# DESIGN OF A SENSOR PROTOCOL SUITE FOR VLSI IMPLEMENTATION

Thomas Hammel and Mark Rich
Fantastic Data
San Francisco, California

Charles J. Graff
U.S. Army RDECOM-CERDEC
Ft. Monmouth, New Jersey

## ABSTRACT

*This paper describes the design and development of an integrated, VLSI based node for use in distributed sensor networks. The objectives of the effort include the design, simulation, and development of an energy aware protocol stack that scales to thousands of nodes, and is easily implemented in commercial VLSI chips. Each node in the sensor network includes a passive infrared sensor for target detection and tracking, a GPS receiver, a flexible, programmable System on a Chip processing element, and a commercial ISM band transceiver. The software supports target detection and tracking, energy efficient network initialization, medium access, and routing protocols to provide multi-hop packet communication. This paper provides a description of the hardware platform, the software environment, and the simulation results for a typical 100 node sensor deployment.*

## MILITARY SENSOR NETWORKS

Very large networks of small, semi-autonomous sensors are required for future combat operations in many arenas—for situation awareness, intelligence preparation of the battlefield, control of semi-automatic munitions, and command and control. Sensor nodes must be small, cheap, plentiful, and long lasting. The sensor node must include sensing, processing, and communication elements. The resulting sensor network will be very large and self organizing. It must be able to accept tasking from outside authority, develop high-level data products, and communicate the results to the outside authority. In many situations the military sensor network must be capable of carrying out its tasking autonomously and storing the results, because the communication link to the tasking authority operates intermittently.

Many companies want to deploy large sensor networks to monitor physical plants, equipment, and stock. Sensors for these commercial networks must also be small, cheap, plentiful, and long lasting, however, the processing and network requirements are not as severe as for military networks. Commercial sensors are often hand emplaced and may be supported by larger infrastructure nodes strategically placed to organize and control the network. Systems postulated for commercial use are often simple data collection systems without sophisticated signal processing that report infrequently to a defined central data repository, unlike the military systems that may include autonomous, in-network processing to deliver real-time alerts.

A number of companies have developed communication products for sensor networks, but have not developed complete nodes. While many of these partial products are quite small and long lasting, it is unlikely that they will meet the more severe requirements of a complete sensor node, especially in the areas of real-time reporting, in-network processing, network formation, and mesh data distribution.

The deployment of large scale sensor networks is prevented because of lack of effective protocol support, network spectrum capacity limits, energy consumption, node size, and cost. An integrated hardware and protocol suite, capable of supporting 1000+ nodes, flexible to adjust to different requirements, and suitable for inexpensive very large scale integrated (VLSI) circuit implementation, is required to realize these sensor network goals.

In this paper, we describe the design, simulation, and testing of such an integrated suite carried out under a U.S. Army SBIR project.

## APPROACH

The major components of a sensor node are shown in Figure 1. As these components and their usage are interrelated, we are pursuing a whole system approach to reducing total node energy consumption. A complete protocol suite that simultaneously applies low duty cycle

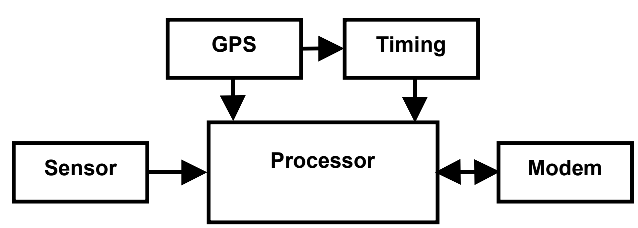


Figure 1. The major components of a sensor node. Software that carefully manages the components lowers overall node energy consumption.

| | |
|---|---|
| Sensor calibration time | 1-10 s |
| Target speed | 5 m/s |
| Node spacing | 10-30 m |
| Node location accuracy | 1-5 m |
| Critical data latency | 1 s/hop or less |
| Critical data bandwidth | 100-200 Byte/s/node |
| Overhead data latency | 30 s |
| Overhead data bandwidth | 0 Byte/s |

Figure 2. Protocol suite design guidelines.

protocols and power saving algorithms to a hardware suite of low power components achieves the lowest energy consumption. By incorporating detection and tracking signal processing with the communication protocol in the same processor, we achieve better performance with lower energy consumption than the typical discrete approach.

