1

Multimedia Data Processing and Delivery in Wireless Sensor Networks

Javier Molina, Javier M. Mora-Merchan, Julio Barbancho and Carlos Leon
University of Seville (Department of Electronic Technology)
Spain

1. Introduction

In the last few years, multimedia processing in Wireless Sensor Networks (WSNs) has become a promising technology. It has the potential to enable a large class of applications, most of them related to surveillance and locating (i.e. target detection and tracking, border protection, patient and elderly assistance, people and object identification, environment monitoring, fire detection, industrial control, ...). To achieve an effective Quality of Service (QoS) in multimedia applications, special node and network capabilities are required. For example, compared to normal WSN nodes, multimedia nodes need additional hardware resources for memory, processing capability, transmission rate and energy. A Wireless Multimedia Sensor Network (WMSN) is a special WSN made up of several multimedia sensor nodes, specially designed to retrieve multimedia content such as video and audio streams, still images, and scalar sensor data from the environment.

In this paper, we focus on a different, but also very practical and common, sort of wireless network: the Heterogeneous Networks, where multimedia and non-multimedia nodes deliver data. In this scenario, the non-multimedia node constraints have to be taken into account to deliver multimedia data through multi-hop paths.

Generally, collaborative processing makes no sense in this case. Although in-sensor multimedia processing is a fundamental topic in order to obtain an effective data reduction.

Currently, research challenges in designing multimedia applications on WSNs include, but are not limited to the following:

- QoS requirements. Streaming media, system snapshots, audio/video store and playback applications have different requirements with respect to delay, jitter, and loss tolerance.
- Bandwidth. WMSNs require a bandwidth that is orders of magnitude higher than that supported by currently available sensors.
- Power. Compared to traditional WSNs, power consumption is greater in multimedia applications because of high volumes of data, high transmission rates, and extensive processing.
- In-network processing support, to efficiently extract relevant information from multimedia data (e.g. panoramic image fusion, target identification and location).

- In-node processing support, to compress data, signal analysis and features extraction (singular points, region segmentation, object detection, ...)
- Cross-layer design. An effective optimization of all the above parameters involve cross-layer protocol design ranging from Application to Physical Layer.
- New hardware design to better manage the energy with high QoS (i.e. power supply, microcontroller (MCUs) architectures and energy harvesting).

In this paper, we propose a new WMSN model that includes the Heterogeneous Network. For this reason, main topics of Wireless Sensor Networks are described in section 2, and the most relevant issues in multimedia networks are summed up in section 3. These issues are: standard network architectures, node constraints, compression techniques, memory, bandwidth, and energy consumption.
Different hardware structure alternatives for heterogeneous nodes are introduced in section 4. Moreover, a simple application algorithm is proposed to enable the low-end MCUs to capture multimedia data, or collaborate in delivering.

2. General vision on the design of Wireless Sensor Network Applications

Because of cost and size, the nodes of a WSN exhibit resource constraints in terms of CPU processing capacity, memory, bandwidth and energy. Figure 1 shows a node hardware structure, and table 1 sums up a comparative study of popular MCUs, RF chips, and platforms.

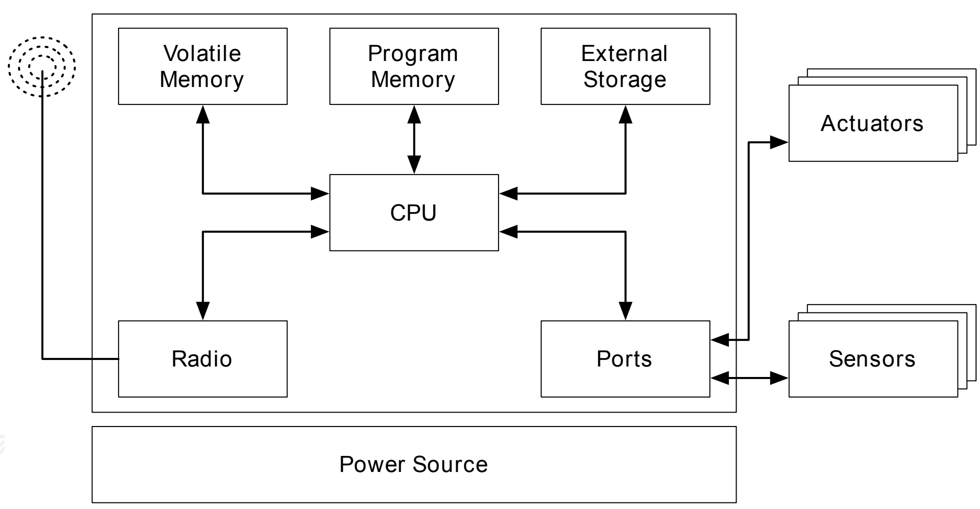

Fig. 1. Node hardware architecture (Soro & Heinzelman, 2009)

