

Wireless Multimedia Sensor Networks: Applications and Testbeds

Network testbeds allow the effectiveness of algorithms and protocols to be evaluated by providing a controlled environment for measuring network performance.

By IAN F. AKYILDIZ, *Fellow IEEE*, TOMMASO MELODIA, *Member IEEE*, AND
KAUSHIK R. CHOWDHURY, *Student Member IEEE*

ABSTRACT | The availability of low-cost hardware is enabling the development of wireless multimedia sensor networks (WMSNs), i.e., networks of resource-constrained wireless devices that can retrieve multimedia content such as video and audio streams, still images, and scalar sensor data from the environment. In this paper, ongoing research on *prototypes of multimedia sensors* and their integration into *testbeds for experimental evaluation* of algorithms and protocols for WMSNs are described. Furthermore, open research issues and future research directions, both at the device level and at the testbed level, are discussed. This paper is intended to be a resource for researchers interested in advancing the state-of-the-art in experimental research on wireless multimedia sensor networks.

KEYWORDS | Distributed smart cameras; experimental testbeds; multimedia sensor networks; video sensor networks; wireless sensor networks

I. INTRODUCTION

In our recent survey [1], we discussed the state-of-the-art and research challenges that characterize the so-called wireless multimedia sensor networks (WMSNs), that is, networks of wireless embedded devices that allow retrieving video and audio streams, still images, and scalar

sensor data from the physical environment. With rapid improvements and miniaturization in hardware, a single embedded device can be equipped with audio and visual information collection modules. In addition to the ability to retrieve multimedia data, WMSNs will also be able to store, process in real-time, correlate, and fuse multimedia data originated from heterogeneous sources.

The notion of WMSNs can be understood as the convergence between the concepts of wireless sensor networks and distributed smart cameras. A WMSN is a distributed wireless system that interacts with the physical environment by observing it through multiple media. Furthermore, it can perform online processing of the retrieved information and react to it by combining technologies from diverse disciplines such as wireless communications and networking, signal processing, computer vision, control, and robotics.

The main unique characteristics of WMSN that call for new research in this field can be outlined as follows.

- *Resource Constraints.* Embedded sensing devices are constrained in terms of battery, memory, processing capability, and achievable data rate.
- *Application-Specific QoS Requirements.* In addition to data-delivery modes typical of scalar sensor networks, multimedia data include *snapshot* and *streaming multimedia* content. Snapshot-type multimedia data contain event triggered observations obtained in a short time period (e.g., a still image). Streaming multimedia content is generated over longer time periods, requires sustained information delivery, and typically needs to be delivered in real time.
- *High Bandwidth Demand.* Multimedia contents, especially video streams, require data rates that are orders of magnitude higher than that supported by commercial off-the-shelf (COTS) sensors.

Manuscript received December 20, 2007; revised June 9, 2008. First published October 17, 2008; current version published October 31, 2008. This work was supported by the National Science Foundation under Contract ECCS-0701559.

I. F. Akyildiz and **K. R. Chowdhury** are with the Broadband Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ian@ece.gatech.edu; kaushikc@ece.gatech.edu).

T. Melodia is with the Department of Electrical Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA (e-mail: tmelodia@eng.buffalo.edu).

Digital Object Identifier: 10.1109/JPROC.2008.928756

Hence, transmission techniques for high data rate and low power consumption need to be leveraged.

- *Variable Channel Capacity.* Capacity and delay attainable on each link are location dependent, vary continuously, and may be bursty in nature, thus, making quality of service (QoS) provisioning a challenging task.
- *Cross-Layer Coupling of Functionalities.* Because of the shared nature of the wireless communication channel, there is a strict interdependence among functions handled at all layers of the communication protocol stack. This has to be explicitly considered when designing communication protocols aimed at QoS provisioning on resource-constrained devices.
- *Multimedia Source Coding Techniques.* State-of-the-art video encoders rely on *intraframe* compression techniques to reduce redundancy within one frame, and on *interframe* compression (also *predictive encoding* or *motion estimation*) to exploit redundancy among subsequent frames. Since predictive encoding requires complex encoders, powerful processing algorithms, and high energy consumption, it may not be suited for low-cost multimedia sensors. However, it has recently been shown [2] that the traditional balance of complex encoder and simple decoder can be reversed within the framework of so-called *distributed source coding*. These techniques exploit the source statistics at the decoder and by shifting the complexity at this end, enable the design of simple encoders. Clearly, such algorithms are very promising for WMSNs.
- *Multimedia In-Network Processing.* Processing of multimedia content has mostly been approached as a problem isolated from the network-design problem. Similarly, research that addressed the content delivery aspects has typically not considered the characteristics of the source content and has primarily studied cross-layer interactions among lower layers of the protocol stack. However, processing and delivery of multimedia content are not independent, and their interaction has a major impact on the achievable QoS. The QoS required by the application will be provided by means of a combination of *cross-layer optimization of the communication process* and *in-network processing of raw data streams* that describe the phenomenon of interest from multiple views, with different media, and on multiple resolutions.

While we refer the reader to [1] for a detailed treatment of the research challenges in WMSNs at all layers of the communication protocol stack, in this paper, we describe applications of WMSNs and discuss ongoing research on *prototypes* and *testbeds* for experimental evaluation of algorithms, protocols, and hardware for the development of wireless multimedia sensor networks. Furthermore, we discuss open research issues and outline future research directions. This

paper is intended to be a resource for researchers interested in advancing the state-of-the-art in experimental research on wireless multimedia sensor networks.

Experiments in wireless networking in general and with wireless multimedia sensors in particular are inherently complex, typically time-consuming to set up and execute, and hard to repeat by other researchers. They become even more complex when considering mobile devices [3]. For these reasons, simulation has been the methodology of choice for researchers in the wireless networking domain. However, the research community is becoming increasingly aware of the fact that current simulators are unable to model many essential characteristics of real systems. For this reason, and also due to an apparent degradation in scientific standards in the conduct of simulation studies, simulation results are often questionable and of limited credibility [4]. This gap between simulated and experimental results may result in significant differences between the behavior of the simulated system with respect to the real system. Hence, we argue that for complex systems like wireless multimedia sensor networks, it will be of fundamental importance to advance theoretical design and analysis of networking protocols and algorithms in parallel with sound experimental validation. For such experiments to be commonplace, the cost to setup and maintain an experimental testbed must be decreased as much as possible [3]. A consensus has to be reached among researchers as to the common characteristics that an experimental platform should have, in terms of means for programming the devices, tools for collection and statistical analysis of experimental data, and techniques to ensure that motion is performed accurately when necessary.

While several researchers have proposed application-layer coding techniques, transport-layer rate control algorithms, multimedia packet prioritization at the link layer, and wide-band physical-layer models, the practical implementation of these systems remains an open challenge. Hence, we review the existing experimental work in the context of WMSNs and discuss how they address the challenges of experimental evaluation.

We broadly classify experimental platforms on two levels: i) device level and ii) testbed level, as shown in Fig. 1. Device-level research mainly deals with interfacing

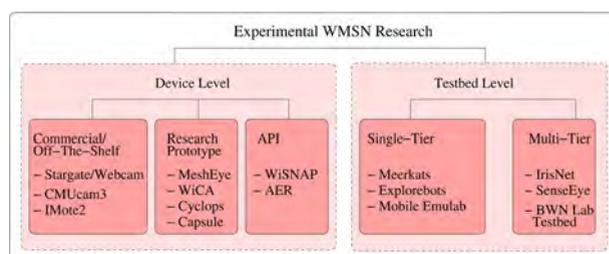


Fig. 1. Classifications in experimental WMSN research.

the video camera, audio, and other sensing circuits with processor, memory, and communication chipset, both in hardware and through software-defined application programmer interfaces (APIs). The small form factor of the sensor nodes, coupled with the need of larger buffer memories for audio-visual sensing, poses several system design challenges.

Research at the testbed level attempts to integrate several individual sensors on a common wireless platform to achieve application-level objectives. Thus, research on experimental testbeds allows evaluating application-level and network-level performance metrics, such as probability of detection of a target, end-to-end delay performance of video/audio streams, or observed jitter in the playback and quality of the received media.

The remainder of this paper is organized as follows. In Section II, we discuss and classify potential applications for wireless multimedia sensor networks. In Section III, we discuss ongoing research on commercial and academic prototypal devices that can find applicability in research on WMSNs. In Section IV, we describe existing software and APIs, while in Section V, we describe the integration of these devices in experimental testbeds. Finally, in Section VI, we conclude this paper.

II. APPLICATIONS

Wireless multimedia sensor networks will enable several new applications, which we broadly classify into five categories.

- *Surveillance.* Video and audio sensors will be used to enhance and complement existing surveillance systems against crime and terrorist attacks. Large-scale networks of video sensors can extend the ability of law-enforcement agencies to monitor areas, public events, private properties, and borders. Multimedia sensors could infer and record potentially relevant activities (thefts, car accidents, traffic violations) and make video/audio streams or reports available for future query. Multimedia content such as video streams and still images, along with advanced signal processing techniques, will be used to locate missing persons or to identify criminals or terrorists.
- *Traffic Monitoring and Enforcement.* It will be possible to monitor car traffic in big cities or highways and deploy services that offer traffic routing advice to avoid congestion. Multimedia sensors may also monitor the flow of vehicular traffic on highways and retrieve aggregate information such as average speed and number of cars. Sensors could also detect violations and transmit video streams to law-enforcement agencies to identify the violator, or buffer images and streams in case of accidents for subsequent accident scene analysis. In addition, smart parking advice systems based on WMSNs [5] will allow monitoring available parking spaces and provide drivers with automated parking advice, thus improving mobility in urban areas.
- *Personal and Health Care.* Multimedia sensor networks can be used to monitor and study the behavior of elderly people as a means to identify the causes of illnesses that affect them such as dementia [6]. Networks of wearable or video and audio sensors can infer emergency situations and immediately connect elderly patients with remote assistance services or with relatives. Telemedicine sensor networks [7] can be integrated with third-generation multimedia networks to provide ubiquitous health care services. Patients will carry medical sensors to monitor parameters such as body temperature, blood pressure, pulse oximetry, electrocardiogram, and breathing activity. Furthermore, remote medical centers will perform advanced remote monitoring of their patients via video and audio sensors, location sensors, and motion or activity sensors, which can also be embedded in wrist devices [7].
- *Gaming.* Networked gaming is emerging as a popular recreational activity. WMSNs will find applications in future prototypes that enhance the effect of the game environment on the game player. As an example, virtual reality games that assimilate touch and sight inputs of the user as part of the player response [8], [9] need to return multimedia data under strict time constraints. In addition, WMSN application in gaming systems will be closely associated with sensor placement and the ease in which they can be carried on the person of the player. An interesting integration of online and physical gaming is seen in the game, *Can You See Me Now* (CYSMN) [10], wherein players logging onto an online server are pursued on a virtual representation of the streets of a city. The pursuers are real *street players* who are equipped with digital cameras, location identification, and communication equipment. The feedback from the devices on the body of the street players is used to mark their position and their perception of the environment. The online players attempt to avoid detection by keeping at least 5 m away from the true locations of the street players. The growing popularity of such games will undoubtedly propel WMSN research in the design and deployment of pervasive systems involving a rich interaction between the game players and the environment.
- *Environmental and Industrial.* Several projects on habitat monitoring that use acoustic and video feeds are being envisaged, in which information has to be conveyed in a time-critical fashion. For example, arrays of video sensors are already used by oceanographers to determine the evolution of

