Analyzing the Performance of Randomized Information Sharing

Prasanna Velagapudi
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
pkv@cs.cmu.edu

Oleg Prokopyev
Dept. of Industrial Engineering
University of Pittsburgh
Pittsburgh, PA 15261
prokopyev@engr.pitt.edu

Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
katia@cs.cmu.edu

Paul Scerri
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
pscerri@cs.cmu.edu

ABSTRACT

In large, collaborative, heterogeneous teams, team members often collect information that is useful to other members of the team. Recognizing the utility of such information and delivering it efficiently across a team has been the focus of much research, with proposed approaches ranging from flooding to complex filters and matchmakers. Interestingly, random forwarding of information has been found to be a surprisingly effective information sharing approach in some domains. In this paper, we investigate this phenomenon in detail and show that in certain systems, random forwarding of information performs almost half as well as a globally optimal approach. We present analytic and empirical results comparing random methods with theoretically optimal sharing in small-worlds, scale-free, and random networks. In addition, we demonstrate a method for modeling real domains that allows our results to be applied toward estimating information sharing performance.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Coherence and Coordination, Multiagent Systems*; G.3 [**Probability and Statistics**]: Probabilistic Algorithms; H.1.1 [**Models and Principles**]: Systems and Information Theory—*Value of Information*

General Terms

Algorithms, Performance

Keywords

information dissemination, random walk, tokens, utility

Cite as: Analyzing the Performance of Randomized Information Sharing, Prasanna Velagapudi, Oleg Prokopyev, Katia Sycara, Paul Scerri, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 821–828

1. INTRODUCTION

Exciting applications are emerging that involve large, heterogeneous teams acting in complex environments. Examples include search and rescue [5], disaster response [15], and military applications [4]. In such domains, team members will often collect local information that is necessary or useful to other members of the team. For example, in urban search and rescue operations, an aerial robot might be able to locate a victim, but be unable to assess their condition or determine a route to them. A ground robot on the team could perform these tasks, but might be unable to locate the victim by itself. Efficiently getting such information from those collecting it to those requiring it is one of the keys to effective team performance. However, teammates often have limited information about which, if any, team members require particular pieces of information. Thus, the team member collecting some piece of information needs to determine whether and where to send collected information with limited knowledge of who might need it and how important it is to them. At the same time, team members must also be careful about what they communicate as the volume of incoming information is typically dramatically higher than available communication bandwidth.

In small teams or static environments, a variety of approaches have been applied to the information sharing problem. One example, STEAM [16], requires team members to keep others informed of their current state, allowing members to intelligently reason about which teammates need which information. Approaches using matchmakers [9] allow team members to keep some central point informed of their state, while the control point is responsible for directing information as required. More recently, for applications such as sensor networks, algorithms drawing on intuitions of how human gossip [3] works have been shown to be effective, but often wasteful with bandwidth. Token-based algorithms have also been shown to be effective for large-scale team coordination [18] and belief sharing [17] in some domains.

An interesting feature of both gossip and token algorithms is that little knowledge of the team is known or assumed. Nearly random communication coupled with local reasoning is sufficient to produce surprisingly competitive results. It is this surprising effectiveness of lightweight, decentralized, and largely random algorithms that is the focus of this paper. The intention in this work is to understand and quantify when and how these simple strategies will be effective.

Our first contribution is an upper bound on the average performance of information sharing in a network. By conditioning the bound by the number of communications used, it can be generalized to any information sharing approach on a peer-to-peer network of a given structure and distribution of utility. Comparing this bound to empirical results, we find that under certain conditions token-based information sharing methods can achieve near-optimal performance. We present a similar derivation for computing expected performance of a random information sharing policy.

The second contribution is an empirical comparison of simple, random token policies against an omniscient token policy on a highly abstracted information sharing problem. For many distributions of utility and network types, it is found that random policies attain a significant portion of the utility of a policy planned with perfect knowledge. Moreover, random policies incorporating simple heuristics to locally avoid previously traversed links or nodes are observed to do substantially better than purely random policies. Adding noise to the information used by the omniscient policy further reduces the gap between the omniscient and random policies. This suggests that if the cost of maintaining the knowledge to do intelligent information sharing is high, simply using a random policy may be a better approach.

The final contribution of this work is a real-world example of robots sharing a map modeled within a utility distribution framework. The resulting distribution is applied in simulation, and results are compared with those of the canonical distributions studied in the previous section. The resulting analysis predicts that certain network types will yield higher performance for token-based information sharing methods.

2. INFORMATION SHARING

In this section, we formally describe the information sharing problem. Consider a team of agents, A, working to achieve some goal. Suppose there is a piece of information η obtained by team member $a \in A$. If any other member $b \in A$ were to obtain that information, it would impact their ability to perform the team goal. Define the utility $\xi_a(\eta)$ as the quantification of this change in performance. In order for b to get the information, it must be communicated across the network, N. This requires some members of the team to spend resources such as time and power on communicating the information. Thus, there is some communications cost $\kappa(a, b, \eta)$ associated with transmitting η from a to b.

