Volume 3, Issue 2, February 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Comprehensive Wireless Sensor Node Structural Design

Jitendra Joshi, Keshav Dev Gupta
Research Scholar
Department of Computer Science and Engineering
Jayoti Vidyapeeth Women's University
Rajasthan, India

Nisha Singh, Priya, Jaya Sachan
M.Tech Student
Department of Computer Science and Engineering
Jayoti Vidyapeeth Women's University
Rajasthan, India

Abstract— The focus of this research paper is to develop a general structural design that addresses the needs of wireless sensor node. The structural design is not tied to any particular radio or processing technology but rather details how the computation and communication should brought together. To communicate over a wireless link, protocols must be built up from the raw electro-magnetic signalling primitives. There are three key issues that must be addressed by the system architecture such as Concurrency, Flexibility, and Synchronization.

Keywords— Wireless Sensor Node (WSN), Radio Frequency (RF), Communication, Protocol, Processing, Decoupling.

I. INTRODUCTION

A key step in exploring the challenges a building of low-power wireless communication platform is in understanding the system requirements for the communication mechanisms necessary to construct the multi-hop networks that are envisioned [1]. The focus of this research paper is to develop a general architecture that addresses the needs of wireless sensor network. The structural design is not tied to any particular radio or processing technology but rather details how the computation and communication should brought together. It is designed to have communication subsystem that allows for flexible, application specific optimization of communication protocols while simultaneously obtaining high bandwidth and efficiency. To set the stage for the design, we first describe the basic operations associated with communication across a radio link. This exposes several of the core design challenges including the concurrency intensive nature of wireless sensor node. To address the set of core challenges presented we develop a comprehensive structural design for a wireless sensor node. In this paper we analyse specific implementations of this general structural design.

II. WIRELESS COMMUNICATION REQUIREMENTS

To communicate over a wireless link, protocols must be built up from the raw electro-magnetic signalling primitives. A transmitter must carefully modulate the radio frequency (RF) carrier while receiver performs demodulation and signal analysis. Figure 1 illustrates the key phases of a packet-based wireless communication protocol. It is important to note that many of the operations must be performed in parallel with each other. This can be seen in the distinct layers that overlap in time.

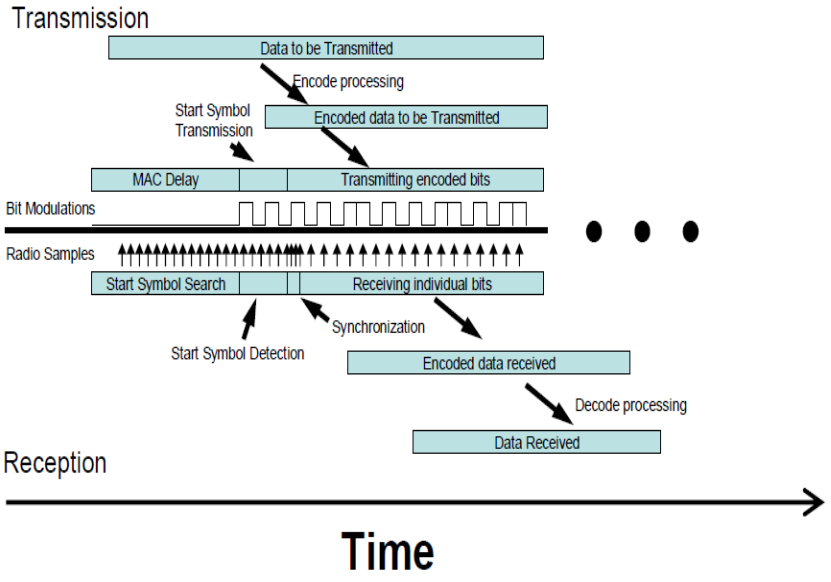

Figure 1: Phases of wireless communication for transmission and reception.