sandbars via image processing techniques [11]. Multimedia content such as imaging, temperature, or pressure, among others, may be used for time-critical industrial process control. For example, in quality control of manufacturing processes, final products are automatically inspected to find defects. In addition, machine vision systems can detect the position and orientation of parts of the product to be picked up by a robotic arm. The integration of machine vision systems with WMSNs can simplify and add flexibility to systems for visual inspections and automated actions that require high speed, high magnification, and continuous operation.

III. DEVICE-LEVEL FACTS

A. Architecture of a Multimedia Sensor

A multimedia sensor device may be composed of several basic components, as shown in Fig. 2: a sensing unit, a processing unit (CPU), a communication subsystem, a coordination subsystem, a storage unit (memory), and an optional mobility/actuation unit. Sensing units are

usually composed of two subunits: sensors (cameras, microphones, and/or scalar sensors) and analog-to-digital-converters (ADCs). The analog signals produced by the sensors based on the observed phenomenon are converted into digital signals by the ADC, then fed into the processing unit. The processing unit executes the system software in charge of coordinating sensing and communication tasks and is interfaced with a storage unit. A communication subsystem interfaces the device to the network and is composed of a transceiver unit and of communication software. The latter includes a communication protocol stack and system software such as middleware, operating systems, and virtual machines. A coordination subsystem is in charge of coordinating the operation of different network devices by performing operations such as network synchronization and location management. An optional mobility/actuation unit can enable movement or manipulation of objects. Finally, the whole system is powered by a power unit that may be supported by an energy scavenging unit such as solar cells.

We next describe the major component blocks of a multimedia sensor device and the factors that determine their design choices.

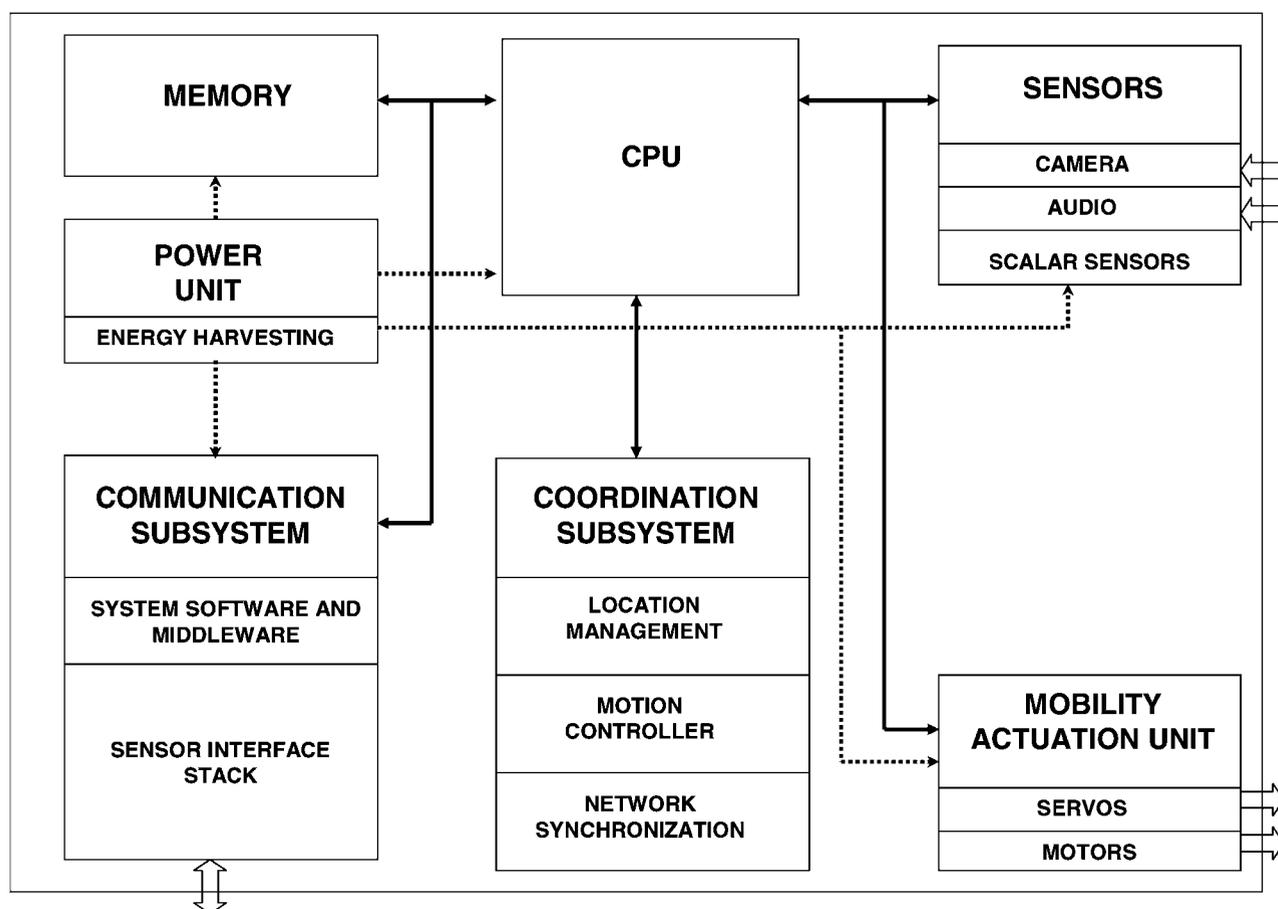


Fig. 2. Internal organization of a multimedia sensor.

1) *Imaging and Sensing Device*: As compared to traditional charge-coupled device (CCD) technology, there is a need for smaller, lighter camera modules that are also cost-effective when bought in large numbers to deploy a WMSN. In a CCD sensor, the incident light energy is captured as the charge accumulated on a pixel, which is then converted into a voltage and sent to the processing circuit as an analog signal. Conversely, the complementary metal-oxide-semiconductor (CMOS) imaging technology [12] is a candidate solution that allows the integration of the lens, an image sensor, and image compression and processing technology in a single chip, thus increasing its complexity but at the same time considerably simplifying the task of interfacing with the other chip components. Here, each pixel has its own charge-to-voltage conversion and other processing components, such as amplifiers, noise correction, and digitization circuits. Depending upon the application environment, such as security surveillance needs or biomedical imaging, these sensors may have different processing capabilities.

2) *Processor*: The wide spectrum of application areas for WMSNs brings about the design choice of processor type and power. In sensor network applications, microcontrollers have typically been preferred over application-specific processors such as digital signal processors (DSPs), field-programmable gate arrays, or application-specific integrated circuits because of their flexibility and ease of reprogramming. For simple, general-purpose applications such as periodic sending of low-resolution images, microcontrollers with limited instruction sets may suffice. However, for streaming video and more complex event-based monitoring tasks, it is important that the data be adequately processed, compressed in volume, and the key information features extracted at the source itself. This calls for more powerful processing platforms and the related tradeoff in power consumption and cost with computational ability. Sometimes, as in the case of Intel/Marvell's PXA271, the microcontroller can be enhanced with a DSP-based coprocessor to accelerate multimedia operations.

3) *Low-Performance Microcontrollers*: Low-cost and simple instruction sets make these processors an attractive option for basic monitoring tasks. For example, the TI MSP430 family of microcontrollers [13] is often used in battery-operated devices with its ultra-low-power 16-bit reduced instruction set computer (RISC) architecture. Used on the TelosB motes, it drains 250 μA for a million instructions per second (MIPS) at 3 V, making it possible to extend battery life to several years. However, its small instruction set (27 instructions) and the limited 10 KByte of RAM may not be sufficient for more involved tasks. For applications involving moderate computations, 8-bit microcontrollers such as the ATMEL ATMEGA128L may be preferred. These microcontrollers, which are used on

MICA2 and MICAz motes, have 128 KByte of program-mable Flash memory, in addition to 4 KByte EEPROM and 4 KByte of internal SRAM. They provide a throughput of up to 16 MIPS at 16 MHz with 2.7–5.5 V [14]. Apart from this, the additional RAM and debug support provided at the hardware level helps in efficient code implementation.

4) *High-Performance Microcontrollers*: For resource-intensive applications, processors that can handle a higher degree of parallelism for every instruction cycle may be preferred. For example, the 32-bit Intel/Marvell PXA255 Processor is targeted at low-power devices and supports fast internal bus speeds up to 200 MHz [15]. In addition, it provides embedded command libraries to optimize performance-intensive applications like MPEG4 video, speech, and handwriting recognition. The more recent Intel/Marvell PXA271 processor is a 32-bit architecture designed for mobile and embedded applications and can be clocked at 312 and 416 MHz. It includes wireless MMX, a set of 43 new instructions that can be used to boost speed in encoding and decoding operations.

Another design approach is to have multiple processors dedicated for video analysis on the same chip but linked to a low-power microcontroller for interfacing with the transceiver and imaging modules of a sensor node. The IC3D [16], a member of the Xetal family of processors, conforms to this design. A key feature of this processor is the use of single-instruction multiple-data (SIMD) that allows one instruction to operate in parallel on several data items instead of looping through them individually. This is especially useful in audio and image processing and considerably shortens the processing time. The IC3D has a linear array of 320 RISC processors, with the function of instruction decoding shared between them. In addition, one of the components, called global control processor (GCP), is equipped to carry out several signal-processing functionalities on the entire data. The lower power application consumption (below 100 mW) and the ease of programmability through a C++ like high-level language makes this processor useful for WMSN applications.

