Communicating Effectively in Resource-Constrained Multi-Agent Systems

Partha S. Dutta
School of ECS
University of Southampton
Southampton SO171BJ, UK.
psd@ecs.soton.ac.uk

Claudia V. Goldman
Samsung Telecom Research Israel
Herzliya, Israel.
c.goldman@samsung.com

Nicholas R. Jennings
School of ECS
University of Southampton
Southampton SO171BJ, UK.
nrj@ecs.soton.ac.uk

Abstract

Agents with partial observability need to share information to achieve decentralised coordination. However, in resource-constrained systems, indiscriminate communication can create performance bottlenecks by consuming valuable bandwidth. Therefore, there is a tradeoff between the utility attained by communication and its cost. Here we address this tradeoff by developing a novel strategy to make communication *selective* based on *information redundancy*; ensuring communication only occurs when necessary, while maintaining acceptable coordination. We apply this strategy to a state-of-the-art communication protocol to evaluate its resource saving benefit in a distributed network routing problem. Furthermore, we design a mechanism to adapt its selectivity level to the prevailing resource constraints to ensure further improvements. Empirical studies show our selective strategy achieves relative savings in bandwidth usage of 50-90%, with only a 5-10% relative reduction in coordination effectiveness and the adaptive strategy further improves relative bandwidth usage by up to 10% and also relative coordination effectiveness by up to 12% over the non-adaptive approach.

1 Introduction

In many distributed systems, such as sensor networks, P2P systems, and grid applications, decision-making is performed by individual agents in a decentralised fashion under incomplete global knowledge. Now, to effectively coordinate their actions under these circumstances, the agents need to share information [Xuan *et al.*, 2001; Goldman and Zilberstein, 2003]. Specifically, the fact that such exchanges can significantly improve distributed task processing has been shown in [Dutta *et al.*, 2005] where a state-of-the-art information-sharing protocol is developed. However, their solution does not consider communication *cost*, which is the extra bandwidth required for communication and is limited in all practical applications. Given this, this paper develops a novel decentralised strategy that agents can employ to decide *when* to exchange information. This strategy is practical for realistic applications and balances the need to communicate (to improve coordination) with the need to reduce overhead (to save valuable bandwidth).

In more detail, the issue of planning when to communicate is a non-trivial problem. In particular, Goldman and Zilberstein [2004] show that solving this problem optimally is intractable. Given this, a number of non-optimal solutions to the problem have been proposed. For example, in [Gmytrasiewicz and Durfee, 2001], agents communicate to improve their own utilities, but this may be at the expense of the overall system performance. Zhang *et al.* [2004] require *all possible* communication acts of agents to compute team utility gained due to communication: an impractical proposition in real applications. The algorithms to solve the DEC-MDP formulation in [Shen *et al.*, 2003] need expensive offline planning and so cannot readily be applied to dynamic environments. Finally, centralised solutions are equally inapplicable [Haas *et al.*, 2000] as they are not scalable and present a single point of failure.

Against this background, this paper presents a compromising but *scalable* approach: agents exchange information *selectively* such that it improves coordination, but without incurring the high costs associated with computing exactly when it is best to communicate. Here, the main challenge is to find the *right* level of selectiveness that will restrict resource usage, thereby allowing its use for realistic task processing, while limiting performance degradation to non-significant levels compared to strategies that permit unconstrained communication.

