SENSOR NET SOFTWARE TECHNOLOGY (SNST)

ISI Technical Contacts/Co-PI's: Deborah Estrin, John Heidemann, Allison Mankin

Principal Investigator: Deborah Estrin, Professor

Technical topic areas:

2. Sensor algorithms and applications
3. Distribution and aggregation technology
4. Nanocryptography
5. Prototyping kits

A. Innovative Claims

USC/ISI proposes to create a software environment for the development and analysis of communication and coordination architectures for sensor nets and similar networks of small specialized devices (including robotic or actuation devices). ISI proposes to distribute this Sensor Net Software Technology (SNST) package to the research community, developing the software in phases (and in conjunction with initial off-the-shelf hardware) such that an initial distribution can occur very shortly after work begins. SNST will be targeted for at least two hardware platforms, also to be distributed to the community: 1) a low-power, low-cost, highly flexible sensor preprocessor device under development by Dr. William Kaiser (Sensor.Com); and 2) a low-power, low-cost radio object location sensor assembled from COTS components. Platform 1 is detailed in Dr. Kaiser's companion to this proposal, entitled WINS Next Generation. Platform 2 is detailed in the technical rationale below. Both hardware platforms will enable tracking of large numbers of actively tagged objects; the COTS platform allows early software development to proceed in parallel with hardware development.

Two fundamental aspects of the software are innovative. First, the communications subsystem will meet the requirements for zero configuration and maintenance operation and adaptivity to low-power, low bit-rate, through extensions of IPv6. We propose to extend autoconfiguration with mutual authentication and network clustering, and to extend neighbor discovery with power-awareness and adaptive strong header compression. The result builds on the existing plug-and-play capability of IPv6 to allow automatic formation of large, capable unattended nets. Another benefit of IPv6 is that it permits global reachability of any node, which means that the large sensor net may deliver its results via any of its nodes. Industry standards for deeply networked systems may be influenced by our work, if it begins soon, as there is industry interest already in the IPv6 address space, but our demonstrations of autonomous operation and power awareness are critical as well.

Second, we will experiment with a data diffusion model, driven by the application, and built with adaptive multicast (over IPv6), to meet the requirements of sensor nets and their relatives to perform coordinated information technology tasks with little or no human intervention, among extremely large numbers of processors, with task coordination, and often with conditions of failing communication or unexpectedly present nodes. IPv6 extensions will be able to make basic networks out of the nodes under these dynamic conditions. Then data diffusion algorithms will use localized communication and aggregation to avoid sensing large amounts of data over long distances and to create an application structure that adapts to dynamic topologies, sink points, and data sources. We propose to deliver protocols, libraries and example applications for exploring this new computing paradigm.

SNST deliverables will be developed in the Windows CE operating system. IPv6 will be derived from a completed conventional IPv6 stack that ISI has developed with Microsoft Research. The code will have its basic communications as a core and significant ability to be reconfigured with diffused instructions and new code.

ISI will distribute code and the COTS hardware spec to the DARPA sensor net and deeply networked systems community frequently, starting in the first quarter of performance. Through the SNST software (and the heterogeneous hardware it will run on), we propose to jump-start collaboration on self-organizing nets and data diffusion for the SenseIT and deeply networked systems programs.

B.Technical Rationale

The ISI effort will consist of three tasks:

1) creating the development and experiment environment
2) extending IPv6
3) developing data diffusion protocols, libraries and applications

Another proposal (SCADDS) will cover the more extensive design and evaluation of diffusion-based coordination architectures. The SCADDS work hopes to use SNST as its experimental platform in order to experiment in the context of power-aware communication. SNST will benefit from SCADDS algorithms as added application drivers. SEEDS, a third proposal from us and UC Berkeley, will develop simulation software (derived from our DARPA-funded VINT/ns) to allow the most futuristic exploration of this area. SEEDS hopes to use SNST nodes for emulation experiments (mixed real and virtual nodes) and to validate simulation models.

B.1 Development and experiment environment

The development environment for SNST will be adapted from the Microsoft Device Driver Kit (DDK) for Windows CE, with support (contributed) from Microsoft Research, under their university outreach program, in which USC/ISI is already a participant. The goal of working with a commodity operating system (Windows CE) as the base is to obtain industrial penetration of our results at early stages. The DDK and Microsoft Developer Support Network are available for a nominal fee or free to researchers. The Windows CE source code will be available under an open license that was discussed in detail at DARPA's Windows NT Workshop (October 5, 1998). Collaborators under the same license can freely exchange their code and new modules may be published. The IPv6 code itself will not require a license for use and will be freely distributible. It is expected to be like the Windows NT code in not requiring source code for installation, operation, or modification. The data diffusion code and applications will also be freely distributible.

It is important to develop a means for researcher collaboration on the powerful new computing model of deeply distributed networks using data diffusion as concretely and early as possible. Therefore, the SNST package will include specification of hardware nodes (and experimentation with them) in a simple but powerful configuration:

1) Intelligent sensor nodes based on PC and PDA technology, in the form of stationary PC's and mobile WIN-CE handhelds.

2) Very numerous, more inexpensive nodes, serving as dense sensor net nodes. The simplest form of these is an active tag.

There are two types of inexpensive, numerous nodes for the dense sensor net. One is described in the companion proposal, *WINS Next Generation*, PI's William Kaiser and Greg Pottie (Sensor.Com). It is a military sensor preprocessor, adaptable to numerous types of sensing and actuation, which interfaces to a handheld computer or may stand alone. Besides the variety of sensors and actuators it hosts, it includes slots for RF devices. It emphasizes experimental control of power management and allows instrumentation of the power and radio characteristics. The IPv6 and application software will reside mainly on the back end handheld computer, but will be able to fully exploit its power and channel management and instrumentation, through a low-level API provided by the members of that project. Figure 1 is a schematic from the WINS Next Generation proposal. In our descriptions, the WINS processor is a Windows CE handheld computer.

