ExScal: Elements of an Extreme Scale Wireless Sensor Network

Anish Arora
Rajiv Ramnath
Emre Ertin
Prasun Sinha
Sandip Bapat
Vinayak Naik
Vinod Kulathumani
Hongwei Zhang
Hui Cao
Mukundan Sridharan
Santosh Kumar
Nick Seddon
Chris Anderson

The Ohio State University

Ted Herman
Nishank Trivedi
Chen Zhang
University of Iowa

Mikhail Nesterenko
Romil Shah
Kent State University

Sandeep Kulkarni
Mahesh Aramugam
Limin Wang
Michigan State University

Mohamed Gouda
Young-ri Choi
The University of Texas at Austin

David Culler
Prabal Dutta
Cory Sharp
Gilman Tolle
University of California at Berkeley

Mike Grimmer
Bill Ferriera
Crossbow

Ken Parker
Mitre Corporation

Abstract

Project ExScal (for Extreme Scale) fielded a 1000+ node wireless sensor network and a 200+ node peer-to-peer ad hoc network of 802.11 devices in a 1.3km by 300m remote area in Florida, USA during December 2004. In comparison with previous deployments, the ExScal application is relatively complex and its networks are the largest ones of either type fielded to date. In this paper, we overview the key requirements of ExScal, the corresponding design of the hardware/software platform and application, and some results of our experiments.

1. The Application and its Requirements

The *ExScal* concept of operation is to deploy a dense wireless sensor network "tripwire" that detects, tracks, and classifies multiple intruders of different types (such as people and vehicles) in a long perimeter region. Application of this concept is envisioned for protection of pipelines that are vulnerable to sabotage, borders between nations that are prone to illegal crossing, and areas abutting critical plants/thoroughfares that are vulnerable to terrorist threat.

The primary requirements for this application are:

1. *Low cost of covering a long perimeter over the mission lifetime*. This translates to selection of: sensing and communication modalities that have desirable range and enable low power operation, appropriate packaging, as well as node layouts that avoid nodal redundancy. Mission lifetime is 1-6 months.
2. *Accurate, timely, reliable, and robust operation*. This translates to low false alarm and loss omission rates in detection, tracking, and classification. Since the physical terrain is not assumed to be constrained, the network must deal with breaches anywhere along the long perimeter. Response must be in near real-time, within a few seconds of intruder events, over the mission lifetime, even if intrusions are random, rare or ephemeral. Quality of operation is required even if some nodes in a region are misplaced or their components fail, during deployment or operation.
3. *Low human effort*. This applies to all phases, including the placement of the nodes as well as in the operation, monitoring, maintenance, and reconfiguration of the network.

The *ExScal* project builds upon an earlier field demonstration, *A Line in the Sand* [1], where we hand emplaced one laptop base station and 90 Mica2 "motes" (with magnetometer and micropower impulse radar sensors) over a 25m by 10m grassy area. For *ExScal* to scale the *A Line in the Sand* perimeter area by 1000 to 10000 times while still meeting its complex requirements, we chose to adopt the following architecture design principles.

1. To contain cost, we design nodes that have sensing and communication ranges substantially larger than Mica2 motes; Section 2 describes these nodes: XSM (for Extreme Scale Mote) sensor nodes and XSS (for Extreme Scale Stargate) backbone communication nodes.

   Also, we emplace these nodes in a *planned topology*, specifically a *regular, hierarchical structure*, to efficiently cover the region. Tier 1 of the hierarchy consists of a grid of the XSMs, Tier 2 consists of grid of the XSSs, and Tier 3 consists of one master operator

node; Section 3 describes the detailed topology. Application services exploit knowledge of this topology for efficiently utilizing system resources.

2. To meet the desired quality, we decompose *ExScal* into *multiple subapplications*; these execute in different phases of operation; Section 4 describes these applications. Decomposition simplifies the design, allows us to configure and manage each subapplication separately (at run time if need be), and reduces operational resource requirements. Importantly, it frees us from using common services for all subapplications, instead we can use different services optimized for subapplication needs.
3. For cost-effective human manageability, *ExScal* operates by *command and control*: Tier 3 initiates, monitors, and regulates the operations of all XSSs; in turn, each XSS likewise monitors and manages a section of (normally 20-50) XSMs. The operator maintains/reconfigures *ExScal* effectively based on the feedback obtained from the network; Section 5 describes the management.

