Novateur Publication's
International Journal of Innovation in Engineering, Research and Technology [IJIERT]
ICITDCEME'15 Conference Proceedings
ISSN No - 2394-3696

WIRELESS SENSOR NETWORKS: A SURVEY

Mr.Chavan Rahul Raghunath
Department of Electronics and Telecommunication AIT Vita, Maharashtra

Mr.Chavan Rahul Barama
Department of Electronics and Telecommunication Dept. AIT Vita, Maharashtra

ABSTRACT

A wireless sensor network (WSN) (sometimes called a wireless sensor and actuator network (WSAN) are spatially distributed autonomous sensors to monitor physical or environmental conditions,[2] such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on. The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding The relentless development of wireless technology paves the way to novel applications, based on sensor networks or wireless systems. This paper aims at bringing together specialists of application areas along with researchers of mobile computing and networking. A goal of the paper is to exchange information about the most promising upcoming applications. In this paper we cover three applications based on wireless sensor network.

INTRODUCTION

A new generation of networked embedded systems incorporating controlled and precise mobility is now being explored. These Networked Info mechanical Systems (NIMS) directly address the fundamental objective of self-awareness by enabling motion of sensor node networks to circumvent obstacles, probe sensing fidelity, and optimize sensor and sample distribution. NIMS introduces a new networked embedded system capability that provides the ability to explore large volumes, adds new networking flexibility and functionality, and new logistics for support of distributed sensors, as well as the capability for self-awareness. This requires, in turn, the development of new methods for scalable and optimized coordination of mobility among nodes for many possible objectives. NIMS also introduce infrastructure-supported mobility to enable low energy transport and retain inherent low operating energy, rapid deployment characteristics, and environmental compatibility of distributed sensors. This paper will describe the development of complete NIMS systems that link mobile and fixed sensor nodes in a hierarchical network. This will include description of NIMS architectures matched to applications in environmental monitoring and are now deployed in the natural environment. Environment characterization methods that exploit autonomous coordination of sensing, motion, and fixed sensor network event detection will also be discussed. Finally, future directions for NIMS research and applications will be discussed

COOPERATING SENTIENT VEHICLES FOR NEXT GENERATION AUTOMOBILES

The cooperating sentient vehicle application is a key demonstrator of the technology developed by the EU Framework V IST funded CORTEX project1 [7].The sentient vehicles are 'context-aware' cooperating autonomous cars, which autonomously navigate to a given destination. CORTEX is concerned with developing middleware support for constructing 'proactive applications' based on a paradigm we call real-time sentient objects. CORTEX proposes a sentient object model based on anonymous event-based communication. Generally speaking, the systems consist of an environment and a set of sentient objects that are capable of independently sensing this environment, deriving context and inferring autonomous action.

Applications built from sentient objects may communicate using event channels to establish higher-level contexts and thus cooperate with each other. We have chosen to focus on a particular proactive application, the 'cooperating sentient vehicles'. This application has been made possible with the recent technological advances including, wireless networking such as 802.11b capable of operating in ad hoc mode with a bandwidth of up to 11Mbps, improved location accuracy provided by GPS which is capable of 3m-5m relative positional accuracy, ultrasonic sensors for obstacle sensing, magnetic digital compasses and Pocket PCs. However, there are key research issues that need to be addressed to realize such applications. We have identified the challenges posed by cooperating sentient vehicle application, and built a resultant Component Framework (CF) based middleware architecture. The middleware architecture addresses the many challenges raised by mobile context-aware applications operating in MANETs. We have implemented this middleware platform and used it to build a prototype of the cooperating sentient vehicle application. The remainder of the paper is structured as follows. First, section 2 briefly describes the application scenario. Section 3 discusses the challenges raised by the cooperating sentient vehicle application and outlines our approaches to address the challenges using component framework based middleware technology. Section 4 briefly describes our sentient vehicle test bed. Finally, section 5 provides our conclusions

