DESIGN AND DEVELOPMENT OF ADVANCED TRANSCEIVER UNIT FOR WIRELESS MOBILE SENSING SYSTEMS *

Daniel Doonan, Ronald Iltis, Hua Lee and Ryan Kastner
Department of Electrical and Computer Engineering
University of California,
Santa Barbara, CA 93106-9560
E-mail: {doonan,iltis,hualee,kastner}@ece.ucsb.edu

ABSTRACT

Sensor technology is continually advancing to meet demands of a wide range of potential applications. Many of these applications could be better served by distributed sensing than by traditional centralized sensing. To support these emerging applications, it is important to design and develop a unified framework for communication and network infrastructure capable of supporting various sensing functions. A research prototype operating in the 915 MHz Industrial, Scientific, and Medical band (ISM band) has been developed as potentially the core component of this infrastructure. In this paper, we will present the design and optimization of the system, data processing procedures, system parameters, network protocols, and experimental results.

KEYWORDS

ISM Band, Frequency Hopping, Ad Hoc Networking

INTRODUCTION

There is currently great interest in applying wireless distributed sensor networks (DSNs) to a broad range of problems in telemetry, surveillance, and control. Past research efforts have been largely focused on sensor development, while conventional wireless network solutions are generally lacking in appropriate features[1]. In order to realize the full potential of distributed sensing, a more unified and flexible approach is necessary.

*This work was supported in part by grants from Cubic Defense Systems, UC MICRO and the International Foundation for Telemetering.

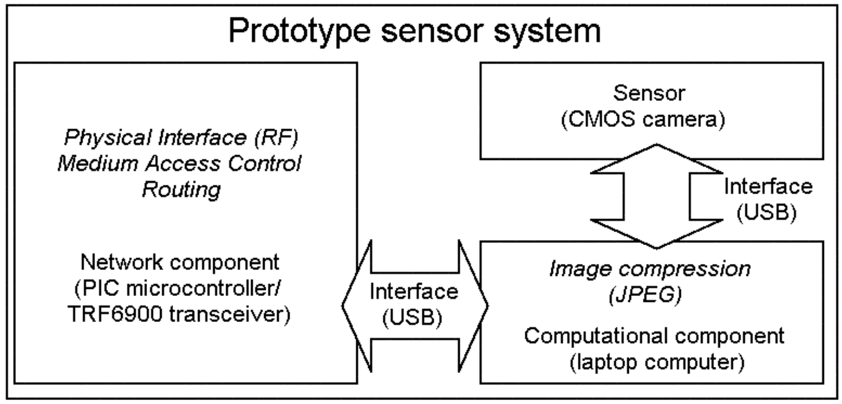

Figure 1: Prototype sensor unit block diagram. *Note: see Figure 3 for network component detail.*

NODE PROPERTIES

We recognize that the network component can be separated from the computational component of the smart sensor, as shown in Figure 1. This is an example of orthogonalization of concerns [2]. Thus, a focus on the networking component should allow the production of a transceiver unit with wide applicability. Accordingly, our prototype sensor unit uses a laptop computer whereas application devices would likely use a Digital Signal Processor (DSP) for computation. At present we merely compress the image data before transmission, whereas an actual sensor would perform operations such as feature extraction and event classification. In the context of mobile communications, many of the traditional problems of wireless networks (link or node dropout, splits, joins) are exacerbated. Additionally, conditions can change relatively quickly, potentially over timescales of minutes or seconds as opposed to hours or days [3]. Under these conditions, it is not feasible to use master/slave or tree architectures, since they require substantial initialization and use nonredundant paths. Accordingly, it is desirable to use a nonhierarchical design [4].

Every transceiver unit will also act to relay messages between other sensors, allowing the creation of a homogeneous network, where any node may act as a source and/or sink of data as well as routing it; protocols for the latter are already emerging [5],[6],[7]. This implies that queries to the network may be posed at any node, which is a postulate of distributed computation based on sensed data. Homogeneity is also important since the use of relatively simple local routing strategies [8] can then produce robust scalable dynamic networks as well as simplifying the system design by reducing variability.