The choice of a processor should be driven by the desired tradeoff between processing capabilities and energy consumption. Traditional first- and second-generation "scalar" motes¹ are based on simple 8-bit microcontrollers, designed to perform basic operations with low energy consumption. However, while this is certainly a good design choice for scalar sensors performing simple scalar operations, for processor-intensive multimedia operations the choice of the right processor needs careful deliberation, even when energy efficiency is the major concern, and 32-bit microcontrollers often prove to be the most desirable choice. For example, it has recently been shown [17] that the time needed to perform relatively

¹Crossbow MICA2 and MICAz mote specifications, <http://www.xbow.com>.

Table 1 A Comparison of the Multichannel Capability Supported by Existing Devices and Standards

Device Type	Frequency Band	Number of Channels	Channel Spacing	Data Rate	Modulation Type
IEEE 802.11b	2.4 GHz	11 (DSSS)	5 MHz	1 Mbit/s	BPSK
	⋮	⋮	⋮	2 Mbit/s	QPSK
	⋮	⋮	⋮	5.5 Mbit/s	CCK
	⋮	⋮	⋮	11 Mbit/s	⋮
IEEE 802.15.4	868 MHz	1	—	20 kbit/s	BPSK
	915 MHz	10	5 MHz	40 kbit/s	⋮
	2.4 GHz	16	⋮	250 kbit/s	OQPSK
Bluetooth1.2 + Basic Data Rate	2.4 GHz	79 (FHSS)	1 MHz	1 Mbit/s	GFSK
Bluetooth2.0	⋮	⋮	⋮	2 Mbit/s	$\frac{\pi}{4}$ -DQPSK
Bluetooth2.0 + Enhanced Data Rate	⋮	⋮	⋮	3 Mbit/s	8 - DPSK
MICA2	300 – 900 MHz	4/50	⋮	38.4 kbit/s	FSK
MICAz	2.4 GHz	16/83	5/1 MHz	250 kbit/s	OQPSK

complex operations such as two-dimensional convolution on an 8-bit processor such as the ATMEL ATmega128 clocked at 4 MHz is 16 times higher than with a 32-bit ARM7 device clocked at 48 MHz, while the power consumption of the 32-bit processor is only six times higher. Hence, although less expensive, an 8-bit processor ends up being slower and more energy-consuming.

5) *Memory*: Memory on embedded devices can be broadly classified as *user memory*, necessary for storing sensed data and application-related data, and *program memory*, used for programming the device. For low-power devices, on-chip dedicated memory (RAM) is typically used on the microcontroller, and lower cost Flash memories are used to store executable code. Static random-access memories (SRAM), which do not need to be periodically refreshed but are typically more expensive, are used as dedicated processor memory, while static random-access dynamic memories (SDRAM) are typically used for user memory. A higher dedicated RAM memory helps in speeding up computations significantly. As an example, the chip used as a microcontroller on MICA motes, ATMEL ATMEGA103, has a 32 KByte RAM. This considerably limits the data available to the processor during computation, especially when compared to more powerful platforms. For example, in Imote2, the Marvell PXA271 is a multichip module that includes three chips in a single package, the CPU with 256 KByte SRAM, 32 MByte SDRAM, and 32 MByte of Flash memory.

6) *Communication Module*: Transceiver modules are based typically either on WLAN transceiver cards, such as those following the IEEE 802.11b standard, or on the Texas Instrument/Chipcon CC2420 chipset, which is IEEE 802.15.4 compatible. The key difference between them stems

from the i) number of channels that can be used, the ii) bandwidth of the channels, iii) the energy consumption, and iv) the modulation type. In addition, modulation schemes based on the binary phase-shift keying (BPSK) are popular as they are easy to implement in hardware and are resilient to bit errors. Hence, for WMSNs, there is a tradeoff between using a physical layer module that provides high data rates (e.g., 802.11b cards at 11 Mbit/s) against a basic communication chipset with a lightweight protocol (e.g., 802.15.4 on CC2420 radio at 250 kbit/s). The maximum transmit power of the 802.11b cards is higher, which results in greater range but also consumes more power. As an example, the Intel Pro/Wireless 2011 card has a typical transmit power of 18 dBm but typically draws 300 and 170 mA for sending and receiving, respectively. The CC2420 chipset, however, only consumes 17.4 and 19.7 mA respectively, for the same functions with the maximum transmit power limited to 0 dBm, with comparable voltage supply. Also, it should be the design goal of the protocol stack to utilize the maximum number of nonoverlapping channels, with the largest allowed bandwidth and data rate per channel to have the best performance in a multimedia application. The channels, bandwidths, and modulation types for the 802.11b, Bluetooth, and 802.15.4 standards are summarized in Table 1.

B. Commercial Products

Several commercial products are available that can function as a WMSN device, although they differ in the amount of processing power and communication capability and energy consumption. As an example, the Stargate² and the Imote2³ processing platforms discussed in this section can provide IEEE 802.11b- and 802.15.4-based networking

²<http://www.xbow.com/Products/Xscale.htm>.

³<http://www.xbow.com>.

Table 2 An Overview of the Features of the Hardware Platforms for WMSNs

Device Name	Manufacturer	Processor	Memory	Multimedia Support	Wireless
Stargate ²	Crossbow	Intel PXA-255 Xscale processor at 400 MHz	32 MByte Flash 64 MByte RAM	High computation power, embedded linux OS	802.11 Compact Flash, 802.15.4 through MICA2/z interface
Imote2 ³	Intel	32-bit PXA271 Marvell processor at 13 – 416 MHz	32 KByte Flash 64 KByte RAM	MMX co-processor for audio/video imaging acceleration	Integrated 802.15.4
CMUcam3 [18]	CMU	32-bit NXP LPC2106 microcontroller at 60 MHz	128 KByte Flash 64 KByte RAM	On-board cc3-open source image processing library	—
MeshEye [19]	Stanford Univ.	32-bit ARM7TDMI RISC processor at 55 MHz	128 KByte Flash 64 KByte RAM	Multiple resolution support	—
WiCa [20]	XNP and Philips Research	IC3D Xetal II processor at 84 MHz	10 Mbit RAM	Dedicated parallel processor, multiple camera modules	—
Cyclops [21]	Agilent Technologies	8-bit ATMEL ATmega128L microcontroller	512 KByte Flash 512 KByte RAM	On-board image processing, low power, cost and size	—

connectivity. While the above require interfacing with a separate camera, the CMUcam3 is a specialized product that performs both the imaging and processing tasks together. A summary of the products, in terms of both sensor platforms and camera functionalities, is given in Table 2.

1) *Multimedia Sensor Platforms*: For higher performance, the Stargate board designed by Intel and manufactured by Crossbow may be used. It is based on Marvell's PXA255 XScale RISC processor clocked at 400 MHz, which is the same processor found in many handheld computers. It additionally includes 32 MByte of Flash memory and 64 MByte of SDRAM. It can be interfaced with Crossbow's MICA2 or MICAz motes as well as PCMCIA Bluetooth or compact Flash IEEE 802.11 cards. Hence, it can work as a wireless gateway and as a computational hub for in-network processing algorithms. When connected with a Webcam or other capturing device, it can function as a medium-resolution multimedia sensor, although its energy consumption is still high, as documented in [22]. Moreover, although efficient software implementations exist, the onboard processor does not have hardware support for floating point operations, which may be needed to efficiently perform multimedia processing algorithms.

The Imote2 platform, also designed by Intel, is built around an integrated wireless microcontroller consisting of the low-power 32-bit PXA271 Marvell processor, which can operate in the range 13–416 MHz with dynamic voltage scaling. It includes 256 KByte SRAM, 32 MByte Flash memory, 32 MByte SDRAM, and several I/O options. As previously mentioned, specially catering to multimedia requirements, the PXA271 includes a wireless MMX coprocessor to accelerate video/imaging operations and adds 30 new signal-processing-based processor instruc-

tions. The software architecture is based on an ARM port of TinyOS [23]. Alternatively, a version of the Imote2 based on the .NET micro framework from Microsoft has recently become available. The Imote2 can run the Linux operating system and Java applications through a virtual machine. In addition, the Imote2 provides additional support for alternate radios and a variety of high-speed I/Os to connect digital sensors or cameras. Its size is limited to 48 × 33 mm. The 802.15.4-compliant Texas Instruments/Chipcon CC2420 radio supports a 250 kbit/s data rate with 16 channels in the 2.4 GHz band. With the integrated 2.4 GHz surface mount antenna, a typical range of 100 feet (30 m) can be achieved. For longer range requirements, an external antenna can be connected via an optional subminiature version A (SMA) coaxial radio-frequency connector.

2) *Cameras*: The CMUcam3 is an open-source low-cost camera, developed by researchers at Carnegie-Mellon University, and is now commercially available [18]. It is approximately 55 × 55 mm² and 30 mm in depth. The CMUcam3 uses an NXP LPC2106 microcontroller, which is a 32-bit 60 MHz ARM7TDMI with built-in 64 KB RAM and a Flash memory of 128 KB. The comparatively low RAM necessitated the development of a lightweight open-source image-processing library named cc3, which resides onboard. This allows several image-processing algorithms to be run at the source, and only the results may be sent over to the sink through the wireless channel. In addition, the developer tools include the *virtual-cam* software that can be used to test applications designed for the actual camera in a simulated environment. It provides a testing library and project code and can be used on any standard PC by compiling with the native GCC compiler. The CMUcam3 comes with an embedded camera endowed

with a common intermediate format () (i.e., having a 352×288 pixel resolution) RGB color sensor that can capture images at 50 frames per second. It can be interfaced with an 802.15.4 compliant TelosB mote.³

The size and power consumption of the imaging device need further consideration. We recall that the CMOS technology allows fusing several imaging and processing components into a single chip. Thus, the CMOS-based Cyclops framework is designed to address both of the above concerns of WMSNs [21]. It provides an interface between a CMOS camera module and a wireless mote such as MICA2 or MICAz and contains programmable logic and memory for high-speed data communication. Cyclops consists of an imager (CMOS Agilent ADCM-1700 CIF camera), an 8-bit ATMEL ATmega128L microcontroller (MCU), a complex programmable logic device (CPLD), an external SRAM, and an external Flash. The MCU perform the tasks of imaging and inference, while the CPLD complements it by providing access to the high-speed clock. Thus, the CPLD works as a frame grabber, copying the image from the camera to the main memory at a speed that cannot be provided by the ATMEL microcontroller. Cyclops firmware is written in the nesC language [24], based on the TinyOS libraries. The module is interfaced to a host mote to which it provides a high-level interface that hides the complexity of the imaging device to the host mote. Moreover, it can perform simple inference on the image data and present it to the host.