COOPERATING SENTIENT VEHICLE APPLICATION SCENARIO

The demonstrator application is divided into two sub-problems: 1) Cooperative behavior without human control, and 2) Autonomous vehicle navigation from a given source to predetermined destination. The autonomous vehicles have the objective of travelling along a given path, defined by a set of GPS waypoints (a 'virtual' circuit). Every vehicle that travels along the path cooperates with other vehicles by inter-vehicle communication. Each vehicle needs to build a real-time perception (an RTImage) of its surrounding environment within some bounded error to make informed decisions regarding its next move. The cooperation between vehicles is critical to avoid collisions, to follow a leading vehicle and to travel safely. The vehicles must obey external traffic signals and give way to pedestrians who cross the road (by sensing their presence). The intelligent vehicles can be deployed in any outdoor arena. Vehicles need to travel from one location to another with minimum driver assistance. Before a journey, vehicles are notified about the virtual circuit waypoint information and bearings.

CHALLENGES AND MIDDLEWARE ARCHITECTURE

The key research challenges, that need to be addressed, to enable the application scenarios described in section 2, are: communication model, routing protocol, context-awareness, end to-end QoS (Quality-of-Service) and fail-safety. We address these challenges and provide our solutions as different component frameworks (CF). The middleware platform consists of, Publish-Subscribe CF, Group communication CF and Context CF. Component frameworks (CF) enforce the functional and non-functional properties of the system, and keep consistency across adaptations triggered by applications. A particular instantiation of middleware components for the MANETs is shown in figure 1.

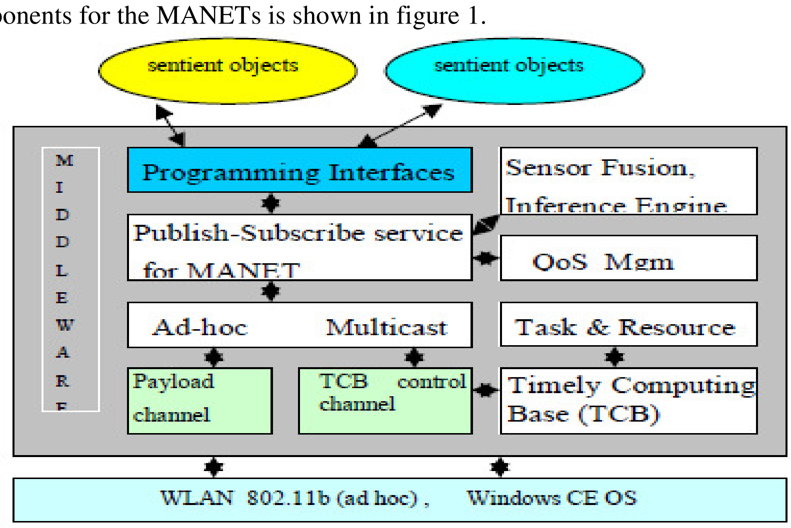

Figure 1 – Middleware Platform for MANET

1) COMMUNICATION MODEL: A key challenge that needs to be addressed by the cooperating sentient vehicles application is the suitable communication model. The client-server or RPC-based paradigms supported by the state-of-the-art object oriented middleware such as CORBA and DCOM are well suited for

ISSN No - 2394-3696

fixed infrastructure based wired networks, but they are not well suited for MANETs. In mobile ad hoc networks any centralized infrastructure based highly coupled, synchronous communication model is not well suited. Since there is no fixed infrastructure to host centralized services, disconnection is the norm and communicating nodes are generally anonymous. To address this, a loosely coupled, asynchronous, anonymous and a fully decentralized communication model is required. The Publish- Subscribe communication model has the aforementioned properties. However, most of the state of the art publish subscribe or event based middleware are based on centralized event brokers. Our middleware takes the approach of having a public-subscribe (event based) communication model, with all the aforementioned required properties and was especially designed for MANET. The design of Publish-Subscribe CF was inspired by STEAM. Events are transported via selectable ad-hoc multicast protocol described below.

2) ROUTING PROTOCOL: Routing in mobile ad-hoc networks is a challenging issue because of frequent topological changes in networks. Publishers and subscribers move frequently, posing a challenge for routing of events in wireless ad-hoc networks.