MEDIUM ACCESS CONTROL (MAC)

Most available networking protocols use time-division multiple access (TDMA) since it can reduce both collisions and power consumption [4],[9],[10],[11], but at the expense of requiring network synchronization. In a network of the type described here, there is no way to transmit a time signal to all nodes simultaneously. Some nodes may have accurate clocks (e.g. Global

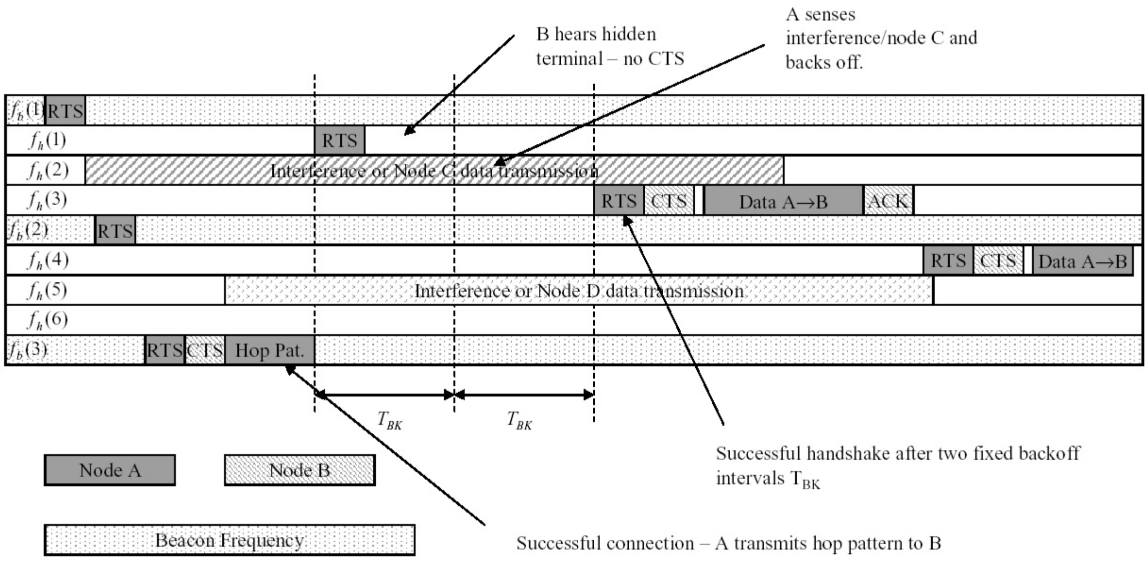

Figure 2: AAFH protocol example with 3 beacon frequencies and 6 data frequencies.

Positioning System receivers) but this cannot be depended upon. Synchronization is still possible using beacon and/or control signals, but such transmissions require a possibly appreciable fraction of the network capacity.

The use of Adaptive Asynchronous Frequency Hopping (AAFH) eliminates the need for TDMA, while simultaneously minimizing hidden terminal problems and fading. Adaptive hopping uses a small number of beacon frequencies for making RTS-CTS-based connections, and fixed backoff timers to enable joint receiver-transmitter (R-T) hopping when carrier sensing detects interference. The asynchronous nature of AAFH makes it a good candidate protocol for ad-hoc wireless networks where TDMA is difficult to implement.

An example of the AAFH protocol timeline is shown in Figure 2, where there are $N_b = 3$ beacon frequencies $\{f_b(1), f_b(2), f_b(3)\}$, and $N_h = 6$ data hops $f_h(1), \ldots, f_h(6)$. Node A has packets to send to Node B, and hence transmits a RTS first on $f_b(1)$. However, Node B is currently receiving in idle mode on hop $f_b(3)$, and only receives the RTS on A's third hop. Node B then transmits a CTS, and A responds with the data hopping pattern $\{f_h(1), f_h(2), \ldots, f_h(6)\}$. At this point, both nodes start the fixed backoff timer with duration T_{BK}. Node A transmits the first data RTS on $f_b(1)$, but B hears a hidden terminal, and does not sent CTS. Nodes A and B having failed to connect, then jump to the next data hop $f_h(2)$ when their backoff timers T_{BK} expire. However, on $f_h(2)$, Node A detects interference while carrier sensing, and does not transmit RTS. Again, both nodes started a backoff timer at the beginning of hop $f_h(2)$, and after T_{BK} sec. together hop to frequency $f_h(3)$. Finally, A and B exchange a RTS-CTS handshake on $f_h(3)$ and A transmits the first data packet. An ACK from B triggers a jump to the next hop $f_h(4)$, and the data transmission procedure repeats.

The operating band extends from 902 to 928 MHz, for a bandwidth of 26 MHz. If 400 kHz channel spacing is used (conservative for a 100kbps system [12]) we obtain 65 channels. Thus,