Many WSNs are designed for environmental sensing applications, acquiring data from scalar sensors (temperature, humidity, light, etc). In those cases where vector data sensing are required, the vector dimension is often low, for example movement sensing (accelerometers), wind data (anemometers), velocity and positioning (GPS sensors). In contrast, multimedia sensor nodes are characterized by audio and video streaming, and still image data.
Software development for WSNs nodes is a complex issue. Many researchers program the modes from scratch, using operating system components, specific middleware, or by higher programming abstractions (Mottola & Picco, 2010). Figure 2 summarizes a general software model on a WSNs node platform.

Sensor Node Name	Micro controller	Transceiver	Progra+DataMemory
BTnode	ATmega128L	CC1000	64k+180K
EPIC mote	MSP430	CC2420	10k
EyesIFX v2	MSP430F1611	TDA5250	10k+48k
FireFly	ATmega128L	CC2420	8k
GWnode	PIC18LF8722	BiM	64k
Imote 2.0	PXA271ARM	CC2420	256k
Mica2	ATmega128L	CC1000	64k+180k
TelosB	MSP430F1611	CC2420	10k+48k
TinyNode	MSP430	XE1205	8k
XYZ	ML67	CC2420	32k

Table 1. WSN Platforms

MAC- Medium Access Control is placed jointly to the operating systems because, in contrast to other wireless technologies where MAC protocol is done in hardware, in WSNs it is typically implemented mostly in software, using the low-level language associated to the operating system. MAC protocols must guarantee an efficient access to the communication media while carefully managing the energy budget available in the node. The latter goal is typically achieved by switching the radio to a low-power mode based on the current transmission schedule.

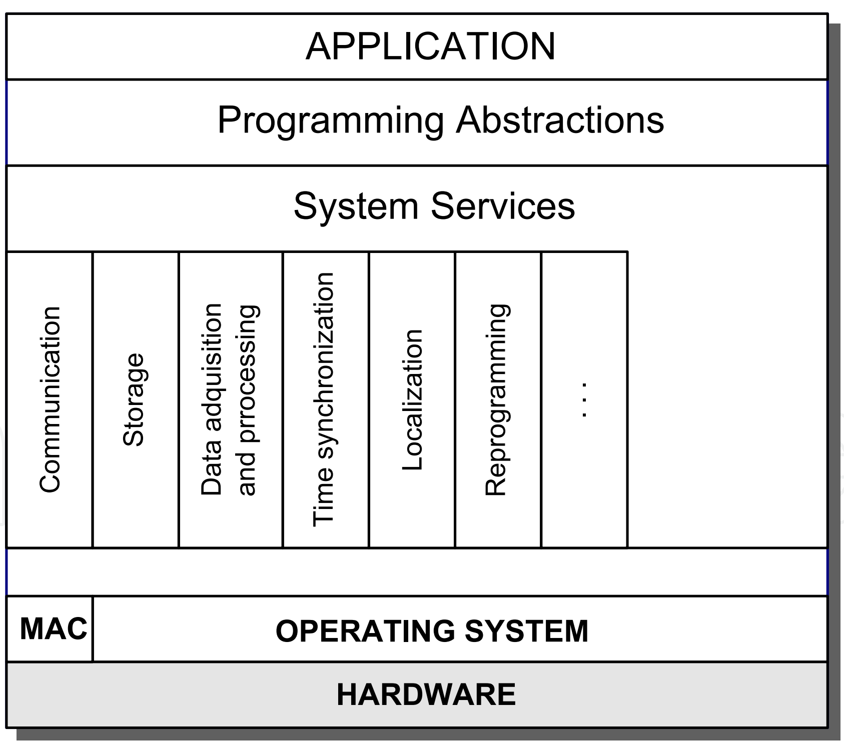

Fig. 2. Node software model

The operating system is essentially a library linked to the application code to generate an executable binary code. This way, the program memory resource is reduced to a minimum, and only necessary hardware specific routines are included. So far, many operating systems

