Using Stand-In Agents in Partially Accessible Multi-agent Environment

Martin Rehák, Michal Pěchouček, Jan Tožička, and David Šišlák

Department of Cybernetics,
Czech Technical University in Prague,
Technická 2, Prague 6, 166 27 Czech Republic
{rehakm1, pechouc, tozicka}@labe.felk.cvut.cz
sislakd@feld.cvut.cz

Abstract. This contribution defines a metrics and proposes a solution for the problem of agents inaccessibility in multi-agent systems. We define the stand-in pattern for knowledge maintenance and remote presence in distributed agent systems with communication inaccessibility. Our implementation has been designed and tested in the $\mathcal{A}$-**globe** agent platform. We also present a set of measurements quantifying agents' inaccessibility in our domain and comparing the usefulness of different solution in the environments with different inaccessibility.

1 Introduction

Nowadays, most agent systems are physically localized in one location or connected by fixed networks. Therefore, the inaccessibility is coped with on the lower parts of the protocol stack. The agents themselves treat the inaccessibility situations as rare, error causes and can not react appropriately in such situations. With the increasing use of physically distributed agent systems in the external environment and the appearance of mobile or static mesh networks [1] for their connection, agent developers will have to solve accessibility or inaccessibility related problems to make their systems more reliable and useful.

Therefore, agents' inaccessibility [2] in a multi-agent community is an uneasy problem of a high practical importance. Agents become inaccessible when they want to communicate but it is not possible. There are several different reasons why an agent may become inaccessible from the other members of the multi-agent community - such as malfunction of the communication links, communication traffic overload, agent leaving the communication infrastructure for accomplishing a specific mission, agent failing to operate, etc.

Consequently, there is a need for an unified and general technology for maintaining social stability/sustainability in multi-agent system with inaccessible agents.

Within the frame of our work we have been comparing the original concept of the **stand-in agents** with the classical relaying approaches. While relaying provides a simple re-direction technology used in computer networks (e.g. it provides only routing of messages in order to implement accessibility between two

M.-P. Gleizes, A. Omicini, and F. Zambonelli (Eds.): ESAW 2004, LNAI 3451, pp. 277–291, 2005.

agents where direct connection is not possible), deployment of stand-in agents represents more advanced concept and suggest a whole set of interesting research problems. Conversely, the stand-in agent is a distant representative of a respective agent – the owner. Stand-in agents are created by their owner and they migrate to different segments of the communication infrastructure that may become inaccessible in the near future. When inaccessibility occurs the stand-in agent acts on the owners behalf.

In this article we will discuss the problem of inaccessibility and suggest specific quantities for measuring inaccessibility. In the section 3 we will discuss possible solutions for inaccessibility. Selected approaches will be then compared in section 4, together with validation of applicability of theoretical concepts presented in the section 2.

2 Measuring Inaccessibility

Systematically we distinguish between several classes of inaccessibility. Inaccessibility can be caused e.g. by unreliability of the communication infrastructure, balancing the cost of the communication, dynamic changes of the communication infrastructure topology, etc.

Quantification of inaccessibility in a multi-agent system is an important problem. In the following we discuss several metrics of inaccessibility that we have been using throughout our research project.

Let us introduce a **measure of inaccessibility**, a quantity denoted as $\overline{\vartheta} \in [0;1]$. This measure is supposed to be dual to the **measure of accessibility** – $\vartheta \in [0;1]$, where $\vartheta + \overline{\vartheta} = 1$. We will want ϑ to be 1 in order to denote complete accessibility and ϑ to be 0 in order to denote complete inaccessibility. In the following text we will mostly describe the agents' accessibility while the inaccessibility is its complement.

We will use the random graph theory [3] in order to describe some general properties of communication inaccessibility in multi-agent systems. Random graph theory has been recently successfully used for theoretical studies of complex networks [4]. Let us represent the multi-agent community as a graph. The agents are represented by nodes and available communication links – connections where the information exchange is possible – by edges. Unlike in the general case of agents inaccessibility, the random graphs theory works with an assumption that all edges are present with the same probability p. In our domain, this probability is represented by **link accessibility**: $p = \vartheta$.