The best team performance is achieved when information is shared with the set of team members that have a higher utility for the information than the cost of communicating it to them. If the communication cost is expressed in the same units as the utility, this is represented by the maximization:

$$A^* = \arg\max_{A \subseteq T} \sum_{a \in A} \xi_a(\eta) - \kappa(\cdot, a, \eta) \qquad (1)$$

In order to address the specific problem of information sharing in a large, dynamic team, we make several key assumptions. First, communications are assumed to be peer-to-peer and of a constant cost per transmission. Rather than representing the communications costs of every pair of agents, $\kappa(a, b, \eta)$ can be compactly represented as some fixed cost κ when agents are neighbors in the network, and infinite when they are not. This is reasonable for domains with peer-to-peer communications, as transmission between distant teammates in a network can be decomposed into a sequence of transmissions to their intermediate neighbors. The only solutions that are lost in this decomposition are solutions where teammates forward information along but do not make use of it, an illogical case for a team.

Given this assumption, it is possible to condition the performance of an information sharing algorithm by the number of communications it has used. In many domains, the tradeoff between communication cost and utility is not well characterized or fixed. Avoiding it allows results to be generalized by removing the second term from Equation 1 and allowing us to compare performance across algorithms which make multiple communications per time step.

Second, while the utility of some information will change over time, we assume that communication is sufficiently fast that utility is constant while a single piece of information is being shared across the network. For example, in a search and rescue domain, the utility of knowing where a fire is will change if the fire spreads or team members move relative to it. However, the speed of these changes is orders of magnitude slower than the millisecond scale transmission speeds of a modern wireless network connecting the team members.

Finally, in a large team, rather than modeling the utility explicitly for each member, we assume that it can be summarized in a *utility distribution* over all agents. This distribution represents the probability that a team member a has some utility ξ for a piece of information η. For a given domain, the utility distribution can be computed empirically by conditioning on relevant variables and sampling utility as information is shared in the team. For analytic purposes, we can approximate this distribution by a number of canonical probability distributions such as normal, uniform, and exponential distributions.

2.1 Token Algorithms

Given these assumptions, we consider two extremes of token-based algorithm design in addressing this problem. In these algorithms, a token is created that contains some information η. This token is atomically passed from teammate to teammate. When a team member receives the token, it can make use of the information inside, then decide to either forward the token to a neighbor or delete it. Since tokens use exactly one communication per time step, token algorithms can control the number of communications by using tokens that are deleted after a fixed number of steps.

If we take advantage of all possible knowledge of agent utility and network properties, the optimal approach is to directly solve the maximization in Equation 1. This is done using an exhaustive search of all possible network paths of length t. We call this a t-step *lookahead* approach.

On the other hand, if we ignore all available knowledge of agent utility, we can propose a simple algorithm of randomly passing information from neighbor to neighbor. This equates to simply performing a random walk across the network. We therefore call this the *random walk* approach.

Given that no knowledge of utility is used in routing a random walk, its efficiency is primarily determined by its coverage of the network. We therefore introduce two intermediate algorithms that are equally naïve with regards to utility, but significantly more intelligent about coverage. A token will maximize coverage if it never revisits the same agent in a network. Thus, a straightforward improvement to the random walk approach is the addition of a history of nodes carried within the token. As the token moves around the network, visited agents are marked in this history, and when the token is being routed, this history is used to exclude visited agents from selection. If all neighbor nodes are visited, the algorithm selects a link at random. This approximates a self-avoiding walk over the network, so we term this the *random self-avoiding walk* approach.

This approach is reasonable when the size of the history is expected to be bounded to a reasonable size. However, in very large teams or systems where tokens visit many agents of the team, this is not a practical solution. In these cases, if there are a bounded number of tokens in existence at any given time time, an alternative solution is to maintain a local history at each agent for each active token, consisting of its previously used incoming and outgoing network connections. Similarly to the random self-avoiding approach, agents that receive a token multiple times will attempt to send it to different neighbors each time, selecting randomly from the outgoing edges if all of them have been previously used. We designate this the *random trail* approach.

3. ANALYTIC BOUNDS

Given a utility distribution for a piece of information and the size of the network, it is possible to compute an upper bound on the expected value of information sharing for a fixed number of communications. Consider sharing a piece of information η. Under the peer-to-peer communications assumption, each transmission of η, regardless of originator, will send η to exactly one team member, a. If we ignore connectivity constraints, then in the optimal case a has never seen the information before and has the highest utility ($a = \arg\max_{\alpha \in A} \xi_\alpha(\eta)$) of any member of the team. Once this member receives the information, however, $\xi_a(\eta) \leftarrow 0$. Thus, the next communication should ideally pass information to a member b that has also never seen the information before and has the highest remaining utility ($b = \arg\max_{\alpha \in A \setminus a} \xi_\alpha(\eta)$) of any team member for the piece of information. The optimal sequence of transmission over the members of the team is to simply go in order by descending utility. For any specific number t of transmissions, the optimal path will be the first t members in this sequence.

For the case of simultaneous communications this optimum still holds, as each still incurs a communication cost. Simultaneous transmissions can be considered to be ordered arbitrarily, but it is clear that the above sequence will be an upper bound for any such sequence of transmissions.