3) *Academic Research Prototypes*: Security and surveillance applications are an important focus area of camera equipped sensors. Depending upon the size of the object under study, it may be preferable to use a single camera with different resolutions, or more than one camera, but with a constant imaging capability.

The MeshEye mote proposed in [19] addresses this challenge by a two-tiered approach. A low-resolution stereo vision system is used to gather data that help to determine the position, range, and size of moving objects. This initial step, in turn, triggers a higher resolution imaging that can be processed later. The mote can support up to eight kilopixel imagers and one VGA camera module. Common to the MICA family of sensors, this architecture also uses a CC2420 IEEE 802.15.4 compliant radio that can support a maximum rate of 250 kbit/s. At such low data rates, video streaming is possible only if sufficient preprocessing steps, such as dimensional reduction and descriptive representations (color histograms, object shape) are undertaken. This approach of taking images with dual resolution works best for small to moderate object sizes.

For larger objects, the WiCa vision system described in [20], with two independent on-mote cameras, is more suited. It consists of two VGA camera modules, which feed video to a dedicated parallel processor based on a vector single-instruction multiple-data (SIMD) architecture. For large objects, the increased processing involved

in the object detection, ranging, and region-of-interest extraction functions is better accomplished with the SIMD architecture.

The advantages of the above two approaches can be combined in the node architecture proposed in [17]. The mote is designed to allow interfacing up to six different cameras of different resolutions on the same board. In addition, the ARM7 32-bit CPU clocked at 48 MHz is shown to be more power efficient than the 8-bit ATmega128 microcontroller that is commonly used for generic sensing motes. The mote is also equipped with an external FRAM or Flash memory and the CC2420 radio. The image sensors can be the midresolution ADCM-1670 CIF CMOS sensors or low-resolution 30×30 pixel optical sensors.

Finally, there exist imaging applications in which the design goals of simplicity, small size, and node lifetime are of the highest importance. These nodes may experience a constant operational environment, where the processing and transmission parameters do not change. As an example, biomedical applications require image sensors that are nonobtrusive and use minimum energy, as they cannot be easily replaced once inserted in the test subject. A CMOS-based single chip sensor for capsule endoscopy is described in [25]. The capsule, less than 5 mm on a side, is implanted in the human body and can return images through a wireless transmitter within it. It consists of a 320×240 pixel array, timing generator, cyclic ADC, and BPSK modulator. However, while the current hardware returns promising results, tests on actual subjects have not been carried out. The receiver station can display images obtained at a signal power of -90 dBm, in open air. The human body, as an example, may induce spurious charge fluctuations in the vicinity of the capsule circuit or result in low signal propagation due to its heterogeneous nature. Thus, the performance of these low-cost wireless implantable sensors merits further study.

IV. SOFTWARE AND APPLICATION PROGRAMMING INTERFACE

The development of efficient and flexible system software to make functional abstractions and information gathered by scalar and multimedia sensors available to higher layer applications is one of the most important challenges faced by researchers to manage complexity and heterogeneity of sensor systems.

As of today, existing software driver interfaces and libraries are often of a proprietary nature. Thus, solutions developed for a particular device cannot be easily ported to another, as there exists no common instruction set. This is an impediment to the widespread use of WMSNs, and we believe there is a need for establishing a basic set of functionalities that can be accessed by the use of APIs. Thus, the application program merely calls these APIs that are well documented and, in turn, recognizes the underlying hardware and controls its drivers appropriately.

However, platform independence is usually achieved through layers of abstraction, which usually introduce redundancy and prevent the developer from accessing low-level details and functionalities. Hence, there is an inherent tradeoff between degrees of flexibility and network performance, while WMSNs are characterized by the contrasting objectives of optimizing the use of the scarce network resources and not compromising on performance. The principal design objective of existing operating systems for sensor networks such as TinyOS is high performance, i.e., perform complex tasks on resource-constrained devices with minimal energy consumption. However, their flexibility, interoperability, and reprogrammability are very limited. There is a need for research on systems that allow for this integration.

We believe that it is of paramount importance to develop efficient, high-level abstractions that will enable easy and fast development of sensor network applications. An abstraction similar to the famous Berkeley TCP sockets, which fostered the development of Internet applications, is needed for sensor systems. However, differently from the Berkeley sockets, it is necessary to retain control on the efficiency of the low-level operations performed on battery-limited and resource-constrained sensor nodes.

As a first step in this direction, the Wireless Image Sensor Network Application Platform (WiSNAP) presents an easy-to-use application interface to image sensors [26]. Several camera-specific parameters can be accessed through simple function calls. Moreover, it integrates the high-level language environment of MATLAB transparently with the camera and the communication module. This allows users to access the rich set of image-processing tools provided by MATLAB without being involved with the minute details of systems programming. Although only the Agilent ADCM-1670 Camera Module is currently supported, the open source architecture of the API allows extension to products made by other vendors. The WiSNAP framework consists of two subparts—i) the image sensor API through which the user can identify the device, the number of frames, and receive the data captured by the desired sensor in form of an image array; and ii) wireless mote API that facilitates mote initialization and medium access control. The work in [26] describes applications that use the WiSNAP APIs for event detection and node localization by tracking the pixel difference between adjacent frames and camera orientations, respectively.

A new approach to sensing called *address event image sensing* (AER) is a software tool to identify the occurrence of an event without sending back real images [27]. The sensors can be visualized as nodes of a large neural network which can independently signal the event. The onboard camera is used as a detection tool by the node to check if the event has occurred. Signal-processing techniques like comparing the used pixels, edge detection

algorithms, and centroid matching algorithms are some of the techniques that are used. The binary decision of the node, along with that of the other sensors, is checked against a prior known event pattern by the AER tool. This approach avoids sending raw data over the wireless link, thus improving energy savings and security. The AER classification is done in a manner similar to the hidden Markov models (HMMs) used in speech processing and handwriting recognition. This tool has been implemented over the Imote2 nodes using the OmniVision OV7649 camera, which can capture color images at 30 fps VGA (640 × 480) and 60 fps QVGA (320 × 240). In an experiment with sensor nodes, the AER could successfully distinguish between the actions of *cooking* and *cleaning* in the kitchen. Hence, the camera nodes do not return the images of the event area. Rather, they merely send back whether they could detect an event or not. This, in turn, is used to form the approximation of the event shown by the projection of the edge nodes.

Switching to different cameras with varying resolution is one way to adaptively reconfigure the working of the WMSN node. Another approach is to decide, on a need basis, the compression algorithm to be used and the transmission parameters such as modulation and coding so that the end-to-end multimedia flow performance is optimized. This marks a shift in complexity from hardware to software design and can lead to cost-effective solutions. Apart from the classical processing and communication blocks, the mobile multimedia architecture presented in [28] has a run-time *reconfiguration* system responsible for understanding the current network conditions and service requirements, and configuring the other blocks accordingly. To this end, the authors have also implemented an on-chip hardware/software system that can apply different compression algorithms on demand while maintaining high levels of energy efficiency. The cross-layer hardware and software design greatly reduces the time and energy consumed for carrying out the image transformation, quantization, and encoding functions for the final goal of image compression.

V. TESTBEDS

In this section, we describe and classify the main functionalities of testbed architectures for WMSNs and classify existing experimental platforms. In particular, in Section V-A, we describe the architecture of typical WMSNs, while in Section V-B, we outline the design space of a testbed architecture for WMSNs. In Section V-C, we describe the desirable features of a testbed. In Section V-D and E we describe existing single-tier and multiple-tier testbeds, respectively.

A. Network Architecture

A typical WMSN architecture is depicted in Fig. 3, where users connect through the Internet and issue