   Autonomous functions, specifically, configurable recoverability for components, tolerance to several classes of faults (often by self-stabilization), and adaptation to certain classes of variable, non-uniform environments all support containment of human effort.

ExScal **status** We started work on *ExScal* in September 2003 with a mandate to cover a 10km by 1km perimeter with 10000 sensor nodes. To this end, we designed the two types of nodes and had 10000 XSMs and 300 XSSs manufactured (these nodes are now commercially available from Crossbow). The footprint of the code we designed for *ExScal* is ∼200KB for an XSM and ∼2MB for an XSS. We designed *ExScal* scenarios for node configuration at the factory, for field marking, for node deployment at site, for network configuration in the field, for *ExScal* operation, and for network teardown.

We executed these scenarios over a two week period in December 2004, during which we collected data on field marking accuracy, deployment yield, localization accuracy, sensing performance and variability, environment data (especially wind data collected via microphones), communications and network management performance at each tier, and intruder traces. Our experiments were conducted with 1000+ XSMs and 200+ XSSs deployed over a 1km by 300m opening in a forest in Florida, USA. The scale of the final experiment was mandated by a change in the security policies of our sponsor, as a result of which *ExScal* is now being transitioned to a classified setting.

Section 6 describes some of the results of these experiments; data and other literature on the project is being made available at the *ExScal* website, http://www.cse.ohio-state.edu/∼exscal .

2. Hardware Platform

XSM We designed the eXtreme Scale Mote (XSM) [2] for *ExScal*, especially to obtain (a) increased sensing range with respect to persons and vehicles as well as increased communication range (as compared with extant motes), (b) long-lived, retaskable operation for timely detection of rare, random, and ephemeral events. Figure 1 shows a picture of the XSM enclosure and internals.

Figure 1. The eXtreme Scale Mote. The XSM circuit board has a 3"x3" footprint and the enclosure has dimensions of 3.5"x3.5"x2.5". The *one-touch* input (on/off switch) and *one- listen* output (buzzer) are mounted on the base next to the batteries. The XSM *platform* integrates an Atmel ATmega128L microcontroller, a Chipcon CC1000 radio operating at 433MHz, a 4Mbit serial flash memory, quad infrared, dual-axis magnetic, and acoustic sensors, weatherproof packaging. XSMs are commercially available under the tradename of MSP410CA Mote Security Package.

The XSM infrared and acoustic sensors are designed for low-power continuous operation and include asynchronous processor wakeup circuitry. Based on signal processing techniques which we describe in Section 4, the sensing ranges that our detectors achieve for various intruders is described in Table 1.

Sensor	Intruder	Sensing Range
Magnetometer	SUV	7 m
PIR	SUV	30 m
	Person	12 m
Acoustics	ATV	50 m

Table 1. Sensing ranges for intruders. SUV abbreviates Sport Utility Vehicle and ATV abbreviates All Terrain Vehicle.

The reliable communication range between on-the-ground XSMs typically exceeds 30m in outdoor settings (although there is variability in this range based on ground conditions, humidity, etc.). The XSM lifetime approaches 1,000 hours of continuous operation on two AA alkaline batteries. Recoverable retasking is addressed by using a grenade timer that periodically forces a system reset.

XSS As its name suggests, the eXtreme Scale Stargate [3] includes a Linux-based Stargate computer [4]. It also includes a GPS unit, which connects via the Stargate daughter card USB port; the sensor we chose has up to 10m positioning accuracy.

(a) packaged XSS

(b) deployed XSS

Figure 2. The eXtreme Scale Stargate. Each XSS has an Intel 400 MHz XScale® processor (PXA255) with 64 MB SDRAM, 32 MB FLASH, type II PCMCIA slot, USB port, and 51-pin mote connector; packaging is watertight.