To design our information-sharing algorithm for efficient task processing and resource usage, we use *information redundancy* as the parameter for agents to decide when to communicate. Specifically, we build upon the information-sharing algorithm of [Dutta *et al.*, 2005] which represents the state-of-the-art in this area (having been shown to outperform Littman's Q-routing, Stone's TPOT-RL, and broadcast protocols). This benchmark algorithm distributes local knowledge among cooperative agents after they jointly complete a task. We call this approach *non-selective information exchange* (NSIE) as information is shared *every* time a task is completed without considering resource constraints (Section 2). Here, we extend this approach by including a level of decision-making among agents *before* distributing local knowledge. Thus, after jointly completing tasks, the agents now choose to communicate or not depending on whether the information transmitted would be useful or redundant to the recipient (Section 3). Such redundancy is captured using a *selectivity threshold*, Δs, which is by how much the current information (the value of an agent's local state) differs from what was previously transmitted. Now, various selectivity values affect a system's performance differently: the higher the threshold, the more restricted the communication, which lessens coordination quality but takes up less resource. To improve resource usage still further, we then develop an algorithm for adaptively setting the selectivity threshold so that its level can be chosen according to the prevailing context. This allows more informed communication decisions, thereby reducing bandwidth usage and also increasing coordination effectiveness compared to using a fixed threshold. We then empirically assess the effectiveness of our *selective information exchange* (SIE) and *adaptive selective information exchange* (ASIE) protocols by comparing their performances against NSIE (Section 4). In particular, for a distributed task processing problem, we measure the amount of degradation (if any) in the quality of solutions and the amount of savings (if any) in resource usage achieved by our strategies compared to NSIE. The evaluations are carried out on an exemplar problem: distributed network routing. This is the same domain used in [Dutta *et al.*, 2005] which allows valid comparisons. That our SIE approach extends the state-of-the-art significantly is confirmed by these empirical studies: SIE achieves a 50-90% saving in bandwidth usage while reducing throughput by only 5-10% compared to NSIE across various network environments. ASIE further improves relative bandwidth usage by about 10% and also the relative throughput by about 12% compared to SIE.

2 Routing in Resource-Constrained Networks

This section outlines the call routing application on which our selective information exchange protocol is evaluated. Note, however, our protocol is generic and can be used to conserve bandwidth in any system with resource constraints and using information-sharing at some level to operate efficiently. Also note, in systems where communication is different from exchanging state values (as happens in our application), we need a suitable way of formalising the redundancy metric.[1] In our application, each node communicates directly with those that are within its communication range. The system's task is to connect a call between the node where the call originates and the one where it is destined. The originating node forwards a call setup request to one of its neighbours. As each agent only has local information, it chooses that neighbour for which it estimates that the call would be placed via the most efficient overall route.

Specifically, each node i maintains a routing table RT_i, where each element $RT_i(d, n)$ (d: the destination node, n: a neighbour of i) represents i's estimate of the best end-to-end bandwidth availability in going from n to d. Node i chooses a neighbour j using a Boltzmann distribution over its RT_i values: $exp(\frac{-RT_i(d,j)}{\tau})/\sum_n exp(\frac{-RT_i(d,n)}{\tau})$, where τ is the importance given to higher RT values. Each node chooses the next hop until the call setup request reaches the destination. After this happens, the destination sends "upstream" along the call route an `ack` message indicating a successful call connection. Each node i also appends its state value to the `ack`. In this application, a node's state is the bandwidth available expressed as a fraction of the maximum availability. Thus, a node k receiving the `ack` from the immediate "downstream" node l gets the set of state values that were appended by all "downstream" nodes on this call route. Node k then updates its RT as: $RT_k(d, l) \leftarrow (1-\alpha)RT_k(d, l) + \alpha\Gamma(s_l, ..., s_d)$, where $\Gamma()$ is an aggregation function on a set of node state values, and α is a discount factor ($0 < \alpha < 1$). Here, we choose Γ as the *minimum* of the set of states since it is the minimum available bandwidth that determines the maximum number of calls on a given path.

The above description of information exchange after *every call connection* is called *non-selective* information exchange (NSIE) which is the approach of [Dutta *et al.*, 2005] and, therefore, treated as the baseline algorithm in this paper. In NSIE, information exchanged via the `ack` messages is significant in updating RTs and, hence, performing decentralised routing efficiently. On the other hand, these messages consume valuable bandwidth and, hence, act as overhead. The larger the `ack's` size is, the more the bandwidth it consumes. To counter this, we make information exchange selective to restrict the `ack` size. Thus, we are essentially solving the problem of controlling resource usage by communication, while retaining system performance.

3 The Information Exchange Strategies

The state value that a node transmits to another at a given time can be identical to what it sent to that node on the previous occasion. This happens if, since the last transmission, exactly as many new calls have been connected through this node as existing calls on this node have terminated and, hence, the resultant bandwidth availability on this node is unchanged. In general, an agent's local state can be identical or similar in terms of information content to what it perceived when it last communicated its state. In such cases, sending state information is redundant and can be discarded. The goal, therefore, is to communicate only when there is a *significant enough* change in an agent's state. Making information exchange *selective* (SIE) in this way, the `ack` size can be reduced. To this end, we discuss the requirements and the design of SIE, how SIE affects system performance, and the notion of communication selectivity. Finally, the adaptive (ASIE) protocol is introduced.