Using the utility distribution, this optimal sequence can be probabilistically modeled as a descent through the *order statistics* of the team members. The value of the kth smallest of n samples from a distribution is defined to be the kth order statistic of the distribution. If the team size is n, we assume that the team members constitute n i.i.d. samples from the utility distribution. Thus, an ideal communication sequence under full connectivity will, on average, first visit an agent with a utility corresponding to the expected value of the nth order statistic of the utility distribution for a sample size of n, then the $(n-1)$th, and so forth, until all team members have been visited. This sequence, by definition, will yield the highest possible average utility for any fixed number of communications.

The expected order statistics for many canonical distributions of interest are well studied. We review the results for three common distributions: uniform, normal, and exponential. For the uniform distribution, the expectation of the kth order statistic over n samples, $U_{k:n}$, follows a Beta distribution of the form $\beta(k, n+1-k)$, and so it can be directly solved as the mean of the distribution [1].

$$E[U_{k:n}] = \frac{k}{n+1} \tag{2}$$

While slightly more complex in the general case, the expectation of the kth order statistic for the standard exponential distribution over n samples, $X_{k:n}$, follows a simple recurrence relation [1].

$$E[X_{k:n}] = \frac{k}{n} \tag{3}$$

$$E[X_{k:n}] = E[X_{k-1,n-1}] + \frac{k}{n} \tag{4}$$

The normal distribution does not have a neat closed form for its expected order statistics. However, a number of high-precision approximations exist. For this paper, algorithm AS177 [14] is used to approximate expected order statistics for the reasonably large ($n = 1000$) sample sets.

With the expectations of the utility distribution's order statistic $E[O_{k:n}]$ computed, the upper bound B can be generated using the following sum:

$$B = E\left[\sum_{n-k}^{n} O_{k:n}\right] = \sum_{n-k}^{n} E\left[O_{k:n}\right] \tag{5}$$

where O_k is the kth order statistic of the utility distribution.

This bound is tight when the network is fully connected. Under this condition, it is also possible to model the expected performance of information sharing via a completely random walk.

Given a team of size n where we have a set A_u of team members that have already received the token (including the member that currently has the token). Let $|A_u| = u$. We can compute the probability that the token will be sent to a team member a that has already received the token:

$$Pr(a \in A_u | n, u) = \frac{u-1}{n-1} \tag{6}$$

We can derive a recurrence that computes the probability that during the kth step of a random walk, the size of the visited set will be x:

$$\begin{aligned} Pr(u = x|n, k) &= Pr(a \in A_u|n, x)Pr(u = x|n, k-1) \\ &+ Pr(a \notin A_u|n, x-1)Pr(u = x-1|n, k-1) \end{aligned} \tag{7}$$

When the information is collected locally, we know that already, one team member must know of it. Thus, we have

$$Pr(u = 1|\forall n, k = 0) = 1 \tag{8}$$

Similarly, at any later point, at least one agent must know of the information.

$$Pr(u < 1|\forall n, k \geq 0) = 0 \tag{9}$$

Using this recurrence relation, we can compute the expected number of distinct team members that will be visited after k communications of a token as

$$E[u|k] = n\left(1 - \left(1 - \frac{1}{n}\right)^{k+1}\right) \tag{10}$$

Since each distinct team member is independently sampled from the utility distribution, this leads to the expected performance of a random walk on a complete network.

$$E[U|k,n] = E[\text{utility for single agent}] \cdot E[u|k,n]$$
$$= E[\text{utility for single agent}]n\left(1-\left(1-\frac{1}{n}\right)^{k+1}\right)$$

Expected performance can similarly be computed for other networks and utility distributions, provided that agents can be considered to be sampled independently and the expectation $E[u|k,n]$ can be computed.

4. EXPERIMENTAL RESULTS

A highly abstracted information sharing token simulator was created to empirically test the performance of token-based information sharing methods. The simulation consists of a network of agents that are assigned utilities for a given piece of information from a specified distribution. A token representing that information is initialized at a randomly chosen agent within the network. The agents propagate the token around the network according to some routing policy, and the accumulated utility is recorded at each step until the simulation executes some fixed number of steps. Results are averaged over 5 runs.

Five canonical network types were examined: small-worlds, scale-free, hierarchical, lattice, and random. Each was generated to contain 1000 nodes with an average degree of 4. The small-worlds network was generated by adding random links to a doubly-connected ring. The scale-free network was generated using a tunable variant of the Barabási-Albert algorithm [10]. The hierarchical network was formed by adding nodes evenly to a balanced tree. The lattice was a four-connected 2D grid with wraparound, and the random network was created by adding random links until the average degree was reached. In most cases, results for random networks were analogous to those of scale-free networks, thus most results for random networks were omitted for brevity.

Three canonical distributions were examined: uniform, normal, and exponential. The uniform distribution was over the interval $[0, 1]$. The exponential distribution had a rate parameter of $\lambda = 1.0$, but was scaled by a factor of 0.2. In the case of the normal distribution, the mean and variance of the distribution were sometimes altered for various trials, but the nominal parameters were $\mu = 0.5$, $\sigma = 0.2$.

Four information sharing methods were considered: optimal, random walk, random trails, and self-avoiding walks. The optimal policy was approximated using a finite lookahead policy with global knowledge. Every m-steps, an exhaustive search of paths of length m was executed to determine an optimal path. This path was executed fully, followed by another m-step planning phase. Ideally, this stage would consist of a single path search of the final path length, but computing this path is extremely expensive due to the non-Markovian nature of the utility function (as agents are visited, their utility drops to zero, so the joint distribution of utility is always dependent on complete network state). Instead, smaller values of m were chosen empirically from the results of early experiments.

