networking approaches: *heterogeneity, scalability and complexity* [29].

Heterogeneity stems from the ongoing differentiation in the devices which will form the future ubiquitous network. In practice, we can devise at least two opposite trends. On one hand, portable devices (e.g., laptops, PDAs, smart-phones etc.) are becoming more and more performing, with a large amount of processing power. They usually carry satisfactory communication capabilities even for intense data transfers and are enforced with energy aware software and hardware design aimed to much longer lasting communication and computations than before. In the opposite direction, there is a technological trend toward miniaturized devices with sensing/identifying and basic communication capabilities. Such devices could be embedded in the objects surrounding us in our ev-

[1] These kind of networks are sometimes described by the term pervasive, in order to underline that we expect a true *continuum* of such devices embedded in the environment.

eryday life and represent the ideal interface to the variables describing the surrounding environment.

We notice that according to the Internet philosophy, any application running on such devices would require a full communication protocol stack such as a TCP/IP pile plus a network interface, a link layer, a MAC and PHY layer, and all this severely impairs the possibility of sizing the hardware and software complexity on their true communication requirements.

As concerns *scalability*, the end-to-end paradigm typical of Internet-based communications, suffers from insurmountable scalability problems when applied to large-scale wireless environments. The first concrete concerns on the feasibility of a flat ad-hoc sensor network can be traced back to the seminal work of Gupta and Kumar on the capacity of wireless networks [30]. Subsequent works have dig further into such topic, finding the conclusion that imposing strict connectivity requirements may negatively impact network capacity [31,32]. In the case of pervasive environments the problem is further exacerbated by the fact that a massive sensor deployment will introduce a huge number of data sources in the network, so that several flows sourcing from the same spots would congestion further the available paths.

The third issue is *complexity*, related to the need of controlling and maintaining network functionalities. On one hand, in the pervasive environment, one challenging issue is the matter of numbers, since the system may potentially comprise several millions of interacting nodes. This has a huge impact on the complexity and on the scalability of the control mechanisms: in large-scale systems the amount of regulation needed increases as a superlinear function of the number of nodes, and with the orders of magnitude considered here this is per-se a serious threat.

We conclude that in the case of heterogeneous and large-scale sensor networks, conventional centralized solutions cannot easily be adapted to such networks, and we need to resort to a (rather efficient) distributed management paradigm. This is a direct consequence of the fact that, as we already stressed, in such environments, full connectivity cannot be grated apriori. Anyway, in order to organize the complexity of such environments into a purposeful system, it is needed a framework for providing stable operations and service management functionalities (i.e., configuration, performance, accounting, fault and security) in a fully distributed and decentralized way. In the following we describe the main features of a design solution able to jointly solve these problems for certain delay tolerant applications.

3.1. A Two Tier Architecture

Instead of coping against the unavoidable heterogeneity of devices, the Nomadic Sensor Network solution described in [29,33,34] leverages heterogeneity and splits the nodes of the future pervasive communication environment into two categories. The split is made following the different logical role and the different technical features of the nodes in the network. This split into a two-tier architecture represents the main functional hierarchy of the nomadic wireless networking paradigm [29,33,34]. In the end, this functional hierarchy based on the role of network devices ends up in a precise separation of the requirements of processing/communication/storage capabilities of the devices. As depicted in Fig 2, in particular we expect two kind of nodes:

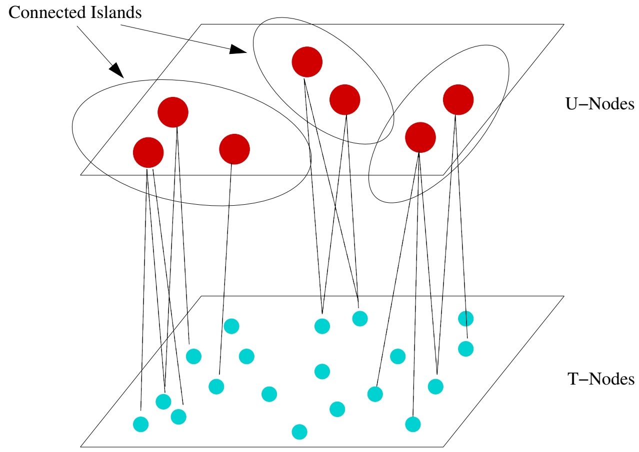

Figure 2. The 2-tier hierarchy of Nomadic Sensor Networks.