XSSs communicate with each other via their 2532W-B high power IEEE 802.11b card, which is connected to a 9dBi antenna which is 1.82m long. With this setup, we observe over 700m reliable communications in the field at full power. XSSs communicate with nearby XSMs via the Chipcon CC1000 radio in a Mica2 that is connected to Stargate via a 51-pin connector.

The current required for various stargate operations is significant: 70mA for processor operation, 90mA for GPS operation, a total of 440mA when in 802.11b receive mode and 810mA in the 802.11b send mode. Given this requirement and the Stargate limitations for wake-up-on-radio and fast processor duty cycling, we chose to power each XSS by a lead-acid battery that provides 6V DC with total current draw of 105Ah, and focused our attention on the energy efficiency of the XSS protocols/programs.

3. Topology, Coverage, and Deployment

As we discussed in Section 1, *ExScal* uses a *planned* topology for node placement so as to efficiently cover the protected region. Note that since 7m is the lowest sensing range for an intruder (cf. Table 1, for magnetometer-based detection of SUVs) it is straightforward to see that no deployment of 10,000 nodes over a 10km by 500m area or of 1000 nodes over a 1.3km by 300m area can cover all points in the region.

Fortunately, *ExScal* application scenarios do not require sensor coverage at all points within the interior of the region; barrier coverage [5] is sufficient. That is, if intruders are detected multiple times soon after they enter the region — they can thus be classified and finely-tracked initially— and they remain undetected only in bounded regions in the interior —they can thus still be coarsely-tracked within the interior. We therefore deploy sensors more densely at the boundary of the region than in its interior, where we assume the high-value asset being secured lies (see Figure 3).

The "thick" line of sensors at the outer boundary of the region consist of 5 rows of XSMs. For ease of deployment, alternate rows are shifted by half the spacing between consecutive sensors in a row so as to provide close to optimal coverage.

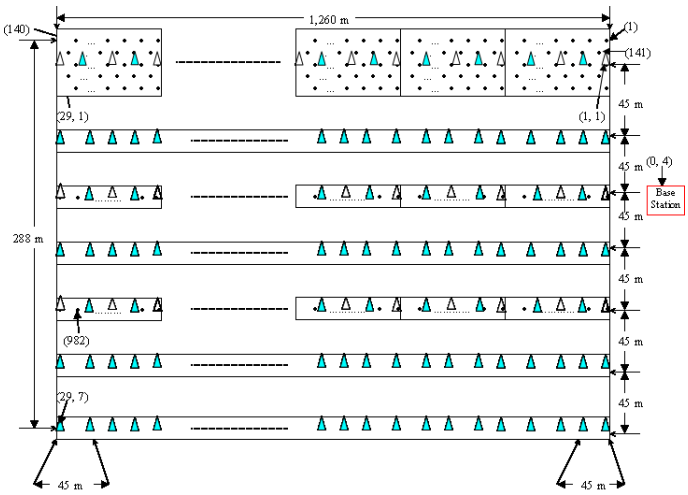


Figure 3. *ExScal* Topology. Dots represent XSMs and triangles represent XSSs. Only the XSSs represented with empty triangles are used as communication bridges between the XSM network and the base station. The tags in parentheses denote the names of devices they point to.

Coverage If any intruder crosses the thick line, it is detected by least 5 sensors; i.e., it is at least 5-barrier covered. More specifically, an SUV is detected by at least 5 magnetometer sensors and 30 PIR sensors; a person is detected by at least 10 PIR sensors; and an ATV is detected by at least 55 sensors. The net result is that even if some intruder detection messages are lost due to network unreliability, classification and fine-grain tracking of intruders is still possible when they enter the region through the thick line.

The interior of the region consists of a grid of "thin" lines, each consisting of a single row of sensors. These enable bounded-uncertainty tracking of intruders as they cross from one thin line region to another.