4.1 Optimality of the Lookahead Policy

In order to determine a sufficient approximation of optimality, an experiment was conducted in which the depth of the lookahead policy was varied over the four network types with a normal utility distribution ($\mu = 0.5, \sigma = 0.2$), and the utilities of the resulting communication paths computed. The results of these tests can be seen in Figure 1, where the utility obtained by each token is plotted against the number of communications the token was allowed. As lookahead depth increases, the obtained utility converges asymptotically to the optimal. Interestingly, while lookahead depths of 1, 4 and 8 converge toward an asymptote, 2-step lookaheads, denoted by the circle symbols, appear to perform pathologically poorly. It is possible that this is due to a negative interaction between the width of the networks and the depth of the search pattern, where the lookahead policy may consistently make myopic routing decisions. From these results, a lookahead policy with a depth of 4 was selected as a baseline for future experiments, as a compromise between computational complexity and optimality of performance.

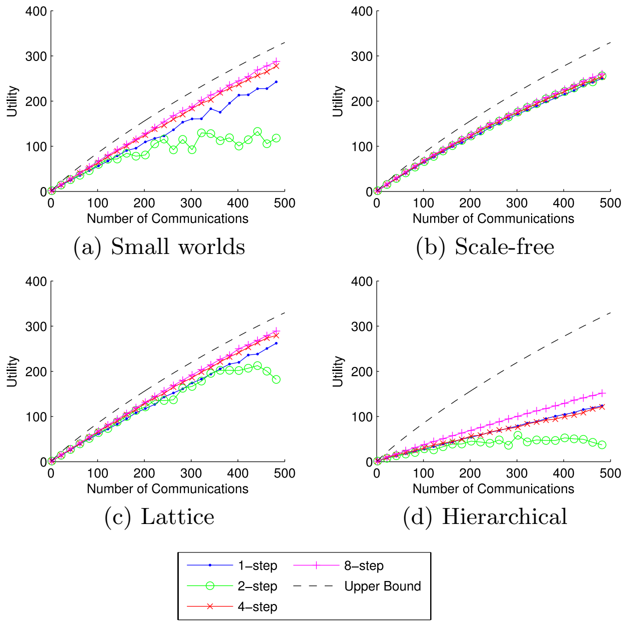


Figure 1: Optimality of n-step lookahead over four network types with a normal utility distribution ($\mu = 0.5, \sigma = 0.2$). The utility obtained by each token is plotted against the number of communications used.

It is also possible to evaluate the optimality of token-based lookahead policies against the upper bound established in the previous section. The dashed lines in in Figure 1 correspond to these bounds. Two key characteristics are immediately evident. First, the lookahead policies often converge very closely to the upper bound on performance, suggesting that in the ideal case, token routing methods can perform very close to optimal. Second, while the bound is independent of network structure, clear differences are visible in the optimality of the lookahead policy over different network types. Most notably, in Figure 1d, the hierarchical network performs much worse than the upper bound, suggesting that information propagation via tokens in this type of structure is either highly inefficient or requires an extremely deep lookahead depth.

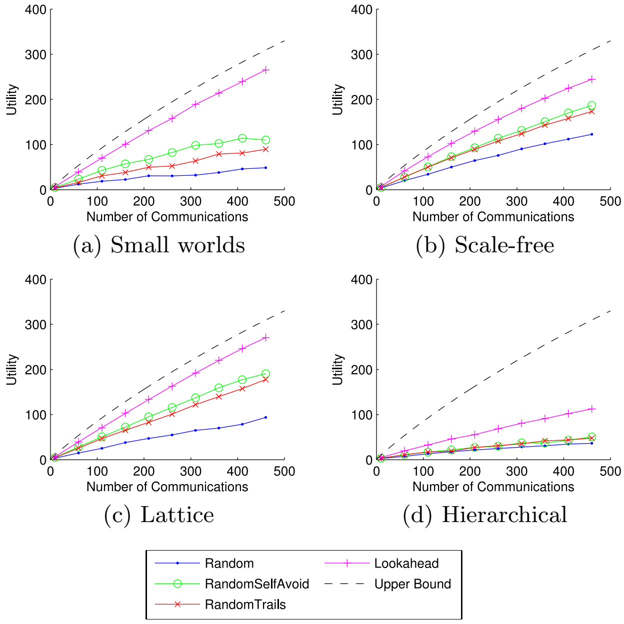


Figure 2: Performance of random and lookahead policies over four network types with a normal utility distribution ($\mu = 0.5, \sigma = 0.2$). The utility obtained by each token is plotted against the number of communications used.