- *T-Nodes*, which will be simple tiny devices deployed in the environment, with sensing and basic communication functionalities. These nodes should be low-power and as simple as possible, in order to minimize costs for repair or maintenance. The unique role of T-nodes is to act as *source of information* on the status of the environment: the term pervasivity should be in this context interpreted as the need for fine-grained environmental data. We expect T-nodes to run a minimal protocol with no need to store and forward. The only communication requirements are short-range exchange of raw data, obtained from measurements, to U-nodes in proximity.
- *U-nodes*, which correspond to users' devices (e.g., PDAs, cell phones etc.). Their role will be to both gather information from T-Nodes deployed in the environment, and diffuse information to other U-Nodes in proximity. Clearly, U-nodes perform more complex operations on raw data and run complex information exchange protocols. U-nodes will be moving in the environment as a consequence of the physical movement of the users carrying them around, and will collect information from T-Nodes and store this information in their local memory.

The effect of such a hierarchical split is apparent because no communication among T-Nodes is encompassed and, compared to the conventional sensor network approach, T-nodes are freed from the overhead caused by store-and-forward operations. This is expected to allow for smaller, cheaper and longer-lasting devices [35]. U-Nodes can "poll" the nearby T-Nodes and communicate among them when they get into mutual communication range. Based on the distinction, all the information available to U-nodes will be available either from proximity T-Nodes or from other U-Nodes. Notice that this relaxes the need for the network to be always connected, since T-nodes will generate data when interrogated by U-nodes, and this will heavily reduce the required number of links.

The Nomadic Sensor Network overcomes the scalability issues of large scale sensor networks, since no end-to-end communication is employed. In fact, under an end-to-end paradigm, the scalability issues of a large scale ad-hoc sensor network are due to two key features: the exponential growth of the end to end traffic along with the number of sensors and the concurrent decrease of the capacity at the increase of the number of communicating nodes.

The key is to adopt an infrastructure-less networking paradigm tailored to pervasive computing environments, based on the 2-tier hierarchy devised before and a suitable data management mechanisms. The term nomadic , in particular, refers to the mechanism to spread the information once raw data are collected by T-nodes: as it has been shown by Grossglauser and Tse, it is possible to obtain a scalable network by dropping the connectivity requirement and exploiting nodes mobility to convey information [36]. Thus, the first mechanism used to cope with scalability issues is related to the role of U-Nodes where we exploit the mobility of devices in order to let information flow. In this way network connectivity is not required *a priori*, so that, in principle, the topology of the nomadic network can and up into an archipelago of connected islands of nodes (see Fig 2). We notice that in the case of the pervasive environment, the relevant key feature is that a huge fraction of the information should be exchanged locally. Thus, we let U-nodes exchange information in a peer-to-peer fashion through single-hop broadcasting: to some extent, our model may somehow resemble Delay Tolerant Networking [37]. But, Delay Tolerant Networking aims at maintaining the end-to-end semantics (typical of the Internet protocols) in a disconnected environment, thus applying store-and-forward policies for the delivery of data to the end user.

Differently, in this framework, the information comes locally from the environment, and it is limited both in space and time, so that end-to-end communications are replaced by localized peer-to-peer exchanges.

In the end, and depending on the trade offs of density and mobility of devices, information is diffused by either multihopping among U-nodes, and this is the conventional way MANETs [38] work, or just through opportunistic exchanges when U-nodes come into communication range [39]. Thus, most of the communications are restricted to fraction of the overall network, i.e. they are local. We remark that the peer-to-peer exchange described above is completely different from the end-to-end conventional communication approach and addressing and routing are not an issue. Conversely, the trigger to U-nodes to require information and process information, will be dictated by the services mounted on the U-nodes, and, so to speak, driven locally by the services.

3.1.1. Data Management in Nomadic Sensor Networks: Information Filtering

We can then exploit the locality (in both space and time) of information coming from the environment: the basic concept is that data originating from sensors loose their usefulness (i.e., information content) as soon as they spread (in both time and space domain). In other words: assume that we transmit sensors-gathered information in an end-to-end fashion, then it is quite understood that we would overload the network with data carrying a potentially low information content. Then, a further mechanism is introduced, called Information Filtering [29]. The Information Filtering mechanism aims at reducing the overhead of data with low information content, by filtering the packet flows based on their age and traveled distance. A first technique is clearly to act on a threshold basis, i.e. dropping information older than a given age or information which traveled more than a certain distance. In general, the problem is to determine the optimal policy for a node, i.e. if a U-node should drop the information or continue the diffusion of the received information: larger thresholds will impact negatively network capacity and overload U-node memory, small thresholds will cancel a large fraction of sensor information.

