A Case Study of Organizational Effects in a Distributed Sensor Network *

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

Abstract

We describe how a system employing different types of organizational techniques addresses the challenges posed by a large-scale distributed sensor network environment. The high-level multi-agent architecture of real-world system is given in detail, and empirical and analytic results are 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, as well as disproportionate load on individuals or increased load on the population as a whole. The presence of such tradeoffs motivates the need for a better understanding of organizational effects.

1. Introduction

Distributed vehicle monitoring as an example application of distributed situation assessment and more generally distributed resource allocation has been studied in the MAS community since its infancy [5, 2]. 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, making purely centralized control inefficient. Each network member would have some type of data producing or interpretation capabilities, resulting in a potentially overwhelming amount of information requiring analysis. Shared resources and potentially conflicting goals add further complications. These challenges make it an ideal candidate for multi-agent techniques.

We propose using organizational structures to address these problems, which can appear in different forms in a variety of domains. This belief is based on our experiences working with a large-scale, realistic distributed sensor network over the past four years, both in detailed simulations and on real hardware [3]. Rather than employing a single organizational scheme, we have found that exploiting the strengths of a collection of heterogeneous organizational styles can be quite effective. By varying just one aspect of such an organization, we will show that the performance of the system can be greatly influenced by the organization's design parameters. We will present examples of such effects, and the methods used to discover and analyze them.

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 targets which move along arbitrary paths in an area. In this work, we use detailed sensor models based on a three-head, MTI Doppler radar system [3]. No individual sensor is capable of solving the goal by itself, or else there would be little need for coordination. 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, and geographically distinct groups of such coordinated sensors used to produce a continuous track. More measurements, and particularly more measurements 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, uncertain environment. This environment is covered in detail in [3].

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

* Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Air Force Materiel Command, USAF, under agreement number F30602-99-2-0525. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. 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.

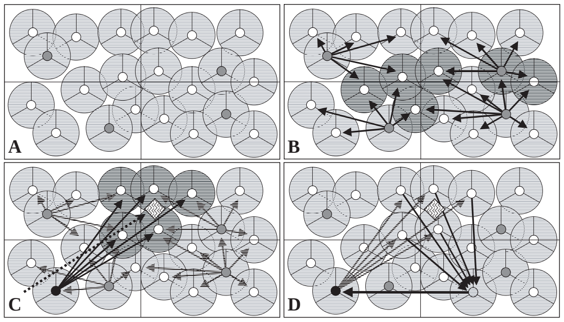

Figure 1. High-level architecture. A: sectorization of the environment, B: distribution of the scan schedule, C: negotiation over tracking measurements, D: tracking data fusion.

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 classes of both control and data messages propagate. The environment's most limiting resource is the wireless communication medium, and we will therefore use this resource to describe the effects of the organization. Our design uses environmental partitioning to create localized regions of interaction, called sectors. Within these sectors, agents take on different responsibilities which dictate their individual behaviors. The number of sensors in these sectors affects how efficient the system is, as large regions may create unwelcome disparities in communicative or processor load, and small regions cause a more global increase in overhead. Specifically, we will see how sector size affects the overall communication load, load disparity between agents, average communication distance, and the quality of tracking.

2. Framework Overview

The tracking environment consists of a number of sensors arranged in an area. Each sensor is equipped with a processor, on which is run an agent process that controls the sensor. The sensors are connected with a FM-based wireless network, which is divided into eight communication channels. Each channel has limited capacity, and agents may communicate over only one channel at a time, so the assignment of agents to channels can affect the performance of the system as a whole. As targets move through the area they will be tracked by the sensor-controlling agents as outlined below. We provide an architectural overview below, and a detailed description of the entire framework be found in [3].

There are three types of responsibilities which exist in the framework: *sector manager*, *track manager* and *sensor manager*. A sector manager will be created for each sector in the environment, which serves as an intermediary for much of the local activity. For example, they will generate and distribute plans needed to scan for new targets, store and provide local sensor information as part of a directory service, and assign track managers. Each detected target will have such a track manager, which is responsible for identifying the sensors needed to gather target information, gathering the resulting data, and fusing it into a track. Track managers obtain some information from their originating sector manager, but can also interact directly with other sector and track managers. The sensor manager controls how the local sensor is used. In response to sector or track manager requests, it will take measurements in particular areas at specified times and deliver the measured data. Each of these three responsibilities corresponds to a *role* in the organization, which must be assigned to a particular agent. Agents may work concurrently on one or more of these roles, so a viable organizational design must ensure that each agent has sufficient resources to meet the combined demands of the roles it is assigned.