Selective Information Exchange: To implement SIE, a node should have the following:

Transmitter memory ($\mathcal{T}$): To store the state values sent to *each neighbour* on the most recent transmission. So, $\mathcal{T}_i(n)$ is the most recent state-value transmitted by i to neighbour n. These are compared against the current state value to determine a redundancy. Thus, a node i needs a memory of size $\mid K_i \mid$, where K_i is its set of neighbours, for its decision of when to transmit.

Receiver memory ($\mathcal{R}$): To store the state values received from *any other node* on the most recent transmission from the latter. So, $\mathcal{R}_i(j)$ is the most recent state-value received by i from j. Thus, i uses this stored value to update its RT when

[1] E.g., a measure of the "distance" between information elements.

it receives an `ack` that does not include the current state of j. This happens if j detects a redundancy and does not transmit. On the other hand, when it receives the state value of j (when j has not identified a redundancy), it replaces $\mathcal{R}_i(j)$ with this value and updates its RT with this new information. Thus, a node i needs a memory of size $\mid A \mid -1$, where A is the set of all nodes, for its RT-update decision.

Given $\mathcal{T}$ and $\mathcal{R}$, a formal description of the selective information exchange decision and RT-update action follows:

Transmitting State Value Selectively. In SIE, a node i ($1 < i \leq d$) transmits its state to its "upstream" node $\{i-1\}$ if: $s_i \neq \mathcal{T}_i(i-1)$ (1). If condition (1) holds and i transmits its state s_i to $\{i-1\}$, then i updates $\mathcal{T}_i(i-1)$ to: $\mathcal{T}_i(i-1) = s_i$. However, if condition (1) is not true, then i does not append s_i to the `ack` before sending it to $\{i-1\}$.

Updating RT. A node i ($1 \leq i < d$) uses the following set of state values of the "downstream" nodes to update its RT: $\{s'_{i+m} | m = 1, ..., (d-i) \wedge s'_{i+m}$ is the latest of $\mathcal{R}_i(i+m)$ and $s_{i+m}\}$ (2). In the values in (2), i determines the "recency" of state information by checking if the value of node $\{i{+}m\}$ is present in the message. If it is, then $\{i{+}m\}$ has transmitted this value by determining that this transmission is not redundant (using condition (1)). Then, i uses this value to update its RT (as stated in Section 2) and also updates $\mathcal{R}_i(i+m)$ to be s_{i+m}. However, if i does not find the state value of $\{i+m\}$ in the message, then it uses the last value received from $\{i+m\}$, $\mathcal{R}_i(i+m)$, to update its RT.

Using the above-mentioned, we now analyse the effects of SIE on decentralised routing along different dimensions:

Memory size. Each node i needs a memory of size $\mid K_i \mid + \mid A \mid$-1 to store $\mathcal{R}$ and $\mathcal{T}$. Thus, memory size is *linear* with the number of nodes, but remains constant at run-time. Now, hardware memory is cheap and can be easily incorporated into the nodes.

Message size. This has a positive correlation with bandwidth use. It increases only if a node appends a state value. *Using* SIE, *therefore, the message size cannot be any more than* NSIE. In fact, node states remain unchanged over time intervals when new calls do not originate and when existing calls are ongoing. In these scenarios, SIE would restrict the size of `ack`, thus reducing overhead and freeing up valuable bandwidth.

Quality of routing table. Information exchange allows RTs to remain up-to-date with the actual end-to-end bandwidth availabilities. Now, since SIE restricts communication, the concern is whether it adversely affects the quality of RTs (hence, the overall routing quality). However, SIE prevents information exchange only if it is identical to what has been communicated previously. Thus, by storing in memory the latest transmissions ($\mathcal{T}$), any new information is guaranteed to be transmitted. *Hence,* SIE *keeps the RTs up-to-date in exactly the same way as* NSIE*; so routing performance should be as good as* NSIE.

