Using and Maintaining Organization in a Large-Scale Sensor Network *

Bryan Horling, Roger Mailler, Mark Sims and Victor Lesser
Multi-Agent Systems Lab
Department of Computer Science
University of Massachusetts, Amherst, MA
{bhorling,mailler,msims,lesser}@cs.umass.edu

Abstract

In this article we describe how different types of multi-agent organizations can be used to address the challenges posed by a large-scale distributed sensor network environment. The high-level architecture is given in some detail, and empirical data is provided showing the various effects that organizational characteristics have on the system's performance. We show how partitioning of the environment can lead to better locality and more constrained communication. It is demonstrated how the size of these partitions can lead to disproportionate load on individuals in the population or increased load on the population as a whole, so this parameter must be assigned with care. Details are also provided on the creation and maintenance processes associated with the different organizational structures being used.

1 Introduction

Distributed vehicle monitoring as an example application of distributed situation assessment and more generally distributed resource allocation has been a problem studied extensively in the MAS community since its infancy [8, 4, 3]. This environment is particularly interesting when investigating issues of scale, because practical scenarios can be envisioned employing distributed sensor networks that are arbitrarily large both in number and geographic size. Each member of such a network would have some type of data producing or interpretation capabilities, resulting in a potentially overwhelming amount of information requiring analysis. In large numbers or constricted environments, the sensor population may also become constrained by shared resources or conflicting goals, creating the possibility of performance degradation when these bounds are reached or exceeded. In this paper, we propose using organizational structure to address these problems, which can appear in different forms in many different large-scale domains.

The goal of a distributed sensor network is most generally to employ a population of sensors to obtain information about an environment. In this paper, we will focus on using such a network to track one or more moving targets, although they may also be used to monitor weather conditions, traffic patterns and computer networks. Inherent in such a system are the limitations imposed by the sensors themselves. We assume no individual sensor is capable of solving the goal by itself, or else there would be little need for the network. Instead, the sensors, each of which is under the control of an agent, must collaborate in some way to achieve their common goal. In our target tracking example, the sensors' measurements consist of only simple amplitude and frequency values, so no one sensor has the ability to precisely determine the location of a target by itself. The sensors must therefore be organized and coordinated in a manner that permits their measurements to be used for triangulation. More measurements, and particularly more measurements that are taken in groups at approximately the same time, will lead to better triangulation and a higher resolution track. Additional hurdles include a lack of reliable communication, the need to scale to hundreds or thousands of sensor platforms, and the ability to operate within a real time, fault prone environment. The characteristics of the environment are covered in detail in [5].

The notion of "organizational design" is used in many different fields, and generally refers to how entities in a society act and relate with one another. This is true of multi-

*Effort sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Air Force Materiel Command, USAF, under agreements number F30602-99-2-0525 and DOD DABT63-99-1-0004. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. This material is also based upon work supported by the National Science Foundation under Grant No. IIS-9812755. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Air Force Research Laboratory or the U.S. Government.

agent systems, where the organizational design of a system can include a description of what types of agents exist in the environment, what roles they take on, and how they interact with one another. The objectives of a particular design will depend on the desired solution characteristics, so for different problems one might specify organizations which aim towards scalability, reliability, speed, or efficiency, among other things. To date, relatively little work has been done in the multi-agent community analyzing the characteristics and tradeoffs of different organizational types. We will provide quantitative results of our design to address this.

The organizational design used in this solution primarily attempts to address the scalability problem, by exploiting locality of reference and organizational constraints to impose limits on how far certain classes of information must propagate. As will be seen, the parameters guiding the creation of the organization can have a dramatic impact on the performance of the system. In our design, we use an environmental partitioning technique to create localized regions of interaction. The number of agents contained by these partitions effects how efficient the system is as a whole, as large regions may create large disparities in agent load, and small regions cause a more global increase in overhead. In sections 3 and 4 we will show quantitative evidence of these effects, and describe the tradeoffs that exist between them.