To see how the organization works in practice, consider the scenario in Figure 1. The environment is divided by the agents into a series of sectors, each a non-overlapping, identically sized, rectangular portion of the available area, shown in Figure 1A. The purpose of this division is to limit the interactions needed between sensors. Each sensor has a local agent, and all agents take on a sensor manager role. A single agent in each sector also takes on the sector manager role, represented by shaded inner circles. Each sensor manager starts by recognizing its sector manager, and sending it a description of its capabilities (position, range, etc.). The sector manager then uses this information to generate scanning schedule for detecting new targets in its sector, which it disseminates in Figure 1B. These tasks are not strictly assigned - the agents have autonomy to resolve conflicts and decide locally what action gets performed when.

Once the scan is in progress, individual sensors report any positive detections to the sector manager which assigned them the scanning task. Internally, the sector manager maintains a list of track managers currently in its region and location estimates for their targets. If the positive detection does not match any of these targets, the manager selects an agent in its sector to be the track manager for that target. Not all potential track managers are equally qualified, and an uninformed choice can lead to very poor tracking behavior if the agent shares communication bandwidth with garrulous agents. Therefore, in making this selection, the manager considers each of its agents' estimated load, communication channel assignment, geographic location and history.

The assigned track manager (shown in Figure 1C with a blackened inner circle) is responsible for tracking 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. The track manager then determines where and when the data should be collected, and negotiates with the appropriate sensor managers (see Figure 1C). As with scanning, conflicts can arise between the new task and previously existing commitments, which the agent must resolve locally or elevate to the conflicting managers. This process is described in detail in [4].

The data produced by the sensors must be collected and analyzed (see Figure 1D). Although this activity is logically a separate role, it is a relatively lightweight process, and as a simplification our organizational design adds it to the track manager's responsibilities. Once the track manager has received the measurements, the data are used in a triangulation process. 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 derive an absolute position. The data point is then added to the track, which is used as to predict the target's future location.

At this point the track manager must again decide which sensors 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 role 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. Organizational Types

Below we describe two of the organizational constructs used in this system, geographic coalitions and functional differentiation. The system also uses structures with characteristics similar to peer-to-peer and hierarchical organizations. An integral part of each of these 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 encourage locality at the expense of time or flexibility. In other cases, as with target tracking, locality is exhibited naturally through the domain. There are many data flows in this framework which affect its quantitative characteristics. We will informally describe them as needed in the following section, and provide more concrete descriptions in Section 6.

3.1. Geographic Coalitions

The partitioning described in Section 2 forms an organization based on the geographic location of 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, affecting both the manager and any other local sensors which use the same channel. If the sector is too small, then track managers may spend excessive effort sending information to different sector managers as its target moves through the environment. Deciding on the correct sector size is analogous to finding the correct membership of a coalition. We 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 show results exhibiting these characteristics, and in section 5 we show how this evidence supports our initial hypothesis.

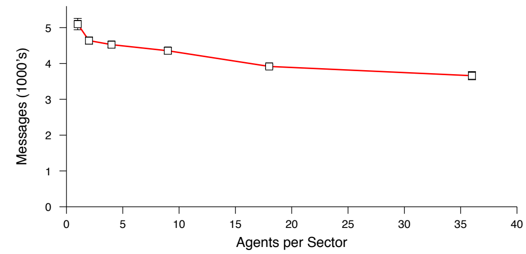

Figure 2. Affect of sector size on messaging.

In these experiments and those that follow in this section, a group of 36 sensors were organized into between 1 and 36 equal-sized sectors with 4 mobile targets. The sensors are arranged in a grid pattern and the targets' location and movement spread evenly through the environment to normalize results and simplify analysis. The results were observed over 10 runs per configuration in a simulation environment which closely models the performance of the physical MTI sensors (the same agent code was used for both the simulation and actual hardware tests). Note that these experiments were performed under idealized conditions. Reducing the rate at which simulation time passes prevents bottleneck and overloading conditions from occurring, so behavioral and organizational phenomena associated with the partition changes can be better isolated. Under bounded conditions, excessive message load could cause performance degradation in many areas of the system. It is our intent to deduce what the issues behind these tradeoffs are, so an informed decision can be made when these limiting effects are present. Conversely, with the same information one could predict the improvements that might be achieved if those limits were relaxed (for instance, if a higher capacity communication mechanism were used).