In terms of reliable communication connectivity, this topology ensures that each XSM can reliably communicate with 10 to 32 other XSMS in the thick line and 3-6 other XSMs in the thin lines. That said, since multi-hop reliability in the bandwidth-constrained Tier 1 network can be insufficient after 5-6 hops, we partition the Tier 1 topology into sections of 20 to 50 XSMs each and emplace an XSS in each section to serve as a communication bridge between its XSMs and the Tier 3 base station. XSSs are thus spaced 90m apart and each XSM can reach two XSSs within 5-6 Tier 1 hops. A total of 45 XSSs are needed for the Tier 2 communication backbone network.

Additional XSSs are emplaced for executing experiments to validate that the XSS network scales to meet the

original objective of the 10000 node *ExScal* network. The full peer-to-peer ad hoc 802.11b Tier 2 network consists of the 203 XSS arranged in a 7×29 grid. At full power, each XSS in this grid communicates reliably with 109 to 202 other XSSs; at low power, the numbers are substantially lower but is still sufficient to tolerate the failure of several XSS without partitioning the Tier 2 network.

Deployment For field ground truth measurement and node emplacement, after considering several alternatives involving lasers, walking meters, and ropes for triangulation, we settled on using simple surveying equipment, which not only provided us with submeter accuracy (within 0.2 m and with high likelihood within 0.1m), but also saved us both time and effort. We used a Leica Total Station 307, 3 Leica Reflectors mounted on prism poles, and a 45m nylon rope with 9m markings on it. A surveying expert helped us with marking; see [6] for details.

The submeter accuracy we achieved in marking out the locations is not necessitated by the ExScal application, since our planned topology satisfies its coverage and connectivity requirements even if the actual separation between consecutive XSMs on the ground was off by 5m (i.e., 14 m instead of the ideal 9m). However, the submeter accuracy gave us finer ground truth so as to measure the accuracy of the localization process and the effects of less accurate placements on the quality of the application.

We completed marking all locations (983 XSMs and 203 XSSs) in 27 working hours with 8 persons. Laying out the equipment took longer, about 24 working hours with 14 persons, near the marked locations.

4. Software Architecture

As we discussed in Section 1, *ExScal* meets its quality requirement by being decomposed into several subapplications. Specifically, these are a Trusted Base program, a Deployment Application, a Localization Application, and a Perimeter Security Application.

4.1. Trusted Base and Deployment Applications

At Tier 1, each XSM has a trusted base program, *Nucleus*. Nucleus includes a bootloader that executes every time a node starts and that offers an API to the running program by which to switch to another application binary (including Nucleus itself); the switch involves rebooting the XSM.

For convenience, we bundle with Nucleus the deployer response application as well as power management features. During deployment, we desire basic confirmation that each node is awake and functioning as we placed it on the ground. To this end, Nucleus exercises the sounder each time it is booted, and sends out multiple radio messages containing the node's unique identifier and network address. This serves as feedback to the deployer about when to move on and begin installation of the next XSM. Since the network deployment can take several hours, to avoid battery depletion in the interim, the power-saving sleep system in Nucleus is immediately enabled after the startup confirmation. This system uses low-power listening [7], making it possible to receive radio messages while asleep.

Nucleus also includes an application that provides node testing and network management functionality: specifically, it includes Deluge [8] for dissemination of new programs and SNMS [9] for sending commands and collection of health/status information to all XSMs in a section via their Tier 1 network.

Nucleus' dissemination component enables sending commands to all XSMs in a section, e.g., to wakeup the section from sleep mode. The wakeup command is sent as a normal message, but with a long preamble that would trigger a sleeping node to wake up. After awakening, the dissemination component in Nucleus would periodically retransmit the long-preamble wakeup message, to help wakeup the rest of the section. Nucleus' other network-based commands implement sleep, reboot, and switch.

Once a Tier 1 section is woken up, its health is determined by querying over the node identification and status over the multihop network. Queries are injected by the XSSs associated with that section into the Tier 1 network using the dissemination component described above; this forms a routing tree rooted at the XSS; a separate data collection component then returns the results to the XSS.

Finally, Nucleus uses the grenade timer to provide protection against Byzantine failure in our task-specific applications. After being rebooted by an expiring grenade timer, the Nucleus bootloader always falls back to executing Nucleus. This ensures that after a failure, the nodes are left in a known state from which we can recover the system.