The definition of SIE implies that information exchange is restricted only when there is an *exact equality* between the current state-value and the one transmitted most recently. This is a very restrictive condition for defining redundancy and, hence, for information exchange being selective. It can be relaxed from zero difference (condition (1)) by incorporating a "threshold difference" between present and past state values. Such relaxation would further restrict information exchange resulting in further bandwidth savings, although it may deteriorate the quality of RTs.

In more detail, a threshold difference (TD) is the amount of change in state-value that a node needs to observe before transmitting its value. Thus, a TD of Δs_i for node i implies that i transmits to its neighbour j iff: $\mid s_i - \mathcal{T}_i(j) \mid \geq \Delta s_i$. (3) The higher the value of Δs, the more selective the communication, the lower the number of transmissions, and the smaller the message size (a large Δs implies nodes communicate only when there is at least an equally large state-change). However, when Δs is small, nodes communicate for both large and small state changes.

Adaptive Selective Information Exchange: In SIE, all agents use a common, pre-set, threshold to communicate. However, agents can choose different levels of communication selectivity and each such combination can affect the system performance differently. Thus, SIE searches only a limited region of the solution space. We now alleviate this by letting each agent choose adaptively its own TD over time (thus, exploring a larger number of combinations of communication behaviours). By so doing, we could have a more effective system by improving upon the efficiencies of both task processing and resource usage than achieved by SIE. To use such adaptive selective information exchange (ASIE), we need to determine *how* an agent should choose a particular TD at a given time. To do this, we use stored performance profiles [Zilberstein and Russell, 1996] of a network.

In more detail, to generate performance profiles, we let a single agent use SIE (with different TD values) and all others use NSIE. Then, for each TD value, we measure the difference of this system's performance from that where all agents use NSIE. This determines how a single agent's communication selectivity affects global performance over time and, in turn, defines our performance profiles. Subsequently, ASIE is implemented when each agent selects, at a given time, the TD which generated the best performance profile. Thus, the stored profiles are re-used at run time by each agent to dynamically select its own TD.

However, in selecting TD dynamically as described above, the following points should be noted. (1) Our ASIE approach is myopic since each agent considers only how its *own* communication action influences global performance. Using estimates of the actions of other agents could generate better performances, but is likely to cause practical difficulties in terms of the computational complexity associated with such estimation [Goldman and Zilberstein, 2004]. (2) The time-indexed performance profiles are re-used in the ASIE system, where, in fact, the system could be at a different global state than it was at the same time when the profile was generated. Thus, the TD that generated the best profile at a given time (and, thus, is selected in ASIE) can be different from the one required in the ASIE system at the current time. To alleviate this potential limitation, an agent needs to estimate the global state: again, a practical problem. (3) We generate performance profiles by letting one agent use SIE. Now, in a distributed system, the communication actions of different agents can impact global performance differently. In this case, the performance profiles of different agents will be different. Nevertheless, we assume that by choosing a "representative" agent, we can generate a performance profile that will be fairly close to those of any agent. Moreover, performance profiles of multiple agents can be stored in case of systems with heterogeneous agents.

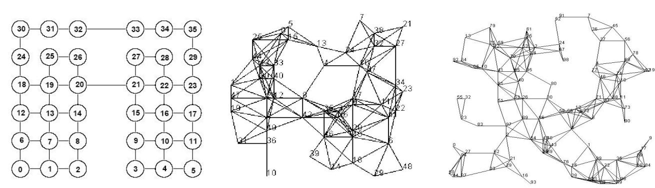
4 Experimental Evaluation

For our experiments, we use the network simulator of [Dutta *et al.*, 2005]. We create different network environments by changing the following simulation parameters so that the results on performance tradeoffs can be generalised.

• **Topology:** Different network structures can affect system performance differently, although we would want to establish some general conclusions about performance trends. To this end, we use a 36-node irregular grid topology (fig 1(a)) used previously in [Boyan and Littman, 1993] to make valid comparisons. Also, two random topologies of 50 nodes (fig 1(b)) and 100 nodes (fig 1(c)) are used.

• **Traffic patterns:** Both *uniform* and *non-uniform* network traffic is used. In the former, a common call origination probability (COP) exists for all nodes and remains constant throughout an experiment. Thus, the network faces a steady traffic inflow. In the latter, the COP is selected randomly per node and then used to generate a call.

• **Information exchange algorithms:** We compare SIE against NSIE (with various TD values) by comparing their routing quality and message overheads across different topologies and load patterns. ASIE and NSIE are compared similarly.