By far the most limiting resource present in the environment is the communication medium, and we will therefore use this metric to describe the effects of the organization. The radio frequency, channel-based communication module available on the sensors is distance limited, has relatively low bandwidth and throughput and provides no reliable or broadcast protocols[1]. Loss and corruption can occur when multiple messages overlap on the same communication channel, and only eight channels are available - each providing roughly the same throughput as a 14.4 modem. It is infeasible under these conditions to route all measurements from a reasonably large population to a central authority. In such an environment, the scope of agents' interactions also comes into play, because of the aggregate effects of communication events by those agents.

Thus, the goal of our organization is twofold: to facilitate gathering and interpreting measurement data, and to limit the range of interactions agents have in general. By imposing behavioral guidelines on agents' behaviors, the proposed solution will scale to large-scale populations.

2 Organizational Overview

The environment is first divided by the agents into a series of partitions or sectors, each a non-overlapping, identically sized, rectangular portion of the available area, shown in figure 1A. The purpose of this division, as will be shown below, is to limit the interactions needed between sensors, an important element of our attempt to make the solution scalable. In this figure, sensors are represented as divided circles, where each 120 degree arc represents a direction the node can sense in. As agents come online, they must first determine which sectors they can affect. Because the environment itself is bounded, this can be trivially done by providing each agent the height and width of the sectors. The agents then use this information, along with their known position and sensor radius, to determine which sectors they are capable of scanning in.

Within a given sector, agents may work concurrently on one or more of several high level goals: managing a sector, tracking different targets, producing sensor data, and processing sensor data. The organizational leader of each sector is a single sector manager, which effectively acts as a hub within a nearly-decomposable hierarchical organization, by serving as an intermediary for much of the local activity. For example, they will generate and distribute plans (to the sensor data producers) needed to scan for new targets, store and provide local sensor information as part of a directory service, and assign target managers. They also concentrate nonlocal information, such as target disambiguation and sensor status data, facilitating the transfer of that knowledge to interested parties. Individual track managers initially obtain their information from their originating sector manager, but will also interact directly, though less frequently, with other sector and track managers, and thus do not follow a fixed chain of command or operate solely within their parent sector as one might see in a fully-decomposable organization. Track managers will also form commitments with one or more agents to gather sensor data, but this relationship is on a voluntary basis, and that gathering agent's behavior is ultimately determined locally.

To see how the organization works in practice, consider a scenario starting with agents determining what sectors they can affect, and which agents are serving as the managers for those sectors. In Figure 1, these sector managers are represented with shaded inner circles. Once an agent recognizes its manager(s), it sends each a description of its capabilities. This includes such things as the position, orientation, and range of the agent's sensor. The manager then has the task of using this data to organize the scanning schedule for its sector. The goal of the scan schedule is to use the sensors available to it to perform inexpensive, fast sensor sweeps of the area, in an effort to discover new targets. The manager formulates a schedule indicating where and when each sensor should scan, and negotiates with each agent over their respective responsibilities in that schedule (see Figure 1B). The manager does not strictly assign these tasks - the agents have autonomy to decide locally what action gets performed

[1] A limited broadcast capability does exist, which can reach all sensors listening on a single channel. It is not possible in this architecture to broadcast a single message to agents which are using different channels.