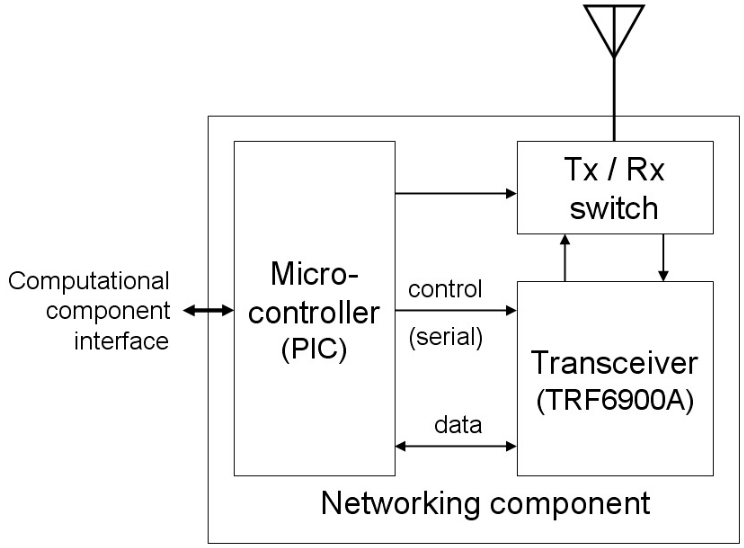

Figure 3: Networking component block diagram as implemented in prototype.

even with several channels reserved for beacon use and others unusable due to interference, the remaining channels will be sufficient to support a relatively high throughout. Spatial reuse of frequencies will occur automatically; also, reduction of transmit power can be implemented on low-loss (short) links to minimize interference.

The present system uses uniform pseudorandom channel selection. Ranking channels to favor those which are more reliable could improve performance and may be implemented if simulation results indicate it will. This approach depends on the assumption that interference will be sustained over time on some subset of frequencies whose use can then be reduced. Transmitters can then compile lists of frequencies with high incidences of interference or corrupted transmissions, and weight their choice of frequency sets for transmission by this data.

PACKET CONSTRUCTION

The data format for the prototype system was based on that of [13] and will be revised to improve network capability and forward compatibility. Several features of the transmission format are required by the transceiver IC [14]. The basics of the data packet format in use are shown in Figure 4.

Each transmission mustbegin with a preamble which has no DC component. This is required for the proper operation of the demodulation circuitry. Following the preamble is the information-bearing part of the packet. This consists of headers, data, and a checksum. The headers contain a transmission ID code, format version info, and datatype specifiers. The data section contains 64 bytes, some or all of which may be meaningful data as specified by the header. The packet ends with a 2 byte checksum, implemented as a standard cyclic redundancy checksum of 16 bits (CRC16).

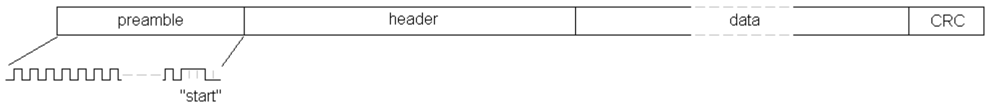

Figure 4: Packet format diagram.

RECEIVER SYNCHRONIZATION

The first part of the synchronization algorithm operates as a very simple correlator in the method of [12]. During the packet preamble, alternating ±1 symbols are transmitted to yield a DC-free sequence. The end of the preamble contains a longer +1 ”start” symbol so that the beginning of the data part can be detected easily. This will insure initial synchronization of the symbol clocks to within the edge jitter time. However, the transmitter and receiver clocks will not run at exactly the same speed, so they will gradually skew apart.

The clock drift is a relatively slow process, but important over the packet length. Correct symbol recovery in spite of it requires tight oscillator tolerance, small packet size, or repeated synchronization. The first two options are unsatisfactory, as the prototype nodes had mismatched oscillators, yielding a significantly greater clock drift than would occur in production models. This would have limited the allowed interval between re-synchronizations to less than 6 bytes. Adding additional synchronization symbols to the packet is possible, but inelegant.

The symbol clock, however, is present in the data to a significant extent, in the sense that the expectation of a transition is small everywhere except at symbol boundaries. Therefore, comparing the actual and predicted times of a symbol transition will show whether the receiver clock is ahead of or behind the received stream. The receiver clock is then adjusted slightly in the appropriate direction, yielding a software phased-locked loop. This technique has been proven to correct abnormally large clock drift, and consumes few enough processor resources to be practical.

RESULTS

The prototype pair of transceivers (one shown in Figure 5) successfully carried fairly high-rate data in the form of a series of JPEG images produced by a PC camera, effectively providing low-quality video (0.8 frame per second at 160x120 pixel resolution). This link operated at only 57.6 kbit/s due to the filter settings on the transceiver evaluation board - adjusting appropriate component values would have allowed operation at the tentative goal rate of 100 kbit/s. Uncorrected error rate averaged ≈ 0.5 bits per packet even on links approaching dropout, due to the ”threshold” effect of FSK deodulation. The CRC16 is guaranteed to correct 1 and detect 2 bit errors, and on a packet this short ($< 1\%$ of the checkable length) is highly likely to detect many more; thus, the system can be considered reliable in delivering error-free data.