(a) 36nodes (b) 50nodes (c) 100nodes

Figure 1: Random topologies

We choose the following performance measures (as per [Dutta *et al.*, 2005]):

• **Call success rate (CSR):** This is the fraction of the number of calls originating that actually get connected to their destinations. Given a network load and maximum node bandwidth, the more accurate the routing tables, the more likely it is that calls will be connected. Thus, CSR is a measure of routing quality (the coordination effectiveness).

• **Message size (MS):** The size of an ack message is the number of state-values appended to it. The smaller the message size, the higher the bandwidth saved (the more efficient the information-sharing algorithm).

We study the impact of communication selectivity (TD) on network performance. Such empirical results allow a system designer or user to examine performance tradeoffs across different network environments. In this context, we can form subjective hypotheses such as: (i) small TD generates good bandwidth savings and only slightly deteriorates call routing; or (ii) high TD generates significant bandwidth savings but significantly deteriorates call routing. In our experiments, first, the CSR and MS of the SIE and ASIE are compared against those of NSIE using uniform load, followed by using non-uniform load. The following parameter values are used (as per [Dutta *et al.*, 2005]): $\alpha = 0.03$, $\tau = 0.1$, and maximum number of simultaneous calls per node is 10.

Table 1: NSIE performance: uniform load

COP	Topology 1				Topology 2				Topology 3			
	CSR		MS		CSR		MS		CSR		MS	
	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev
0.1	0.487	0.003	3.86	0.022	0.493	0.003	3.52	0.015	0.287	0.003	4.026	0.042
0.2	0.328	0.001	3.16	0.014	0.363	0.002	2.95	0.009	0.217	0.002	3.31	0.023
0.4	0.213	0.0001	2.58	0.01	0.247	0.001	2.32	0.012	0.158	0.001	2.676	0.016
0.6	0.164	0.0004	2.26	0.009	0.191	0.0005	1.996	0.009	0.128	0.0006	2.365	0.016
0.8	0.134	0.0004	2.06	0.006	0.158	0.0004	1.80	0.007	0.108	0.0007	2.159	0.008

4.1 Uniform Load

Each experiment uses a constant COP chosen from the set $\{0.1, 0.2, 0.4, 0.6, 0.8\}$. Note, the higher the COP value, the more calls originate and, hence, the higher the network load.

Performance of NSIE: Our baseline results (NSIE) are summarized in Table 1, showing the steady state average CSR and average MS values for different topologies. For all topologies, the average CSR decreases as the COP increases (calls originating increase). With limited bandwidth (10 units per node), as the COP increases, larger percentages of originating calls fail to be connected. Hence, the CSR decreases. Also, the average MS decreases with increasing COP. At higher COPs, when all nodes generate large number of calls, only the short-distance calls connect. Having limited bandwidth, a node gets saturated with the calls that it originates. Then, the extent to which a node routes calls originating elsewhere decreases with increasing COP. Since ack size is determined by the route length, MS decreases with less long-distance calls connecting at high COP.

Performance of SIE: Table 2 shows the steady state average CSR and MS in the different topologies using SIE when $\Delta s = 0.1$. This is the *minimum value that* Δs *can take to have any communication selectivity*, generating a baseline SIE. For all other Δs, we get the same trends (excluded due to limited space).

In Table 2, the trends in both CSR and MS are identical to those in Table 1. However, both of these in Table 2 are lower than the corresponding values in Table 1. Hence, SIE *uses less bandwidth than* NSIE, *but it has a slightly reduced call success rate*. We then compare these across all COP and Δs values by generating the relative differences of the average CSRs and MSs as: $\text{CSR}_{diff} = (\text{CSR}(\text{NSIE}) - \text{CSR}(\text{SIE}))/\text{CSR}(\text{NSIE})$, and $\text{MS}_{diff} = (\text{MS}(\text{NSIE}) - \text{MS}(\text{SIE}))/\text{MS}(\text{NSIE})$. Thus, the higher the value of CSR_{diff}, the poorer is the solution quality of SIE compared to NSIE. However, the higher the MS_{diff}, the more efficient the communication of SIE.