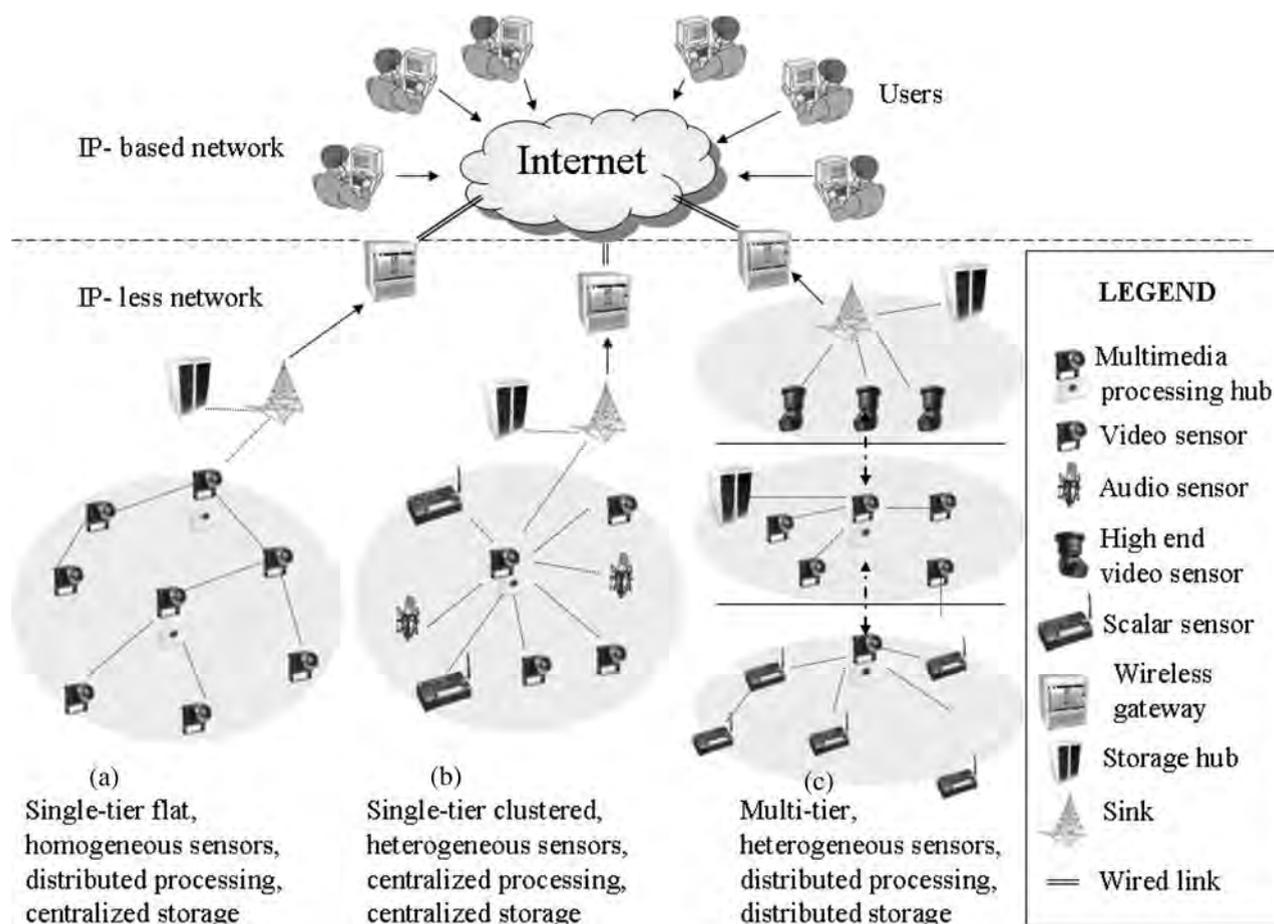


Fig. 3. Reference architecture of a wireless multimedia sensor network.

queries to a deployed sensor network. The functionalities of the various network components are summarized in a bottom-up manner, as shown below.

- *Standard Video and Audio Sensors.* Sensors capture sound, still or moving images of the sensed event. They can be arranged in a single-tier network, as shown in the first cloud, or in a hierarchical manner, as shown in the third cloud.
- *Scalar Sensors.* These sensors sense scalar data and physical attributes like temperature, pressure, humidity and report measured values. They are typically resource-constrained devices in terms of energy supply, storage capacity, and processing capability.
- *Multimedia Processing Hubs.* These devices have comparatively large computational resources and are suitable for aggregating multimedia streams from the individual sensor nodes. They are integral in reducing both the dimensionality and the volume of data conveyed to the sink and storage devices.
- *Storage Hubs.* Depending upon the application, the multimedia stream may be desired in real time or after further processing. These storage hubs allow data-mining and feature extraction algorithms to identify the important characteristics of the event, even before the data is sent to the end user.
- *Sink.* The sink is responsible for packaging high-level user queries to network-specific directives and return filtered portions of the multimedia stream back to the user. Multiple sinks may be needed in a large or heterogeneous network.
- *Gateway.* This serves as the last mile connectivity by bridging the sink to the Internet and is also the only IP-addressable component of the WMSN. It maintains a geographical estimate of the area covered under its sensing framework to allocate tasks to the appropriate sinks that forward-sensed data through it.
- *Users.* Users are the highest end of the hierarchy and issue monitoring tasks to the WMSN based on geographical regions of interest. They are typically identified through their IP addresses and run application-level software that assigns queries and displays results obtained from the WMSN.

B. Testbed Design Space

A WMSN may be equipped with multiple types of cameras with varying resolutions and image-processing ability. In addition, the capability of the underlying communication hardware, especially the transceiver characteristics, may also differ. As an example, a testbed may consist of a limited number of high-end pan-tilt-zoom digital cameras and a comparatively larger proportion of low, fixed-resolution Webcams. The communication chipset may support multiple channels and transmission rates. Thus, testbeds can be characterized in the following ways based on the devices used, communication system, and level of heterogeneity.

- *Imaging Ability.* The presence of different types of cameras (CCD and CMOS) results in a varying ability to capture images by the nodes in the testbed. The choice of camera technology is directly related to the cost of deployment and the application needs, with many existing testbeds choosing to use COTS Webcams to keep the costs low. CCD imagers consume power of an order of magnitude higher than their CMOS counterparts but offer better quality imaging. CMOS imagers offer more on-chip image processing and lower power dissipation and are smaller in size. However, they are less sensitive to light, as part of the sensor is covered with the noise-filtering circuits and the production costs of these sensors are comparatively high. Another key difference in image quality is the higher range of pixel values that can be clearly detected by the CCD sensors compared to CMOS and the uniformity in the readings in the individual pixel positions. CMOS compensates for these drawbacks by its ability to work at higher speeds and greater reliability owing to its on-chip integration of various processing elements.
- *Communication Hardware.* A sensor node may be equipped with multiple communication transceivers. As an example, the Crowssbow Startgate boards can have an operational 802.11 card along with an interfaced MICAz mote that follows the 802.15.4 standard. The number of channels, power restrictions, and channel structure are different in the two cases. The testbed must choose which of the several available transceiver designs and communication protocol standards may be followed to optimize the energy saving and the quality of the resulting communication.
- *Heterogeneity and Hierarchical Organization.* A single testbed may comprise different types of sensors, equipped with cameras of varying resolutions and processing power. Thus, the sensors form a hierarchical arrangement, with the powerful but resource-consuming nodes at the higher levels and simple, low-quality imaging nodes at the lower

levels, such as the Senseye platform [29]. Depending upon the information needed by the user and the quality of the retrieved data, a progressively higher level of sensors could be activated. We call such a tiered network as a multiple-tier WMSN.

- *Scale.* Depending upon the resource constraints, testbeds vary in the number of nodes. From the earlier example, the functioning Senseye testbed comprises four simple sensor nodes for the lower level and two comparatively powerful Stargate boards at the next higher level of hierarchy. The classical WMSN paradigm envisages the number of nodes to be on the order of hundreds or more, but this imposes considerable demands on both the communication protocols and the initial setup costs.
- *Testbed Tools.* The presence of data-monitoring tools, such as packet sniffers and signal analyzers, may enhance the capabilities of a testbed. It is a research challenge to integrate these tools, that may possibly be designed by third-party developers, into the normal functioning of the testbed in a nonintrusive manner.

C. Testbed Features

Testbeds allow observing the performance of the WMSN in a controlled environment. Hence, the effect of different types of inputs, physical operating conditions, and subjects for sensing can be studied, and the functioning of the devices in the testbed may be changed appropriately for accurate measurement. An example testbed architecture is shown in Fig. 4. This setup consists of the WMSN base station (BS), sensor nodes, and supporting tools for monitoring the system performance and storage of the retrieved data. The BS may itself comprise component blocks including the:

- i) *User Interface* block that allows the user to issue commands and view the final data;
- ii) *Configuration* block that alters the functioning of the transceiver circuits and communication protocols based on the state of the network;
- iii) *QoS Monitor* block that evaluates if the received data matches the user performance constraints;
- iv) *Receive and Transmit (Rx/Tx)* block;
- v) *Data Storage* block, which regulates where the large incoming data should be stored and the format for storage;
- vi) *Data Analysis* block, which provides an interface to extract and analyze useful information from the received data and present it in a user-friendly fashion.

The limited internal storage in a sensor may necessitate an external dedicated tertiary storage (storage hub). In addition, the packets sent over the wireless network may be monitored by external trace collectors. By using the information contained in the packet headers, an estimate

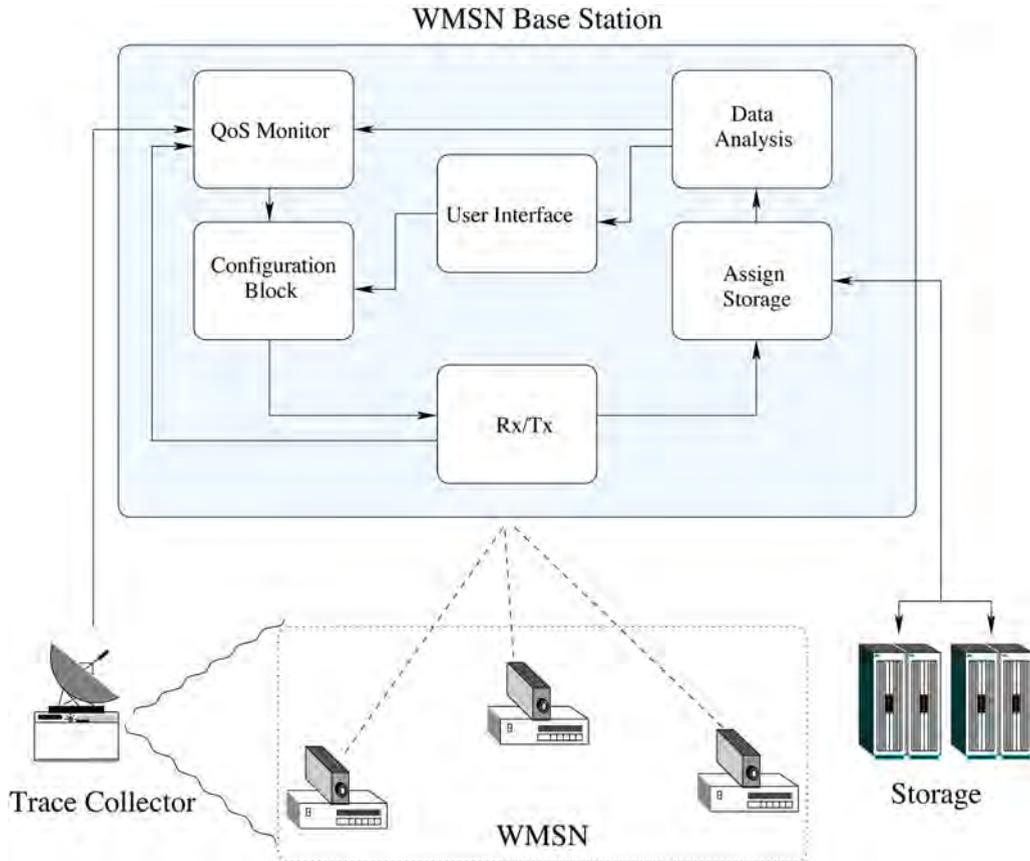


Fig. 4. Testbed features and components.

of the network performance can be obtained, without introducing additional complexity to the BS. We describe these main features of a testbed below and discuss how these features may enhance its capability in collecting the sensed information, storage, and subsequent analysis.