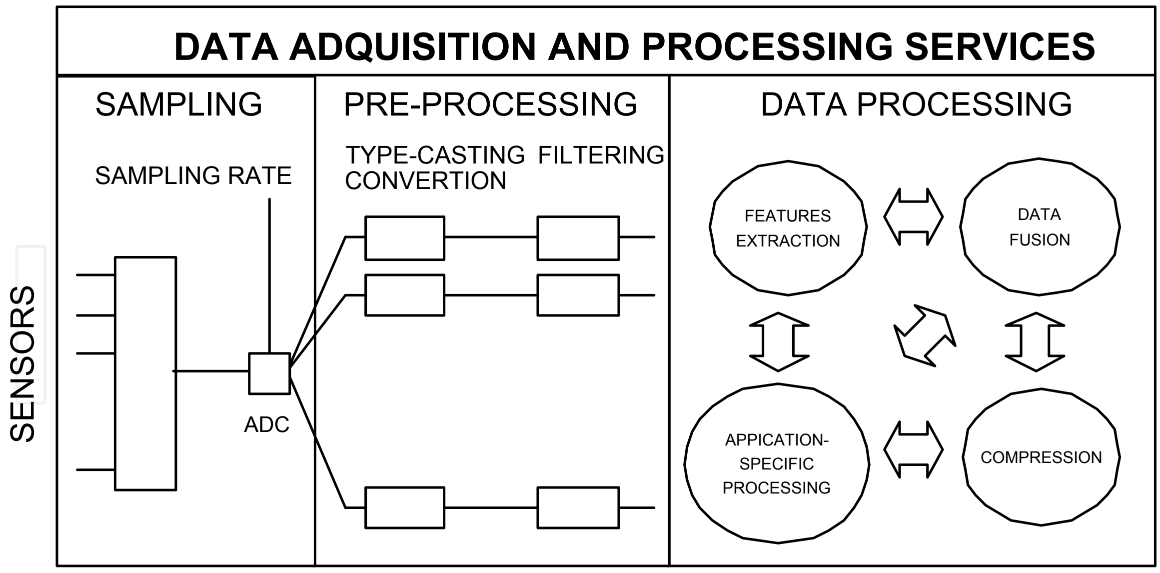

Fig. 3. Data acquisition and processing service

and specific program languages have been proposed. NesC language (Gay et al., 2003) for TinyOS (Hill et al., 2000) are the most popular. Alternatives include Contiki (Dunkels et al., 2004), SOS (Han et al., n.d.), Mantis (Abrach et al., 2003), RETOS (Cha et al., 2007), LiteOS (Cao & Abdelzaher, n.d.), t-Kernel (Gu, 2006), or NANOrk (Eswaran et al., 2005). Some of the above operating systems (e.g., SOS, LiteOS, and Contiki) also provide dynamic linking capabilities, i.e., new code modules can be added at run-time to the application running on a node. Dynamic linking is particularly important in supporting wireless reprogramming of the WSN.

System Services are programs designed to support applications. Typically, an application requires Location, Time Synchronization, Storage, Communication and Data acquisition/processing services. Location services estimate the own node position or the location of a target (He et al., 2003; Mao et al., 2007; Patwari et al., 2005). Time synchronization is usually required for data time-stamping, to measure the TOA (Time of Arrival) of a signal, to manage hibernate-wake up cycles, etc. Many algorithms have been proposed for this (Dai & Han, 2004; Elson & Römer, 2003; Maróti et al., 2004). Communication services deliver data reliably, manage the network and optimize QoS and energy savings. The researchers' community has developed many different algorithms for routing, cluster management, etc. Although, the industry has developed the Zigbee standard, it is probably the most widely accepted one so far.