Multicast routing based on proactive and reactive ad-hoc routing, using shared state kept in the form of routes and adjacent information, is useful in environments with low node mobility. However, in scenarios with high node mobility such protocols are unsuitable as shared state and topology information can quickly become outdated. For this reason as part of the Group communication CF, we have built a Probabilistic Multicast Protocol for Wireless Ad Hoc Networks. The protocol specifically targets ad hoc environments with high node mobility and a frequently changing topology of group members. The design of the probabilistic multicast protocol was inspired by previous research on multicast algorithms (both proactive and reactive) for ad-hoc networks, which shows that most existing algorithms (AM Route, CAMP, MCEDAR, AODV, etc.) perform inadequately when high node mobility is present in the environment. Our multicast routing protocol is based on a probabilistic flooding algorithm with damping, which does not maintain shared state in nodes. The Group communication CF also includes IP multicast protocol. Publish-subscribe CF can dynamically reconfigure to utilize a multicast protocol from the Group communication CF.

SENTIENT VEHICLE TEST BED

A sentient vehicle used in the demonstrator application is shown in figure2

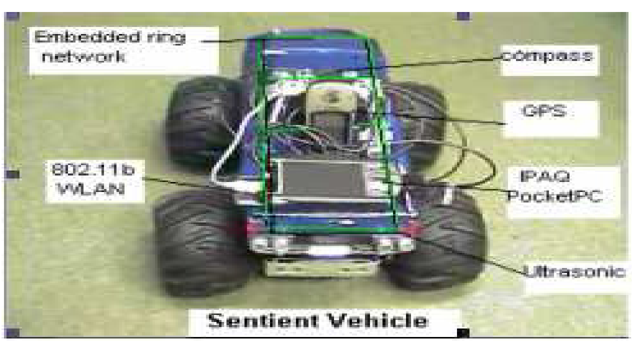

Figure.2

The sentient vehicle is a modified Juggernaut 2 remote control (RC) Car. The RC module has been removed and the car is controlled by an HP iPAQ Pocket PC via a RS232 connector. The car is augmented with a GPS module for location sensing, an electronic compass for sensing its orientation, 8 units of ultrasonic sensors (with 3cm to 3m range) to sense the presence of neighboring physical objects. The Pocket PC has 2 WLAN cards (configured for ad-hoc mode use). The more powerful WLAN card is for the exclusive use of TCB control channel (which requires a lightly loaded network) and the other WLAN card is for the event channels (payload) used for inter-vehicle communications. The two WLAN cards operate in no overlapping 802.11b channels. An on-board CAN is used on the sentient vehicle to broker data between the sensors, actuators and iPAQ. The on-board network is a bespoke ring topology with single break failure resilience. Importantly, the design enables addition of further devices on to the ring in a plug and play fashion making it extensible. The test bed contains a small number of sentient vehicles and laptop acting as a sentient traffic light. The pocket PC mounted on the sentient vehicles has the Windows CE version of the middleware platform, which was implemented using embedded Visual C++. This application has described an interesting and challenging application domain involving the co-ordination of autonomous vehicles in a

MANET environment. Based on the experience of constructing this application, the key results are: i) sentient object model has proved to be an excellent programming abstraction for the development of such applications, particularly because of their intrinsic support for context awareness; ii) there is a real need for middleware in this area to ease the burden on the application developer and also to provide support for the management of non-functional concerns such as timeliness properties; iii) the properties of configurability and re-configurability inherent in our approach are highly suited to this domain, for example to encourage the construction of adaptable or autonomic systems.

WIRELESS AD HOC DISCOVERY OF PARKING METERS

Locating a suitable parking spot is a common challenge faced by millions of city-dwellers every day. As common is the revenue generation by fee and _ne collection in these municipalities. Wireless ad hoc networking technologies over a new and ancient means to both simplify the process of parking and find collection as well as extending the convenience for drivers. In this paper we describe a multi-hop wireless parking meter network (PMNET) that, when coupled with a GPS receiver, allows a user (driver) to quickly locate and navigate to an available parking space. Our solution is achieved by equipping existing parking meters with wireless radio frequency (RF) transceivers and auxiliary hardware and software. We believe that this is a compelling application that applies wireless ad hoc networking and low-power, short range RF technologies. The attractiveness of the proposal stems from the fact that such a network of nodes can function without any fixed wired or wireless infrastructure such as cellular or satellite networks. In this work, we model a PMNET as a special class of ad hoc networks characterized by a combination of static, immobile nodes (parking meters) and mobile nodes (vehicles). We propose scalable techniques for satisfying a mobile user's query in a distributed fashion.