The protocol suite can be adjusted for different uses. We have adopted as our motivating mission a surveillance system that performs detection and tracking of targets in an area. Surveillance systems exhibit wide variation in the demands placed on the nodes. For example, estimated target speed affects node spacing which in turn affects the required sensor and radio range, which affect energy consumption and node lifetime. A set of nominal parameters for testing purposes is shown in Figure 2. These parameters are not limits on the performance of the protocol; they are design guidance to avoid mistakes. For example, the overhead bandwidth guideline says that techniques relying on a steady stream of messages are inappropriate, or at least have added energy consumption costs.

## PROTOCOLS AND SOFTWARE

A complete sensor software suite is shown in Figure 3. Power saving protocols have been developed for all of the node components, including a real time task scheduler that adjusts the processor clock rate as required to execute tasks or to save energy and firmware that is enabled or disabled as needed.

The transceiver is the most power hungry component; thus the communication protocol is the most effective at saving energy. The Synchronous/Asynchronous (*Sas*) medium access protocol is capable of rapidly switching from asynchronous, contention service to synchronous, collision free service depending upon node state, data load, and data criticality. The hybrid approach allows optimization for performance when that is critical and optimization for low power consumption when performance is not important.

The network operates a time division multiple access (TDMA) scheme with support for dynamic frequency selection (DFS), transmit power control (TPC), and spatial diversity through the use of directional antennas. Many configuration variables exist to control the protocol operation and tailor it to a specific use.

**Slot scheduling.** The protocol operates by scheduling the use of time slots for a specific purpose. Slot scheduling is fully distributed—any node can schedule a slot at any time based upon its knowledge of the state of the neighboring network, its load, interference statistics, data criticality, and other factors.

Slots may be scheduled for communication to or from a specific node, subset of nodes, or all nodes. Before scheduling a slot, the node checks its knowledge of slot schedules for conflicts. After scheduling a slot, the node informs its neighbors, which then pass the information on to their own neighbors. Slots may be scheduled for transmit or receive and may be scheduled for point-to-point, fan-in (1 receiver from multiple potential transmitters) or fan-out (1 transmitter to multiple receivers) operation. Collision detection and random backoffs manage multiple access on fan-in slots.

Schedule conflicts are rare since the number of slots is far

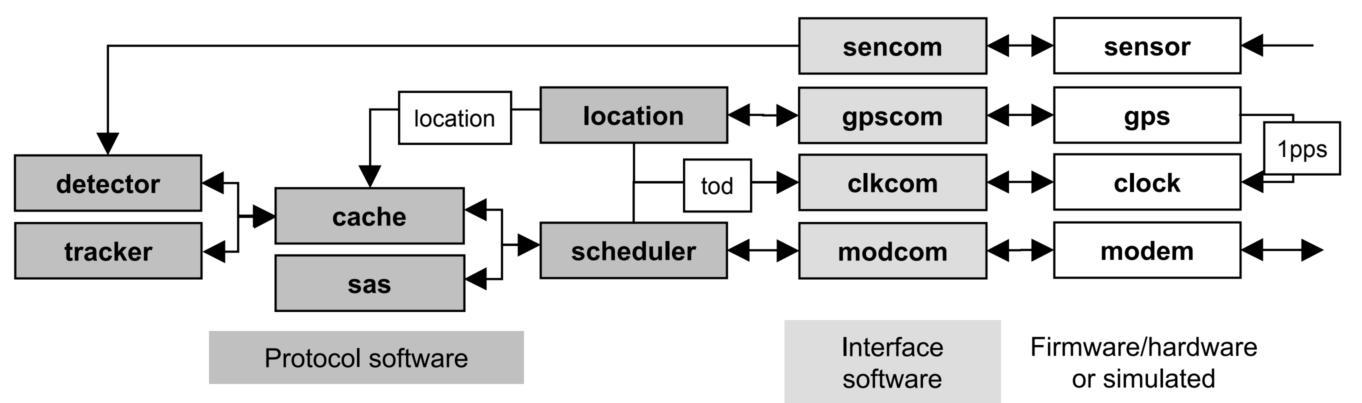


Figure 3. The major modules of the sensor protocol suite and the data flow between them.