In most applications, sensor networks are deployed once and intended to operate unattended for a long period of time. Reprogramming capabilities facilitate the management of application changes or updates (Rubio et al., 2007). Because of the data memory constraints in the nodes, it is not possible to save a monolithic program with many applications before being burned. Instead of this, small packets of code can be reprogrammed and can also be reliably delivered through the network. There are two basic schemes to reprogram sensor networks: code dissemination and code acquisition. The first one is initiated by the administrator to reprogram all the devices in the network. In the second one, the nodes request and reprogram a packet dynamically. Data acquisition is not always considered and programmed as a service. However, it is especially relevant in multimedia applications. Figure 3 shows a software

Phenomena	Sample rate (/Hz)
Very low frequency	
Atmospheric temperature	0.017–1
Barometric pressure	0.017–1
Low frequency	
Heart rate	0.8–3.2
Volcanic infra-sound	20–80
Natural seismic vibration	0.2–100
Mid frequency (100Hz – 1000 Hz)	
Earthquake vibrations	100-160
ECG (heart electrical activity)	100–250
High frequency (>1 kHz)	
Breathing sounds	100–5k
Industrial vibrations	40k
Audio (human hearing range)	15–44k
Audio (muzzle shock-wave)	1M
Video (digital television)	10M

Table 2. Sensor sampling rates of different phenomena.

model for data acquisition and processing service with three stages: Sampling, Pre-processing and Data processing.

In the sampling stage, a calibrated set of sensors are sampled, and the row data are time-stamped, if necessary. Sensor calibration is an important issue. Traditional methodologies can not be applied because of the cost of a manual and node-by-node adjusting. The sampling process is fired by either an event or periodically. Following the Nyquist information theory, sampling rate depends on phenomena dynamics. In Table 2 are shown data rates for different applications.

More modern theories, like compressive sensing, allow the reduction of the sampling rate for specific signals (e,g images).

Sensor calibration is an important issue. Traditional methodologies can not be applied because of the cost of a manual and node-by-node adjusting. Several collaborative techniques have been proposed to allow self-calibration (Feng et al., 2003; Ihler et al., 2004; Miluzzo et al., 2008).

Accommodation stage involves data transformation into engineering units, and data-type casting. To pick the right data type, value ranges and quantization errors should be taken into account. Although, using high precision arithmetic, like floating-point, makes the processing performance drastically fall in small CPUs (Zoumboulakis & Roussos, n.d.).

Filtering, if present, reduces or removes undesirable time tendencies, spurious interferences or frequency bands to enhance later processing algorithms. This stage is especially relevant in audio processing applications.

The application-specific processing stage often includes algorithms for data compression, data fusion and feature extraction. In-node data fusion allows the qualification and validation of sensor data, making it easier to associate different physical measures to complex phenomena. Feature extraction is highly application dependant. Its main goal is to obtain a pattern to better represent, detect, identify or predict events and complex phenomena.

Data storage strategies usually distinguish between short-term and long-term storage. Uncompressed, lossless compressed and labeled fused data are often necessary in short-term storage . Otherwise, lossy compression and ancient data loss (amnesic algorithms) are relevant topics in long-term storage (Girao et al., 2007; Sheng et al., 2006).
All the stages can interchange information for feedback and adaptive processing. In-network processing involves data exchange among several sensor nodes. At the sampling stage, for example, several collaborative algorithms have been proposed to calibrate the sensors (Feng et al., 2003; Ihler et al., 2004; Miluzzo et al., 2008). A smart analysis of fused data or features from different sensor nodes is necessary to process localized phenomena (i.e. target detection, tracking and positioning (Arora et al., 2004; Cao et al., 2005)). Collaborative in-network processing can also increase application reliability and performance (i.e. using voting schemes and/or analyzing different target features in different nodes (Bénézit, 2009)).

3. Wireless Multimedia Sensor Networks: specific characteristics

Classic Wireless Sensor Networks have been initially thought up to manage ambient data at low data rates. In general, for multimedia signals, high computing capabilities, high transmission rates, large memory and an unlimited power source supply are necessary. Wireless Multimedia Sensor Networks do not meet all of these conditions, though they do support ambient sensing and multimedia applications, but with a limited quality of service.

3.1 Architectures

The node model shown in figures 1 and 2 is also suitable to describe a node of a Wireless Multimedia Sensor Network (WMSN). However, multimedia signals require higher sampling rate and more complex data processing. These topics make multimedia applications high consumers of resources, in terms of memory, bandwidth, processing capabilities and energy. Limited resources in most hardware platforms are some of the more relevant constraints for multimedia applications. Other restrictions depend on the application. They can be described in terms of Quality of Service. For example, in image delivery, topics like transmission reliability or image quality can be relevant, and video streaming can require security, and low delay. (Faheem et al., 2010)
Overall network architecture is also another relevant topic in order to understand practical constraints. The research community has proposed three architectures, as described in the survey (Akyildiz et al., 2008). From this paper, figure 4 depicts these.
Elements of these architectures are described below:
Single-tier flat architecture is made up of homogeneous multimedia sensors connected to a sink that bridges data to a storage hub or a gateway. The function of these elements is described below:

- Standard Video and Audio Sensors capture sound, still or moving images of the sensed event.
- Scalar Sensors acquire data from physical variables such as temperature, light, humidity, etc. They are very resource-constraint devices.
- Multimedia Processing Hubs. These devices have larger resources compared to sensor nodes. They add data streams and reduce the volume of data delivered to the sink.
- Storage Hubs save still-images or multimedia streams for data-mining and feature extraction, even before the data is delivered to the end user.

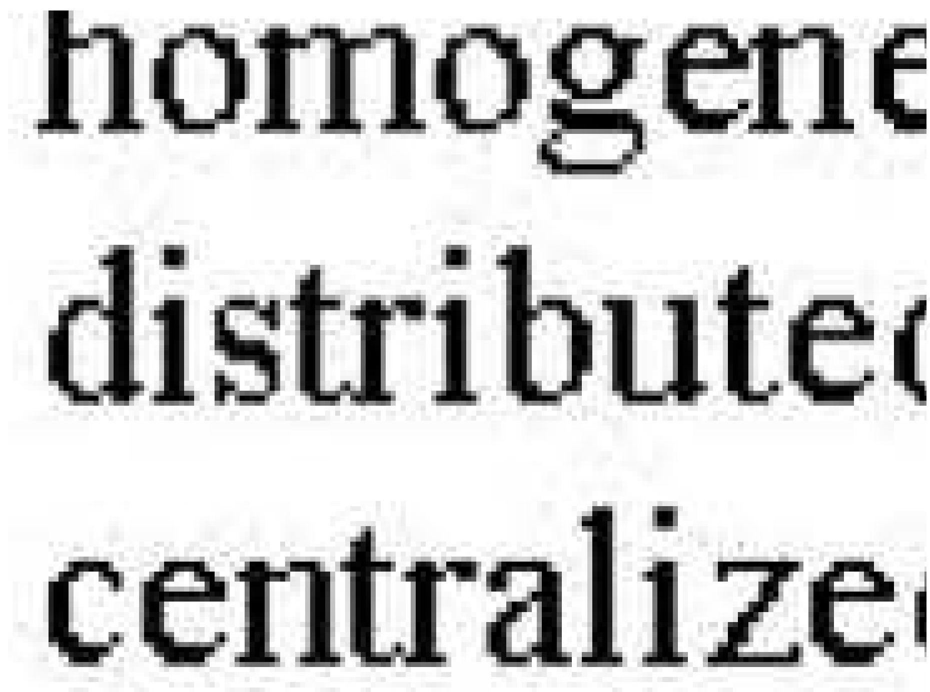

Fig. 4. Standard WMNs architectures (Akyildiz et al., 2008)

- The Sink is responsible for translating high level user queries to network-specific directives and returning the right chunk of the multimedia stream. Several sinks may be needed for large and heterogeneous networks.
- the Gateway bridges the sink to the Internet. IP is only assigned to the gateway, even with multiple sinks. It manages geographical information to allocate the right nodes and sinks for the selected area.

These architectures depict multimedia sensor networks where data flows between devices with the same hardware capabilities, or from lower-end to higher-end devices. Complex in-node and in-network processing is also performed in the multimedia processing hubs or high-end multimedia nodes. Scalar data acquisition and processing are managed in scalar nodes which are not involved in multimedia data processing or delivery. But, most environmental sensor networks manage scalar data and, currently it is becoming more frequent to include audio, sill-image and video delivery when an event has been taken place. This also means that image and audio processing capabilities are needed. This situation can be described as the fourth architecture: single-tier flat heterogeneous sensors (STFH), where low-end or high-end nodes must process and deliver data (streams are not often) through the deployed scalar sensor nodes (low-end devices). This contrasts with standard architectures, where each tier is in fact a sub-network with different performances.
Figure 5 describes the new architecture.
In the single-tier flat heterogeneous architecture, the constraints of the lowest-end nodes limit the performance of the overall network, no matter if it contains high-end video nodes. For this reason, only a limited set of services of image and audio processing make sense. This architecture will be the reference for this study, because it represents both: a very frequent situation in environmental applications, and the worst and most restrictive case for multimedia