Studying the CSR_{diff} values (figure 2(a)) of topology 1, we learn: (1) *At any* COP, *for small* Δs, *the* CSR *of* SIE *is very close to that of* NSIE. Thus, when $\Delta s < 0.4$, CSR_{diff}

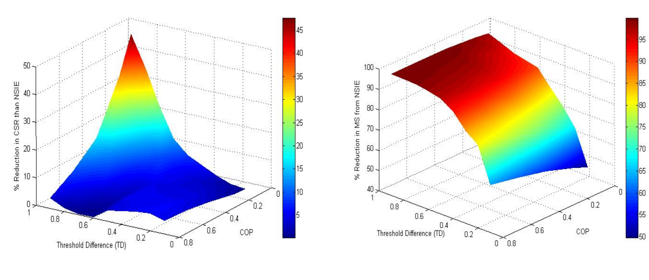
Table 2: SIE performance (Δs = 0.1): uniform load

COP	Topology 1				Topology 2				Topology 3			
	CSR		MS		CSR		MS		CSR		MS	
	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev
0.1	0.484	0.002	1.938	0.015	0.492	0.004	2.455	0.016	0.274	0.003	1.89	0.02
0.2	0.325	0.001	1.50	0.011	0.363	0.002	2.067	0.011	0.212	0.001	1.60	0.011
0.4	0.207	0.0006	1.098	0.004	0.246	0.0009	1.612	0.006	0.156	0.0008	1.298	0.006
0.6	0.157	0.0005	0.891	0.006	0.190	0.0007	1.352	0.004	0.127	0.0006	1.124	0.005
0.8	0.128	0.0005	0.755	0.004	0.156	0.0005	1.18	0.003	0.107	0.0004	0.985	0.005

is less than 4.0% for any COP. When Δs is low, communication occurs for both small and large state-changes. Now, small state-changes occur at any COP value because the origination and termination of few calls are more common than a larger number. Larger state-changes, however, occur more at higher COPs. Thus, for any COP, a low Δs value allows frequent communication. This keeps the RTs updated fairly well, thus, generating good call success rates (close to NSIE). (2) *At larger Δs, the* SIE CSR *is significantly lower than* NSIE *when* COP *is small, but remains close to* NSIE *at higher* COP*s.* Thus, $\Delta s = 0.9$ causes a 32.0% CSR$_{diff}$ when COP is 0.1, but it is only about 3.0% when COP is 0.8. High Δs allows communication only for large state-changes. Now, large state-changes are rare at small COPs when nodes are sparsely occupied, and so is communication. Thus, the RTs are not up-to-date, causing poor routing and reducing the CSR of SIE. But, at higher COPs, large state-changes (and communication) are more frequent; hence CSR of SIE is close to NSIE.

Studying the MS$_{diff}$ values (figure 2(b)) of topology 1, we learn: (1) *At any* COP*, the larger the Δs, the larger is the* MS$_{diff}$ *value.* Since communication reduces with Δs, so does the MS of SIE (but not influencing the MS of NSIE). These results confirm that larger bandwidth savings are achieved by making communication ever more selective. (2) *At any Δs, a change in* COP *does not significantly change the* MS$_{diff}$ *value.* Thus, SIE maintains a uniform advantage over NSIE in terms of saving bandwidth for all COP values.

In summary, the significant result is that the bandwidth usage improvement of SIE (up to 99%) clearly offsets its degrading in routing (as low as 35%). More typically, we obtain 30-80% improvement in bandwidth efficiency at the expense of less than 5% degradation in routing quality. The results from the other topologies show similar trends.