The first step in the communication process is to encode the data for transmission. The coding schemes are designed to increase the probability of a successful transmission by preventing and correcting slight errors. For efficiency reason the encoding process is pipelined with the actual transmission process. Once the first byte is encoded, transmission may begin. The remaining bytes can be encoded as preceding bytes are transmitted.

Coding schemes can range from simple direct connect (DC) balancing schemes, such as 4b-6b or Manchester encoding, to complex code division multiple access (CDMA) schemes [2]. In either, a collection of one or more data bits, called a symbol, are coded into a collection of radio transmission bits called chips. Manchester encoding has two chips per symbol which represents 1 bit of data. Direct sequence spread spectrum and code division multiple access (CDMA) schemes often have 15 to 50 chips per symbol with each symbol containing 1 to 4 bits of data.

The actual transmission begins with the execution of a media access control protocol (MAC). MAC protocols are designed to allow multiple transmitters to share a single communication channel. One of the simplest MAC protocols is carrier sense media access (CSMA) where each transmitter first checks for an idle channel prior to each transmission [3]. If the channel is busy, it waits for a short, random, delay after which it reinitiates the transmission.

III. KEY ISSUES ARCHITECTURE MUST ADDRESS

There are three key issues that must be addressed by the system architecture. First, from our basic description of a wireless communication protocol we can see that several operations must occur in parallel. The channel must be continually monitored, data must be encoded, and bits must be transferred to the radio. The ability to deal with fine grained concurrency is required in order to perform these operations in parallel [4].

Secondly, the system must be flexible to meet the wide range of target application scenarios. Third, the architecture must provide precise control over radio transmission timing. This requirement is driven by the need for ultra-low power communication in the data collection application scenario. Finally, the system must be able to decouple the data path speed and the radio transmission rate. We show how a direct coupling between processing speed and communication bit rates can lead to sub-optimal energy performance.

A. Concurrency

The structural design must provide an efficient mechanism to handle fine-grained concurrency. In both the transmission and reception case, the computations associated with wireless communication must occur in parallel with application-level processing and potentially with intermediate level packet processing. Sensor events and data calculations must continue to proceed while communication is in progress [5]. This is particularly true when considering overlapping start-symbol detection and application processing.

For example, start symbol detection must be performed continually when waiting for incoming communication. If any other processing is to be performed in must be done in parallel with the start symbol search. In addition to the parallel operation of protocol processing and application-level processing, we it is also required that several different steps of protocol processing must be performed in parallel.

B. Flexibility

The structural design must provide the flexibility to support a wide range of application scenarios. For deeply embedded systems, it is important that a system is flexible enough to support a wide range of application-specific protocols [2]. Unlike cell phones, wireless local area networks, or Bluetooth devices, wireless senor networks do not have a fixed set of communication protocols that they must adhere to. Deeply embedded wireless networks can exploit tradeoffs between bandwidth, latency, and in network processing to reduce power consumption by order of magnitude.

For example: if sensor data is sampled only once per minute, it might be acceptable to delay transferring the data for several seconds, allowing the network to coordinate many such flows efficiently while operating at a low duty cycle. Such ‘optimizations can extend battery life from weeks to years.

In addition to supporting flexible communication protocols, it is also important to support flexible interfaces between protocols and applications [6]. A rich interface between application processing and communication processing allows programmers to create arbitrary protocol decompositions. Internal protocol state can be exposed up to applications and applications can have fine-grained control over the protocols.

For example: The need for flexible protocols that allow applications to view the underlying channel characteristics can be seen in [7]. They discovered that many 802.11 cards abstract away signal quality from application level processing which eliminates their ability to support radio frequency (RF) localization applications. Had the protocols for interfacing between applications and the communications hardware been flexible, that information could have been easily exposed.

C. Synchronization

The structural design must provide the ability to achieve precise node-to-node synchronization and give the application direct control of radio timing. This plays a critical role in many wireless sensor network applications. Our data collection application scenario relies on time synchronization to schedule the child to parent communication windows and the temporal accuracy of data collection [8]. The short duration and frequency of the transmission window makes it important for the application to have precise temporal control over the radio.