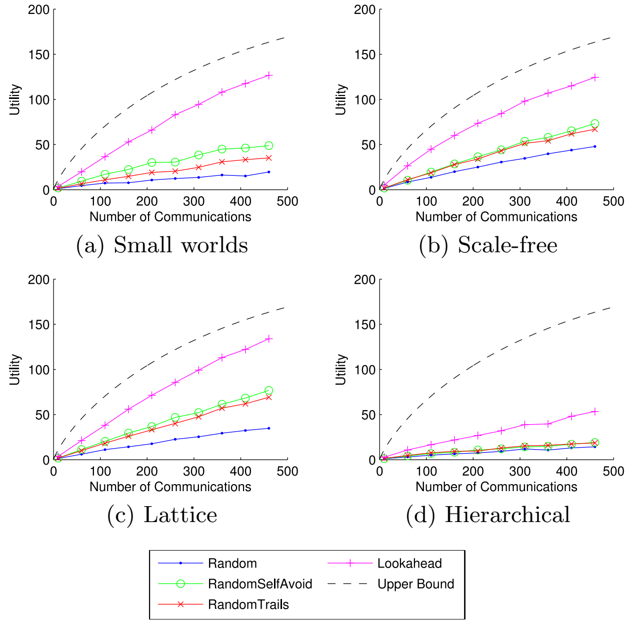


Figure 3: Performance of random and lookahead policies over four network types with an exponential utility distribution ($\lambda = 1.0$, scale factor of 0.2). The utility obtained by each token is plotted against the number of communications used.

4.2 Optimality of the Random Policies

The four information sharing methods were tested in normal and exponential distributions. Figures 2 and 3 show the results of these experiments. It is evident that performance is heavily influenced by network type, with all policies performing significantly worse in hierarchical networks with both distributions and random policies performing proportionally much worse in the small worlds and hierarchical networks (Figures 2d and 3d). Random policies perform best in the scale-free and lattice networks, with purely random walks attaining almost half the utility of the lookahead policy in the scale-free network with a normal utility distribution. The addition of self-avoiding heuristics appears to improve random policy performance significantly, primarily in the lattice and scale-free networks. The similar performance of the random trail and self-avoiding walk policies suggests that they are comparably effective at avoiding previously visited agents when covering the network. Further experiments focused on the random trail policy, as it typified the performance of the heuristic random policies.

As an example of the surprisingly efficient performance of random policies, consider Figure 2b. In it, we find that a utility of 175 is attained by a lookahead policy using an average of 300 communications. The same utility is obtained by a random self-avoiding policy in 425 communications. However, the random self-avoiding policy has no computational or structural overhead as it is *completely unaware of utility*. This suggests that under these conditions, if the cost of maintaining the necessary knowledge to perform an optimal strategy is on par with the cost of the extra 125 communications, a randomized policy is in fact a competitive strategy for information sharing.

4.3 Effects of Noisy Estimation

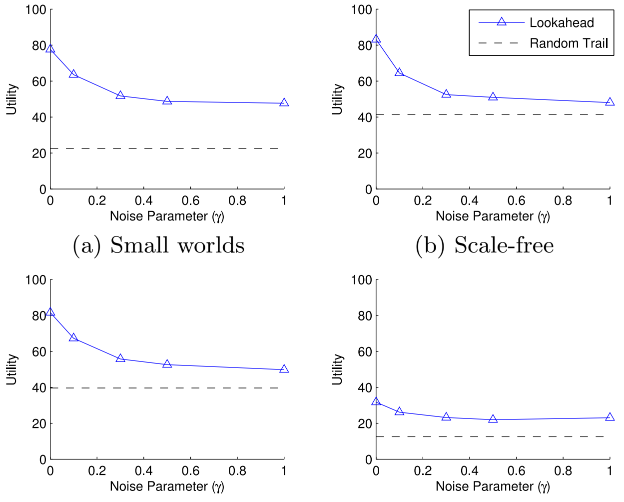
Exploring this tradeoff further, we examine the effects of noisy estimates of utility on performance of the lookahead policy. Gaussian noise was introduced into the utility estimates of unvisited team members used by the lookahead policy. The utility of visited members was fixed at zero and not affected by this noise. To simulate the compounded inaccuracy of estimating the utility of team members further away in the network, the standard deviation (σ) of the noise was scaled exponentially by the network distance between teammates ($d_{a,b}$) using the following equation.

$$\sigma_{a,b} = (\sigma + 1.0)^{d_{a,b}} - 1.0 \qquad (11)$$

As seen in Figure 4, as the amount of noise was increased, the performance of the lookahead policies degraded. This suggests that even when using an ideal routing policy, incorrect estimates of utility can disrupt intelligent routing policies. However, at $\sigma = 1.0$, the noise was so large that estimates of utility were approximately random. The only usable information available in this condition was that the utility of visited teammates was fixed at zero. Without the ability to discern high- and low-utility team members, the remaining difference in performance between the lookahead and random policy in the high noise condition cannot be attributed to the selection of higher utility paths. However, it may be the result of the lookahead policy's ability to avoid myopic routing decisions that might force future communications to pass through visited teammates.

4.4 Properties Affecting Optimality

Another observation was that the proportional gap between the lookahead policy and the random trail policy var-



(a) Small worlds (b) Scale-free

(c) Lattice (d) Hierarchical

Figure 4: Effects of noise on lookahead and random trail policy over four network types with a normal utility distribution ($\mu = 0.5, \sigma$ varied). The utility obtained by each token is plotted against a noise scaling parameter γ.

ied repeatably over networks and utility distributions across trials, suggesting that a combination of network structure and utility distribution properties affect the efficiency of the random trail policy. To explore this further, a wide array of networks and utility distributions were tested across a constant number of communications of $t = 250$ to study how the optimality of the random trail policy was affected by various properties. A cross section of these results can be found in Figures 5 and 6. It was found that certain characteristics clearly affected optimality, while most had negligible effects.