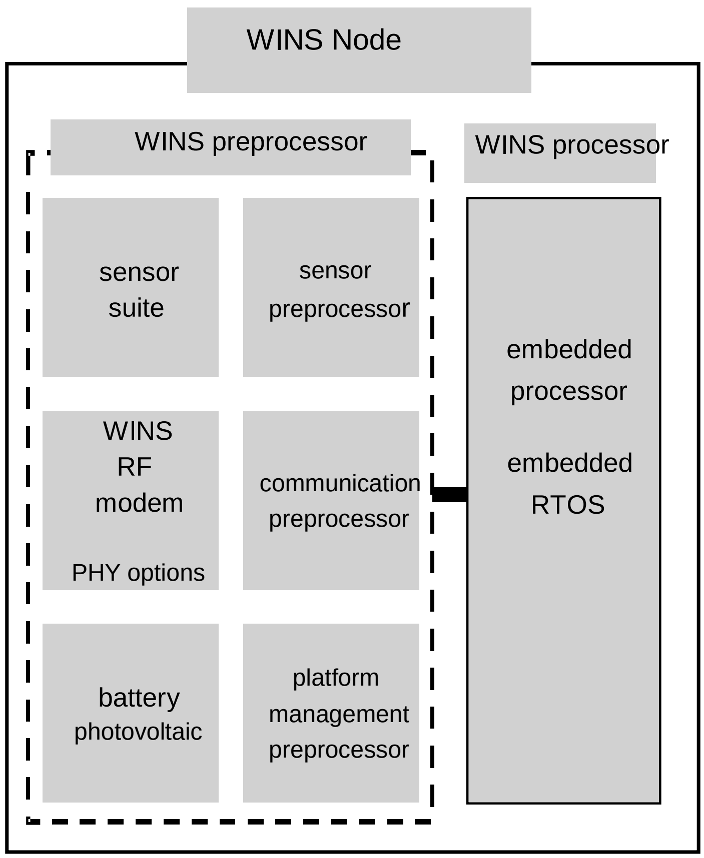

Figure 1. WINS Next Generation schematic (due to Kaiser)

A drawback of WINS Next Generation is that it is developmental hardware and will not be available at the immediate start of SNST. The COTS hardware described next will provide immediate experimental capabilities. In addition, ISI will be open to working with other platforms similar to WINS Next Generation, as and when they become available. SNST will be designed to accommodate heterogeneity of hardware. The WINS Next Generation hardware's provision of multiple sensors and communications channels at low power will be the standard to be met. Our applications will be designed around such capabilities as the goal.

The second type of experimental hardware for the numerous, inexpensive nodes is based on a COTS radio component from the British company, Radiometrix (www.radiometrix.co.uk). We are preparing nodes for our initial, internal experimentation within the first quarter of 1999, before the start of SNST. When purchased in the hundreds, the radios cost around $50.00, and their size is approximately 1"x2"x0.5". While not yet approaching the $1.00 per node goal set by disconnected embedded processors, this is pricing that can approach the lower mark when there emerges a bulk market for radio nodes like the bulk market for disconnected nodes. The imminent availability of digital pulse wireless radio [1], which permits high per node bandwidths for very dense configurations of RF nodes, will contribute to the emergence of a bulk market.

For SNST, we have coupled the Radiometrix card (carrying two components, a BiM-418F transceiver and RPC-418-A radio controller) with a 6811 micro-controller in a very simple configuration. The addition of the micro-controller provides a small memory and a small amount of programmability beyond the programmable features of the radio controller. We plan to implement a small IPv6, with a basic features described in the next section, partly on the radio controller's registers and processor and partly on the micro-controller.

Figure 2 shows a block diagram of the COTS sensor node. When used as the transmitter for an intelligent node, the node receives data from the host via its serial or IR interface and then passes it to the 6811 processor. The processor then buffers the data as needed and sends the data to the radio module for transmission. As a receiver, a node wakes up from a low-power state when it detects incoming data. The radio module receives, formats and performs a checksum on the data. The data is then output to the 6811 processor, which buffers the data and transmits the data via the serial or IR port to the intelligent node. The CPLD in the diagram is used to contain any necessary glue logic.

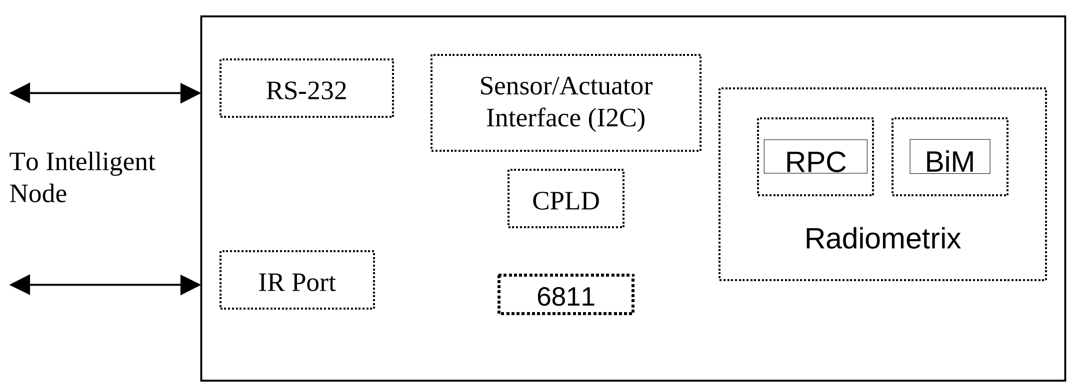

Figure 2. Block diagram of COTS sensor node

For tag nodes, the RS-232 serial port connector and IR port are not populated. The 6811 processor will be programmed (using a COTS compiler development environment that we will specify for others to purchase as well) to perform as an active tag. The participation of the active tag in the location protocols and tracking applications described in B.3 make it a location sensor node. The potential density of these active tags means they yield a very useful environment for experimentation and thus a key part of SNST, though their sensor capabilities are limited. The sensor/actuator interface shown on the diagram is an I2C fitting (like those provided for memory on motherboards). One or more sensors or actuators could be attached (by collaborators) using this very common interfacing style. One collaborator expects to use this node as the base for development of underwater temperature sensors, for instance.

The radio module is an off-the-shelf product from Radiometrix and consists of two components, a radio packet controller (RPC) and a bi-directional RF modem (BiM). The RPC is a single integrated circuit which communicates with the on-board processor (in this case the 6811). The RPC configures and controls the BiM. The RPC is responsible for packet framing and error checking.

The BiM is a single hybrid module, which contains a 433 MHz UHF modem. The BiM is designed to operate in-doors and includes a sensitive superheterodyne receiver as well as many power saving modes. The BiM communicates with the RPC via a serial interface. The BiM transmits packets received from the RPC using a small external antenna.

The COTS nodes are active tags, operating alone (tagging a dumb object, such as a piece of office equipment or military inventory) or as the front-end of an intelligent node (a handheld computer). Standing alone, they will be capable of forming limited local nets and of adapting to peers arriving or departing. They will be able to send small amounts of data slowly, but they will be able to adapt to changes in the sink for the data (simple data diffusion) with comparative ease.

These tags represent “hundreds of moving parts”, while the WINS Next Generation front-ended handhelds represent general, more capable sensor nodes. It is important, in the end, that all the nodes have the potential to be first class members of data diffusion algorithms. The multiplied power of a sensor net is enhanced to the degree that hundreds of nodes are potentially intelligent participants. All nodes in the system will participate in simple IPv6 and simple data diffusion.

The specification, some nodes, and initial code for working with the COTS system will be released to a group of DARPA collaborators at the beginning of the SNST work.

B.2 IPv6

This task will focus on provision of an IPv6 protocol stack with extensions suited to sensor IT and deeply networked systems, along with a network security prototype for these systems. Our base will be the freely distributible source code of IPv6 for Windows NT, which ISI has collaborated on with Microsoft Research (www.research.microsoft.com/msripv6). The port to Windows CE is an early deliverable of the project.