ARCHITECTURE OF A PARKING METER NETWORK

Existing parking meter function needs to be augmented in order to be able to support the application described in Section 1. Parking meters (or pm-nodes) will require a low cost, low-power embedded processor. Programmable memory and operating system. They require an inexpensive infra-red (IR) sensor to detect occupancy; and a short-range, low-power RF transceiver to communicate with neighboring meters and nearby mobile vehicles. Since the transmission range of Bluetooth's radios (10 m) may be too limiting to maintain a connected ad hoc network, we believe that Millennial Net's i-Bean is a better candidate technology since it can support transmission ranges up to 40 m and has nice energy saving features. We leave the detailed investigation of how to leverage i-Bean technology for this application as future work. A stationary pm-node has a _xed geographical location attribute which can be exploited during the discovery process. Instead of putting GPS receivers into every parking meter, meter deployment can include the use of a portable GPS receiver to permanently burn the geographical coordinates of each pm-node into its memory. A locally unique ID along with the GPS location attribute can serve as a globally unique ID (GUID). These attributes and some others such as \street location" are non-volatile attributes. Other attributes that are likely to be stored at a pm-node at any time instant include the size of the spot, its current avail ability, the fee for use, and the time limit. These are referred to a volatile attributes. Wirelessly enabling these devices results in their connection to a virtual network of devices thus allowing them to function as resources that can be discovered remotely, instead of physically. In addition to supporting drivers seeking parking, PMNETs can be used by the municipality to simplify and optimize revenue generation from fees or penalties. Arguably a game theoretic analysis of the distinct goals of individuals vs. municipalities can yield a variety of approaches to parking rules Some representative queries that might be posed by the fee collector or driver are: Q0: Which streets in the locality have vacant spots right now? Q1: Which meters are about to expire within 200 yards of my location (GPS coordinate specified)? Q2: Is there any parking spot scheduled to become available soon on Main St. until midnight? Q3: Locate all vehicles that reside in expired spots. A query processing software module running on a user's unit structures a query in a suitable query language before it is dumped in the network via pm-nodes that are RF-reachable from the user's current location

AD HOC PARKING SPOT DISCOVERY

Although the pm-nodes are themselves static, the content served by them changes dynamically. In addition to the non-volatile attributes such as geographic location, it is this dynamic content (e.g., is the spot available now?) which is often of great interest to the users. Most mobile user queries require both static and dynamic attributes for resolution. A query processing module receives structured queries from mobile users, determines if a query matches the current values of its relevant attributes and takes appropriate action. If a query can be satisfied locally, the pm-node responds to the user using the underlying unicast routing protocol described in Sec. 3.3. In the basic version of the system, if a pm-node is unable to satisfy the query itself, it rebroadcasts the query to its neighbors. In a more sophisticated version of the system as proposed in Sec. 3.1, query processing is performed in a hierarchical fashion on pm-nodes which act as Cluster Heads (CH) for a set of other pm-nodes. The status update processing module receives status updates (values of volatile attributes) from other pm-nodes and updates the local data structures accordingly. These data structures need to be queried repeatedly while responding to user queries. We advocate a hybrid approach (partly proactive and partly reactive) for facilitating querying and status update so as to minimize bandwidth usage as well as user latency.

CLUSTER BASED STATUS UPDATES

A hybrid cluster-based approach which utilizes the natural geographical clustering in PMNETs is ideal for facilitating quick status updates and speeding up the discovery process. Non-volatile attributes such as street name can be used to cluster a set of pm-nodes together. The radio transmission range is assumed to be large enough so that the pm-nodes on opposite sides of a street can communicate with each other. Election of Cluster heads: In our approach, exactly one of all pm-nodes with the same street attribute serves as a cluster head (CH) of the set. A CH can be elected to its role during installation or can automatically be selected by a simple leader election process executed by all pm-nodes2.