Figure 2 shows that as the number of agents per sector increases, and there are correspondingly fewer sectors overall, the amount of communication traffic decreases. Because each

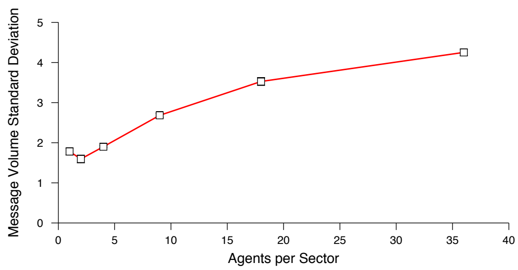

Figure 3. Messaging disparity vs. sector size.

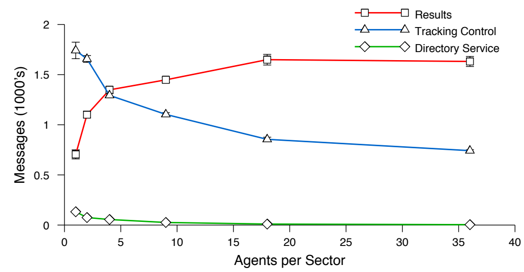

Figure 4. Message types vs. sector size.

sector requires a certain amount of control messages, the total number of messages is reduced as the number of sectors decreases. A more detailed view of the effects this change has on messaging will be shown later in Figure 4.

Partitioning can also reduce reactivity, because an extra step may be required to fetch information. A track manager must perform queries to obtain sector information as its target moves to new sectors, so smaller, more numerous sectors will result in delays caused by the additional queries. This delay will be revisited in Section 6.

3.2. Functional Differentiation

The varied assignment of roles forms a different, functional organization [1] 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 [6]. Concentrating the track management functionality into individual agents serves a similar role, by limiting the number of interactions necessary to resolve conflicts 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 exhibited by individual agents. This metric captures how much agents in the population differ in their communication habits. If all agents are roughly the same they will have a low deviation, while a population that has a handful of outlier agents with significantly higher message traffic will have a high deviation. As the number of agents in each sector increases, this graph shows an increase in disparity, because a few agents are communicating much more than their peers. As the sector size grows, specialized agents become "hotspots" of activity. In a bounded environment this could lead to data loss as the communication channel becomes overloaded. In conjunction with Figure 2 which shows the average *total* communication, we see a tension between sector sizes: smaller sectors lead to increased message traffic, and larger sectors imbalance load in the population. Since the environment and target spacing are uniform, the differences can be attributed to the roles those agents take on. The slight rise in deviation when there is a single agent per sector represents the coexistence of the sector and track manager roles at a single sensor.

4. Maintaining Organizations

Although our organization helps localize interactions, we must also consider the additional communication overhead associated with its creation and maintenance, as it has the potential to minimize these benefits. The most frequently updated organization in the environment is the hierarchy formed between track and sensor managers, because the sensors needed by the track manager change as the target moves. This results in a class of control messages dependent on sector size. For example, as the target moves into part of the environment the track manager is not familiar with, the manager must query the sector manager of that area to discover local sensors. Once those sensors are known, additional messages are needed to establish data collection commitments. Finally, as the target is tracked, the relevant, nearby sector managers must be notified of the target's estimated position.

Figure 4 provides a quantitative view of this overhead. As sector size increases, fewer directory and tracking control messages are necessary, because there are a fewer sectors to interact with as the target moves. In addition, the number of measurements increases as the sector size increases, which produces a lower root-mean-squared (RMS) error between the measured and actual track, as seen in Figure 5. This is also due to communication control overhead; the reduced time spent by the manager interacting with the additional sector managers allows more time to be spent collecting data. This is caused primarily by front-loading sensor discovery, and the reduced probability of track manager migration.