B.2.1 Extending IPv6

IPv6 is an important internetworking protocol for deeply networked systems. First, it has a huge address space, offering ample global addresses (and network identifiers). This has been cited as an attraction for sensor networks by the IEEE P.1451committee [2]. Global addresses for deeply networked systems such as dense sensor nets are desirable for a simple reason: the contact from a conventional system to the deeply networked system often needs to take place opportunistically. A sensor net example is drifting ocean acoustic or temperature sensors, which may be envisioned as sharing data with each other continuously, but relying on one or a few of their number to upload the data to a passing ship. Currently, each ocean sensor has its own satellite uplink or other communications link to the collector of its data.

Less familiar than the address supply of IPv6 is the extensibility of the IPv6 protocols, the aim being to accommodate new paradigms as they arise [3]. This extensibility is in the form of:

1) Extension headers that may be daisy chained, so that new low-level protocols are easy to incorporate
2) Extensible option syntax within the headers for autoconfiguration, neighbor discovery, and security

ISI will start with some capabilities of conventional IPv6 not found in IPv4, autoconfiguration and generalized header compression. We will meet the further paradigm challenges of deeply networked systems computing with extensions of IPv6, using the built-in extensibility of the protocol stack.

IPv6 addresses consist of 128 bits with a structure of 64 high order bits for the network identifier and 64 low order bits for the end-system identifier. Address autoconfiguration is based on building a local address (without global significance) from a conventional prefix and the end-system identifier, which is conceived to be a flat identifier from the IEEE EUI (End User Identifier) space [4]. Nodes that come up on a single physical link have plug-and-play connectivity with each other. Addresses for use beyond the immediate physical link are developed automatically in the presence of any router to another link. The router uses the multicast advertisement protocols to install prefixes in the end-systems [5].

A crucial design point for IPv6 for sensor nets is header compression. Because of the need for low power/long battery life, and the limitations of most available wireless communications media (e.g. micropower radio, sonar), bits should not be wasted. The model of this proposal is to operate primarily in an IPv6 header compression mode. IPv6 header compression [6] is a multi-access design that allows the IP header to be reduced to two or three bytes. We will design a scheme that uses full IPv6 for initialization and maintenance of the net and for global access, but not for general data transport. The maintenance of the net will use power-aware neighbor discovery (see below) to limit the cost of full IPv6 communications throughout the battery life of the nodes.

The paradigm challenges we will meet with the extended IPv6 stack (in a limited fashion on the active tags and a full fashion on the intelligent nodes with the COTS or WINS Next Generation sensors) include:

Autonomous operation –

It is necessary for deeply networked systems nodes to initialize with no in situ configuration and to adjust to changes in their surrounding network without human intervention. We propose to start with conventional IPv6 autoconfiguration and neighbor discovery for the nodes' autonomous operation. Extensions are needed to cluster the nodes into "link groups" for the link-local plug-and-play and to choose those nodes that coordinate (at the low-level) the "off-link" communications. The goal is for the sensor net to self-organize into a hierarchy of groups, formed with stochastic algorithms.

IPv6 will be extended to perform clustering based on power minimization. Nodes with the highest signal level (proximity) will form the link groups or 0^{th} level groups. Link-local addresses will be used among these, while one or more nodes in each group participates in a larger-scale algorithm to form the site-local group (named for the type of IPv6 address which is not global but allows multiple links within a site to be addressed uniquely). A single node (usually the one with best transmission conditions) will be used to maintain communication between two link groups, but many nodes should be able to perform the role in case of failure due to exhaustion of battery or other problems. The result of the operations sketched here is automatic formation of large, capable unattended nets. Only very simple clustering is performed because data diffusion algorithms above the IPv6 layer will form application-specific neighborhoods based on local multicast.

Dynamic topology and peer changes –

IPv6 maintains neighbor lists dynamically in the existing technology, on one physical link, or for a small number of nodes that are mobile relative to the whole. The traditional mobility models developed for IPv4 and IPv6 will be inadequate to represent the mobility patterns of some deeply networked systems. The SNST IPv6 stack will be developed with routing and mobility models in which diffusion-based multicast (see below) takes the place of conventional approaches. After "link groups" have been formed, their intercommunication will make use of localized multicast and limited multicast forwarding. Peer changes or changes in the group will be accommodated by the design of the algorithms, though in format they will be handled as IPv6 (or compressed IPv6) neighbor discovery events.

Power-aware neighbor discovery –

We will explore power-aware, self-stabilizing algorithms not only for minimizing the occurrence of full IPv6 headers, but also to minimize the use of power in maintaining the neighbor set for localized networking. The neighbor discovery protocol will be extended to evaluate the node-node power-distance, based on the received signal strength. We will devise an adaptive algorithm, which will determine the optimized threshold of the signal level for use by the neighbor discovery protocol. When reached, a threshold neighborhood will attain both effective aggregation and reliable, low-power transmission.

An example of power-awareness is choosing connectivity to the link group among the neighbor set whose transmissions are characterized by highest power. Another example is to perform neighbor unreachability detection, one of the functions of neighbor discovery in IPv6, only to nodes whose transmission recently were above a certain power threshold, so as to avoid wasting power on nodes that have moved to the edge of range. Neighbor discard would be a concept added to conventional IPv6 in the context of deeply networked systems. It is a viable concept because of the paradigm of data diffusion for transport in these systems (see below). Multiple paths are expected and the functional loss of nodes is well tolerated.

3.2.2 Sensor node mutual authentication

Security is an important, ground-floor requirement of sensor nets. Unattended nodes will be used in sensitive applications, for instance, in the network operating an automobile or tank. It is desirable to be able to add nodes to such a net using plug-and-play, but dangerous to allow unauthorized nodes to join. In short, extensive plug-and-play means opportunities for malicious nodes to join and subvert a deeply networked system. In the context of military environmental sensors, this is a substantial risk.

ISI proposes to demonstrate the feasibility of integral security because it is critical that the commercial world incorporate secure mechanisms into deeply networked computing as it emerges. The feasibility will be shown by two basic building blocks:

1) mutual authentication among sensor nodes
2) integrity of the network-organizing communications.

This research cannot cover prevention of hardware cracking, which is also a substantial risk for unattended military sensors, but is an orthogonal problem. The target here is only prevention of spoofing, which otherwise is made very easy by the broadcast medium on which the nodes communicate and by their dependency on each other. Dense communications are desirable for data diffusion, therefore security methods involving low probability of intercept (aside from some security obtained from the multihop transmission path) are not applicable.

The intelligent nodes in the SNST experimental environment will demonstrate (over both COTS and WINS Next Generation RF) a method of mutual authentication during autoconfiguration and self-organization. We will demonstrate effective security in which diffusion multicast forwarders are mutually authenticated with data collectors, but not all data collectors are mutually authenticated with each other, because both the extended autoconfiguration and data diffusion use this hierarchy.