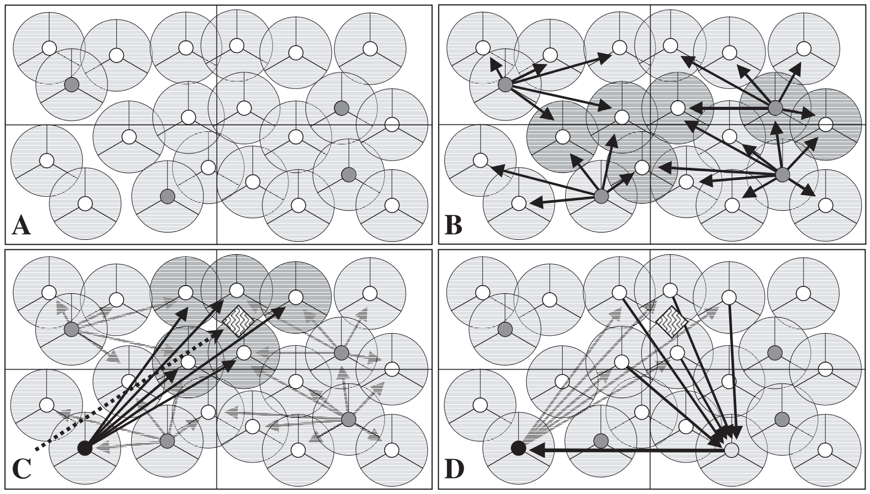

Figure 1. High-level architecture. A: sectorization of the environment, B: distribution of the scan schedule, C: negotiation over tracking measurements, and D: fusion of tracking data.

when. This is important because sensors can potentially scan in multiple sectors; thus there is the possibility that an agent may receive multiple, conflicting requests for commitments from different sector managers. The agent's autonomy and associated local controller permit the agent itself to be responsible for detecting and resolving these conflicts. If one receives conflicting requests for commitments, it can elect to delay or decommit as needed. Shaded sensors in the previous figure show agents receiving multiple scan schedule commitments.

Once the scan is in progress, individual sensors report any positive detections to the sector manager which assigned them the scanning task, which can then be used to spawn a new track manager. Internally, the sector manager maintains a list of all local agents that currently perform the role of track manager, and location estimates for the targets they are tracking. This location estimate is used to determine the likelihood of the positive detection being a new target, or one already being tracked. If the target is new, the manager uses a range of criteria to select one of the agents in its sector to be the track manager for that target. Not all potential track managers are equally qualified, and an uniformed choice can lead to very poor tracking behavior if the agent is overloaded or shares communication bandwidth with other garrulous agents. Therefore, in making this selection, the manager considers the agents' estimated load, communication channel assignment, geographic location and activity history. Ideally, it will select an agent which has minimal channel overlap, is not currently tracking a target, but has tracked one previously. This will minimize the potential for communication collisions, which occur if two agents on the same channel attempt to send data at the same time, but maximize the potential amount of cached organizational data the agent can reuse. As we have seen previously, this notion of limited communication is an important motivating factor and recurring theme in this architecture which contributes to the organizational structure, role selection, protocol design and the frequency and verbosity of communication actions.

The assigned track manager (shown in Figure 1C with a blackened inner circle) is responsible for organizing the tracking of the given target. To do this, it first discovers sensors capable of detecting the target, and then negotiates with members of that group to gather the necessary data. Discovery is done using the directory service provided by the sector managers. One or more queries are made asking for sensors which can scan in the area the target is predicted to occupy. The track manager must then determine when the scans should be performed, considering the desired track fidelity and time needed to perform the measurement, and negotiate with the discovered agents to disseminate this goal (see Figure 1C). As with scanning, conflicts can arise between the new task and existing commitments at the sensor, which the agent must resolve locally. The source of a given commitment can identify how important its task is to it, which is normalized in such a way that it has the correct importance relative to others in a more global sense. For instance, if a track manager determines that a sensor is particularly useful, based either on its location relative to the estimated position of the target or the scarcity of viable alternative sensors, this can be reflected in the importance value of the commitment. These importance values then allow the local agent to effectively discriminate among conflicted tasks with an eye towards global social welfare.