Putting together the structure outlined above, from the communication point of view, the net result is a quite simple system, which relies on two layers, i.e. the U-Nodes layer the T-Nodes layer, U-nodes communicate among them when they get within radio range, the information spreads according to the user mobility pattern and it is filtered to preserve the system from overflowing.

Of course, since we do not assume a priori any backbone support, the information flow is generated solely by the physical movement of users, together with the opportunistic exchange of data. Contextual information, which is generated from sensors, is also diffused by means of the users' physical mobility: clearly, this means that, for efficiently running services based on this mechanism, an adequate level of users mobility is needed in order to provide a sufficient flow of information in the environment [29].

4. Conclusions

Wireless Sensor Networks are emerging as one of the most promising technologies for bringing the pervasive computing vision into reality. Due to the possibility of sensing the physical world with a granularity unimaginable before, a vast class of innovative applications becomes possible.
However, WSNs come at the cost of totally new technological challenges to be faced. Sensor nodes need to work in an extremely frugal energy budget, since they are often deployed in remote locations where the replacement of batteries is not an option, and this calls for innovative techniques capable of prolonging the network lifetime. Data management in WSN deals with the challenging task of *exploiting* the relevance of the information in order to reduce the communication needed for conveying the data outside the network. In this chapter, we have briefly reviewed the reference WSN data management techniques adopted, with a particular attention to distributed storage, distributed querying and data aggregation.
The proposed techniques, while reducing the communication overhead, thus alleviating the constraints on the sensor nodes, do not solve the communication problems deriving from a a network composed by thousands of nodes applying a multi-hop communication paradigm for delivering data. This is the major result of the seminal work of Gupta-Kumar [30], where it is shown the throughput of a large-scale network is O of the inverse of the number of nodes. Hence, as the number of nodes increases, alternative solutions has to be considered. In the second part of this chapter, we have reviewed a 2-tier network architecture tailored to the provisioning of pervasive services in urban environments. The proposed architecture, by assuming mobile users as nodes of the network, leverages on their movement and on the opportunistic relaying of data for relaxing the constraints on the sensor nodes. The proposed network architecture has been described, together with the principles of a suitable data management technique, called Information Filtering.

5. Acknowledgements

The architecture of Nomadic Sensor Networks was conceived and developed under the support of the EU within the framework of the BIONETS project EU-IST-FETSAC-FP6-027748 [40].

References

[1] E. C. Whitman, “Sosus: The "secret weapon" of undersee surveillance,” *UnderseeWarfare*, vol. 7, no. 2, Winter 2005.

[2] D. Culler, D. Estrin, and M. Srivastava, “Guest editors’ introduction overview of sensor networks,” *Computer*, vol. 37, no. 8, pp. 41–49, August 2004.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, pp. 393–422, 2002.

[4] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater acoustic sensor networks: Research challenges,” *Ad Hoc Networks (Elsevier)*, vol. 3, no. 3, pp. 257–279, May 2005.

[5] J. Lifton, D. Seetharam, M. Broxton, and J. A. Paradiso, “Pushpin computing system overview: A platform for distributed, embedded, ubiquitous sensor networks,” in *Pervasive ’02: Proceedings of the First International Conference on Pervasive Computing*. London, UK: Springer-Verlag, 2002, pp. 139–151.

[6] “Crossbow technology inc.” [Online]. Available: http://www.xbow.com/

[7] C.-Y. Chong and S. P. Kumar, “Sensor networks:evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, August 2003.

[8] K. Romer and F. Mattern, “The design space of wireless sensor networks,” *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 54–61, Dec 2004.

[9] J. M. Kahn, R. H. Katz, and K. S. J. Pister, “Next century challenges: Mobile networking for ”smart dust”,” in *Proc. of ACM MobiCom*, Seattle, 1999, pp. 271–278.

[10] D. Arvind and K. Wong, “Speckled computing: Disruptive technology for networked information appliances,” in *in Proceedings of the IEEE International Symposium on Consumer Electronics (ISCE’04)*, September 2004, pp. 219–223.