1) *WMSN Base Station*: The WMSN BS makes the various functionalities of the WMSN available to the user through a convenient interface. The stagewise functioning of the BS is shown by the arrows marked in Fig. 4 and is described as follows.

- *User Interface*. The user may run a client application that remotely connects to the BS server through the infrastructure network. This interface gives the user freedom to issue queries to the network, retrieve data from the storage, and assign QoS metrics by remote access. Possible QoS metrics may be the acceptable jitter (variance in the packet arrival times) in the data stream, the end-to-end latency for smooth playback of audio or video data, the resolution of the still images, among others.
- *QoS Monitor*. During the operation of the testbed, by monitoring the packet header information,
- *Configuration Block*. Based on the user-specified preferences and the observed network QoS, the configuration block modifies the receiver and transmitter parameters of the BS and also of the deployed sensor nodes. Examples of reconfiguration may include choice of packet length and forward error correction, channels for use, and transmission rate that can meet the delay and jitter requirements of the user, for a given network size and topology.
- *Receive and Transmit (Rx/Tx)*. The primary purpose of this block is to send commands to the sensor nodes and receive data from them for further storage and processing. The specific standard followed for communication and the tuning of the exact transmission parameters is undertaken by the configuration block. In addition, the design

information about the packet loss over the links, the sources originating the data, and other performance parameters can be obtained in real time. This estimate, along with the processed information about the previously stored data, is used to evaluate the performance of the network by the QoS block.

choices of a collision-based multiple access, such as CSMA, or the use of multiplexing in time and frequency may be made in this block. It covers the wireless protocol stack whose functioning may be adapted based on the application requirements.

- *Data Storage.* The large volume of generated data needs to be efficiently distributed among the available storage resources. The limited storage on the sensor nodes may result in additional storage devices external to the network. The storage assignment block decides which of the several external storage centers should be used for the incoming data based on the volume of packets generated by the stream and their residual capacity.
- *Data Analysis.* The relevant information from the large received volume of data needs to be extracted and presented to the user in a simple, understandable form. The analysis block links with the user interface and serves as the last mile processing before the data is presented to the end user. Additionally, the ability to sift through the received data and extract the performance indicating metrics of the network is a challenge. Thus, this block also analyzes the current QoS of the network on the basis of the received data and provides feedback to the QoS monitoring block. The retrieval function of the data from the storage must be fast for both the back-end data analysis and providing this QoS feedback, necessitating dedicated high-capacity wireline connection to the storage.

We next describe the external components of the system—the trace collection mechanism, storage, and backchannel—in detail.

2) *Trace Collection:* By measuring the types of packets sent in the network, their transmission and reception times, and the packet loss over the links, an assessment can be made on the network performance in terms of throughput, delay, and reliability. Trace collection serves as a valuable tool, in which the networking aspects of the testbed may be monitored nonintrusively by an external device. Owing to the large file sizes generated by multimedia applications, a single sensed video or still image may be split up into a large number of packets. It may not be feasible to wait for the reception of all the packets that comprise the file before making an evaluation of the system performance. In addition, energy considerations require that a stream that does not meet the user-specified QoS requirements be identified as early as possible. In such cases, sniffing on the packets sent over a link and storing relevant header information can yield insight into the operation of the network. As an example, the commonly used HTTP request used to download content from a Web site running at a server has the fields

User-Agent, Content-Type, and Application-Type that are used to identify the type of the browser, the multimedia type, such as video or audio, and the encoding used, such as MP3 or MPEG, respectively. Similarly, by defining and standardizing these fields for a WMSN protocol, the type of sensor in a heterogeneous network and the nature of the multimedia data can be identified. By monitoring these fields, as is done in classical trace collection, a relatively fast estimation of the protocol performance can be undertaken.

3) *Data Storage:* The large volume of generated data in a typical multimedia application brings in the associated challenge of efficient storage. Experimental setups are constrained in the available external storage, and the analysis can only be undertaken on the information that can be contained within the storage bound. This can be partly addressed by recent advances in memory technology, in which the cost per unit of storage volume is steadily decreasing. In addition, the logical structures, such as graph models, may be used [30] for efficient storage. Here, the nodes of the graph represent the events of a phenomenon that generate multimedia data, and the edges capture the temporal relationship between them. These graphs may be further classified into graphlets that link together relevant and in some cases, redundant data. By creating a hierarchy of such graphlets into a tree structure, storage and retrieval functions of the multimedia data are considerably simplified. The higher levels of the tree represent the general trend of the events, and this information gets progressively specific as the levels are traversed from top to bottom. If the storage space is limited, the hierarchy tree may be pruned to include few graphlets at each level, or entire levels may be removed altogether. It remains an open challenge to address the problem of efficient multimedia storage and efforts are underway to achieve this from both the device and an algorithmic point of view.

4) *Backchannel:* Apart from the wireless channels used for data retrieval and control messaging, there is a need for a separate backchannel for sending the performance evaluation data to the BS from possible onsite monitoring devices. As an example, the trace collector provides the QoS block with real-time network performance details. This transmission must be undertaken either in a wireless channel orthogonal to those used by the network or through wired connections. Devices that monitor the system must do so in a nonintrusive manner and not introduce additional interference. For this reason, the wired feedback channel is often preferred in testbeds and implemented through Ethernet or USB.

The testbeds that exist in the literature generally incorporate a subset of the above-mentioned blocks. Further, the WMSN itself may comprise levels of hierarchy. We next describe the existing single and

Table 3 An Overview of the Testbed Features of WMSNs

Testbed Name	Imaging Device	Pixels	Interfacing Mote	Additional Features
Meerkats [22]	Logitech Quick Cam Pro 4000	640 × 480	Stargate	Camera on-off switching for energy conservation
Explorebots2 [31]	XCam2	320 × 240	MICA2	Mobile robots Sonic sensors
Mobile Emulab [27]	Overhead Hitachi KP-D20A	768 × 494	Stargate MICA2	Mobile robot Remote Reconfiguration
IrisNet [33]	Logitech Quick Cam Pro 3000	640 × 480	Stargate	Network animator Highly scalable
SensEye [29]	Cyclops, CMUcam pan-tilt-zoom webcams	352 × 288	Stargate MICA2	3-4 camera tiers Boundary/motion detection and tracking
BWN Lab Testbed	Logitech Quick Cam Pro 4000	640 × 480	Stargate MICA2	Mobile robot

multiple-tier WMSN testbeds and their research abilities and challenges. A summary of the features of the testbeds is given in Table 3.

D. Single-Tier Testbeds

A visual sensor testbed is developed in [22], as part of the Meerkats project to measure the tradeoff between power efficiency and performance. It consists of eight visual sensor nodes and one information sink. The visual sensors are realized by interfacing the Crossbow Stargate boards with the Logitech QuickCam Pro 4000 Webcam, giving a resolution up to 640 × 480 pixels. With respect to multimedia processing, tests were performed with the objective of measuring the energy consumed during i) the idle times of the sensor with its communication circuits switched off, ii) processing involving high computation, iii) storage and retrieval functions, and iv) visual sensing by the Webcam. Results reveal that there is significant energy consumption in keeping the camera active, and writing the image to a Flash memory followed by switching the camera off conserves energy. There is also a finite instantaneous increase in the energy consumption due to state transients. Another observation pertains to the switching times. In their experimental setup, suspending and resuming the Webcam took nearly 0.21 and 0.259 s, respectively, thus indicating that these parameters cannot be neglected in protocol design for WMSNs. Interestingly, the processing-intensive benchmark results in the highest current requirement, and transmission is shown to be only about 5% more energy-consuming than reception.

Expandable, vision-, and sensor-equipped wireless robots with MICA sensor motes for networking are designed in the *Explorebots* architecture in [31]. The robots are equipped with a Rabbit 3000 programmable microprocessor with Flash memory, a 320 × 240 pixel XCam2 color camera, and a 11.1 V and 1500 mA-h lithium-polymer batteries to power the robot. There are also other custom-designed velocity and distance sensors, motor movement control, an in-built magnetic two-axis compass,

and sonic sensors. The target localization experiments on the testbed, composed of these mobile robots, uses the onboard multimedia sensors. Here, by processing the sound and light sensors outputs, the robots may be guided towards the target source. It may also communicate with other stationary MICA2 generating a specific acoustic tone or light source that signals the event area.

The Mobile Emulab [3] network testbed provides a remotely accessible mobile wireless and sensor testbed. Acroname Garcia Robots carry motes and single-board computers through an indoor field of sensor-equipped motes. All devices run a software that can be reconfigured and remotely uploaded by the user. A remote user can position the robots, control all the computers and network interfaces, run arbitrary programs, and log data in a database. The path of robots, which are also equipped with Webcams, can be planned, and a vision-based system provides positioning information with accuracy within 1 cm. Precise positioning and automation allows evaluating the effects of location and mobility on wireless protocols.

E. Multiple-Tier Testbeds

There is an increasing trend to leverage the capabilities of a heterogeneous WMSN so that the monitoring is undertaken at the optimal tradeoff between performance requirement and energy cost. This is seen in the use of different resolutions in the single camera [19], multiple cameras in the same node [20], and, finally, different cameras altogether within the same network [29], [32] (Fig. 5).