The data gathered from individual sensors is collected by an agent responsible for fusing the data and extending the computed track (see Figure 1D). The different measurements are used in a triangulation process, where amplitude and frequency values can place the target's location and heading relative to their source sensor, and several of these relative values can be combined to triangulate an absolute position. In a general sense, this data fusion agent could be any agent in the population able to communicate efficiently with both the data sources and the ultimate destination of the tracking data. However, the data fusion itself is fairly lightweight in this application, and thus does not benefit from distribution for load balancing purposes, and transferring the fusion data results in an additional delay while it is being communicated to the track manager. Therefore, our organization assigns this fusion task to the track manager itself, which avoids this delay with relatively little overhead. If the data values returned are of high enough quality, and the agent determines those measurements were taken from the correct target, then they are used to triangulate what the position of the target was at that time. This data point is then added to the track, which itself is distributed back to the track manager to be used as a predictive tool when determining where the target is likely to be in the future.

At this point the track manager must again decide which agents are needed and where they should scan. Under most situations, the process above is simply repeated. However, if the target has moved far from where the track manager is, the track managing task may be *migrated* to a new agent in a different sector. This is done to avoid the penalty associated with long-distance communication, which may cause unwanted latency or unreliability transferring information. This technique is covered in more detail in section 4.3.

3 Organizational Types

Several different types of organizational constructs are used in this system, each with different behavior appropriate for some specific task, and each designed to meet our scalability requirements. As will be shown, an integral part of each of these structures is the notion of locality. Information propagates and is made available to only the agents which have need of it. In some cases, such as with the environmental sectorization, artificial boundaries are created to produce the notion of locality at the expense of time or flexibility. In other cases, as with target tracking, information locality is exhibited naturally through the domain.

This class of organization reflects the behaviors at the "agent level" of the sensor's software, and affects the roles and goals taken on by the sensor. Lower level organizations, which govern such characteristics as communication protocol details and location discovery[1], are also important but beyond the scope of this article.

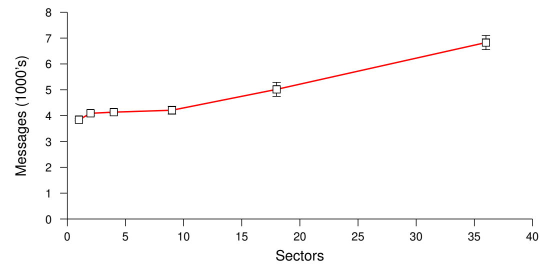

Figure 2. Affect of sector size on messaging.

3.1 Geographic Organization

The most obvious structure is the partitioning described above, which forms an organization based on the geographic location of the sensors. Because much of the information being communicated is contained within sectors, the size and shape of the sector has a tangible effect on some aspects of the system's performance. If the sector is too large, and contains many sensors, then the communication channel used by the sector manager may become saturated. If the sector is too small, then track managers may spend excessive effort sending and receiving information to different sector managers as its target moves through the environment. We initially hypothesized that a reasonable sector would contain from 6 to 10 sensors, although the physical dimensions of such a sector depend on the density of the sensors, and in different environments one would need to take into account sensor range, communication medium characteristics and maximum target speed. In the following sections, we will discuss and show empirical evidence describing these characteristics, and in section 5, we will show how this evidence supports our initial hypothesis.

Figure 2 shows some of the effects of varying partition size. In these experiments, a group of 36 sensors were organized into between 1 and 36 equal-sized sectors with 4 mobile targets. The results were observed over 10 runs per configuration in a simulation environment which closely models the performance of physical sensors. The graph shows that a larger numbers of sectors (and correspondingly fewer sensors in each sector) can significantly increase the amount of communication traffic, primarily because of overhead from additional control messages. We will see in Figure 5 how individual message types contribute to this increase.

Partitioning the environment reduces the amount of communication and processing agents must perform for several different tasks. For example, generating a coherent scan schedule is simplified by only taking into account a tractable number of sensors. Similarly, when a new target is detected during scanning that information is sent to only the appropriate sector manager, which can determine directly if it is

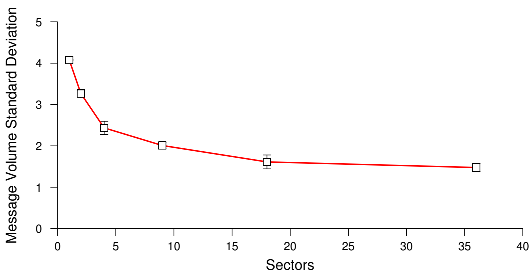

Figure 3. Affect of sector size on agents' communication disparity.