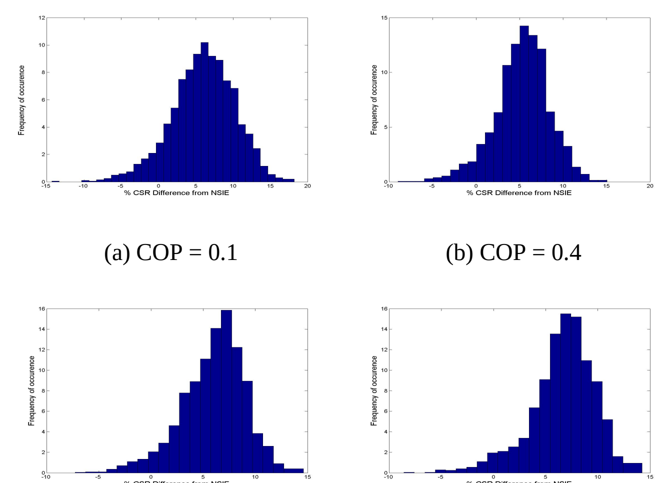
(a) CSR difference (b) MS difference

Figure 2: CSR and MS differences between SIE and NSIE

Performance of ASIE: We study ASIE on a regular 6×6 grid where the centroid is the representative node (n_R) (section 3). Here, performance profiles (*pp*) represent time-variations of the percentage deviations of global CSR and MS between NSIE and the one where only n_R uses SIE and all others use NSIE. In ASIE, node i selects that Δs_i which generates the least degradation in CSR for the COP value, at that time, in the profile. It then uses this Δs_i for its communication decision (equation (3)) until it chooses a new Δs. We select a period of 1000 time steps for storing performance profiles and, thus, for choosing a new Δs.

For each COP, we studied the frequency with which the CSR of ASIE deviated from that of NSIE (figure 3). In each graph, using a different COP, the x-axis shows the different (percentage) CSR deviations between ASIE and NSIE. The y-axis shows the (percentage) duration of a simulation run for which the corresponding CSR deviation (the x-value) was observed. Thus, in figure 3(a), a 6% CSR deviation was observed (ASIE worse than NSIE by 6%) for 10.2% time of the simulation. We conclude: *(1) At smaller* COP *values,* ASIE *performs better than* SIE *by generating* CSR*s that are closer to the baseline* NSIE. Thus, the majority of the CSR differences between ASIE and NSIE is 4-6% for COP less than 0.6. For the same COP values, the mean CSR differences between SIE and NSIE is 8-16% (measured on the same topology but omitted here due to lack of space). So, the CSR of ASIE is up to 12% closer to the benchmark than SIE, implying a further improvement in the quality of task processing. *(2) At higher* COP *values,* SIE *performs slightly better than* ASIE *by achieveing* CSR*s closer to* NSIE. Thus, for COPs 0.6 and 0.8, the CSR difference in ASIE is mostly 6-9%, while for SIE it is 4.5-5.6%. The slight decrease in ASIE performance compared to SIE is due to the limitation of a myopic algorithm (ASIE) in a dynamic environment (large COP).

Now studying the MS of ASIE and SIE, we observe: *For all* COP*,* ASIE *ensures higher bandwidth savings than* SIE: ASIE saves up to 85-100% compared to NSIE, which is greater (by up to 10%) than the bandwidth savings of SIE compared to NSIE (plots omitted due to limited space).



(a) COP = 0.1 (b) COP = 0.4

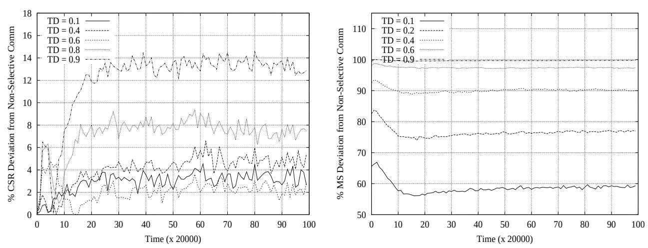
(c) COP = 0.6 (d) COP = 0.8

Figure 3: CSR difference between ASIE and NSIE

4.2 Non-Uniform Load

Having a dynamic COP (varies between 0.1 and 0.9), we measure performances over time instead of steady-state averages.

<u>Performance of SIE</u>: Figure 4 shows the percentage differences of the CSR and the MS between SIE and NSIE for topology 1 (other topologies show similar trends). We conclude: *(1) The higher the* TD, *the smaller is the* CSR *of* SIE *than* NSIE (poorer routing than NSIE, figure 4(a)), and *(2) the smaller is the* MS *of* SIE *than* NSIE (more bandwidth savings than NSIE, figure 4(b)). More typically, 60-90% bandwidth saving is achieved incurring less than 10% loss in CSR; further confirming that SIE is more bandwidth efficient and its advantage thoroughly offsets its limitation.