IrisNet (Internet-scale resource-intensive sensor network services) [32] is an example software platform for a heterogeneous WMSN testbed. Video sensors and scalar sensors are spread throughout the environment and collect potentially useful data. IrisNet allows users to perform Internet-like queries to video and scalar sensors that spread throughout the environment. The user views the sensor network as a single unit that can be queried through a high-level language, through a simple query statements

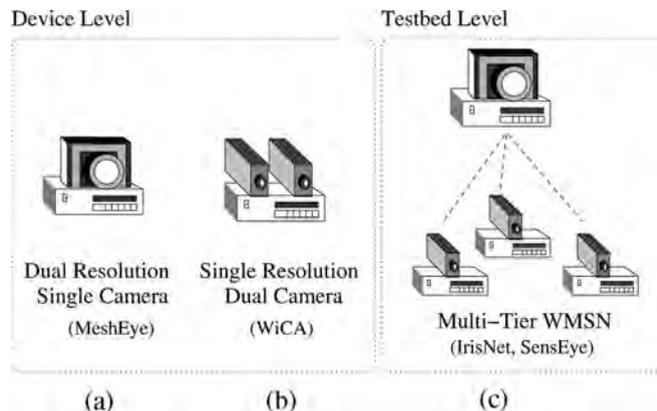


Fig. 5. The device level and testbed level heterogeneity in WMSNs is shown.

or more complex forms involving arithmetic and database operators. The architecture of IrisNet is two-tiered: heterogeneous sensors implement a common shared interface and are called *sensing agents* (SAs), while the data produced by sensors are stored in a distributed database that is implemented on *organizing agents* (OAs). Different *sensing services* are run simultaneously on the architecture. Hence, the same hardware infrastructure can provide different sensing services. For example, a set of video sensors can provide a parking-space finder service, as well as a surveillance service. Sensor data are represented in the Extensible Markup Language (XML), which allows easy organization of hierarchical data. A group of OAs is responsible for a sensing service, collects data produced by that service, and organizes the information in a distributed database to answer the class of relevant queries. Irisnet also allows programming sensors with filtering code that processes sensor readings in a service-specific way. A single SA can execute several such software filters (called *senselets*) that process the raw sensor data based on the requirements of the service that needs to access the data. After senselet processing, the distilled information is sent to a nearby OA.

In [29], the design and implementation of SensEye, a multiple-tier network of heterogeneous wireless nodes and cameras, is described for surveillance applications (Fig. 6). Each tier comprises nodes equipped with similar cameras and processing ability, with increasing resolution and performance at each stage. The lowest tier consists of low-end devices, i.e., MICA2 Motes equipped with 900 MHz radios interfaced with scalar sensors, e.g., vibration sensors. The second tier is made up of motes equipped with low-fidelity Cyclops [21] or CMUcam [33] camera sensors. The third tier consists of Stargate nodes equipped with Webcams that can capture higher fidelity images than tier 2 cameras. Tier 3 nodes also perform gateway functions, as they are endowed with a low-data-rate radio to communicate with motes in tiers 1–2 at 900 MHz and an 802.11 radio to communicate with the tier 3 Stargate

nodes. An additional fourth tier may consist of a sparse deployment of high-resolution high-end pan-tilt-zoom cameras connected to embedded PCs. The overall aim of this testbed is to efficiently undertake object detection, recognition and tracking by triggering a higher tier into the active state based on a need basis.

1) *BWN-Lab Testbed*: The WMSN-testbed at the Broadband Wireless Networking (BWN) Laboratory at Georgia Tech is based on commercial off-the-shelf advanced devices and has been built to demonstrate the efficiency of algorithms and protocols for multimedia communications through wireless sensor networks. The testbed is integrated with our scalar sensor network testbed, which is composed of a heterogeneous collection of Imote sensors from Intel and MICAz motes from Crossbow.

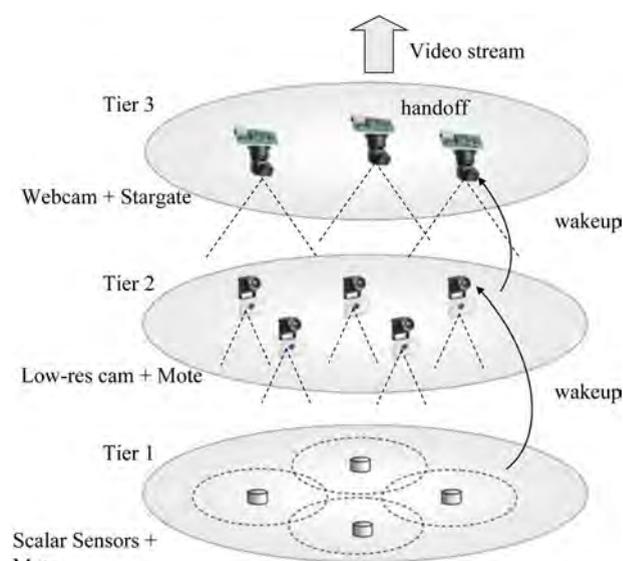


Fig. 6. The multiple-tier architecture of SensEye [29].



Fig. 7. Stargate board interfaced with a medium resolution camera. Stargate hosts an 802.11 card and a MICAz mote that functions as a gateway to the sensor network.

The WMSN testbed includes three different types of multimedia sensors: low-end imaging sensors, medium-quality Webcam-based multimedia sensors, and pan-tilt cameras mounted on mobile robots. Low-end imaging sensors such as CMOS cameras can be interfaced with Crossbow MICAz motes. Medium-end video sensors are based on Logitech Webcams interfaced with Stargate platforms (see Fig. 7).

The high-end video sensors consist of pan-tilt cameras installed on an Acroname GARCIA robotic platform,⁴ which we refer to as actor, and shown in Fig. 8. Actors constitute a mobile platform that can perform adaptive sampling based on event features detected by low-end motes. The mobile actor can redirect high-resolution cameras to a region of interest when events are detected by lower tier, low-resolution video sensors that are densely deployed, as seen in Fig. 9. The testbed also includes storage and computational hubs, which are needed to store large multimedia content and perform computationally intensive multimedia processing algorithms.

2) *Experiments at the BWN-Testbed:* We have developed software framework to perform experiments on the above-mentioned hardware. A sensor network consisting of MICAz devices is deployed in a building, scattered in several rooms. Each sensor runs a tinyos-based protocol

⁴<http://www.acroname.com/garcia/garcia.html>.

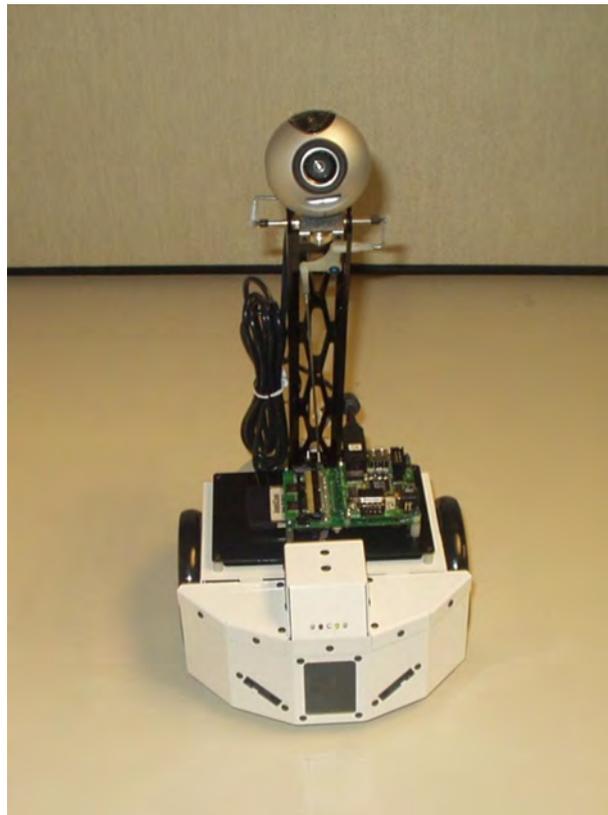


Fig. 8. Acroname GARCIA, a mobile robot with a mounted pan-tilt camera and endowed with 802.11 as well as 802.15.4 interfaces.

stack that allows it to create a multihop data path to the sink. Sinks are built on Stargate board that receive sensor information through a MICAz mote hosted on the



Fig. 9. GARCIA deployed on the sensor testbed. It acts as a mobile sink and can move to the area of interest for closer visual inspection. It can also coordinate with other actors and has built-in collision avoidance capability.

Stargate. There are two types of sinks, i.e., static sinks, which are basically video sensors connected to static Stargates as shown in Fig. 7, and mobile sinks or actors, where Stargate is mounted on the Acroname Garcia robotic platform, as shown in Fig. 8. Each sensor has identifiers that associate it with a specific room on the floor. Sensors periodically measure physical parameters in the room (light). If the value of measured light goes above a predefined threshold, the sensors start reporting the value by transmitting it to the sink through the multihop path.

When a packet containing light measurements reaches a sink, the MICAz sensor hosted on Stargate forwards the received packet to the serial port on Stargate. Stargate runs a daemon that listens to the serial port, decodes the packet received from the mote, and stores the values contained into an open source Structured Query Language (SQL) database (PostgreSQL⁵) running on Stargate.

A second daemon running on Stargate periodically queries the database and averages in time values of light stored in the database and from different sensors in the same room. If the value increases above a predefined threshold, the room is marked as requiring video monitoring. Sinks form an ad hoc network by communicating through AmbiCom's IEEE 802.11b Wave2Net Wireless Type I CompactFlash cards mounted on Stargate. The mobile sink that is closest to the room requiring intervention is identified. Based on a simplified map of the floorplan, the mobile platform moves to the room requiring intervention. Upon reaching the room, the camera mounted on the robot is activated. JPEG video is streamed through the wireless interface in quarter-CIF

format at 15 frames per second using a Linux-based open source utility for video streaming, camserv.⁶ The video stream is sent back to a remote laptop, possibly through a multihop path, and can be visualized on a graphical user interface on the laptop or stored for subsequent analysis. The robot keeps receiving light samples and returns to its initial position where the values of light in the room are below the threshold.

VI. CONCLUSIONS

We have discussed ongoing research on *prototypes and testbeds for experimental evaluation* of algorithms and protocols for the development of WMSNs. In particular, we have motivated the need for experimental research on wireless multimedia sensor networks to provide credible performance evaluation of existing protocols for wireless multimedia sensor networks. Then, we have discussed and classified existing applications for wireless multimedia sensor networks. We have then reviewed commercially available devices and existing research prototypes that will find applicability in conducting research on WMSNs. Finally, we have discussed examples of integration of heterogeneous devices in experimental testbeds and some successful examples in developing APIs and system software for WMSNs. ■

Acknowledgment

The authors would like to thank the anonymous referees, whose feedback greatly helped to improve the quality of this paper.

⁵<http://www.postgresql.org/>.

⁶<http://www.cserv.sourceforge.net/>.