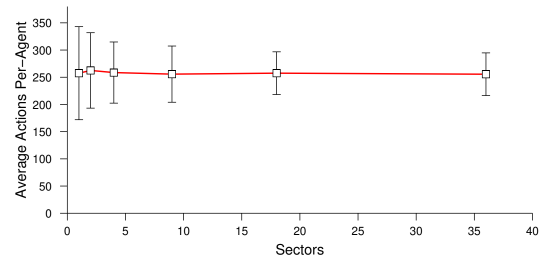

Figure 4. Agents' activity relation to sectors.

a new or existing target based on local information.

As a side effect, partitioning does reduce the system's reactivity, because an extra step may be required to fetch information that is not available locally. Instead of obtaining information about all sensors from a single source, for example, a track manager must perform several queries to obtain information as its target moves. We cope with this problem wherever possible by caching such data to avoid redundant queries, and by assigning new roles whenever possible to agents which have served that same role in the past, to take advantage of that cached data.

3.2 Functional Organization

The varied assignment of roles forms a different, functional organization [2] in the system. Agents specialize their functionality in order to restrict the type of interactions which must take place between agents. For example, to obtain information about available sensors, a track manager must only contact the relevant sector managers, which act in that capacity as an information broker [9]. Concentrating the track management functionality into individual agents serves a similar role, by limiting the number of interactions necessary to resolve conflicts that may arise in sensor usage.

Interestingly, although this type of functional decomposition does reduce the total number of interactions an agent might need to make, it can also increase that number for particular individuals in the environment. For example, we have seen how the sector manager is responsible for disbursing information about the sensors in its sector, thus providing a single point of contact for such data. However, by serving in this capacity, it makes itself a center of attention, which can adversely affect its overall performance.

Consider Figure 3, which shows how sector size affects the standard deviation in communication activity that individual agents exhibit. This metric attempts to capture how broadly agents in the population differ in their communication habits. A population where all agents are roughly the same will have a low deviation, whereas a population that has a handful of outlier agents with significantly higher message traffic will have a high deviation. With fewer sectors, and more agents in each sector, this graph shows a marked increase in disparity, representing a situation where a handful of agents are communicating much more than their counterparts. Therefore, as the sector sizes scale, specialized agents can become "hotspots" of activity, which, in this environment, can lead to significant data loss as the communication channel becomes overloaded. Thus, in comparison with Figure 2 which shows the average *total* communication, we see a tension between sector sizes, as smaller sectors lead to increased message traffic, and larger sectors can imbalance load in the population.

A similar phenomena is observed in the activity of the agents. Figure 4 shows how the deviation among individual agents' activity decreases with sector size, while activity of the population as a whole is relatively unaffected. We did not observe performance degradation because of increased sector size, even in individual members of the population (such as sector managers) which concentrated activity and thus might suffer from excessive load in certain scenarios. This is because the load incurred by the actions and communication activity was relatively low compared to the time provided for them. We expect that an increase in more computationally expensive actions would cause performance to degrade or improve as a result of changing sector characteristics.

Although not required in the scenarios presented here, it is interesting to note the applicability of this organization to situations where agents have an additional limitation or attenuation of communication capability based on the geographic distance separating the participants. In this case, sectors, or the sector managers themselves, could serve as the basis of an ad-hoc network, where messages are routed from one region to the next, using the organizational structure as a guide, until they reach their destination. This further emphasizes the intuitive notion that "local" communication is more efficient, and the locality of information should be exploited by the organization.