| Field Name | Bytes |
|---|---|
| Preamble | 4 |
| Synchronization Header | 4 |
| Source | 2 |
| Destination Mask | 2 |
| Tx Time, seconds | 4 |
| Tx Time, microseconds | 4 |
| Type | 1 |
| Length | 1 |
| Application Data | Length |
| Block Code | 16 |
| Synchronization Trailer | 4 |

Figure 4. The header provides address and time synchronization data while minimizing overhead.

greater than the number required at a particular time. Conflicts are resolved by having one of the conflicted nodes change its schedule. Conflicts occur when 2 nodes in a neighborhood simultaneously schedule the use of the same slot, or more often during network formation when 2 nodes have scheduled the use of the same slot before they become connected. Slots are scheduled with an effective first and last use time and are only used within that time period. Slots scheduled by the transmitter are not used until acknowledged by the intended receiver. Time slots and frequency channels are scrambled on every frame based on a shared key and time of day.

**Mode Switching.** An asynchronous, fan in, receive slot uses the least energy—only a single receiver is on when there is no data to transmit. A synchronous, fan out, transmit slot to all neighbors uses the most energy since many receivers are on even if no data is transmitted. Conversely, such a receive slot is capable of moving very little data; such a transmit slot is capable of moving a lot of data efficiently and with low latency. *Sas* switches operation between the types of slots to best meet the current communications needs of the nodes.

*Sas* may be configured to alter the slot schedule upon a number of types of events. By far the most interesting event type for a sensor network is a cue from an external process. *Sas* reacts to a cue from the surveillance system that sensor data may be forthcoming to produce a broadcast slot. With approximately 1-2 seconds of lead-time, *sas* can reconfigure the network neighborhood from a laid-back, barely active state to a state supporting the rapid movement of sensor data.

**Network Formation.** Operation of a node typically begins by scheduling some (configuration dependent) fan in, receive slots. Every so often (also configuration dependent), *sas* formats a message containing its receive slot information, and announces this in an unscheduled use of a randomly selected, unused slot. If any neighboring node happens to hear this transmission, it now knows how to talk back to the originating node. On subsequent frames it attempts to contact the originating node with a message containing its own receive slots. Once 2-way communication is established, the nodes exchange other information about themselves and their neighbors, so that both nodes may build knowledge of activities in a neighborhood about themselves.

Upon learning of a neighbor of a neighbor, the node attempts to contact that neighbor of a neighbor. This procedure establishes additional links more quickly and more efficiently than the unscheduled search that discovered the first link.

The number of receive slots and the probability of performing an announcement are decreased exponentially from their initial values with the discovery of each neighbor. After network formation, the network typically rests with 1 receive slot for each node and little data traffic.

**Packet structure.** The communicated packets are formatted as shown in Figure 4. The packet header contains addressing and timing information. Each node contains a unique 64-bit hardware id. However, 16-bit dynamically selected and negotiated ids are used for communication to conserve bandwidth. Neighboring destinations are specified very compactly as single bits in the destination mask, which allows dynamic multicast addressing. The type field specifies the type of the application data and is used to control delivery on the receiver. The length field specifies the amount of application data.

The timing fields in the packet header allow *sas* to maintain tight time synchronization between nodes without having to resort to the expensive operation of the GPS receiver.

**Data cache**. Data is stored in a small, memory resident distributed database system. Data records are filtered and routed based upon attributes within the data in order to satisfy the queries expressed on the individual nodes. It efficiently distributes data using dynamic multicast messages. It guarantees delivery as long as the data remains valid and it ensures consistency of the copies.