4.2. Localization Application

This application assigns the grid position labels described in Figure 3 to Tier 1 and Tier 2 nodes. Our design separates the concern of manual deployment of nodes to grid locations from the concern of how nodes acquire grid positions. We use GPS to record spatial coordinates: at Tier 2 this is done by attached hardware, and Tier 2 copies the GPS data to Tier 3. For Tier 1 we do the same, however to economize on equipment, we manually record GPS data for each XSM and then upload results to Tier 3. Algorithms at Tier 3 then compute grid information to associate with each node identifier of the other tiers, and network protocols at these tiers deliver the output to each node.

Snap to Grid Imprecise physical deployment and inaccurate or missing GPS data complicate the transformation from GPS coordinates to grid positions. Errors of four or more meters can be expected for GPS readings at Tier 1,

which makes resolution of a grid with 9 meter spacing and with (say) 1 meter deployment error nontrivial. So, we use standard techniques of regression and geometric algorithms: the resulting algorithm is called *snap to grid* (Snap).

The inputs to Snap are the ideal template of GPS coordinates for the grid, with a grid label for each node, and the set of recorded GPS data and corresponding node ID. The output of Snap is a list of grid labels and the associated node ID, which is subsequently used to determine routing information, later disseminated via Tier 2 and Tier 1 protocols. Briefly, Snap works by iteratively rotating and translating a bounding box of the input set of points until the box roughly aligns with the template; then by using linear regression to obtain a refined estimate of the slope of (say) one column of the input set, a further rotation improves the alignment. Finally, translation is improved by minimizing the sum of distances between input point and its nearest template point(s), again calculated iteratively using bounding boxes. The same technique also yields the required matching between input point and template grid position.

Results Figure 4 shows typical input and output for a small portion of the grid. The plot on the left uses latitude/longitude coordinates, whereas the plot on the right uses synthetic coordinates of Snap only for matching purposes. This example has several points missing as well as errors in GPS position in the input.

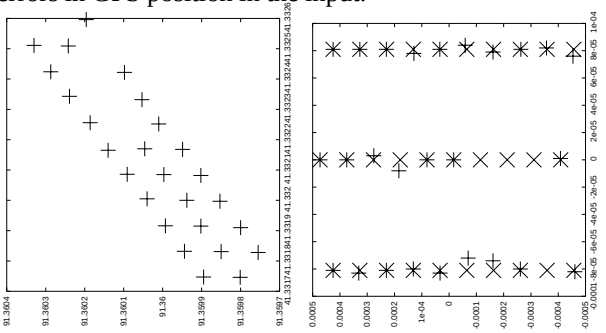


Figure 4. Left: input (plusses) with points missing; Right: matching of input with template points (crosses)

4.3. Perimeter Security Application, *OpAp*

At Tier 1, *OpAp* includes the magnetometer, infrared, and acoustic sensor chains that detect intruders; for lack of space, we describe only one of these chains below. Detection events, alongwith their timestamp (Tier 1 *OpAp* executes timesync with an accuracy of 1 millisecond) and grid location, are sent via the Tier 1 and then the Tier 2 network to the Tier 3 node, which classifies and tracks intruders.

Tier 1 Motion Detection using Passive Infrared Sensors PIR sensors are commonly used for motion detection in automatic light switch and home security products; they are made of a crystalline material whose surface charge varies in response to the received infrared radiation emitted from warm objects such as the human body.

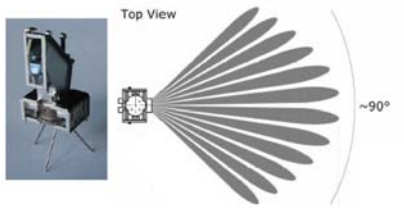
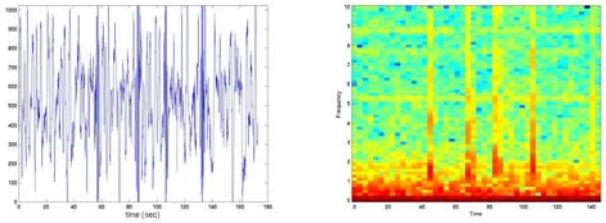


Figure 5. PIR sensor from Kube Electronics with integrated cone optics and 90 degree field of view. Four such sensors provide 360 degree coverage for the XSM.