We will explore refinements for sensor nets to the Initial Mutual Authentication Protocol, DARPA-funded research at the University of Pennsylvania [7]. We will adopt from that work also a reduced-sized representation of certificates, such that they fit within 256 bytes. The mutual authentication depends on loading nodes with an authorization certificate signed by a trusted third party, along with the public key of the trusted third party. The vision is that authorization certificates are loaded into the sensor nodes at the time of manufacture, much as the 64-bit end-system identifier for addressing must be loaded then. Manufacture-time security requirements must be demonstrated to industry or commercial deeply networked systems are likely to minimal security. For example, Bluetooth (www.bluetooth.com) authentication is said to be hardwired without elements that will change over time.

With the authorization certificate, a second certificate called the authentication certificate can be generated. The two certificates are used, one changing based on nonces, one fixed, to verify that node's identity with respect to its private key. In the context of sensor nets, the protocol allows recognition of peers belonging to the same cohort and recognition that peers are consistent over a series of communications. In addition, a shared secret can be generated from the initial mutual authentication and this can be used for operation in compressed mode. The adaptation of Arbaugh's algorithm to the SNST environment will require refinements, including its adaptation to the extended IPv6 autoconfiguration described above and creation of a compressed mode, perhaps using Diffie-Hellman shared secrets in lieu of the 256 byte certificates and their IPv6 authentication header frame.

B.3 Data Diffusion and Applications

B.3.1. Introduction to the use of SNST to support diffusion based applications

One very promising approach to designing robust scalable sensor nets is to view the problem (coordinated behavior among a set of dynamically changing nodes) as one of data diffusion. Data diffusion uses localized communication and aggregation to avoid sending data inefficiently in a network where efficiency may be paramount. The structure of the data network, its topology, sources and sinks, all vary, and data diffusion adapts. We will experiment with diffusion-based transport protocols, which promise robust end-to-end communication in the rapidly changing environments we have described. Diffusion protocols match request and result generation through interest gradients, caching data to optimize network performance. (See the SCADDS (Estrin-Heidemann-Govindan) submission to BAA 99-06/16 and 1998 Simple Systems ISAT study for more detailed discussion of diffusion).

While there are many outstanding design and analysis questions raised by diffusion, this particular proposal will focus on the substrate needed to support diffusion-based applications. In this way, we can support the larger research community with the building blocks for constructing scalable, robust sensor net applications. We expect many of these building blocks will be applicable to a range of distributed sensor network techniques, not just diffusion. However, at this early stage of sensor network research, it seems most useful to start out with a concrete architecture instead of one that attempts to be fully generic. As we and others in the community gain experience, we will augment our initial substrate services.

We next describe the initial application that we will use to drive our substrate development. Following this, we will discuss the network location protocol (NLP) to support this and a range of other sensor network applications. Finally, we describe the sensor network substrate libraries (SENSlib) that we plan to develop and make available to the community as a common platform for sensor network application development.

B.3.2. Object tracking application

Our driving application will be object tracking. Intelligent sensor nodes will be sometimes stationary and sometimes attached to things that move (such as human users). The events being sensed will be the status of objects within the sensing range of the sensor nodes. Status can be as simple as the presence and identity of sensed objects, or as involved as to details of remaining energy, details about the objects movement, etc. Sensor nodes will have a limited sensing range and will exchange data with other sensor nodes with which they communicate. Sensor nodes themselves will operate in power-conserving ways.

This application will force us to deal with dynamics and heterogeneity. The sensor network topology itself will be ad hoc and continuously subject to change as the sensor nodes move around the monitored environment. The change in topology is not just that of a change in the set of neighbors seen by any particular node. In addition, and perhaps more importantly, there will be wide and changing variations in the density of sensor nodes. As the sensor node density varies, the protocol approaches developed in companion work will require that the sensor nodes adapt their behavior to the varying densities. The environment itself will be continuously changing as the monitored objects change position and state. Objects will also have widely varying densities over time.

This application captures key characteristics of a range of envisioned sensor network applications. We will start by developing the basic support for object tracking, and then use this support to develop some interesting applications of object tracking. For example, we will start with a simple mapping application that will build a map at various granularities of object location. Mapping will include display of object location and efficient queries against this distributed data.

Over time, we will develop applications that monitor the more detailed status of objects and generate alerts and perhaps responses to events triggered by significant object-status changes. One use of this will be condition-based maintenance (CBM). By exploring CBM in the context of our diffusion architecture and object tracking, we will experiment with the use of aggregation and scaling. In particular, we will build support for dissemination of aggregate condition, as opposed to single device condition. The aggregate condition might be physical or logical. For example, one might want to know the condition of all equipment on the 11th floor when doing facilities surveys, whereas one might want to know the condition of all photocopiers of a particular vendor type when the repair person from that vendor is on site.

Finally, we will demonstrate distributed actuation and control with a security-related application of object tracking. In this example application we will control locks and other forms of access control based on the presence of sensitive or expensive equipment.

The next two subsections describe some general mechanisms that this sample application and others to follow will require.

B.3.3. NLP

Location awareness is a fundamental middle-layer service for many sensor network applications. While GPS is suitable in many situations, it does not support all kinds of object tracking: it requires an unobstructed antenna, it is relatively expensive ($100) compared to the price of the objects being tracked, and it has relatively limited accuracy. More importantly, GPS provides only physical location (latitude and longitude) as opposed to logical location (in the office, at home, on the highway, visiting a university). We will therefore design a Network Location Protocol to augment GPS and other commercial tracking systems.

Our Network Location Protocols will assume a number of mobile objects, some with known locations (due to on-board GPS or a fixed location). Other objects will estimate their location from these first-stratum objects through data communication and direct measurements of timing and communications signal. Thus, network location becomes analogous to time determination with the Network Time Protocol [8].

There are many dimensions of possible network location protocols. Indoor and outdoor communication have substantially different interference and accuracy requirements. A choice must be made between using existing RF-based communications for location and augmenting them with ultrasound sensors. We assume that several base stations are reachable at any time, and will compare different algorithms using this information, from the simple use of the nearest base station, using the mean or weighted mean, or considering signal propagation effects to do multilateration [9]. Finally, if specialized hardware is used (possibly sensors or high resolution clocks), what are the trade-offs to placing it at the end system or with in more central ‘‘location servers”(possibly co-located with network base stations).

These approaches allow us to determine physical locations: a node’s coordinates in 3-space. End-users don't care about something at latitude 33.97988N, longitude 118.4399W, altitude 56m; they want to know about the printer in room 1120. We will develop mappings between physical and logical locations, and provide multiple, overlapping, hierarchical coordinate systems suitable for logical location.