The technique of migrating the tracking responsibility through the agent population as the target moves is another

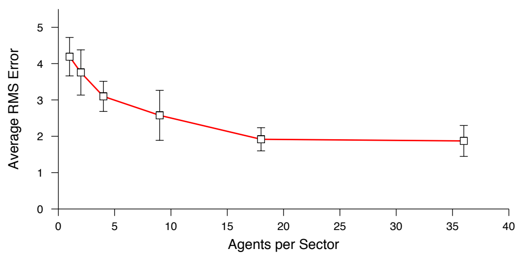

Figure 5. Effect of sector size on RMS error.

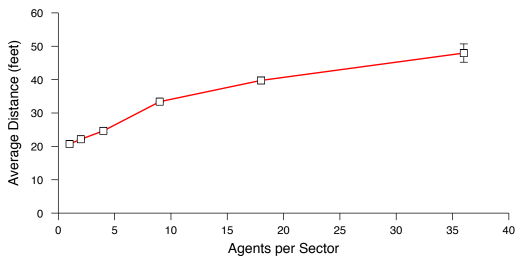

Figure 6. Average communication distance.

aspect of local information exploitation. Signal latency and attenuation conspire to make communication slower and less reliable as distance increases. Lacking the capacity for movement, the initial manager selected to track a target will therefore become less effective as the target moves away from it. By migrating this task to follow the target, the organization is able to retain locality despite the fact that the sensors themselves are immobile. This results in a reduction in the average distance that messages must travel.

Figure 6 shows the effects track manager migration has on the average distance of communication. Because migration is triggered by sector boundaries, the tracking task will migrate less frequently when sectors are large simply because they cover more area. Thus, a lower average communication distance is observed when sectors are smaller.

5. Scalability Results

To explore the generality of these conclusions, we performed experiments with varied target numbers and different sensor population sizes. The first set of tests kept the sensor population static, but contained between 1 and 24 targets. The scenario was otherwise identical to those in Section 3. Figure 7 shows that our original communication disparity profile is maintained if the target density is varied, although the level of disparity is reduced as the number of targets increases. Intuitively, this is because more agents are doing more work, and thus the effect of distinguished overworked agents is minimized. As in Figure 3, when there is only a single sensor per sector, overlapping roles exacerbates the problem. Similarly consistent results are shown in Figure 8. As one would expect, the baseline RMS error increases with the number of targets, since the bounded sensing capabilities result in fewer average measurements per target.

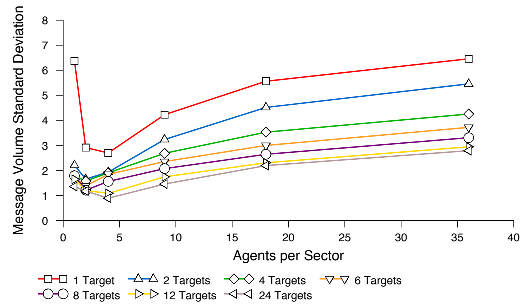

Figure 7. Communication disparity with varied sector sizes and target densities.

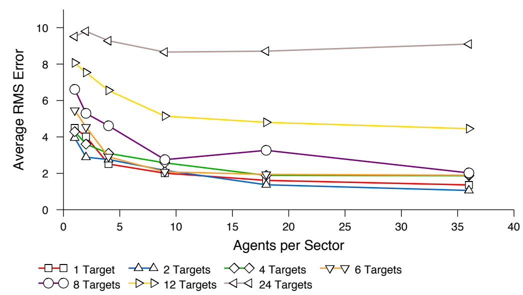

Figure 8. RMS error differences with varied sector sizes and target densities.

In another set of tests, we varied the size of the sensor population from 9 to 81, while maintaining a similar sensor-to-target ratio. The communication deviations shown in Figure 9 are consistent with the earlier findings. The flattening and lowering of the profile as the number of sensors increase is expected, because the larger number of targets spreads work out among more agents. This decrease is somewhat misleading, however, as it hides the true burden that is imposed under these more extreme circumstances. The deviation, although slight, can represent a significant load when the population is large. Figure 10 shows the actual communication burden incurred by different roles for a single-sector environment (i.e. all sensors are in the same partition). The sector manager's burden increases at an undesirable rate, while the track manager and median (non-manager) roles remain relatively constant. Similarly consistent trends were observed in the RMS error, message totals, and average communication distance for other target and sensor population sizes.