For most indoor applications a simple analog detector circuit –typically a multi-stage amplifier and a two-level comparator– has satisfactory performance. However, comparator-based detectors produce frequent false alarms in outdoor environments due to heat drifts and sunlight. Figure 6 shows raw signal values obtained from the PIR sensor for an SUV traveling four times across the field of view of the PIR sensor at 35 km/hr. We observe that simple threshold based detectors cause false alarms even for very high threshold values. A frequency domain analysis of the same data reveals that target signature and the background variation occur in non-overlapping bands, enabling reliable detection. A human walking at a moderate speed of 5 m/sec occupies 0.5-1.5 Hz band, a vehicle traveling 35 km/hr signature is at 1-5 Hz band, whereas background variations are confined to 0-0.5 Hz band.



(a) Time Domain (b) Frequency Domain

Figure 6. Raw PIR sensor values for an SUV (35 km/hr)

To increase detector robustness, we used a polyethylene film in the PIR windows, to reduce the effect of sunlight, and a digital bandpass filter to process raw sensor values to isolate target energy from the slower background variations due to heatdrifts. Raw signal values are first passed through a band pass filter with a passband of 0.4-2 Hz. The energy of the filter output over a sliding time window is calculated by low pass filtering the instantaneous power. A low pass filter with a low cutoff frequency of 0.3 Hz is used to smooth detection events to prevent the PIR activity event from being broken into multiple events. Figure 7 depicts the output signal of the detector. We observe 12 meter reliable range for humans walking at a speed of 3 km/hr and 25 meter reliable range for a midsize SUV traveling 35 km/hr.

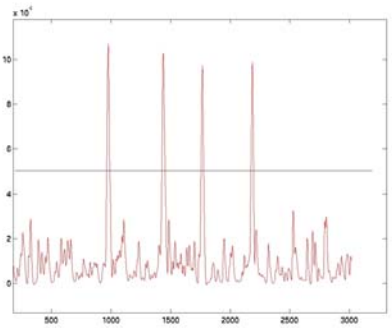


Figure 7. Output signal of PIR sensor signal chain

Tier 1 communications *OpAp* quality requires that XSM detection packets be transported reliably and in real-time to the corresponding XSS. Nevertheless, we find that with the default distance-vector routing and queue management protocols in TinyOS, only 33.7% of packets from XSMs are delivered to the XSSs on average. The causes for such low packet delivery rate include unreliable wireless links and high degree of channel contention in the presence of bursty convergecast, where a huge burst of data packets needs to be transported reliably, in real-time, and simultaneously.

To address the high packet loss rate, we use the LGR routing protocol (for Logical Grid Routing) [10]. LGR maps the Tier 1 network onto a logical grid and only uses links that are reliable in the presence of bursty convergecast. In LGR, each node locally determines, via information on its grid location, its potential parents and then distributes traffic uniformly across all the potential parents. LGR is fault-tolerant and recovers from faults quickly. With LGR, up to 81% of data packets are delivered from XSMs to the corresponding XSSs.

To further improve reliability, we use the RBC transport protocol (for Reliable Bursty Convergecast) [11]. RBC deals the loss of per-hop acknowledgments and retransmission-incurred contention as follows: To improve channel utilization and to reduce ack-loss, it uses a windowless block acknowledgment scheme that guarantees continuous packet forwarding and replicates the acknowledgment for a packet; and to alleviate retransmission-incurred channel contention, it uses differentiated contention control. It also has mechanisms to handle varying ack-delay and to reduce delay in timer-based retransmissions. Testbed experiments show that LGR with RBC delivers on average 99% of the data packets from XSMs to the corresponding XSSs in real-time, sufficing for the requirements of reliability and timeliness in Tier 1 communications.

Tier 2 communications The Tier 2 network is configured in IEEE 802.11 peer-to-peer ad hoc mode. Its architecture is composed of three reliable, power efficient transport services, namely Initd, LOF [12], and Sprinkler [13].