REFERENCES

- [1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Netw. (Elsevier)*, vol. 51, no. 4, pp. 921–960, Mar. 2007.
- [2] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, pp. 71–83, Jan. 2005.
- [3] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau, "Mobile Emulab: A robotic wireless and sensor network testbed," in *Proc. of IEEE Conf. Comput. Commun. (INFOCOM)*, Barcelona, Spain, Apr. 2006.
- [4] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET simulation studies: The incredibles," *ACM Mobile Comput. Commun. Rev.*, vol. 9, no. 4, pp. 50–61, Oct. 2005.
- [5] J. Campbell, P. B. Gibbons, S. Nath, P. Pillai, S. Seshan, and R. Sukthankar, "IrisNet: An internet-scale architecture for multimedia sensors," in *Proc. ACM Multimedia Conf.*, 2005.
- [6] A. A. Reeves, "Remote monitoring of patients suffering from early symptoms of dementia," in *Proc. Int. Workshop Wearable Implantable Body Sensor Netw.*, London, U.K., Apr. 2005.
- [7] F. Hu and S. Kumar, "Multimedia query with QoS considerations for wireless sensor networks in telemedicine," in *Proc. Soc. Photo-Optical Instrum. Eng. Int. Conf. Internet Multimedia Manage. Syst.*, Orlando, FL, Sep. 2003.
- [8] H.-J. Yong, J. Back, and T.-J. Jang, "A stereo vision based virtual reality game by using a vibrotactile device and two position sensors," in *Proc. ACM Int. Conf. Comput. Graph. Interactive Tech. (SIGGRAPH)*, Boston, MA, 2006, p. 48.
- [9] M. Capra, M. Radenkovic, S. Benford, L. Oppermann, A. Drozd, and M. Flinham, "The multimedia challenges raised by pervasive games," in *Proc. the ACM Int. Conf. Multimedia*, Hilton, Singapore, 2005, pp. 89–95.
- [10] S. Benford, R. Anastasi, M. Flinham, C. Greenhalgh, N. Tandavanitj, M. Adams, and J. Row-Farr, "Coping with uncertainty in a location-based game," *Pervasive Comput.*, vol. 2, pp. 34–41, Jul.–Sep. 2003.
- [11] R. Holman, J. Stanley, and T. Ozkan-Haller, "Applying video sensor networks to nearshore environment monitoring," *Pervasive Comput.*, vol. 2, pp. 14–21, Oct.–Dec. 2003.
- [12] D. James, L. Klibanov, G. Tompkins, and S. J. Dixon-Warren, "Inside CMOS image sensor technology," in *Chipworks White Paper*. [Online]. Available: <http://www.chipworks.com/resources/whitepapers/Inside-CMOS.pdf>
- [13] Texas Instruments, "MSP430x1xx family user's guide (Rev. F)," in *Developer's Manual*, Feb. 2006.
- [14] Atmel Corp. "ATtiny24/44/84 automotive preliminary (Rev. B)," in *Datasheet*, Sep. 2007.
- [15] Intel Corporation, "Intel PXA255 processor," in *Developer's Manual*, Mar. 2003.
- [16] R. Kleihorst, B. Schueler, and A. Danilin, "Architecture and applications of wireless smart cameras (networks)," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Hilton, Singapore, Apr. 2007, pp. 1373–1376.
- [17] I. Downes, L. B. Rad, and H. Aghajan, "Development of a mote for wireless image sensor networks," in *Proc. Cogn. Syst. Interact. Sensors (COGIS)*, Paris, France, Mar. 2006.

- [18] Carnegie Mellon Univ., "CMUcam3 datasheet version 1.02," Pittsburgh, PA, Sep. 2007.
- [19] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "MeshEye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Cambridge, MA, 2007, pp. 360–369.
- [20] R. Kleihorst, B. Schueler, A. Danilin, and M. Heijligers, "Smart camera mote with high performance vision system," in *Proc. ACM SenSys Workshop Distrib. Smart Cameras (DSC)*, Boulder, CO, Oct. 2006.
- [21] M. Rahimi, R. Baer, O. Iroezji, J. Garcia, J. Warrior, D. Estrin, and M. Srivastava, "Cyclops: In situ image sensing and interpretation in wireless sensor networks," in *Proc. ACM Conf. Embed. Netw. Sensor Syst. (SenSys)*, San Diego, CA, Nov. 2005.
- [22] C. B. Margi, V. Petkov, K. Obraczka, and R. Manduchi, "Characterizing energy consumption in a visual sensor network testbed," in *Proc. IEEE/Create Net Int. Conf. Testbeds Res. Infrastruct. Develop. Netw. Commun. (TridentCom)*, Barcelona, Spain, Mar. 2006.
- [23] J. Hill, R. Szweczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," *ACM SIGPLAN Notices*, vol. 35, no. 11, pp. 93–104, 2000.
- [24] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language: A holistic approach to network embedded systems," in *Proc. ACM SIGPLAN 2003 Conf. Program. Lang. Design Implement. (PLDI)*, San Diego, CA, Jun. 2003.
- [25] S. Itoh, S. Kawahito, and S. Terakawa, "A 2.6 mW 2 fps QVGA CMOS one-chip wireless camera with digital image transmission function for capsule endoscopes," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2006.
- [26] S. Hengstler and H. Aghajan, "WiSNAP: A wireless image sensor network application platform," in *Proc. IEEE Int. Conf. Testbeds Res. Infrastruct. Develop. Netw. Commun. (TRIDENTCOM)*, Mar. 2006.
- [27] T. Teixeira, D. Lymberopoulos, E. Culurciello, Y. Aloimonos, and A. Savvides, "A lightweight camera sensor network operating on symbolic information," in *Proc. ACM Workshop Distrib. Smart Cameras*, Boulder, CO, 2006, pp. 76–81.
- [28] C. N. Taylor, D. Panigrahi, and S. Dey, "Design of an adaptive architecture for energy efficient wireless image communication," in *Proc. Embed. Process. Design Challenges: Syst., Architect., Model., Simul. (SAMOS)*, Samos, Greece, Jul. 2002, pp. 260–273.
- [29] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "SensEye: A multi-tier camera sensor network," in *Proc. ACM Multimedia*, Singapore, Nov. 2005.
- [30] A. Vakali and E. Terzi, "Multimedia data storage and representation issues on tertiary storage subsystems: An overview," *ACM SIGOPS Oper. Syst. Rev.*, vol. 35, no. 2, pp. 61–77, 2001.
- [31] T. A. Dahlberg, A. Nasipuri, and C. Taylor, "Explorebots: A mobile network experimentation testbed," in *Proc. ACM SIGCOMM Workshop Exper. Approach. Wireless Netw. Design Anal. (E-WIND)*, Philadelphia, PA, 2005, pp. 76–81.
- [32] S. Nath, Y. Ke, P. B. Gibbons, B. Karp, and S. Seshan, "A distributed filtering architecture for multimedia sensors," Intel Res. Tech. Rep. IRP-TR-04-16, Aug. 2004.
- [33] A. Rowe, C. Rosenberg, and I. Nourbakhsh, "A low cost embedded color vision system," in *Proc. IEEE/RJS Int. Conf. Intell. Robots Syst. (IROS)*, Lausanne, Switzerland, Oct. 2002.

ABOUT THE AUTHORS

Ian F. Akyildiz (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981, and 1984, respectively.

Currently, he is the Ken Byers Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, and Director of the Broadband Wireless Networking Laboratory. Since June 2008 he is an Honorary Professor with the School of Electrical Engineering at the Universitat Politècnica de Catalunya, Barcelona, Spain. He is Editor-in-chief of *Computer Networks (Elsevier) Journal*, founding Editor-in-Chief of both journals, *Ad Hoc Networks (Elsevier)* in 2003 and *Physical Communication (Elsevier)* in 2008, respectively. His current research interests are in next-generation wireless networks, sensor networks, and wireless mesh networks.

Prof. Akyildiz is a Fellow of the Association for Computing Machinery (ACM). He received the Don Federico Santa Maria Medal for his services to the Universidad de Federico Santa Maria in 1986. From 1989 to 1998, he was a National Lecturer for ACM and received the ACM Outstanding Distinguished Lecturer Award in 1994. He received the 1997 IEEE Leonard G. Abraham Prize Award (IEEE Communications Society) for his paper entitled "Multimedia Group Synchronization Protocols for Integrated Services Architectures" published in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in January 1996. He received the 2002 IEEE Harry M. Goode Memorial Award (IEEE Computer Society) "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking." He received the 2003 IEEE Best Tutorial Award (IEEE Communication Society) for his paper entitled "A Survey on Sensor Networks," published in IEEE COMMUNICATIONS MAGAZINE in August 2002. He received the 2003 ACM Sigmobile Outstanding Contribution Award "for pioneering contributions in the area of mobility and resource management for wireless communication networks." He received the 2004 Georgia Tech Faculty Research Author Award for his "outstanding record of publications of papers between 1999-2003." He received the 2005 Distinguished Faculty Achievement Award from School of Electrical and Computer Engineering, Georgia Institute of Technology.



Tommaso Melodia (Member, IEEE) received the Laurea degree in telecommunications engineering and the doctoral degree in information and communication engineering from the University of Rome "La Sapienza," Rome, Italy, in 2001 and 2005, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2007.

He is an Assistant Professor with the Department of Electrical Engineering, University at Buffalo, The State University of New York, where he directs the Wireless Networks and Embedded Systems Laboratory. He is the author of about 40 publications in leading conferences and journals on wireless networking. He is an Associate Editor for *Computer Networks Journal* and *Hindawi Journal of Sensors*. He is a member of the Technical Program Committee of several leading conferences in wireless communications and networking. His current research interests are in wireless multimedia sensor and actor networks, underwater acoustic sensor networks, and cognitive radio networks.

Prof. Melodia received the Georgia Tech BWN Lab Researcher of the Year award for 2004.



Kaushik R. Chowdhury (Student Member, IEEE) received the B.E. degree in electronics engineering (with distinction) from Veermata Jijabai Technological Institute, Mumbai University, India, in 2003 and the M.S. degree in computer science from the University of Cincinnati, Cincinnati, OH, in 2006. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta.

He is a Research Assistant with the Broadband Wireless Networking Laboratory, Georgia Tech. His current research interests include multichannel medium access protocols, dynamic spectrum management, and resource allocation in wireless multimedia sensor networks.

Mr. Chowdhury received the Outstanding Thesis award from the ECECS Department, University of Cincinnati, for the year 2006.