3.3 Peer-to-Peer

Track managers form a different, peer-to-peer based organization among themselves. As their respective targets increase in proximity, track managers will "discover" one another and learn which sensors they use. This information is first used to passively avoid contention by simply not requesting commitments from sensors that are known to be in use. If such contention is unavoidable, the track managers communicate as peers to resolve their conflict in a mutually agreeable way. A complex resource allocation protocol, called SPAM, was developed for this process. More details about this protocol can be found in [6].

3.4 Hierarchy

A fourth organization type exists, in the form of a manager-worker hierarchy between track managers and the sensors they intend to use. In this case, the managers send commitments to these sensors, requesting measurements at a particular time and place. The sensors uniformly accept these commitments without complaint, but still have sufficient autonomy to address unresolved conflicts among commitments locally, if they exist. A similar hierarchy also exists between sector managers and sensors, which facilitates scan task assignment and information disbursal.

In other environments, characteristics other than communication may be important. Local computational power, heterogeneous agent capabilities, and other shared, but limited resources can present constraints that can be addressed through organizational constraints. To some extent, we do address bounded rationality through the functional organization; agents attempt to separate roles that are known to be computationally expensive for load balancing purposes. However, in this distributed sensor network, such costs were largely negligible when compared with the effects of constrained, large-scale communication issues.

4 Maintaining Organizations

Although we have seen how the organizations above can be effective in their own right, there are costs associated with creating and maintaining these structures, and one must be sure that these costs do not significantly degrade their benefits. These costs differ from those described in the previous section in that they are more dependent on the dynamics of the environment and organization.

4.1 Partitioning

Thus far, and for the experiments described in this article, the partitions created by the agents are static and predefined. The costs associated with this structure are relatively minimal, as the agents need not discover what sector they belong to, or maintain that information over time. This assumption is overly restrictive for a realistic environment where agents do not necessarily know their position or the names of their sector managers a priori. It also becomes impractical to hand-generate such an organization as the population scales, or if newly arriving or malfunctioning agents make the population dynamic. Therefore, our eventual goal is to move to more arbitrary configurations and dynamic construction of organizational relationships.

In [7] Sims, Goldman, and Lesser discuss more recent simulation work enabling agents in a sensor network to self-organize into nearly decomposable sectors. In this work, the assumption that sectors are non-overlapping, identically sized, rectangular divisions of the area is removed. Instead, a sector's boundaries are defined to be the intersection of the circular viewable areas of each sensor within the sector. The implication of this is that the borders of adjacent sectors most likely overlap. The goal of self-organizing then is to minimize the overlap between the sectors while maximizing the coverage they provide. In other words, a sector that is responsible for tracking a vehicle should be able to track that vehicle while it is in the sector without requiring sensor data from agents outside of the sector.

There are three basic components to the self-organization process: maintenance, discovery, and negotiation. The maintenance process must occur throughout the life of the system since sector managers must make sure that the members of their sectors still exist, and the members must make sure that their managers still exist. In our approach, each member of a sector periodically sends a brief message to its manager. If the manager does not receive a message from a member, the manager assumes the agent is no longer a member of the coalition and adjusts its evaluation of the coalition accordingly. Likewise, the manager periodically sends a message to each of its members. If the member does not receive a message from its manager, that member assumes the manager is no longer active as a manager and joins the nearest coalition to it (as if it were entering the system for the first time).

Discovery occurs when agents become active in the environment, since an agent must make itself known and learn of other agents near it before it can work cooperatively with them. To make this happen quickly, an entering agent joins the nearest sector to it, by listening for beacons on the channel assigned to managers. If there is no sector manager within range, the agent elects itself manager of a new sector and begins attracting entering agents to it by broadcasting a periodic beacon. Because the agents become active asynchronously, this discovery phase results in a very rough division of the area into sectors.

The goal of the negotiation component is to refine the rough sectors formed by the discovery process into the nearly decomposable sectors needed for the organizational structure. We have implemented several negotiation protocols to enable sector managers to exchange sensor agents, and redefine sector boundaries accordingly. Because the discovery process often results in numerous small sectors, the negotiation process results in many fewer sectors ideally composed of eight sensors each.