Initd–An Unstructured Broadcast Service: Initd is used to initialize the XSS network. Initialization consists of the Tier 3 base station contacting each XSS and collecting their GPS locations. Initd uses controlled diffusion to energy-efficiently construct a one-shot tree rooted at the base station; the tree is used in the GPS location collection.

LOF–A Structured Convergecast Service: LOF is used to transport a message from any XSS to the base station. LOF exploits geographic location information to construct a shortest path tree rooted at the base station. The metric to construct the tree is link quality in terms of delay and the geographic distance advanced towards the root. To save power and to improve estimation fidelity, LOF uses data traffic instead of beacon messages to estimate the link quality.

Sprinkler–A Structured Broadcast Service: Sprinkler is used to disseminate bulk data (up to say 200KB) to all XSSs. It exploits geographic information to construct a connected dominating set (CDS) and a transmission schedule for the XSSs in the CDS. The cardinality of the CDS is minimized to optimize the number of transmissions. The CDS XSSs use broadcasts to transmit messages; their schedule avoids the hidden terminal effect to ensure reliability and timeliness, without significant message retransmission.

Tier 3 *OpAp* logic To classify an intruder as person, SUV or ATV, *OpAp* measures for each sensor modality the *influence field* of the intruder on that sensor modality [1, 14], i.e., the area surrounding the intruder within which it is detectable by a sensor of that modality. Thus, for instance, with respect to an ATV, an SUV has a relatively larger magnetic influence field, a relatively smaller acoustic influence field, and a comparable PIR influence field.

To measure influence fields, *OpAp* aggregates detections over a time interval (typically 500msec), since multiple XSMs may detect the intruder at a given location at different times, due to differences in hardware and synchronization. It clusters spatially collocated detections of the same sensor modality in the interval, calculates cluster sizes, and classifies accordingly. The application also maintains a history of decisions made in the recent past intervals, to increase or decrease the confidence of classification. Processing of intervals lags real-time to accommodate network jitter, yet end-to-end classification of intruders is within 5 seconds.

To track an intruder, *OpAp* estimates the centroid of a convex region enveloping all of the nodes detecting that intruder. Depending on the type of intruder and its current estimated position, the tracking module also computes an expected region for the intruder location in the next time interval, based on the velocity of the intruder type. It then correlates the tracked objects from successive windows in order to construct a continuous track per intruder for the entire time it spends in the network. If the estimated location of an intruder does not lie in the expected regions of any of the currently tracked intruders, a new intruder is detected

and a new track is created. When no new information is associated with a particular target for a certain interval of time, the track is removed.

OpAp also maintains health information of the XSMs by keeping track of frequent outlier nodes and nodes that do not respond; it uses this information to refine the classification of intruders.

5. Management

ExScal uses two distinct approaches for application management. The first is a multi-tier command-and-control framework that allows its operator to perform management operations from the Tier 3 base station: operator commands are disseminated using *Sprinkler* over the Tier 2 network to a management daemon running on each XSS. Based on the command type and its scoping rules, XSSs then either locally execute the command or invoke a Tier 1 management process, such as Deluge or SNMS or *OpAp*-specific "dynamic reconfiguration". New configurations range from simple parameter updates to activating or deactivating modules such as sensor chains; an optimized version of the reconfiguration service at Tier 1 piggybacks configuration updates on routing heartbeat messages, thereby conserving network bandwidth. The results of commands, which are likewise obtained locally or aggregated from the XSM network, are then communicated by each XSS to the Tier 3 base station using *LOF*.