**Routing.** Data attribute routing has many advantages for a sensor network. First it removes any need for prior knowledge of the nodes or their addresses. Data messages are disseminated by matching attributes of the data and interests of the nodes. In a sensor network, node interests naturally cluster in a neighborhood allowing the system to

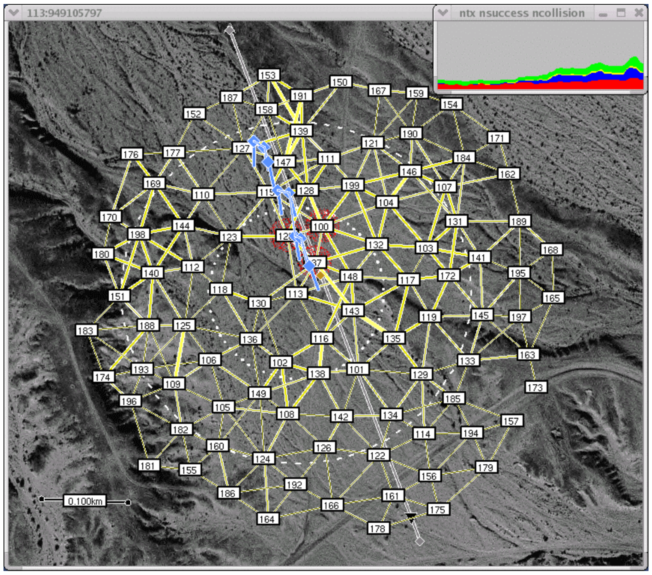


Figure 5. A track (light blue diamond and velocity vector) formed by the protocol suite from simulated detections (red splotches) shown superimposed on the 100 node simulation environment.

take advantage of the natural properties of radio communication to reach many nodes with few messages. Secondly it establishes the primacy of the data over the source node. This is a more natural means of specifying which data is of interest to an application program.

**Detection.** Data from the 3 passive infrared sensors is acquired at 100 sample/s and processed with a simple threshold detector looking for the characteristic pulse shape. The detections are communicated within a neighborhood around the detecting nodes and used to drive the collaborative tracker.

**Tracking.** On a single node, the collaborative tracker employs standard tracking algorithms--a spatial and temporal data association window about the predicted track location and a linear fit to the prior track points and the new detections to produce an estimated target state vector. Tracks are initiated when 2 or more unused detections occur in the same window.

The collaboration between multiple trackers is accomplished with the *data cache* consistency algorithms. Redundant updates to the same track are resolved automatically during network dissemination since track records are keyed by target id and time. Redundant track initiation (2 tracks simultaneously initiated by different nodes) must be resolved by one tracker deleting its track in preference to the other.

**Cueing.** Track estimates are propagated ahead of the predicted track, providing time to cue the nodes to turn on the sensor, start the detector and tracker software, and configure the communication system for low latency dissemination of critical data through the creation of a broadcast slot from the node.

## TEST RESULTS

Simulations were conducted to measure the performance of the software suite under a variety of conditions and operating parameters. The environment provides a simulated clock signal, radio communications, and detections. A small sampling of the test results is presented in this section.

The 100 simulated nodes are laid down randomly inside a 300m radius circle with the constraint that no node can be closer than 50m to another node as shown in Figure 5. Low power (1mW), moderate data rate (1Mb/s), radios are simulated yielding an approximate communication range of 80m. Collisions from simultaneous broadcasts within range are detected and packets may be lost. For the purpose of all of the tests, a 1 second frame was divided into 250 equal slots.

**Network Formation.** In these tests, the 100 node sensor network is started from a cold start. Nodes have no prior knowledge of the existence of other nodes. Knowledge is acquired as the network operates. The tests were conducted with several sets of configuration parameters. Unless very fast network formation is required, the recommended parameter settings use 1 listen slot and an initial schedule announcement probability of 1.0 as shown in Figure 6a.