This is in contrast to the communication patterns of a cell phone where communication occurs infrequently such as a few calls per day and for long durations such as for minutes per call. A 10 ms delay before a cell phone call starts or stops has

a negligible impact on the user experience. However if a packet is just 2 ms late in our data collection application scenario, it will be lost.

To allow applications to achieve precise time synchronization, radio communication protocols must provide exact timing information as to when data packets are received. Additionally, they must allow the application to precisely control when radio communication begins as well as when the radio is switched on or off for reception

IV. DECOUPLING BETWEEN RADIO FREQUENCY (RF) AND PROCESSING SPEED

Finally, the system structural design must allow a decoupling between radio frequency (RF) transmission rates and processing speed. A radio is the most efficient when data transmissions occur at its maximum transmission rate. When transmitting with a fixed power, reducing the transmission time reduces the energy used. However, modern studies in low power processor design and dynamic voltage scaling have shown that processors are the most efficient when they spread computation out in time as much as possible so that they can run at the lowest possible voltage [9, 10]. Ideally, for power consumption reasons, we would want the controller to perform all calculations as slowly as possible and just as the computation is complete, the radio would burst out the data as quickly as possible.

Coupling the speed of the microcontroller to the data transmission rate forces both pieces of the system to operate at non-optimal points. This coupling can be seen in the design of a wireless sensor node where a balance must be formed between the speed of the controller and the speed of the radio. To dig into this requirement further we can analyse a performance shortcoming of the Rene platform. On the Rene platform, only 10% of the radio's 115Kbps capability is used. Increasing the clock speed of the microcontroller could easily increase the radio transmission speed because the speed of Programmed I/O is directly proportional to the data path speed. If the clock speed of the microcontroller were doubled, the resulting transmission rate would be 20Kbps and the energy consumed per bit by the radio would decrease by 50%. However, the speed increase from 4MHz to 8MHz increases the power consumption of the microcontroller from 15mW to 50 mW according to the AT90LS8535 data sheet [11]. Additionally, the idle mode power consumption would increase from 9mW to 28 mW. The savings in radio transmission power is quickly offset by the increase in the controller's power consumption.

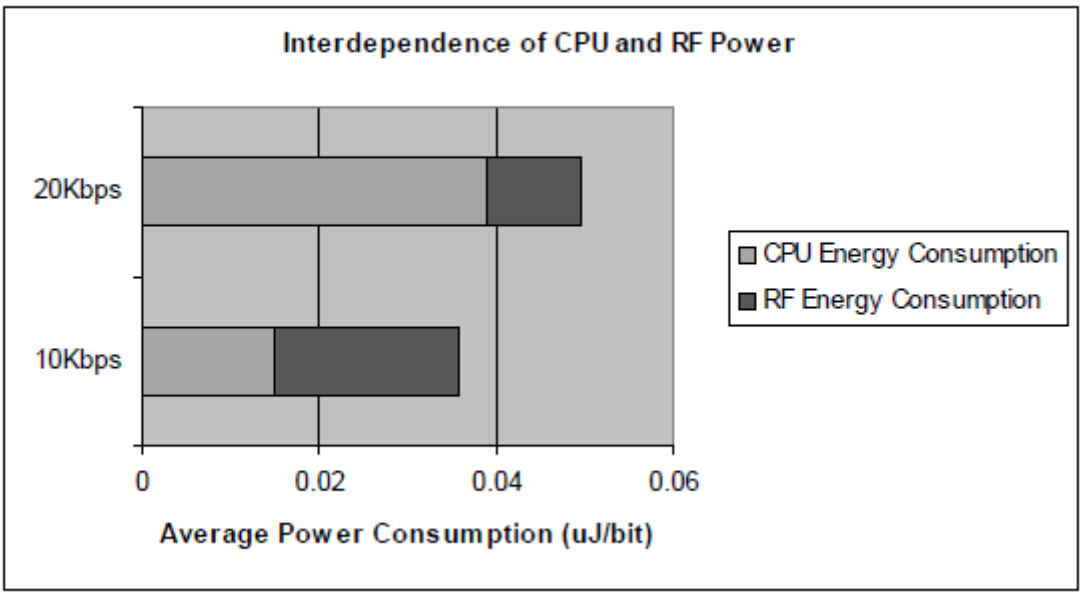

Figure 2: Energy cost per bit when transmitting at 10Kbps and 50Kbps broken down into controller and radio frequency (RF) energy.