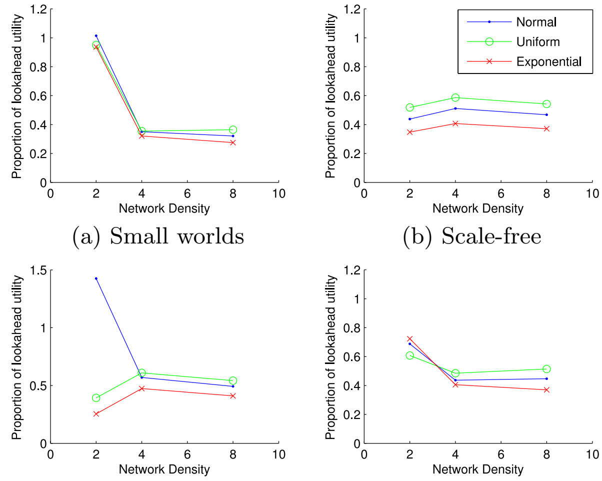
The type of network had a clear impact on optimality. Interestingly, small worlds and hierarchical networks were similar in performance and contrasted with scale-free networks. In this experiment, particularly interesting results were obtained for random networks, so these results are presented in lieu of the grid results.

The *network density* was another property that showed a clear effect on optimality. At low network densities ($\rho = 2$), the average case performance of the random trail algorithm matched or exceeded the optimal policy on the small worlds and random networks. The consistency of this result, and its specificity, suggest that certain combinations of network structure, utility distribution, and network density are pathological for the lookahead policy. At higher densities, the optimality seemed to converge to a constant value dependent on network type.

Aside from inconsistent behavior at low network densities, the variance of the utility distribution also affected optimality, with the random trail policy performing better as variance was decreased. This makes sense, as a perfect self-avoiding policy over a network of members with constant utility (no variance) will always take an optimal path.

4.5 Scaling Properties

The performance of the lookahead and random trail policies was tested under the normal and exponential utility



(a) Small worlds (b) Scale-free

(c) Random (d) Hierarchical

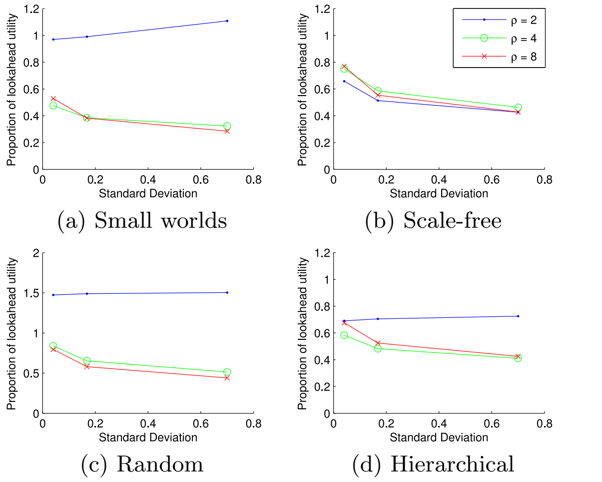
Figure 5: Effects of network density on optimality of random trail policy over four network types with a normal ($\mu = 0.5, \sigma = 0.2$), exponential ($\lambda = 1.0$, scale factor of 0.2), and uniform utility distribution. The proportion of lookahead utility refers to the utility of the random trail policy scaled by that of a 4-step lookahead policy.

distributions as the networks were scaled between 500 and 6500 nodes. As seen Figure 7, performance of both random and lookahead policies was scale-invariant over this range. This suggests that previous results, obtained for 1000 node networks, should scale to much larger team sizes. As the complexity of maintaining accurate knowledge of the team increases with team size, it also suggests that in practice random policies may be more competitive in larger networks.

5. MODELING SPECIFIC DOMAINS

While the previous experimental results are applied to canonical distributions, we can also extend the analysis to more realistic domains. As a practical example, we consider the case of two robots exploring a randomly generated maze-like obstacle field, represented as a 2D cost map. The robots are capable of exchanging their locally-sensed maps. Suppose we have one mobile robot a at some location on the map. It has been randomly assigned a goal somewhere on the map, and its objective is to move to this location via a path of minimum cost by making a series of steps to neighboring map cells. The robot is capable of sensing the cost of map cells within a small Euclidean distance around it, but it has no knowledge of the cost of cells outside of its sensory range. Using the D* lite algorithm [8], the robot can plan a path to the goal and traverse it, recomputing its path as new obstacles are sensed. The path it generates will often be less than optimal, as it will sometimes explore routes that lead to dead ends or circuitous paths.

Now, suppose a second robot, b, is located on the map as well. b is a stationary robot, but has an extended sensing range. It has the option of sending the costs it can locally sense to robot a. In some cases, this may help a avoid exploring an area needlessly. We define the utility of this information to be the change in a's traversal cost with and without



(a) Small worlds (b) Scale-free (c) Random (d) Hierarchical

Figure 6: Effects of variance on optimality of random trail policy over four network types with a normal utility distribution ($\mu = 0.5, \sigma$ varied) and varying network densities. The proportion of lookahead utility refers to the utility of the random trail policy scaled by that of a 4-step lookahead policy.