Physical constraints are critical in both wireless communication and deep networking. The object tracking application thus will serve as an important case study to verify that our design choices, physical constraints, and models are appropriate for real-world scenarios. Furthermore, indoor location tracking (as required for object tracking) provides substantial challenges (due to multipath effects and communications blackouts) and provides a pre-existing scheme for logical location (building number and floor).

Preliminary work suggests that network location protocols are feasible. We have investigated several algorithms (nearest, mean, weighted mean, and multilateral) over an outdoor grid of Metricom Ricochet base-stations. Base stations are statically placed in an irregular grid 350-800m distant from each other. In this environment, we compared our estimated position with true position (based on GPS) for each algorithm. As shown below, choice of algorithm has substantial effect on accuracy. The best of the relatively simple algorithms considered so far can estimate location within 100 meters with 90% confidence. This very preliminary result shows what is possible with a naïve, software-only implementation. We are currently investigating how minor hardware changes (for example, using a high-resolution clock at the base stations) can improve these results while still allowing inexpensive (software-only) mobile nodes.

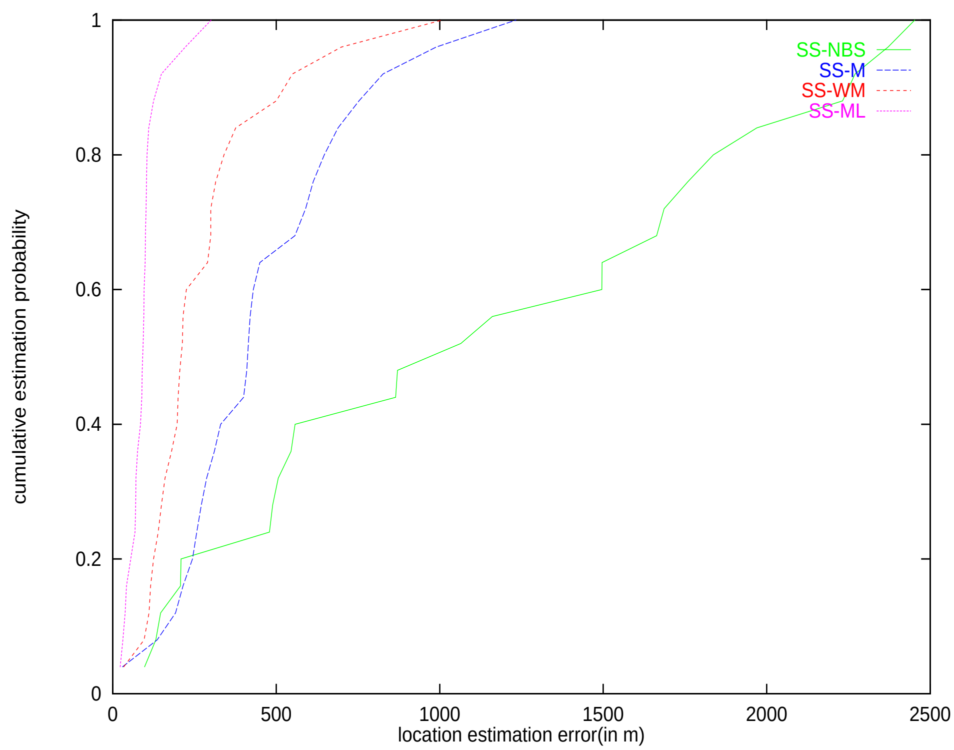

Figure 3. Average estimation error vs. base station density

We expect to use physical and logical network location through the APIs described next.

B.3.4. SENSlib

To support the experimental community, we will develop a Sensor Network Substrate Library (SENSlib) that we hope can serve a function similar to that of Berkeley sockets, promoting the composability, interoperability and reusability of the communities software developments.

SENSlib will offer a set of standard interfaces for applications to use. These interfaces will initially concern data types, attribute matching, application level neighborhood discovery, network location (logical and physical), application relevant power information and control, and simple interest propagation used for diffusion based systems. SENSlib will provide control over the underlying IPv6 network parameters, as well.

Data types will develop as applications are invented, however we can identify basic data types that we know will be needed from the outset. Location in three-space is basic to most deeply distributed systems. Time is also relevant to many applications where physical events need to be correlated. And of course there will be a range of event types and the data names will explicitly encode the modality of the data, both flat (e.g., seismic, motion, audio, infrared, etc.) and hierarchical (e.g., vehicle/truck/hummer, vehicle/boat/life-raft, etc.).

Given the multi-dimensional namespace, we will develop a flexible and efficient matching engine for use in sensor nodes. This matching will be used for local processing, caching, as well as communication with other nodes and local control actions.

Another common mechanism required by these applications will be application level, variable-scope (multihop), neighborhood discovery. While IPv6 will support network level neighbor discovery, sensor nodes of different types and with different capabilities will need to define neighborhoods that are application or data type specific. Building upon the IPv6 support, application-specific neighborhood discovery will be developed. Neighborhood definition needs to be flexible. In some cases, it may be defined strictly as nearest neighbors, or neighbors within some specific physical or logical distance. In other cases will be defined to include wider scopes.

Power management is another area in which IPv6 will support low level control, but additional application level control will be desired. Applications will require information about available power since application level behavior will need to adapt to the availability of remaining energy, or to the signal strength currently received from neighbors. Similarly, the application level will want to exert some controls over low-level behavior such as invoking header compression.

To support the directed diffusion architecture we will also develop a library to support the primitives of that architecture. In particular, an interest propagation mechanism and a gradient establishment mechanism (see SCADDS for more detail). Interest propagation is used by nodes to pull down data flows. Gradients are the abstraction used by nodes to represent the data dissemination rates and resolutions that result from expressions of interests by neighbors.

Finally, we will develop simple diagnostic tools to support debugging of SNST applications. In particular, we will begin tracing and mapping tools. Tracing will support collecting status information for a particular sequence of nodes. Mapping will operate in the background to provide diagnostic information about all currently reachable nodes. We plan to tie this into visualization tools that will be developed under the complementary proposal SEEDS (Heidemann-Fall-Estrin), also submitted to 99-06/16.

C. Deliverables

The deliverables are:

1) The SNST package, produced in snapshots, an alpha, a beta, and a final release (see Schedule of Milestones)
   a) IPv6 software
   b) SENSlib library
   c) Specification of COTS experimental nodes and their use

2) Internet-drafts describing IPv6 extensions and experience with compressed mode, power-awareness and self-organizing networks

3) Demonstrations of experiments with SNST (see Schedule of Milestones)

D. Statement of Work

ISI will perform SNST in three sub-tasks:
1. Development and experiment environment
2. IPv6
3. Diffusion routing and applications

The three sub-tasks will be performed in parallel, though there might seem to be dependency of subtask 3 on subtasks 1 and 2. The COTS hardware and basic IPv6 software are being prepared before the start of SNST. Initial data diffusion software with standard IPv4 multicast is similarly in development before the start of SNST. The result for the project is that the subtasks will be equally active and full-blown from the beginning. The early development of the SENSlib API will allow both ISI and community applications to be supported through successive refinements of the software (IPv6, diffusion routing, network location protocols), which will in turn be supported on heterogeneous hardware.