[11] P. S. Philippe Bonnet, Johannes Gehrke, “Towards sensor database systems,” *Lecture Notes in Computer Science*, vol. 1987, p. 3, 2001.

[12] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong, “Model-driven data acquisition in sensor networks.” in *VLDB*, 2004, pp. 588–599.

[13] Y. Yao and J. Gehrke, “The cougar approach to in-network query processing in sensor networks,” 2002. [Online]. Available: citeseer.ist.psu.edu/yao02cougar.html

[14] ——, “Query processing for sensor networks.” in *In Proc. Conf. on Innovative Data Syst. Res*, 2003, pp. 233–244.

[15] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “Tinydb: an acquisitional query processing system for sensor networks.” *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, 2005.

[16] V. Handziski, J.Polastre, J.H.Hauer, C.Sharp, A.Wolisz, and D.Culler, “Flexible hardware abstraction for wireless sensor networks.” in *In Proc. Conf. on EWSN*, 2005.

[17] D. Ganesan, B. Greenstein, D. Estrin, J. Heidemann, and R. Govindan, “Multiresolution storage and search in sensor networks,” *j-TOS*, vol. 1, no. 3, pp. 277–315, Aug. 2005.

[18] S. Tilak, N. B. Abu-Ghazaleh, and W. R. Heinzelman, “Collaborative storage management in sensor networks,” *CoRR*, vol. cs.NI/0408020, 2004.

[19] L. Tong, Q. Zhao, and S. Adireddy, “Sensor networks with mobile agents,” in *MILCOM 2003 - IEEE Military Communications Conference*, October 2003, pp. 688–693.

[20] T. Liu, C. M. Sadler, P. Zhang, and M. Martonosi, “Implementing software on resource-constrained mobile sensors: Experiences with impala and zebranet.” in *MobiSys*, 2004.

[21] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, 2003.

[22] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, “Data-centric storage in sensornets,” October 2002. [Online]. Available: citeseer.ist.psu.edu/shenker02datacentric.html

[23] T. He, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, “Aida: Adaptive application-independent data aggregation in wireless sensor networks.” *ACM Trans. Embedded Comput. Syst.*, vol. 3, no. 2, pp. 426–457, 2004.

[24] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 575–578.

[25] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[26] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 131–146, 2002.

[27] S. Ray, R. Ungrangsi, F. D. Pellegrini, A. Trachtenberg, and D. Starobinski, "Robust location detection in emergency sensor networks," in *Proceedings of INFOCOM 2003*, San Francisco, 2003.

[28] M. Weiser, "The computer for the 21st century," *ACM Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, 1999.

[29] I. Carreras, I. Chlamtac, F. D. Pellegrini, and D. Miorandi, "Bionets: Bio-inspired networking for pervasive communication environments," *IEEE Transactions on Vehicular Technology*, to appear.

[30] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. on Inf. Th.*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[31] O. Dousse and P. Thiran, "Connectivity vs capacity in dense ad hoc networks," in *Proc. of IEEE INFOCOM*, Hong Kong, 2004.

[32] O. Dousse, M. Franceschetti, and P. Thiran, "The costly path from percolation to full connectivity," in *Proc. of Allerton Conf.*, Urbana Champaign, US, 2004.

[33] I. Carreras, I. Chlamtac, H. Woesner, and H. Zhang, "Nomadic sensor networks," in *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN)*. Istanbul, Turkey: Springer-Verlag, Jan. 2005, pp. 166–176.

[34] I. Chlamtac, I. Carreras, and H. Woesner, "From internets to bionets: Biological kinetic service oriented networks," in *Advances in Pervasive Computing and Networking*, B. Szymanski and B. Yener, Eds. Springer Science, 2005, pp. 75–95.

[35] I. Carreras, I. Chlamtac, H. Woesner, and H. Zhang, "Nomadic sensor networks," in *Proc. of EWSN*, Istanbul, Turkey, 2005.

[36] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. on Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

[37] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. of ACM SIGCOMM*, Karlsruhe, DE, 2003.

[38] I. Chlamtac, M. Conti, and J. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad-Hoc Networks Journal*, vol. 1, pp. 13–64, Jul. 2003.

[39] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding," Univ. of Cambridge, Tech. Rep. UCAM-CL-TR-617, 2005.

[40] The BIONETS IP Project. [Online]. Available: http://www.bionets.org