These experiments suggest a tradeoff exists between the overall volume of message traffic and its distribution over the agent population. Message volume decreases when there are

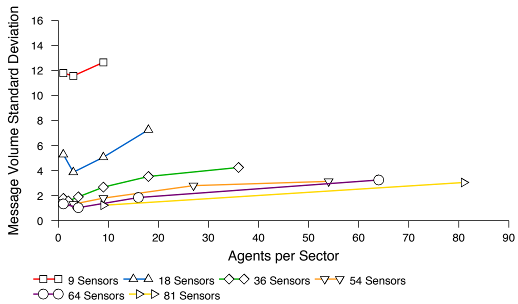

Figure 9. Communication disparity with varied sector and sensor population sizes.

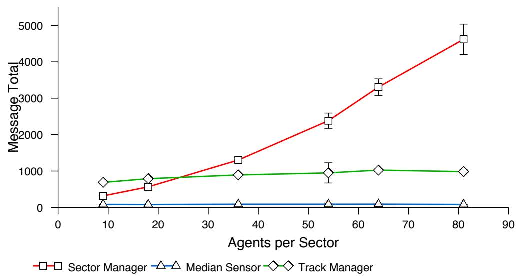

Figure 10. Average communication totals by role for a single-sector environment.

more agents per sector because fewer interactions are needed to obtain information, as shown in Figure 2. However, this shift can cause individual agents to incur a disproportionate communication burden, as shown in Figures 3 and 9. Figure 10 in particular shows the large-sector solution does not scale well. Figures 4, 5, and 6 show that organizational maintenance causes a similar tradeoff - larger sectors have lower overhead and better RMS error, while more track migration in smaller sectors increases communication reliability.

6. Analysis

Our objective is to use these results to make architectural decisions. Normalizing and overlapping the earlier trends produces the graph in Figure 11. By searching for a common inflection point in this diagram, we can conclude that a sector size between 4 and 9 strikes an acceptable balance. This supports our hypothesis that a sector size between 6 and 10 was an appropriate choice. However, the notion of "appropriate" is problem-specific, depending on the characteristics of the agents, the resources they use, and the environment. For instance, if more robust managers were available to handle the increased load, this graph also shows that better RMS performance can be obtained by using larger sector sizes. In general, the requirements imposed by goals and the capabilities

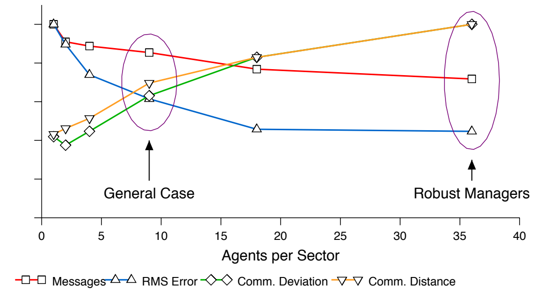

Figure 11. Finding the appropriate configuration from normalized results.

of the system and environment guide an appropriate selection.

Although these profiles give empirical evidence of the system's performance, it is usually preferable to work with a more formal model at design time. Instead deducing metrics from a graph as above, one can create a function which takes requirements and characteristics as inputs, and produces a rating as output. We therefore wish to capture the system's behaviors in an abstract, quantitative model that provides a good approximation of the real system. We will concentrate our analysis on the communication load by role over the lifetime t of the role. As before, we assume that the sensors and targets are uniformly distributed in the environment, and targets move with constant velocity. One could relax these assumptions by estimating interaction probabilities; although the calculations would be more complex, the spirit of the analysis would remain the same. Similarly, one could determine worst-case peak performance by assuming worst-case densities. The formulas presented below do not represent actual message totals, but are meant to reflect relative growth rates. As we will show in Figure 12, quantitative results can be obtained through the addition of appropriate constants.

Consider the sensor manager. Measurements are taken in response to track manager requests, which are in turn prompted by targets in range of the sensor. This role's measurement load ($\mathcal{M}$) is therefore dependent on the likelihood that a target is within its range r. Assume T targets in an environment of area A, each with m measurements per time unit.

$$\mathcal{M} = \sum_t \min(\frac{\pi r^2}{A} Tm, m) \quad (1)$$

So, as the number of targets increase, or the environments area decreases, the number of measurements will approach tm. This model is an upper bound, however, as it does not take into account the track managers' specific behaviors. To better understand this, we will look at how many measurements are generated for an entire track.