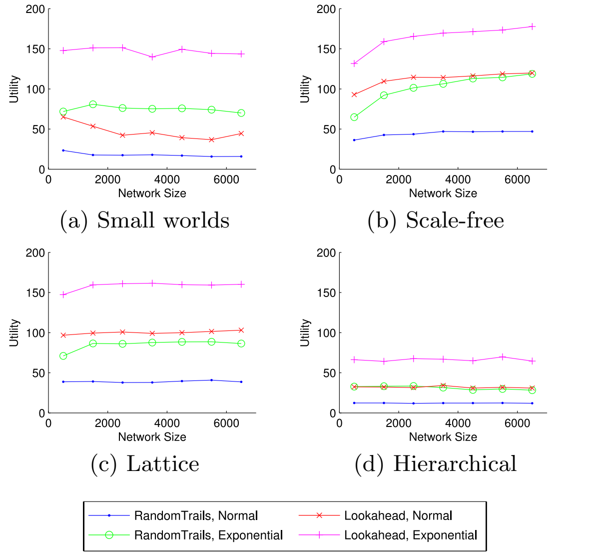
the additional information from b. If we assume this activity is taking place on a very large scale, we can empirically compute the distribution of utility over many random placements of a, a's goal, and b to get a probabilistic estimate of the utility of an information update.

For the case where the grid is 200×200, the sensing radius of a is 5 cells, and the sensing radius of b is 20 cells, the distribution of 40,000 samples can be seen in Figure 8. The distribution is roughly exponential, as most updates cover areas outside of the regions of interest or do not give a much novel information. Some updates do significantly decrease the traversal cost for a, and these are the ones that potentially should be propagated by an intelligent information sharing algorithm. Interestingly, there are also updates with negative utility, which occur when b sends an update that encourages a to explore a dead end that it would not have explored otherwise. If additional information about a were available, a more precise utility distribution could be formed by conditioning on this information.

Once we have the utility distribution computed, we can sample from it in the simulation to estimate the large-scale performance of the various policies. The results in Figure 9 are consistent with the results in the exponential distribution in Figure 3. From these graphs, we can estimate the optimality of the heuristic random policies at various communications limits and over various networks.

6. RELATED WORK

The problem of communication in teams has been well-studied in a variety of fields. Approaches such as STEAM [16] and matchmakers [9] share knowledge about information requirements in order to reason about where to direct information. Gossip algorithms [3] and token passing algorithms [18, 17] use randomized local policies to share information and are thus particularly suited to large scale problems. To address the expense of synchronizing beliefs over teams, several techniques have been developed in conjunction with decentralized Bayesian filtering techniques, including channel managers [2] and query-based particle filters [12].



(a) Small worlds (b) Scale-free (c) Lattice (d) Hierarchical

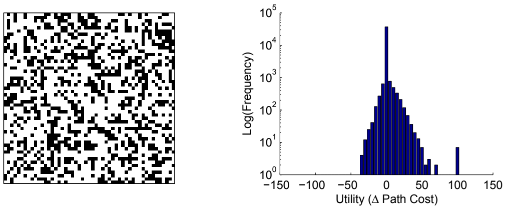
Figure 7: Effects of scale on performance of random trail and lookahead policies over four network types with a normal ($\mu = 0.5, \sigma = 0.2$), and exponential ($\lambda = 1.0$, scale factor of 0.2) utility distribution. Performance of both random and lookahead policies appears scale-invariant over networks ranging from 500 to 6500 nodes.

A number of approaches to communication for multi-agent coordination have evolved around the concept of multi-agent POMDPs. Some approaches augment the decentralized problem with actions for synchronization [11], while others model communications as an explicit choice and seek to maximize the tradeoff between communication cost and reward [19, 6] to achieve goals such as minimizing coordination errors [13].

7. CONCLUSIONS AND FUTURE WORK

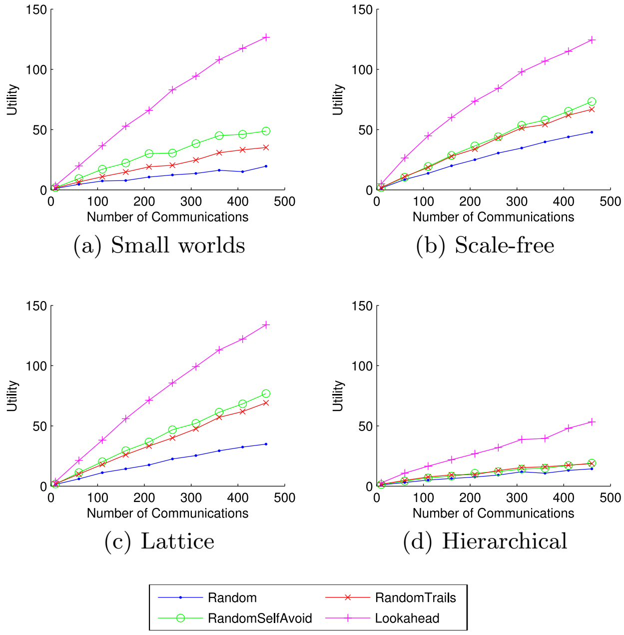
In this paper, we establish an upper bound on average case performance of information sharing in large teams and show that in certain circumstances random policies can achieve a significant portion of that performance. By adding simple heuristics to avoid redundant communications, it is possible to improve the performance of a purely random policy significantly. This means that in domains where network and utility distributions are similar to these cases, random information sharing policies may present an efficient and robust information sharing solution. Furthermore, this performance is scale-invariant, making these policies particularly well suited to large team environments.