To make this work, each sector manager has a utility function that is dependent on the physical size of its sector, the number of agents in it, and how well the sensors within the sector provide coverage of the region within its boundaries. In the most promising protocol, sector managers attempt to transfer sensor agents from one sector to another in such a way that the sum of the marginal change in utility of one manager and that of the other is maximal.

Our results show increases in global utility (defined as the sum of each sector manager's local utility) as high as 70% from the initial division of sectors to the partitioning after organization. Also, according to our model of message traffic, the vast majority of messages occur within sectors rather than between them. Finally, the system shows the ability to adapt to environmental changes such as the loss of sensor agents by reorganizing with the resulting set of agents after the failure is discovered.

The process of self-organization is especially important as the number of agents in the system and the physical size of the environment increase. It enables the agents to find an efficient partition of the region into sectors through a localized, iterative process and focuses communication to within a small area compared to the size of the entire region.

4.2 Measurement Collection

The most dynamic organization in the environment is the manager-worker hierarchy formed between track managers and sensors, because the specific sensors working for a particular manager change continuously as the target moves. This results in an increase in control messages needed to maintain the organization. For example, as the target moves into part of the environment the manager is not familiar with, the track manager must send a directory service query to the sector manager of that area to discover which sensors are available. Once those sensors are found, additional messages are needed to create and maintain data collection commitments with them. Finally, as the target is tracked, the relevant, nearby sector managers must be notified of the target's estimated position.

Figure 5 provides a more quantitative view of this overhead. It shows that as the number of sectors increase additional directory service and tracking control messages are necessary. This is intuitively true, because there are a greater number of sectors which must be interacted with as the environment changes. Somewhat unexpectedly, the

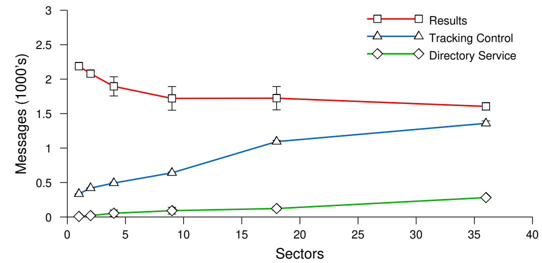

Figure 5. Sector size vs. message types.

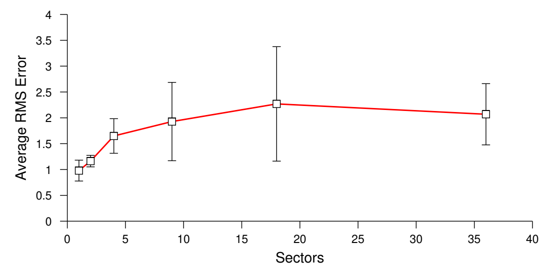

Figure 6. Effect of sector size on RMS error.

number of measurements also decrease as the sector sizes decrease. This is also due to communication control overhead; the increased time spent by the manager interacting with the additional sector managers competes with the time which needs to be spent interacting with the data collection sensors. This is caused primarily by the sensor discovery phase, and the higher probability of track manager migration. This decrease in the number of measurements affects the RMS error of the tracking process, as shown in figure 6. With fewer measurements, we see an increase in the average and variability the track's error.

4.3 Track Manager Migration

The technique of migrating the tracking responsibility through the agent population as the target moves is another aspect of local information exploitation. It should be clear that, lacking the capacity for movement, the initial manager selected to track a target will gradually become less effective as the target moves away from it. Simple signal latency and attenuation conspire to make communication over distance less reliable. Therefore, when a target and the sensors needed to track it are far enough away from the track manager, the track manager contacts the closest remote sector manager and hands off the tracking responsibility. By migrating this task to follow the target, the organization is able to retain locality despite the fact that the sensors themselves are immobile. The observable result from this practice is a

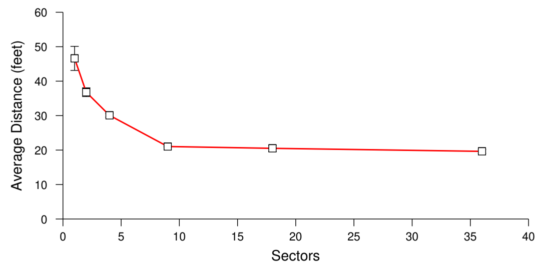

Figure 7. Distance of communication.

reduction in the distance that messages associated with the tracking effort must travel.