To make this concrete, we consider an application where a device is transmitting 100% of the time, doubling the transmission rate would allow the device to spend 50% of its time in the idle mode. With a radio transmission consuming 21mW, the average power consumption of the improved system would be approximately .5*21mW+.5*50mW+.5*28mW or 49.5 mW. On the other hand, our original system only consumed an average of 36 mW. While expecting a decrease in power consumption, we see a net increase.

To address this issue, the system architecture must provide a mechanism for decoupling the transmission rate from the data path speed. This observation provides an indication that there needs to be dedicated special-purpose resources for interacting with the radio. Special purpose hardware can operate at frequencies that are optimal for the radio, while the general-purpose controller is tuned to the optimal rate for computation. Through buffering the special-purpose hardware can perform a rate matching between the general-purpose data path and the radio. The amount of buffering sets the level of decoupling. If the entire packet were buffered the processor speed could be scaled to just meet the packet processing requirements set by the application scenario.

At first the need to decouple the transmission speed from the processing speed seems at odds with the goal of allowing applications precise, direct and flexible control over the radio communications. However, a processor supported by specialized hardware accelerators has the ability to do both.

V. CONCLUSION

The objectives of this paper to provide the structural design wireless sensor node. A transmitter must carefully modulate the radio frequency (RF) carrier while receiver performs demodulation and signal analysis. In the actual transmission begins with the execution of a media access control protocol (MAC). In both the transmission and reception case, the computations associated with wireless communication must occur in parallel with application-level processing and potentially with intermediate level packet processing. In this used of three key issues that must be addressed by the system architecture. The structural design must provide an efficient mechanism to handle fine-grained concurrency. The structural design must provide the flexibility to support a wide range of application scenarios. The structural design must provide the ability to achieve precise node-to-node synchronization and give the application direct control of radio timing.

REFERENCES

1. Hill, J., System Architecture for Wireless Sensor Networks. 2000: Masters Report, University of California Berkeley.
2. Gay, D., et al., The nesC Language: A Holistic Approach to Networked Embedded Systems. 2003: Programming Language Design and Implementation(PLDI).
3. Ye, W., J. Heidemenn, and D. Estrin, An Energy-Efficient MAC Protocol for Wireless Sensor Networks. 2001.
4. Hill, J., et al. System architecture directions for networked sensors. in ASPLOS 2000. 2000.
5. Scott, M.D., An Ultra-Low Power ADC for Distributed Sensor Networks. 2002: Master of Science Theses, UC Berkeley.
6. Ms. Sunita Jyoti Malik Suman Mor, Comprehensive Study of Applications of Wireless Sensor Network, IJARCSSE, Volume 2, Issue 11, PP56-60, November 2012.
7. Bahl, P. and V. Padmanabhan, RADAR: An in-building RF-based user location and tracking system. IEEE Infocom, 2000. 2: p. 775-784.
8. Eiko Yoneki, J.B., A Survey of Wireless Sensor Network Technologies: Research Trends and Middleware'S Role.2005, University of Cambridge: Cambridge. p. 45.
9. Chandrakasan, A.P., S. Sheng, and R.W. Brodersen, Low Power CMOS Digital Design. 1992, University of California Berkeley.
10. Pering, T., T. Burd, and R. Brodersen, The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms. 1998.
11. Atmel Corporation, AT90S/LS8535 Datasheet. 2001: http://www.atmel.com/atmel/acrobat/doc1041.pdf.