As a prototyping kit, SNST is expected to provide software and hardware for the DARPA sensor net community within 4 months of the start of project. With the early start on the subtasks that ISI has made, the COTS hardware specification and the development environment will be available for distribution at the start of the project. The first snapshot of SNST software for this environment will be provided to an initial collaboration community 3 months after the project begins. Thereafter part of the project will be maintenance of snapshots for the part of the community wishing frequent updates, along with alpha and beta releases for the broader community.

E. Schedule of Milestones

Time 0	Initial conventional IPv6 and data diffusion software. Distribution of specs for COTS sensor.
+3 months	Internet-draft on power-aware IPv6. First presentation of extended IPv6 to industry, e.g. Microsoft and Sun.
+3 months	Demonstration of sensor autoconfiguration and initial power-aware IPv6
+3 months	Publication of preliminary API for collaboration between protocol and application designers.
+3 months (as available)	Begin development work on first available WINS sensor node.
+3 months	First code snapshots to DARPA collaborators.
+6 months	Alpha release of IPv6 in Windows-CE
+6 months	Detailed description of diffusion algorithms and network location protocols to the community.
+9 months	Demonstration of remaining basic functions of IPv6 sensor nets, including opportunistic global accessibility.
+12 months	Alpha release of Sensor Net Software package to DARPA sensor net community (working snapshots having been provided for nine months).
+16 months	Demonstration of large-scale autoconfiguration with mutual authentication.
+18 months	Public Beta release of SNST package for both COTS and WINS hardware.
+18 months	Submission of extended IPv6 to IETF.

+20 months	Demonstration of sensor net and data diffusion with large population of highly mobile objects.
+27 months	Demonstration with applications of heterogeneous sensor nodes.
+36 months	Final maintenance of SNST package. Report on technology transfer

F. Technology Transfer

An important goal of SNST is to insert a dynamic, self-organizing network and data diffusion model into the development of military sensors and commercial deeply networked systems. SNST will be developed using an operating system, WIN-CE, that has a large (though not actually dominant) share of the market in handheld computers and that Microsoft targets for embedded systems. Microsoft will be an unfunded collaborator with the project, helping us to port MSRIPv6 into Win-CE from Windows NT and participating in network algorithm development.

The SNST deliverables have as one of their primary goals technology transfer. We will create the software necessary to make the emerging sensor and low-power wireless node technology readily usable by the experimental systems community. Notably, after initial alpha development, daily code snapshots will be made available to the community to encourage collaboration. In addition we will:

- Publish early drafts of APIs and other interfaces to obtain feedback from our targeted user community.
- Make our software available, including source code, and solicit feedback from our users.
- Coordinate incorporation of contributed code where appropriate and in a quality-controlled fashion.
- Work on interoperability and integration issues with other DARPA PI's.

Our research group has a long history of extensive collaboration with other researchers, members of the DARPA community, and commercial industry. We plan to continue this collaborative approach in this new area of work. We believe that collaboration in the definition, execution, and experimentation phases is critical. Other technology transfer activities include:

- Presentation at DARPA PI meetings
- Presentations at workshops and conferences
- Interaction with potential vendors of sensor and tag technology.
- Participating in IEEE P.1451 development
- Participating in (or starting) and Internet Engineering Task Force working group on extended IPv6

G. Comparison with Other Work

Hardware design

The Low Power Wireless Integrated Micro-sensors (LWIM) [10,11] project, the Ultra Low Power Wireless Sensor project [12] and the Wearable Computing project[13] among others, are involved in building devices that have the ability to perform sensing, signal processing and other capabilities in a single module.

Micro-electro-mechanical Systems, or MEMS, are integrated micro devices or systems combining electrical and mechanical components. MEMS devices can range in size from micrometers to millimeters. Reference [14] provides some information on the state of MEMS technology and its potential applications.

IPv6

Aside from the early thinking of IEEE 1451 [2] about IPv6 addresses, the proposal to build sensor nets from IPv6 autoconfiguration is unique to ISI, and is derived from our extensive experience with IPv6 autoconfiguration and neighbor discovery. An embedded network developer, Epilogue Systems, has used IPv6 for embedded devices for traffic signals, but neither in a low power nor in an autoconfiguration context (the devices are on the power grid and are centrally managed).

Neighbor Discovery [5] includes radio-oriented techniques for neighbors with hidden terminal problems.

Some relevant techniques for the site-local level of autoconfiguration (as well as for data diffusion) can be found in web caching such as the self-configuring Adaptive web caching[15] system. In this system, cache servers self-organize themselves into overlapping multicast groups. The mesh of overlapping groups form a scalable, implicit hierarchy. Similar techniques are found in the Scalable Session Messages work (http;//ee.lbl.gov).

Some applications of embedded networks

Various process control and automation tasks in factories have traditionally used networks of embedded systems known as control networks [16] to efficiently perform their monitoring tasks. The control network architecture typically consists of a centralized controller coordinating the communication between sensors and actuators connected to the controller. Many control networks are currently wired networks where the sensors are powered by a common power-line. A transition to wireless communication is expected due to the ease of deployment and maintenance of wireless networks.

Tags or transponders are simple circuits that respond to interrogations from a reader by returning some pre- programmed information. These devices find a wide range of applications in object monitoring scenarios such as anti- shoplifting and inventory tracking systems. Reference[17] provides information about the available tag technologies.

Pottie et al.[10] give a high-level systems point of view on the design of Wireless Integrated Network Sensors (WINS). They discuss the necessary hardware technology, signal processing techniques and some details of a self-organizing network architecture in the context of a military situational awareness problem. The networking aspects discussed include TDMA scheduling, automatic discovery of neighbors by nodes to form a connected topology and cooperative detection by sensors.

The Piconet[18] project at the Olivetti and Oracle Research Laboratory (ORL) is developing a prototype embedded network. Piconet is a low-rate (about 40 Kb/s), low-range (5 meters) ad-hoc radio network. Sensors can use the Piconet module to enable wireless connectivity. A low level radio protocol is presented for communication between various embedded objects that use Piconet. Some interesting applications for Piconet in the home and office environments are also discussed.

Active badges [19] are small computing devices worn by personnel for location tracking purposes. Each badge has a globally unique code that is broadcast using an infrared interface. The signal is picked up by infrared receivers placed at various points in the building. In the above fashion, the badge's owner can be tracked as the person moves around in the building. ParcTab[20], equipment tag[21] also perform similar functions. The ORL location system [22] improves on the performance of infra-red based technologes by using ultrasonic sound.

The MIT smart room project [23] and the Forest of sensors project [24] analyze data from video images to make inferences about the presence or absence of various objects or people. PinPoint Corporation has recently developed a Local Positioning System (LPS)[25] to track various assets in home or office. They use baseband radio communication between tags attached to the various objects and a set of controllers placed at various points in the building. The tags do not communicate among themselves.