Figure 7 shows the effects track manager migration has on the average distance of communication. A track manager will opt to migrate its task when the target has moved away from the track manager's parent sector(s). When there are fewer sectors in the environment, and therefore individual sectors are larger, the tracking task will migrate less frequently. Thus, in the graph we see that a lower average communication distance and better locality is observed when sectors are smaller. Recall, however, that Figure 5 showed that smaller sector sizes also resulted in fewer total measurements and a greater control message burden, so there is again a tension between the extents of the partition size characteristic.

5 Results

Our experiments suggest a tradeoff exists between the overall volume of message traffic and the distribution of this volume over the agent population. Message volume is decreased when there are fewer sectors, and fewer interactions are needed to obtain information and meet objectives, as shown in Figure 2. However, by having more communication over fewer interactions, individual agents may then incur a disproportionate communication burden, as shown in Figure 3.

Figures 5, 6, and 7 show the maintenance of these organizations has a similar tradeoff, since larger sectors require a lower control overhead and better RMS error, while smaller sectors allow track migration to take advantage of information locality. By searching for a common inflection point in these results, we can conclude that a sector size between 4 and 9 is most appropriate. This supports our initial hypothesis that sector sizes between 6 and 10 were "reasonable".

6 Conclusions

The system presented in this paper uses several different organizational paradigms to address challenges posed by a distributed sensor network problem. The primary structure consists of a partitioned environment, where each partition contains sensors managed by agents that are further organized by function. Depending on an agent's function, or role, it will take part in other organizational constructs, using a peer-based or hierarchical organization scheme. Locality and constrained communication are exploited for a scalable solution in a bandwidth-limited environment.

The quantitative conclusions presented here are quite domain specific. They depend on the communication characteristics of the environment, the actions needed to achieve the scenario goals, and the behaviors exhibited by the agents managing the sensors. However, we feel that the types of issues raised by these experiments, such as information locality, specialization bottlenecks and organizational control overhead, are applicable to many different domains, particularly those which are communication intensive. More generally, we can conclude that such organizational parameters can have significant effects on performance, making them interesting candidates for further research.

References

[1] L. P. Clare, G. Pottie, and J. R. Agre. Self-organizing distributed microsensor networks. In *Proceedings of SPIE 13th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls (AeroSense)*, April 1999.

[2] M. S. Fox. An organizational view of distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1):70–80, Jan. 1981.

[3] V. Lesser and D. Corkill. The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. *AI Magazine*, 4(3):15–33, 1983.

[4] V. Lesser and L. Erman. *IEEE Transactions on Computers Special Issue on Distributed Processing*, C-29(12):1144–1163, December 1980.

[5] V. Lesser, C. Ortiz, and M. Tambe. *Distributed Sensor Networks: A multiagent perspective.* Kluwer Publishers, 2003.

[6] R. Mailler, V. Lesser, and B. Horling. Cooperative negotiation for soft real-time distributed resource allocation. In *In Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, Melbourne, AUS, 2003. Also submitted as UMass Computer Science Technical Report 2002-49.

[7] M. Sims, C. Goldman, and V. Lesser. Self-organization through bottom-up coalition formation. In *Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, 2003. To appear.

[8] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transctions on Computers*, 29(12):1104–1113, 1980.

[9] K. Sycara, K. Decker, and M. Williamson. Middle-agents for the internet. In *Proceedings of IJCAI-97*, January 1997.