**Network Join.** In this test a few of the nodes in the network were not started along with the others. These few nodes were later tuned on and allowed to join the network. This process simulates the action of dormant nodes waking, finding the network, and acquiring existing knowledge. The new nodes find the existing network in 4 to 30 seconds and are fully connected in about 80 seconds. The existing network knowledge is passed quickly over point-to-point links created upon discovery of a new node, which has an otherwise negligible effect as shown in Figure 6b.

**Normal State.** This test shows the operation of the network in its normal active state. Network formation has finished and there are no targets traversing the field, so there is no application data that must be communicated. The nodes utilize very little communication bandwidth as shown in Figure 6c.

**Detection and Tracking.** The detection and tracking test involves the complete protocol suite. A simulated target is moved through the sensor field along a specified route. Detections are simulated on each node with a very high

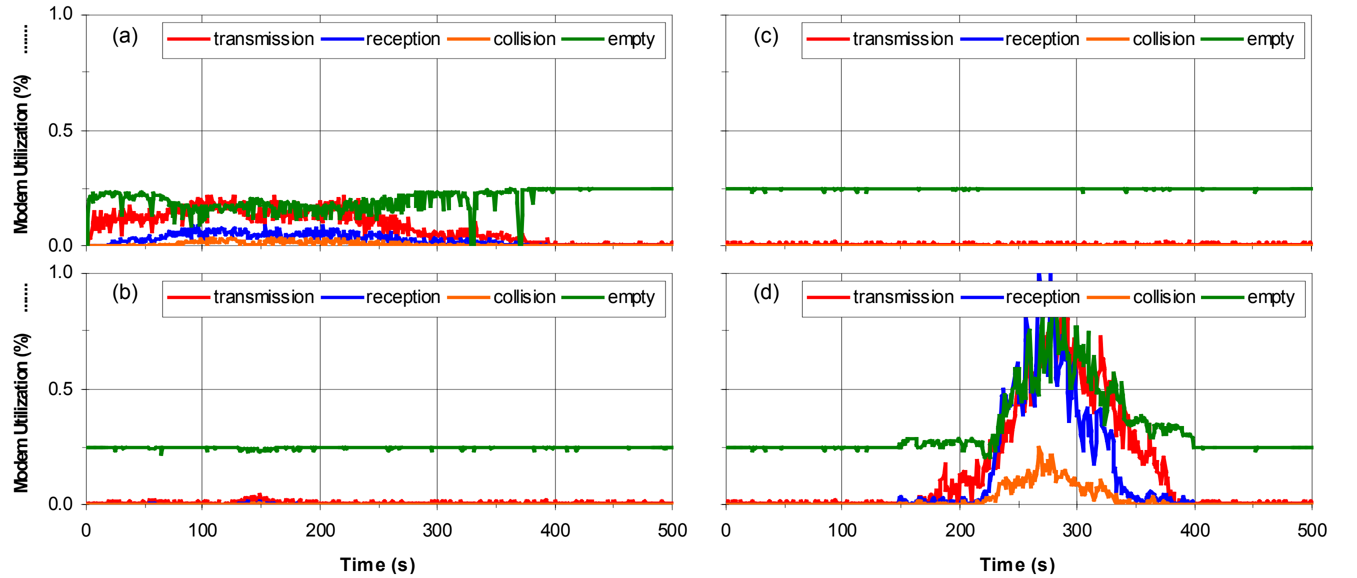


Figure 6. Modem utilization is dramatically lowered by the *sas* protocol as shown (a) during network formation, (b) during network join by a few additional nodes, (c) during the normal active state, and (d) while target tracking. The vertical axis scale is in percent—total utilization of the 1Mb/s modem in all tests is less than 3% and in the non-tracking tests is less than 0.5%.

probability of detection (0.99) and a very low probability of false alarm (0.01) and an expected detection radius of 50m. Tracker performance is highly dependent upon data latency, making tracking an extreme test of the communication protocol. Performance results are shown in Figure 6d. The ground truth, detections, and tracks are shown in Figure 5.