The ϑ link accessibility can be determined in two ways. Firstly as time accessibility ϑ_t:

$$\vartheta_t = \frac{t_{\mathrm{acc}}}{t_{\mathrm{inacc}} + t_{\mathrm{acc}}}, \tag{1}$$

where t_{acc} denotes the amount of time when communication is possible while t_{inacc} denotes time when agents are disconnected.

Similarly, we may measure accessibility as a function of sent communication messages (communication accessibility ϑ_m):

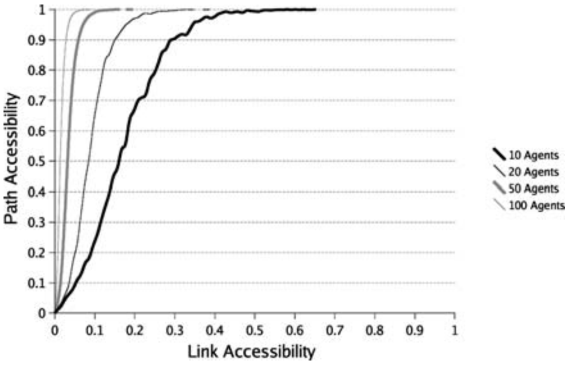


Fig. 1. The dependency of probability of existence path between two agents and link accessibility. This graph is the same for the link accessibility with or without the symmetry

$$\vartheta_m = \frac{|m| - |m_{\text{fail}}|}{|m|}, \tag{2}$$

where $|m|$ denotes the total number of messages sent and $|m_{\text{fail}}|$ the number of messages that failed to be delivered. The accessibility measure ϑ_t is symmetrical between entities A and B

$$\vartheta_t(A, B) = \vartheta_t(B, A), \tag{3}$$

while the accessibility measure ϑ_m is not necessarily symmetrical.

In the following we will discuss ϑ_t while most conclusions apply equally to ϑ_m.

We have been investigating primarily the domain of mobile ad-hoc networking among computational units constantly changing their physical location. In this domain, ϑ_t accessibility depends on the environment agent positions only, while ϑ^m accessibility depends also on other factors, like communication link load or limited social knowledge of the agents.

We have determined the probability of existence path between two agents - **path accessibility** - depending on link accessibility in simple mathematical simulation. The result is shown in Figure 1. Classical result of the random graph theory is that there exists a critical probability at which large cluster appears. In our domain, we assume that there is a **critical accessibility** – ϑ^{c} such that below ϑ^{c} the agent community is composed of several isolated groups but above ϑ^{c} most of agents become mutually path-accessible (using relay agents). The ϑ^{c} value is represented by the quick growth in the Figure 1. This observation is similar to a percolation transition known in the field of mathematics and statistical mechanics [5]. In the field of multi-agent systems, it means that the relay agents are more efficient than isolated stand-in agents for link accessibility bigger than ϑ^{c}. Our testing scenario, presented in section 4 and implemented using actual multi-agent system based on $\mathcal{A}$**-globe** [6] allows to set up and verify properties of both cases.

Table 1. Different cases of accessibility as described by random graph theory

$\vartheta_t n < 1$	The network is typically composed of isolated trees. The diameter is equal to the diameter of trees.
$\vartheta_t n > 1$	A large cluster is formed. The diameter is equal to the largest cluster diameter and if $\vartheta_t n > 3.5$ it is proportional to $\frac{\ln(n)}{\ln(\vartheta_t n)}$.
$\vartheta_t n > \ln(n)$	The graph is probably totally connected and the diameter is very close to $\frac{\ln(n)}{\ln(\vartheta_t n)}$.

Second relevant result of random graph theory is the average length l^* of path between any two vertices and the diameter l^d of a graph (i.e. maximal distance between any two nodes). It holds [4]:

$$l^* \sim l^d = \frac{\ln(n)}{\ln(\vartheta_t n)},$$

where n is number of agents.

In our domain, the length of the path says how many relays has to be used in order to convey a message between the agents A and B. And as a result of random graph theory, the maximal number of relays necessary is not much greater then the number of relays in average case.

Table 1 summarizes several results of random graph theory important for our study of inaccessibility.