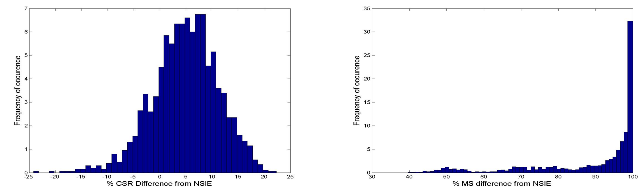
(a) CSR difference (b) MS difference

Figure 4: CSR and MS differences between SIE and NSIE

<u>Performance of ASIE</u>: Here, we use the performance profiles of uniform load experiments using the 6×6 grid topology. This is a myopic choice since, with dynamic load, the COP values are different across nodes which was not the case in uniform load. Now, at run-time, each agent estimates the average COP ($\hat{cop}$) at its node over the past T (=1000) time steps. Then, it determines its Δs value as: $\text{argmin}_j \; pp_{CSR}(\hat{cop}, \Delta s_j, t)$, where pp_{CSR} is the CSR performance profile. The CSR and MS differences between the above-mentioned ASIE system and NSIE are shown in figure 5. We also run SIE to compare it against NSIE (results omitted due to space limitation). Our results show, with non-uniform load, ASIE performed better than SIE over all Δs. Thus, *the mean* CSR *difference in figure 5(a), about 7%, is smaller than the minimum* CSR *difference, about 8.1%, achieved by* SIE *across all* Δs, and *the mean* MS *difference in figure 5(b), about 99.9%, is larger than the maximum* MS *difference, about 99%, achieved by* SIE, *across all* Δs *values*. A fixed TD for all nodes (as in SIE) is not an effective strategy in dynamic load. Rather, the nodes should adapt their communication behaviours; hence, ASIE outperforms SIE.

5 Conclusions

Our novel protocol for *selective* information-sharing effectively trades off communication utility and cost in resource constrained MAS. It uses *information redundancy* for communication decisions to save bandwidth and achieve good coordination. Furthermore, it adapts the selectivity levels using stored performance profiles. Empirical studies on a call routing problem shows, compared to the most effective communication strategy currently available, our protocol achieves significant bandwidth savings, while achieving only a slightly lower throughput. The adaptive strategy achieves further improvements over the non-adaptive one.



(a) CSR Difference (b) MS Difference

Figure 5: CSR and MS differences between ASIE and NSIE

Acknowledgements: This research was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Systems) project and is jointly funded by a BAE Systems and EPSRC (Engineering and Physical Research Council) strategic partnership (EP/C548051/1).

References

[Boyan and Littman, 1993] J. A. Boyan and M. L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In *NIPS*, pages 671–678, 1993.

[Dutta *et al.*, 2005] P. S. Dutta, N. R. Jennings, and L. Moreau. Cooperative information sharing to improve distributed learning in multi-agent systems. *JAIR*, 24:407–463, 2005.

[Gmytrasiewicz and Durfee, 2001] P. J. Gmytrasiewicz and E. H. Durfee. Rational communication in multi-agent environments. *JAAMAS*, 4:233–272, 2001.

[Goldman and Zilberstein, 2003] C. V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *AAMAS*, pages 137–144, 2003.

[Goldman and Zilberstein, 2004] C. V. Goldman and S. Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *JAIR*, 22:143–174, 2004.

[Haas *et al.*, 2000] Z. Haas, J. Y. Halpern, L. Li, and S. B. Wicker. A decision-theoretic approach to resource allocation in wireless multimedia networks. In *Proc. of the* 4^{th} *DIALM Workshop*, pages 86–95, 2000.

[Shen *et al.*, 2003] J. Shen, V. Lesser, and N. Carver. Minimizing communication cost in a distributed bayesian network using a decentralized MDP. In *AAMAS*, pages 678–685, 2003.

[Xuan *et al.*, 2001] P. Xuan, V. Lesser, and S. Zilberstein. Communication decisions in multi-agent cooperation: model and experiments. In *Agents*, pages 616–623, 2001.

[Zhang *et al.*, 2004] Y. Zhang, R. A. Volz, T. R. Ioerger, and J. Yen. A decision-theoretic approach for designing proactive communication in multi-agent teamwork. In *ACM Symp. on Applied Computing*, pages 64–71, 2004.

[Zilberstein and Russell, 1996] S. Zilberstein and S. Russell. Optimal composition of real-time systems. *AIJ*, 82(1-2):181–213, 1996.