Weiser[26] considers a vision of ubiquitious computing where the various computing elements are seamlessly integrated into the environment. Many types of devices will be available around us that will each be capable of adapting their behavior based on the conditions. These devices will however be so tightly integrated with the environment that they will be invisible to common awareness.

Stajano and Jones [27] present a system where users can query and control their personalised computing resources and services from their cell-phones. Their emphasis was to not require an user to carry an additional gadget to access some basic services. Instead, they focus on instrumenting the cell phone that many people carry to be able to access some basic computing resources such as checking email or stock quotes.

Ad-hoc networks

Ad-hoc networks refer to self-organizing networks of mobile wireless nodes that do not depend on any fixed infrastructure. Ad-hoc routing protocols can be classified as reactive or proactive. Proactive routing protocols continuously compute routes to all nodes so that a route is already available when a packet needs to be sent to a particular node. Reactive routing protocols on the other hand start a route computation process only when a packet needs to be sent to some other node.

Proactive routing proposals for ad-hoc networks include distance-vector[28 , 29] and link-state algorithms[30]. These proposals do not apply well to sensor networks as continuous computation of routes to frequently mobile destinations is not energy efficient.

Many of the current reactive routing protocols do not apply well to sensor networks because they flood route requests throughout the network. These proposals[31] include the Temporally Ordered Routing Algorithm (TORA), Dynamic Source Routing (DSR) and Ad-hoc On-demand Distance Vector (AODV) protocols.

The Zone Routing Protocol (ZRP)[32] and Cluster Based Routing Protocol (CBRP)[33] proactively maintain routing information for a local neighborhood while they reactively acquire routes for farther away destinations. These protocols expect to use lesser average control traffic compared to pure reactive or proactive approaches.

Core Extraction Distributed Ad hoc Routing (CEDAR)[34] establishes a virtual backbone consisting of "core"nodes. These "core" nodes are responsible for performing route computations to destinations. This scheme may not apply well to networks of energy constrained nodes since the core nodes on the backbone will be drained of energy much faster than others.

The On-Demand Multicast Routing Protocol (ODMRP)[35] uses a mesh-based rather than a conventional tree-based scheme for multicast communications in ad hoc networks. This approach improves robustness at the expense of redundant packets in the network.

H. Key Personnel

Pursuant to FAR, Subpart 9.5, the University of Southern California/Information Sciences Institute does not support any DARPA offices and therefore has no Organizational Conflict of Interest.

Allison Mankin is Computer Scientist and leader of USC/ISI Computer Network Division's outpost in Arlington, Virginia. She started her Internet research career in 1986, working on transport, congestion and scalability problems. Working initially in the Defense Advanced Research Projects Agency DARTnet testbed, she was one of the inventors of the MBONE, the several-thousand-node overlay network in which wide area multicast routing has been developed and standardized through the 1990's.

She served in the Internet Engineering Steering Group, the body that sets Internet protocol standards, from 1993 to 1997, as Area Director of Transport Services, and Co-Director of the IP Next Generation selection process. She has conducted research on IPv6 under both DARPA and NSF funding in recent years. Her publications include the book IPng (Addison-Wesley, 1996), besides papers and Rocs on IPv6 and various other topics.

In 1996-1997, she designed and launched DARPA's CAIRN (Collaborative Advanced Interagency Research Network) testbed. She has recently founded and co-chairs the Reliable Multicast research group of the Internet Research Task Force (IRTF). She is a member of the IRTF's End-to-End research group.

Deborah Estrin is a Professor of Computer Science at the University of Southern California in Los Angeles, where she joined the faculty in 1986. Estrin received her Ph.D. (1985) and M.S. (1982) from the Massachusetts Institute of Technology and her B.S. (1980) from the University of California at Berkeley. In 1987, Estrin received the National Science Foundation Presidential Young Investigator Award for her research in network interconnection and security. Estrin is a co-PI on the DARPA Virtual Internet Testbed (VINT) project and the NSF Routing Arbiter project at USC's Information

Science Institute, where she spends much of her time supervising doctoral student research.

Estrin is an active member of the IETF multicast routing related working groups and a long-time member of the End-to-End research group. She is a member of the ACM, IEEE, and AAAS. She has served on numerous panels for the National Science Foundation and the National Academy of Engineering, and is currently a member of DARPA's Information Systems and Technology (ISAT) advisory board. Estrin has served as an editor for the ACM/IEEE Transactions on Networks and the Wiley Journal of Internetworking Research and Experience, and as a member of the program committee for many IEEE INFOCOM and ACM Sigcomm conferences, and she was program co-chair of ACM Sigcomm '96.

John Heidemann is a research assistant professor at USC/ISI where he has worked since 1995. He has been conducting ARPA-sponsored operating systems and network research since 1989. His stackable filing interface is deployed in the 4.4BSD operating system. His networking research has included studies of HTTP and TCP interactions, TCP optimizations, and currently network simulation issues with the VINT project. As a part of VINT he has focused on simulation abstraction to improve scaling and extension of the simulator to new domains. He is a member of ACM, IEEE, and Usenix, is an active reviewer for IEEE/ACM ToN, SIGCOMM, NSF, and is currently co-chair of the USENIX '99 invited talks track.

John Heidemann is currently supported at 100% effort on the VINT project at ISI. He is involved in several pending proposals with DARPA, including SEEDS (PI at 75% effort with Deborah Estrin and Kevin Fall), SCADDS (co-PI at 25% effort with Deborah Estrin and Ramesh Govindan), and SNST, all in the area of deeply embedded networking.

Ramesh Govindan is a Project leader of ISI's NSF-sponsored Routing Arbiter project, PI on the DARPA-funded SCAN project, and a Research Assistant Professor of Computer Science at University of Southern California. Dr. Govindan received his Ph.D. (1992) and M.S. (1989) in Computer Science from the University of California at Berkeley, and his B.Tech. (1987) in Computer Science and Engineering from the India Institute of Technology at Madras. He also worked for two years at Bell Communications Research, Morristown, NJ. While at Bellcore, he was a co-PI on the DARPA-funding Pip project and participated actively in the IPng standardization efforts within the IETF.

I. Facilities

USC/ISI occupies 75,392 sq. ft. of space in Marina del Rey, CA, and 13,500 sq. ft in Arlington, VA (ISI-East). ISI maintains a large University research Information Processing Center (IPC). ISI has built a reputation for excellence and efficiency in both experimental computing services and production services. The IPC contains a variety of computers, from mainframes to microcomputers. The computers are networked together on an Ethernet-based local area network (100Mbps backbone and server connections, 100Mbps and 10Mbps workstation connections), and all are provided high-speed access to the Internet via the Los Nettos regional network (also managed by ISI). ISI-East has distinct information processing facilities sized to its smaller population, including T-1 access to the Internet.