These properties are well observable also in our domain (see section 4.2).

3 Solving Inaccessibility

We are now going to analyze existing methods coping with inaccessibility. Two main approaches can be distinguished between them: building **remote awareness** or **remote presence**.

When an agent builds remote awareness, allows the remote agents to update their social knowledge with relevant information about itself and to let them operate using this information. This process may be implemented using either pull or push information retrieval operations. Typical examples are acquaintance models described in section 3.3, matchmaking middle agents (3.2) or synchronization and search in peer-to-peer networks [7].

When an agent builds a remote presence, it does so in order to operate **actively** in the remote location. As a collateral effect of this action, the agent may also build a remote awareness - as in the stand-in case, but this does not necessarily apply when we use middle agents. Examples of this approach are relaying (3.1), stand-ins (3.4) or broker middle agents (3.2).

3.1 Relay Agents and Adaptive Networks

First, and perhaps the most classical solution to the inaccessibility problem are relay agents (or low-level entities), responsible for setting up a transmission path through other elements when the direct contact between parties is impossible. Such protocols are currently widely implemented for routing in various types of networks, like TCP/IP [8] or on lower levels [1]. However, this solution is efficient only if the network is in a "reasonably connected" state (see third row of Table 1 and Figure 1). Besides this limitation, that can be clearly distinguished in the results of our experiments, there are several other factors limiting the use of relayed connection. These factors are for example reduced battery life due to the fact that all the messages must be transmitted several times, or network maintenance overhead, especially in case of mobile networks. Another factor limiting the use of relaying in agent systems is the dynamic nature of their topology if the agent platforms are based on moving entities. In this case, relaying cost increases as the link maintenance and path-finding in dynamic environment is a non-trivial process.

3.2 Middle Agents

Middle agent is a term that can cover a whole range of different facilitators in a multi-agent system. In an overview article [9], authors list different types of middle agents - **Matchmakers** and **Brokers** (Facilitators). Matchmakers may provide remote awareness by notifying interested agents about the presence of service providers, while the brokers can act as intermediaries and pass actual service requests between two mutually inaccessible parties. Even if this solution may perform very well in many situations, it may be unusable if middle agents are difficult to find, unreliable, or can not be trusted with private preferences of different parties. Stand-in agents described later are intended to close this gap.

3.3 Social Knowledge and Acquaintance Models

Social knowledge represent necessary and optional information which an agent needs for its efficient operation in the multi-agent community. The social knowledge is mainly used for reduction of communication, provides self-interested agents with a competitive advantage and allows agents to reason about the others in environments with partial accessibility.

The acquaintance model is a very specific knowledge structure containing agent's social knowledge. This knowledge structure is in a fact a computational model of agents' mutual awareness. It does not need to be precise and up-to-date. Agents may use different methods and techniques for maintenance and exploitation of the acquaintance model. There have been various acquaintance models studied and developed in the multi-agent community, eg. *tri-base acquaintance model* [10] and *twin-base acquaintance model* [11]. In principle, each acquaintance model is split into two parts: **self-knowledge** containing information about an agent itself and **social-knowledge** containing knowledge about other members of the multi-agent system.

While the former part of the model is maintained by the **social knowledge provider** (an owner), the latter is maintained by the **social knowledge requestor** (a client).

Social knowledge can be used for making operation of the multi-agent system more efficient. The acquaintance model is an important source of information that would have to be repeatedly communicated otherwise. Social knowledge and acquaintance models can be also used in the situations of agents' short term inaccessibility. However, the acquaintance models provides rather '*shallow*' knowledge, that does not represent a complicated dynamics of agent's decision making, future course of intentions, resource allocation or negotiation preferences. This type of information is needed for inter-agent coordination in situation with longer-term inaccessibility.

3.4 Stand-In Agent

An alternative option is to integrate the agent self-knowledge into a mobile computational entity that is constructed and maintained by the social knowledge provider. We will refer to this computational entity as a **stand-in agent**. The stand-in agent resides either on the same host where the social knowledge requestor operates or in the permanently accessible location. While using stand-in, the social knowledge requestor does not create an acquaintance model of its own. Instead of communicating with the provider or middle agent, it interacts with the stand-in agent. Therefore, the client agent is relieved from the relatively complex task of building and keeping up-to-date detailed acquaintance model and both provider and requestor may benefit from the full-fledged remote presence. Factoring the acquaintance model out of the each requestor agent internal memory allows it to be shared between all locally accessible agents, further minimizing the traffic and computational resources necessary for model maintenance.