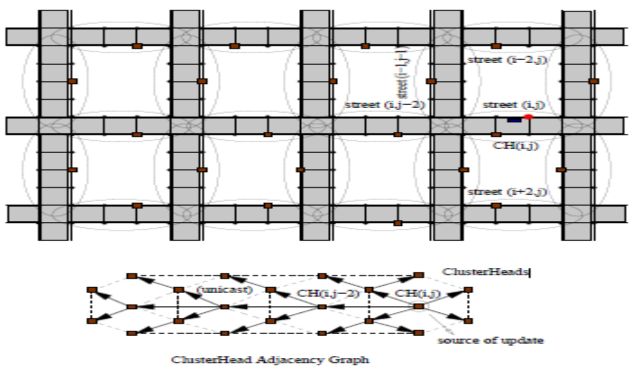

Figure.3: Scalable Status Update Dissemination

Fig. 3 illustrates the process of performing a status up- date with a specific example. When a car leaves a parking spot on street (i,j), the corresponding pm-node (marked by a red filled circle) informs its cluster head, CH(i,j) about this change. Now, CH (i,j) needs to send a status update to all other CH's in the locality. This can be accomplished in a number of ways. In the first approach, a CH attempts to send reliable updates to its neighboring CH are over unicast which then re-forward the updates further downstream. To determine who to forward the updates to, a Cluster head Adjacency Graph (CHAG) is constructed from the topology of street intersections3. Now, unicast forwarding needs to be done only along the edges of a minimum spanning tree of CHAG rooted at CH(i,j). There are several trade-o_s between latency and control overhead that can be exploited in this system. A CH can choose to disseminate only a salient part of the status update to other CH's, e.g., if a spot in street S becomes vacant, CH(S) can inform other CH's of only the fact that there is an additional vacant parking spot available, instead of furnishing the details of the spot. This can help in quick resolution of queries like [Q0]. However, when a user in street X issues a more involved query, CH(X) may not be able to answer that query immediately from this delta-information, but it can query the relevant CHs about the details of the parking spots at the current time. This approach is advantageous as a CH needs to store information only at a coarse level of granularity. The obvious penalty is that of additional time taken to discover a spot. The best approach in a particular scenario would depend on factors such as the predominant nature of queries and their frequency, the frequency of changes, size of the PMNET etc.

CODE BLUE: AN AD HOC SENSOR NETWORK INFRASTRUCTURE FOR EMERGENCY MEDICAL CARE

Wireless sensor networks are an emerging technology consisting of small, low-power, and low-cost devices that integrate limited computation, sensing, and radio communication capabilities. This technology has the potential to have enormous impact on many aspects of emergency medical care. Sensor devices can be used to capture continuous, real-time vital signs from a large number of patients, relaying the data to handheld computers carried by emergency medical technicians (EMTs), physicians, and nurses. Wearable sensor nodes can store patient data such as identification, history, and treatments, supplementing the use of back-

ISSN No - 2394-3696

end storage systems and paper charts. In a mass casualty event (MCE), sensor networks can greatly improve the ability of first responders to triage and treat multiple patients equipped with wearable wireless monitors. Such an approach has clear benefits for patient care but raises challenges in terms of reliability and complexity. While there have been many recent advances in biomedical sensors, low-power radio communications, and embedded computation, there does not yet exist a flexible, robust communication infrastructure to integrate these devices into an emergency care setting. We are developing Code Blue, an efficient wireless communication substrate for medical devices that addresses ad hoc network formation, naming and discovery, security and authentication, as well as filtration and aggregation of vital sign data. Code Blue is designed to operate across a wide range of devices, including low-power "motes," PDAs, and PCs, and addresses the special robustness and security requirements of medical care settings.