Overall random policies were found perform relatively poorly on small-worlds networks, while performing well on scale-free and lattice networks. In addition, hierarchical networks were shown to be ill-suited to even optimal token-based information sharing algorithms. Similar results were obtained for the maze navigation domain, demonstrating that real-world domains can also be modeled through utility distributions, allowing them to be analyzed on large scales.



(a) Patch of map (b) Distribution of utility

Figure 8: Utility in the maze navigation domain. A static and mobile robot capable of sensing local obstacles are placed in a randomly generated map (a). The utility of the stationary robot's information (b) is defined as the difference in the mobile robot's path cost when traveling to a goal while using the stationary robot's information versus traveling without it.



(a) Small worlds (b) Scale-free

(c) Lattice (d) Hierarchical

Figure 9: Performance of random and lookahead policies over four network types with the maze utility distribution. The utility obtained by each token is plotted against the number of communications the token was allowed.

In future work, we will apply these results to a variety of physical domains, including urban search and rescue and mobile mesh networking. Using these analysis methods, it should be possible to determine which information sharing methods are best suited to these domains, including if and when random policies should be used. In addition, by modeling the utility distributions of these domains, it may be possible to gain insight into the fundamental properties of real-world information sharing problems, in turn improving the information sharing algorithms that must address them. Further graph-theoretic and probabilistic analysis should yield tighter bounds on performance, and additional experiments can determine the optimality of other common information sharing algorithms such as classic flooding [7], gossiping [3], and channel filtering [2].

8. ACKNOWLEDGMENTS

This research has been funded in part by the AFOSR grant FA9550-07-1-0039 and the AFOSR MURI grant FA9550-08-1-0356. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

9. REFERENCES

[1] N. Balakrishnan and A. C. Cohen. *Order statistics and inference*. Statistical Modeling and Decision Science, Boston: Academic Press, 1991.
[2] F. Bourgault and H. Durrant-Whyte. Communication in general decentralized filter and the coordinated search strategy. In *Proc. of FUSION'04*, 2004.
[3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. In *Proc. of IEEE/ACM Trans. Netw.*, 2006.
[4] L. Chaimowicz and V. Kumar. Aerial shepherds: Coordination among uavs and swarms of robots. In *Proc. of DARS'04*, 2004.
[5] J. L. Drury, J. Richer, N. Rackliffe, and M. A. Goodrich. Comparing situation awareness for two unmanned aerial vehicle human interface approaches. In *Proc. of IEEE Int. Workshop on Safety, Security and Rescue Robotics*, 2006.
[6] C. V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *Proc. of AAMAS'03*, 2003.
[7] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proc. of MobiCom'99*, 1999.
[8] S. Koenig and M. Likhachev. D* lite. In *Proc. of AAAI'02*, 2002.
[9] D. Kuokka and L. Harada. *Readings in Agents*, chapter Matchmaking for information agents. Morgan Kaufmann, 1997.
[10] C. C. Leary, M. Schwehm, M. Eichner, and H. P. Duerr. Tuning degree distributions of scale-free networks. *Arxiv preprint physics/0602152*, 2006.
[11] R. Nair and M. Tambe. Communication for improving policy computation in distributed pomdps. In *Proc. of AAMAS'04*, 2004.
[12] M. Rosencrantz, G. Gordon, and S. Thrun. Decentralized sensor fusion with distributed particle filters. In *Proc. of UAI'03*, 2003.
[13] M. Roth. *Execution-time Communication Decisions for Coordination of Multi-agent Teams*. PhD thesis, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, 2007.
[14] J. P. Royston. Algorithm as 177: Expected normal order statistics (exact and approximate). *Applied Statistics*, 31(2):161–165, 1982.
[15] N. Schurr, J. Marecki, M. Tambe, P. Scerri, J. Levis, and N. Kasinadhuni. The future of disaster response: Humans working with multiagent teams using DEFACTO. In *AAAI Spring Symposium on Homeland Security*, 2005.
[16] M. Tambe. Agent architectures for flexible, practical teamwork. In *Proc. of AAAI'97*, 1997.
[17] P. Velagapudi, O. Prokopyev, K. Sycara, and P. Scerri. Maintaining shared belief in a large multiagent team. In *Proc. of FUSION'07*, 2007.
[18] Y. Xu, P. Scerri, B. Yu, S. Okamoto, M. Lewis, and K. Sycara. An integrated token-based algorithm for scalable coordination. In *Proc. of AAMAS'05*, 2005.
[19] P. Xuan, V. Lesser, and S. Zilberstein. Communication decisions in multi-agent cooperation: Model and experiments. In *Proc. of AGENTS'01*, 2001.