The Computer Networks Division also maintains research infrastructure in three lab areas, two on the West Coast, and one on the East Coast (in Arlington). The West Coast WAN networking lab includes OC-3c ATM links provided and primarily utilized by the CAIRN project, connected to multiple Fore ASX200 ATM switches and an Ascend/Netstar Gigarouter. The East Coast WAN networking lab includes OC-3c, DS-3 and OC-12 links provided by the CAIRN and RESNETS projects, connected to a large testbed of CAIRN PC routers, Fore ATM switches and hubs, and several commercial routers. Finally, the High-Performance Networking Lab contains 5 Sun 20/71 SPARCs running SunOS, connected by 155Mbps ATM, 100Mbps fast ethernet, and 1.2Gbps Myrinet (previously called ATOMIC LAN) links. It also contains 4 200Mhz Pentium Pro PCs running FreeBSD, with the same set of interfaces, and a high-speed RAM disk with embedded processing. In addition to these labs, both coasts maintain both Wavelan Ethernet and Metricom radio wireless infrastructures for their researchers.

J. Cost Estimate

<u>Equipment Justification</u>

The budgeted equipment includes only pieces needed for development, deployment and demonstration of the SNST software and applications. PC’s are used for two purposes, Windows CE development machines and fixed location sensors in the application. Cassiopeia handheld computers with Wavelan ethernet gear communicate with the fixed PC’s and serve as mobile intelligent sensors. 200 Radiometrix radio assemblies are purchased for creation of the standalone tag sensors and for adding radio communication to the fixed PC’s and the mobile intelligent sensors. The second year duplicates the first year because we will provide equipment to DARPA collaborators during both years.

Additional Information

Bibliography

[1] J. Markoff, F.C.C. Mulls Wider Commercial Use of Radical Radio Technology, New York Times, 21 December 1998, p. C1.
[2] B.H. Lee, Embedded Internet Systems: Poised for Takeoff, IEEE Internet Computing, Vol. 2, No 2, May-June 1998, pp. 24-29.
[3] S. Bradner, A. Mankin, IPng: Internet Protocol Next Generation, Addison-Wesley, 1996.
[4] S. Thomson, T. Narten, IPv6 Stateless Address Autoconfiguration, Network Working Group, RFC 2462.
[5] T. Narten, E. Nordmark, W. Simpson, Neighbor Discovery for IP Version 6, Network Working Group, RFC 2461.
[6] M. Degermark, B. Nordgren, S. Pink, IP Header Compression. Internet Engineering Task Force Work-in-Progress, June 1998.
[7] W. Arbaugh, A. Keromytis, D. Farber, Automated Recovery in a Secure Bootstrap Process, Internet Society 1998 Symposium on Network and Distributed System Security, http://www.cis.upenn.edu/~waa/NDSS.ps.gz
[8] D.L. Mills, Network Time Protocol(Version 2) Specification and Implementation. Network Working Group RFC-1119.
[9] A.Ward, A.Jones, A. Hopper, A new location technique for the Active Office. IEEE Personal Communications, Vol.4, No. 5, October 1997, pp.42-47.
[10] G. Pottie, W. Kaiser, L. Clare and H. Marcy. Wireless Integrated Network Sensors. Submitted 1998.
[11] The WINS project. http://www.janet.ucla.edu/lpe.lwim/.
[12] The Ultra Low Power Wireless Sensors project. http://www-mtl.mit.edu/~jimg/project_top.html.
[13] Wearable Computing. http://www.cs.cmu.edu/ wearable/.
[14] MEMS Technology Applications Center. http://mems.mcnc.org.
[15] L. Zhang, S. Michel, S. Floyd, V. Jacobson, K. Nguyen and A. Rosenstein. Adaptive Web Caching: Towards a New Global Caching Architecture. In Third International Caching Workshop, June 1998. Available from http://irl.cs.ucla.edu/AWC/
[16] Echelon, The LonWorks Company. LonWorks Solutions. http://www.echelon.com/Solutions/.
[17] Transponder News. http://rapidttp.com/transponder.
[18] F. Bennett, D. Clarke, J. Evans, A. Hopper, A. Jones and D. Leask. Piconet: Embedded Mobile Networking. IEEE Personal Communications, 4(5), October 1997.
[19] R. Want et al. The Active Badge Location System. ACM Transactions on Information Systems, January 1992.
[20] N. Adams. An Infrared Network for Mobile Computers. In Proc. USENIX Mobile and Location-Independent Comp. Symp., Cambridge, MA, August 1993.
[21] A. Harter and A. Hopper. A Distributed Location System for the Active Office. IEEE Network, Special Issue on Distributed Systems for Telecommunications, January 1994.
[22] A. Ward, A. Jones and A. Hopper. A New Location Technique for the Active Office. IEEE Personal Communications, 4(5), October 1997.
[23] A. Pentland. Machine Understanding of Human Action. In Proceedings of 7^{th} International Forum Frontier of Telecommunication Tech., November 1995.
[24] W. Grimson, C. Stauffer, R. Romano and L. Lee.Using adaptive tracking to classify and monitor activities in a site. CVPR, 1998.

[25] J. Werb and C. Lanzl. Designing a positioning system for finding things and people indoors. IEEE Spectrum, September 1998.
[26] M. Weiser. The Computer for the 21st Century. Scientific American, September 1991.
[27] F. Stajano and A. Jones. The Thinnest of Clients: Controlling it all via cellphone. ACM Mobile Computing and Communications Review, October 1998.
[28] J.J. Garcia-Luna-Aceves. Loop-Free Routing using Diffusing Computations. IEEE/ACM Transactions on Networking, February 1993.
[29] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In Proc. of the ACM SIGCOMM, October 1994.
[30] P. Jacquet, P. Muhlethaler and A. Qayyum. Optimized Link State Routing Protocol. Internet Draft, draft-ietf-manet-olsr-00.txt, November 1998. Work in progress.
[31] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In Proc. of the ACM/IEEE MobiCom, October 1998.
[32] Z. Haas and M. Pearlman. The Performance of Query Control Schemes for the Zone Routing Protocol. In Proc. of the ACM SIGCOMM, Vancouver, Canada, September 1998.
[33] M. Jiang, J. Li and Y. Tay. Cluster Based Routing Protocol (CBRP) Functional Specification. Internet Draft, draft-ietf-manet-cbrp-spec-00.txt, August 1998. Work in progress.
[34] R. Sivakumar, P. Sinha and V. Bharghavan. Core Extraction Distributed Ad hoc Routing Protocol (CEDAR) Specification. Internet Draft, draft-ietf-manet-cedar-spec-00.txt, October 1998. Work in progress.
[35] M. Gerla, G. Pei, S. Lee and C. Chiang. On-Demand Multicast Routing Protocol (ODMRP) for Ad-Hoc Networks. Internet Draft, draft-ietf-manet-odmrp-00.txt, November 1998. Work in progress.