In our implementation the community of stand-in agents operates in two phases: **stand-in swarming**, **information propagation** and **social knowledge synchronization**.

During the swarming phase, stand-ins propagate through the system to reach the locations that may become inaccessible in the future. First, existing stand-in agent or knowledge provider determines set of currently accessible locations using broadcast-like mechanism of underlying communication infrastructure. It analyzes the locations and decides which entities are interesting for further stand-in agent deployment, either because of the presence of knowledge requestor agent or because it considers the location to be interesting for future spread. Then, it may decide to create and deploy its clones on one or more of these accessible locations. After its creation, each deployed stand-in agent chooses the type of functionality it will provide in its location and repeats the evaluate/deploy process. The swarming propagation strategy is a crucial element of agent system tuning, as we must find a delicate balance between information spread efficiency and resources consumed by stand-ins.

Information propagation between members of the stand-in community is also a challenging process to tune, because the information flows not only from the knowledge provider towards the stand-in community, but also from the stand-in community towards knowledge provider, or even within the isolated parts of the stand-in community.

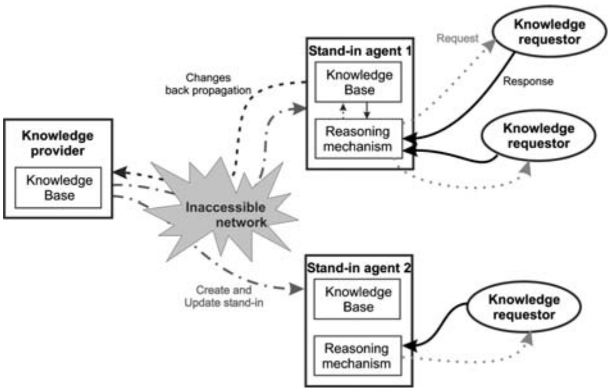


Fig. 2. The concept of the stand-in agent

When a member of the stand-in community receives an update of the shared knowledge, or updates this knowledge after having acted on behalf of knowledge provider, it must determine if the information update is valuable enough to be propagated to other members of the community and eventually to the knowledge provider itself. It determines the list of currently accessible stand-ins in the community to which it will send the updated knowledge set or relevant subset and keeps the updated information ready for future synchronization with currently inaccessible stand-ins.

In our current implementation, we do present two limit approaches to information synchronization. In the first configuration, we consider the cost of communication to be important and the stand-ins therefore synchronize their knowledge only when they encounter. When they receive an information update, they don't propagate it to other accessible members of the community.

Our second approach is based on an assumption that communication is cheap and that all updates are worth to be propagated to all accessible members of the community. In this approach, any stand-in that updates the information or receives more recent version sends this update to all accessible members of the community. When two stand-ins become accessible, they exchange their information and join it into the shared common version, as ensured by domain-specific joining algorithm. This policy ensures an optimum information quality on domain elements, but must be optimized for domains with big number of locations and represented agents, for example using existing results from peer-to-peer networks research domain [7].

The most important added value of stand-in agent is not in providing remote awareness, but in providing rich and proactive remote presence by acting on behalf of knowledge provider. However, as in any system working on the shared data, synchronization problems arise in the agent community when the stand-ins accept commitments in place of knowledge provider. Situation is further complicated by the fact that no reasonable locking protocol may be implemented between the components that are inaccessible in a given moment. Until now, we have not explicitly addressed the synchronization problem. Its solution may use e.g. sophisticated multi-level negotiation protocols, the concept of structured (rich) commitments or advanced methods of synchronization between locally accessible agents.

The concept stand-in agents are currently advantageous in the two very specific situations:

- in the **very dynamic environment**, with relatively low path accessibility (this can be e.g. in situations where a low number of unmanned vehicles are collaboratively inspecting large areas), or
- in the **non-trusted environment** with at least some communication inaccessibility (in these cases the agent do not want to provide sensitive knowledge for sharing while off-line).