CODE BLUE: A WIRELESS INFRASTRUCTURE FOR EMERGENCY RESPONSE

Integration of low-power wireless devices into medical settings raises a number of novel challenges. Current demonstration systems operate with a small number of devices under fairly static conditions. Scaling up to handle a mass casualty scenario and ensuring robust operation with a high degree of mobility and minimal packet loss poses a number of open problems. We do not wish to assume an existing wireless infrastructure; the system must work in an ad hoc manner. We are developing Code Blue, a wireless communications infrastructure for critical care environments. Code Blue is designed to provide routing, naming, discovery, and security for wireless medical sensors, PDAs, PCs, and other devices that may be used to monitor and treat patients in a range of medical settings. Code Blue is designed to scale across a wide range of network densities, ranging from sparse clinic and hospital deployments to very dense, ad hoc deployments at a mass casualty site. Code Blue must also operate on a range of wireless devices, from resource-constrained motes to more powerful PDA and PC-class systems. We are in the early design and prototyping stages of Code Blue's development; the discussion that follows outlines our current design goals and the research challenges that emerge in this environment.

CODEBLUE ARCHITECTURE

Code Blue offers a scalable, robust "information plane" for coordination and communication across wireless medical devices. Code Blue provides protocols and services for node naming, discovery, any-to-any ad hoc routing, authentication, and encryption. Code Blue is based on a publish/subscribe model for data delivery, allowing sensing nodes to publish streams of vital signs, locations, and identities to which PDAs or PCs accessed by physicians and nurses can subscribe. To avoid network congestion and information overload, Code Blue will support filtration and aggregation of events as they flow through the network. For example, physicians may specify that they should receive a full stream of data from a particular patient, but only critical changes in status for other patients on their watch. The use of ad hoc networking will allow the "mesh" of connectivity to extend across an entire building or between multiple, adjacent facilities. Additional coverage, if necessary, will be possible with placement of fixed nodes in hallways, rooms, or other areas. No matter the topology, the network will be self-organizing: loss of a given node or network link can be rapidly detected and data re-routed accordingly. Code Blue will also provide for reliable transmission of critical data through content-specific prioritization and dynamic scaling of transmission power. Code Blue will support a flexible security model allowing a range of policies to be implemented. For example, it is necessary that EMTs who require access to patient data be authenticated by the network before they are able to receive all patient information. One EMT must also be able to hand off access rights to another, as when a new rescue team arrives on the scene of a disaster. Authentication must be performed transparently as the patient is transported from disaster site to hospital, or transferred between hospitals. Access control must be decentralized to avoid reliance on a single authoritative system. Code Blue will simplify application development by providing a rich infrastructure for connectivity of medical devices. In the hospital, data collected from wireless sensors can be relayed to fixed, wired terminals and integrated with patient records in existing hospital information systems. At a mass casualty site, an ambulance-based system can record extensive data streams from each wireless sensor or PDA to support audits and billing. We have completed an initial design of Code Blue and prototypes of several of the components described herein. The pulse oximetry mote has been completed and development of an ECG mote is currently underway. We have explored the use of an adaptive spanning-tree multi-hop routing algorithm, based on the TinyOS Surge protocol, and we have incorporated dynamic transmission power scaling to minimize interference.

CONCLUSION

This paper has described an interesting and challenging application domain involving the co-ordination of autonomous vehicles in a MANET environment, wireless parking meter network in which drivers can quickly locate and navigate to available parking spaces, code blue to provide routing, addressing, security, and prioritization of data. Such an infrastructure is necessary to realize the benefits of these next-generation wireless devices.

REFERENCES

[1] Coulson, G, Blair, G.S, Clark, M, Parlavantzas, N, “The Design of a Highly configurable and reconfigurable Middleware platform” ,ACM Distributed Computing Journal, Vol 15,No 2, pp109-126, April 2002.
[2] C. Trillin, \Tepper Isn't Going Out," New York: Random House, 213 pp.
[3] P. Basu and T. D. C. Little, \Networked Parking Spaces: Architecture and Applications," IEEE Vehicular Technology Conference (VTC), Vancouver, Canada, September 2002.
[4] L. Subramanian and R. Katz, \An Architecture for Building Self-Con_gurable Systems," Proc. MobiHoc 2000, Boston, MA, August 2000.
[5] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David E. Culler, and Kristofer S. J. Pister. System architecture directions for networked sensors. In Architectural Support for Programming Languages and Operating Systems, pages 93–104, 2000.
[6] David Malan. Crypto for tiny objects. Technical Report TR-04-04, Harvard University, January 2004.