## HARDWARE

Prototype sensor nodes are developed to test the protocol suite in a realistic field environment. The sensor node is implemented as a stack of 3 small printed circuit boards and common, commercially available components as shown in Figure 7. After verification of the design, correct hardware sizing and further integration can reduce the number of discrete components, yielding a decrease in size and cost. Processing, memory, and transceiver components can be integrated in a single chip.

Numerous design tradeoffs can be made to achieve the right balance between size, cost, and energy consumption. The use of more capable components may actually lower total node size, cost, and energy consumption by allowing the sharing of the components among multiple uses and allowing more sophisticated protocols and processing. In this design, the processor, modem, and GPS receiver all provide more capability than current commercial designs, but the total node energy consumption is comparable to or better than those designs.

**Processor.** A hybrid general purpose processor (GPP) and programmable logic device (PLD) with support for System on a Chip (SoC) methodology is at the heart of the node. Real time functions are implemented in Verilog for the programmable logic. Non-real-time functions are implemented in C for the GPP. Substantial instantaneous computing power is available (130MIPS), but energy consumption remains low due to processor duty cycle management. The processor is supported by external memory—FLASH to store program instructions and logic configuration when the node is not powered and RAM to store instructions and data during operation.

**Transceiver**. The transceiver is an integrated 1Mb/s FSK modem with a 2.4GHz RF front end. The transceiver is augmented by an external LNA and PA to provide greater link margin for operation at ground level and at greater range (>100m), which may not be necessary in many applications. A compact antenna is included in the circuit or an external antenna may be used.

**Timing**. Timing support is provided by an on-board ±20ppm, 20MHz oscillator. The node performs self calibration of the oscillator using either 1 pulse per second signals from the GPS receiver or time synchronization data embedded in the received packets from other nodes. Self calibration renders timing accuracy better than ±1ppm. Synchronization with an external source is still required, but at a much lower duty cycle.

Figure 7. The fully integrated sensor node includes a PIR sensor, a GPS receiver, an ISM band modem, a processor, and memory powered by 2 AA batteries. The enclosure is weatherproof to allow outdoor operation. The node measures 192cm$^3$ and weighs about 140g. Batteries make up about a third of the volume and more than half the weight. Another third of the volume is empty space to allow for the correct focal length between lens and sensor.

**Location.** A GPS receiver is powered at node startup to provide the node location and to calibrate the node oscillator. It is then turned off to save energy. It is operated only occasionally as necessary to resynchronize the node clock.

**Sensor.** Detections are provided by the PIR sensor. The Fresnel lens mounted in the node case forms a tightly focused detection beam with a range of approximately 30m for small, slow moving targets. Larger or faster targets, such as vehicles, can be detected at a greater range.

| Component | Peak (mW) | Dormant (mW) | Active (mW) | Tracking (mW) |
|---|---|---|---|---|
| Sensor | 10 | 0.00 | 0 | 10 |
| Gps | 160 | 0.01 | 1 | 1 |
| Modem | 264 | 0.06 | 2 | 8 |
| Processor | 142 | 0.35 | 27 | 95 |
| TOTAL | 576 | 0.42 | 30 | 114 |
| | | | | |
| Days (2AA, 2000mAh) | 0.3 | 476.2 | 6.6 | 1.7 |

Figure 8. Estimated node lifetime. The deployed node spends most of the time in the Active state.

## CONCLUSION

Simulation has demonstrated that the largest power consuming component of the system, the radio transceiver, can be reduced to a very small energy consumer by aggressively managing its usage. This can be done by making more sophisticated and autonomous sensor nodes and without sacrificing the ability to perform detection and tracking with realistic parameters.

The node energy consumption and lifetime estimates are shown in Figure 8. These estimates are based on power consumption values from commonly available components (nothing exotic or expensive was selected) and measured communications protocol duty cycles. The expected lifetime of a deployed node using 2 AA batteries is about 6 days.

We expect system energy consumption to decrease by a factor of 10 in the next 2 years through improvements such as increased hardware integration, smaller logic gate sizes, reduced supply voltage, protocol improvements, and battery energy density, providing a reasonable sensor lifetime of a month or two.