In our current work, we are analyzing and optimizing the collaboration patterns of the stand-in community to make the approach scalable.

3.5 Towards Optimization of the Stand-In Approach

The measurements presented in the next section (4) provide us with the **limits** of performance of various inaccessibility solutions depending on the accessibility of the environment. They show that the use of stand-ins (see 3.4) or other remote awareness and presence technologies allows the multi-agent system to operate in highly inaccessible environments. Currently, we are answering many crucial practical questions concerning the efficient stand-in use.

All these questions are related to the scalability of the approach - until now, we have considered the processing power and bandwidth as either cheap and unlimited or very expensive. With the use of stand-ins in larger domains, these basic assumptions are not valid anymore and consequent issues must be addressed.

The first problem is stand-in deployment. In the large domains featuring significant number of nodes, the complete flooding with stand-ins and their deployment in each container would mean that the stand-ins would outnumber all other agents by a large factor, making their use prohibitive. This would not only consume the memory and processing power of the nodes, but it would also increase the bandwidth necessary for information synchronization and action coordination in the system.

Therefore, we must optimize both the stand-in deployment and information synchronization using domain independent methods. We will try to maintain the system performance close to the theoretical limit established in the experiments (sect. 4.2), while minimizing the number of stand-ins and synchronization messages. On the other hand, this optimization shall not decrease system robustness in respect to the failures - it shall adapt rapidly to the changing situation and keep the information up-to-date under most circumstances.

In our current research, we have pre-selected two approaches to system optimization. The pre-selection criteria were very simple ones - efficiency, robustness to the failure of elements and stability in the rapidly changing environment of mobile ad-hoc networks. The first model is inspired by biology, the second one by micro-economy.

Social dominance and altruism models [12, 13] were successfully used to partition the group of agents into those who work for the good of the community and the others who profit from the altruism of the first group. Interestingly, observations during the experiments with rats in the laboratory environment confirm that such approach actually *maximizes* the survival rate for the members of the community and is stable with respect to changes in the groups of observed rats - both properties being of particular interest to us.

During the experiments with rats, it was determined that a sufficient number of individuals behaves in an altruistic manner. They bring the food and share it with the others, who only consume. Surprisingly, when the group is split, half of the previous altruists change their behavior and become passive, while half of the group that was previously lazy becomes altruistic. This behavior is formalized by a simple mathematical model presented in [12].

I our approach, the stand-ins will be split between altruists and "lazy" individuals. Altruists will form a backbone of the community, as they will pass the information to other altruists and adjacent lazy stand-ins. While the configuration of the community changes, we expect the stand-ins to adopt the new social role and maintain the functionality of the community. The main problem to solve is the actual modification of the model and automatic adaptation to various types of environments with diverse accessibility and mobility characteristics.

Simulated micro-payments model. In this model, stand-ins answer the information updates with a micro-payment, indicating the usefulness of the received information. As the agents subtract the virtual price attributed to the sending of the synchronization message from the received payment, the network shall optimize itself if the agents value the information most upon the first reception and decrease the payment for the updates that are already known. To optimize the number of agents, the approach is similar. Agents who don't generate sufficient gain from representing the owner agent or from relaying the updates to the others in the community terminate themselves.

The main challenge in this approach is in fine-tuning the mechanism - optimizing the virtual costs and payments and determining the probabilities which will be applied to sending unsolicited updates or re-creation of already self-destroyed stand-ins by adjacent stand-ins - both parameters are essential for community re-adaptation in changing environment. All the mentioned parameters will be hardly constant - they will undoubtedly vary in function of the accessibility characteristics, defined in section 2.

We are also analyzing the methods how to enhance these essentially emergent models with a global vision - in a way similar to the adaptive adaptation (or meta-adaptation) as proposed for example by Bedau and Packard [14]. Both emergent (bottom-up) and meta-reasoning approaches are analyzed for this task.