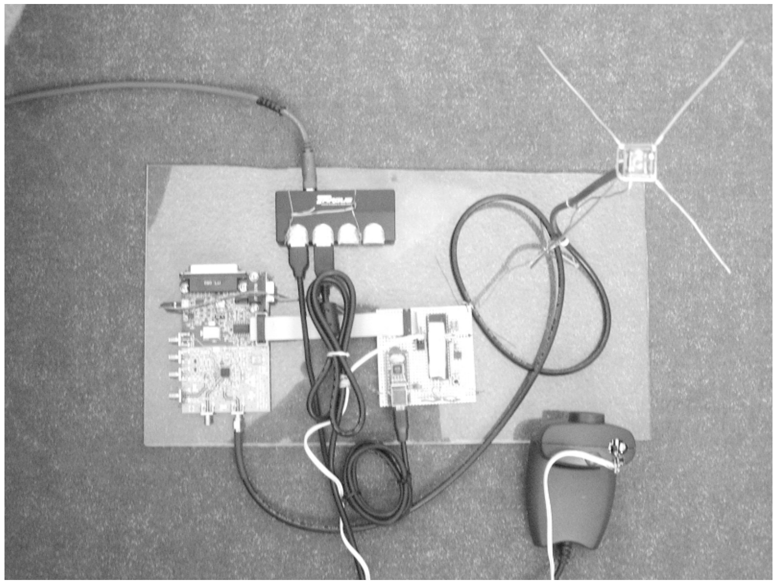

Figure 5: TRF6900 testbed, with USB hub, PIC microcontroller, and camera (lower right).

CONCLUSIONS

This paper presents a system and board-level design for an adaptive frequency hopping radio that is targeted for sensing and telemetry applications. The PIC microcontroller permits flexible control of the TRF6900 transceiver carrier/hopping frequency, data rate and synchronization. An adaptive asynchronous frequency-hopping (AAFH) MAC protocol has also been presented that eliminates the need for network synchronization in wireless sensor networks. Future work will focus on implementation of AAFH in the PIC microcontroller / TRF6900-based radio, and demonstration of sensor/telemetry data transmission.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, pp. 102–114, Aug. 2002.

[2] K. Keutzer, S. Malik, R. Newton, J. Rabaey, and A. Sangiovanni-Vincentelli, "System level design: Orthogonalization of concerns and platform-based design," *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, vol. 19, Dec. 2000.

[3] K. Almeroth, K. Obraczka, and D. DeLucia, "A lightweight protocol for interconnecting heterogeneous devices in dynamic environments," in *IEEE International Conference on Multimedia Computing and Systems*, (Florence, ITALY), June 1999.

[4] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, pp. 16–27, Oct. 2000.

[5] C. Perkins and E. M. Royer, “Ad hoc on-demand distance vector routing,” in *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, (New Orleans, LA), pp. 99–100, Feb. 1999.

[6] C. Tschudin and R. Gold, “Lunar lightweight underlay network ad-hoc routing,” tech. rep., Dept. of Computer Systems, Uppsala University, Uppsala, Sweden, Jan. 2002. http://www.docs.uu.se/docs/research/projects/selnet/lunar/.

[7] C.-K. Toh, M. Delwar, and D. Allen, “Evaluating the communication performance of an ad-hoc wireless network,” *IEEE Transactions on Wireless Communications*, pp. 402–414, July 2002.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *Proceedings of the Sixth Annual Intern’l Conference on Mobile Computing and Networking (MobiCom 2000)*, (New York, NY), pp. 56–67, 2000.

[9] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” *Communications of the ACM*, vol. 43, pp. 51–58, May 2000.

[10] Z. Tang and J. Garcia-Luna-Aceves, “A protocol for topology-dependent transmission scheduling in wireless networks,” in *Proceedings of IEEE WCNC*, (New Orleans, LA), pp. 1333–1337, Sept. 1999.

[11] I. Chlamtac, A. Faragó, A. D. Meyers, V. Syrotiuk, and G. Záruba, “Performance comparison of hybrid and conventional MAC protocols for wireless networks,” in *Proceedings of VTC 2000*, pp. 201–205, May 2000.

[12] C. Bohren, M. Loy, and J. Schillinger, *Designing With the TRF6900 Single-Chip RF Transceiver*. Texas Instruments, 2001. http://focus.ti.com/lit/an/swra033d/swra033d.pdf.

[13] P. Spevak, *Implementing a Bidirectional, Half-Duplex FSK RF Link With TRF6900 and MSP430*. Texas Instruments, 2001. http://focus.ti.com/lit/an/slaa121/slaa121.pdf.

[14] Texas Instruments, *TRF6900A SINGLE-CHIP RF TRANSCEIVER*, 2001. http://focus.ti.com/lit/ds/symlink/trf6900a.pdf.