Ignoring the effects of uncertain measurements or faulty data fusion, the RMS error of the tracking process is dependent on the number of measurements produced for the track over its lifetime ($\mathcal{R}$). In the absence of hindering factors, the

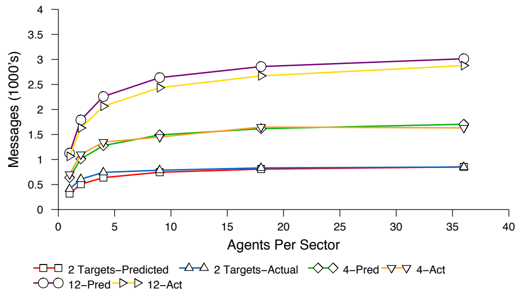

Figure 12. Comparison of predicted and actual results of $\mathcal{R}$ for 2,4 and 12 targets.

track will ideally receive measurements at a uniform rate m from each of c sensors used (we assume c is sufficient for triangulation purposes). The actual rate of measurement is affected by the number of sensors that are used and any delays incurred by overhead tasks. In particular, the collection of sector directory information, and task migration when the target has grown too distant can reduce the total number of measurements are obtained. Competition for sensors by other targets can also reduce the measurement rate.

$$\hat{c} = \min\left(c, \frac{\pi(b+r)^2}{A}N\right) \quad (2)$$

$$l = \min(0, v(\frac{N}{\hat{c}T} - c)) \quad (3)$$

$$\mathcal{R} = \sum_t \hat{c}m\left(\min\left(1, \frac{N}{\hat{c}T}\right)\lambda^l\left(1 - \frac{v}{\sqrt{S}}(d - \frac{g}{2}\right)\right) \quad (4)$$

Equation 2 defines $\hat{c}$, the number of sensors that will actually be used to track the target. It is bounded above by the desired quantity c, and below by the expected proportion of the total number of sensors N that are in range of the target with radius b. The first term of Equation 4 models the proportion of a potentially contended sensor's time usable by the target. If we assume the sensor is shared equally among targets, then the measurement rate obtained by an individual target will be inversely proportional to target density. However, as sensors come under contention, an allocation strategy must be employed to resolve the conflict [4]. An additional reducing factor models this optimization process; l estimates the amount of conflict, while λ controls how much the conflict degrades performance. When the target moves into a new area, there will be a delay d before the appropriate information is received. An additional delay g is incurred during track migrations when the target has moved two sectors away from that of the track manager. The net effect of these delays and the corresponding increase in measurements when sector sizes grow is supported by Figures 4 and 5. Figure 12 shows a comparison of the predicted $\mathcal{R}$ obtained from Equation 4 and the observed load, which was produced by finding appropriate constants for our system.

Returning to the estimated sensor manager load, we can see that $\mathcal{M}$ is more accurately represented by $\frac{\mathcal{R}T}{N}$, as $\mathcal{R}$ models the managing behaviors absent in Equation 1.

Although the detailed results are not presented here, similar analytic models were also created for estimating the load placed on sector and track managers.

7. Conclusions

The quantitative results we have presented 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. 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. For instance, our sector size results can be directly related to the estimated load incurred by a distributed collection of middle agents [6].

More generally, we feel that multi-agent organizations can have significant positive effects on performance. By specifying roles, authority relationships and working groups, the system can both reduce runtime combinatorics by restricting search as well as improve global coherence without requiring a global view. However, we have seen that these benefits come with costs and side effects, which must be well understood for the organization to be used successfully. In this paper, we varied just one organizational parameter, and observed the ramifications of this change across several distinct dimensions. With continued research in this area, the complete space of organizational types and their corresponding characteristics can be more fully understood and exploited.

References

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

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

[3] V. Lesser, C. Ortiz, and M. Tambe, editors. *Distributed Sensor Networks: A Multiagent Perspective (Edited book)*, volume 9. Kluwer Academic Publishers, May 2003.

[4] R. Mailler, V. Lesser, and B. Horling. Cooperative Negotiation for Soft Real-Time Distributed Resource Allocation. In *Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, pages 576–583, Melbourne, July 2003. ACM Press.

[5] 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.

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