The second approach is one of local, autonomous management wherein each node uses several local managers to detect and correct different types of low-level faults and maintain node consistency. For instance, each XSM uses a "mode manager" to handle transitions between different application phases of *ExScal* and uses local decisions to resolve any detected inconsistencies with its neighbors' modes based on policies specified by the network operator. Each XSM also uses multiple component-level managers to monitor health predicates. For instance, the XSM *OpAp* module that aggregates detection events from the sensor chains for dispatch to the base station uses a "rate monitor" to detect violations of its contract to report only a limited number of messages per time unit; this prevents highly sensitive XSMs or XSMs in noisy environments from flooding the network with false detections. Finally, each XSM has an application manager which uses outputs from the component health managers and from a watchdog timer based monitor to detect if the application is in a persistent failure state where it would repeatedly deadlock or its components would repeatedly misbehave. In such cases, the application manager could choose to either run the application in a *safe* configuration by disabling certain modules, or simply return control back to Nucleus whereby the faulty application could be replaced. Again, such decisions are based on policies pre-specified by the operator.

6. Experiments

We measured *ExScal* network yield for realistic intruder scenarios. The end-to- end routing yield was 85.27% (with Tier 1 at 86.72% and Tier 2 at 98.32%). We found the deployment faults (at 5.37%), localization faults (at 11.4%), and reprogramming faults (at 5.5%) to all be uniformly distributed across the region. In most cases, protocol designs were primarily responsible for tolerating the faults; policy-based managers dealt with nodes that experienced "lagger" and Byzantine contract-violation faults. We found the overall Tier1- Tier2 networked sensors reliability to be 73%.

Given the significant coverage and communication redundancy in our planned topology (cf. Section 3), the spatial uniformity of faults, and our architectural design principles (cf. Section 1), we found that with this yield the *ExScal* application was able to meet its requirements at the 1000+ node scale, as was validated in the presence of persons, ATVs and SUVs traversing through its thick line and thin lines, and in management operation tests to reconfigure, change parameters, upload programs, and query healths.

We also studied the scaling of density, size, and path length in the Tier 2 network. Given the predictable patterns we observed, we believe that our hierarchical design will also meet *ExScal* requirements at 10,000+ node scale.

Acknowledgement Project *ExScal* was conceived in and funded by the DARPA NEST research program, with Vijay Raghavan serving as program manager.

References

[1] A. Arora et al. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks Journal*, 46(5):605–634, 2004.

[2] P. Dutta, M. Grimmer, A.Arora, S. Bibyk, and D. Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *4th Intl. Conference on Information Processing in Sensor Networks (IPSN'05—SPOTS Track)*, 2005.

[3] A. Arora, P. Sinha, E. Ertin, V. Naik, H. Zhang, M. Sridharan, and S. Bapat. Exscal backbone network architecture. In *Submission at Mobisys*, 2005.

[4] Intel, http://www.xbow.com/Products/XScale.htm. *Stargate Specifications*.

[5] S. Kumar, T.H. Lai, and A. Arora. Barrier coverage with wireless sensors. In *Proceedings of the Eleventh Annual International Conference on Mobile Computing and Networking (ACM MobiCom)*, 2005.

[6] N.W.J. Hazelton. Report on mark placement for node deployment, 2004.

[7] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems*, 2004.

[8] J.W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems*, 2004.

[9] G. Tolle and D. Culler. Design of an application-cooperative management system for wireless sensor networks. In *Proceedings of the Second IEEE Workshop on Wireless Sensor Networks*, 2005.

[10] Y.-R. Choi, M. Gouda, H. Zhang, and A. Arora. Routing on a logical grid in sensor networks. Technical Report UTCS TR-04-49, University of Texas at Austin, 2004.

[11] H. Zhang, A. Arora, Y.-R.Choi, and M. Gouda. Reliable bursty convergecast in wireless sensor networks. In *ACM MobiHoc*, 2005.

[12] H. Zhang, A. Arora, and P. Sinha. Learn on the fly: Quiescent routing in wireless sensor networks. Technical Report OSU-CISRC-4/05-TR20, The Ohio State University, 2005.

[13] V. Naik, A. Arora, P. Sinha, and H. Zhang. Sprinkler: A reliable and energy efficient data dissemination service for wireless embedded devices. Technical Report ExScal-OSU-EN04-2005-05-11, The Ohio State University, 2005.

[14] S. Bapat, V. Kulathumani, and A. Arora. Reliable estimation of influence fields in unreliable sensor networks. Technical Report OSU-CISRC-8/04-TR49, The Ohio State University, Computer Science and Engineering Department, August 2004.