Using the stand-ins as a part of the system brings another interesting aspect. As the owner gives more power to the stand-ins, it increases the likelihood of identifying the optimum partner for the operation. On the other hand, as the parts of the stand-in community may get isolated, an issue of concurrence must be solved. This problem is very similar to adjustable autonomy in human-agent relationship, as studied by Sierhuis et al. [15].

4 Experiments

In this section, we will describe a set of accessibility experiments with a multi-agent simulation. The goal of the experiments was to validate the relevance of the theory presented in the first part (Section 2) of this contribution on a real multi-agent system and to determine the boundaries of applicability of the solutions to the inaccessibility problem presented in the second part (Section 3). First we will investigate and analyze inaccessibility in our scenario and after this we will study how inaccessibility affects performance of our system. Three techniques for coping with inaccessibility will be analyzed.

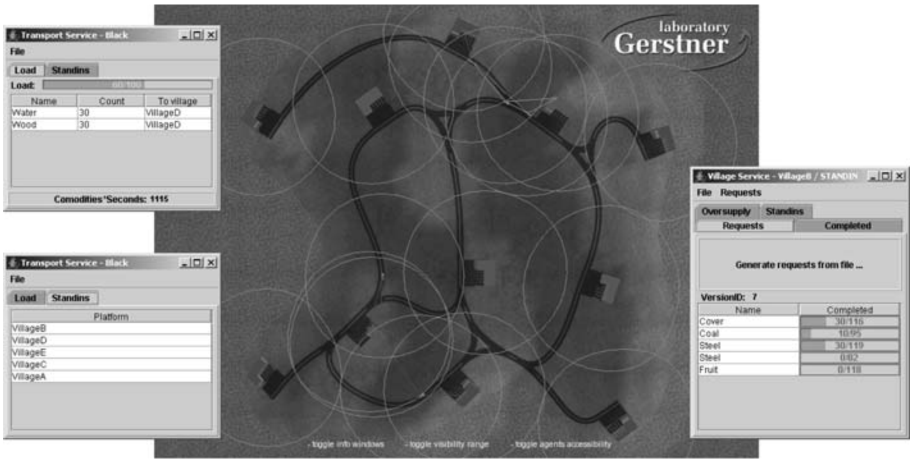
In our measurements, we will validate if the classical random-graph model presented in Section 2 is appropriate for our case, or if we need to apply more realistic network modelling techniques [16]. Then, we will measure the influence the inaccessibility has to the solution of the model domain and the efficiency of three possible approaches dealing with the problem.

4.1 Testing Scenario

For our measurements, we have prepared a simulation featuring a logistics problem in collaborative environment, where the humanitarian aid must be delivered to the zone ravaged by a disaster. In the domain, we will deploy three main types of entities: 5 aid sources, called **ports**, where the material comes in; 5 aid sinks, called **villages**, where it is consumed and 7 **transports** carrying the aid between ports and villages. Each transport has its predefined route that does not change during the simulation. Aid requests in the villages are generated by predefined script to ensure uniformity between simulation runs. They must be transmitted to the ports to ensure that the proper material is loaded on the transport going to the village. The way these requests are transmitted depends on the inaccessibility solution that is currently applied. We suppose that the physical communication links between the entities are limited-range radios, therefore the link exists if the distance is smaller then parameter ϱ. This parameter varies between different scenario runs to model different possible configurations, from complete link accessibility to only local (same position) accessibility Test domain is shown in Fig. 3.

In total, 33 results are presented, with 11 different communication ranges and 3 different approaches solving the inaccessibility problem:

- relaying transmissions by relay agents (3.1) – loading of the goods on a transport is possible only if a communication path exists between the destination village and the port in the moment when port-based entities negotiate the cargo to load,



Name	Count	To village
Water	30	VillageD
Wood	30	VillageD

Comodities*Seconds: 1115

Platform
VillageB
VillageD
VillageE
VillageC
VillageA

VersionID: 7

Name	Completed
Cover	30/116
Coal	10/95
Steel	30/119
Steel	0/82
Fruit	0/118

Fig. 3. Test domain used for the experiments contains 5 ports, 5 villages and 7 transports. Circles represent the accessibility ranges, while the lines the actual accessibility and ongoing communication between nodes

- stand-in agents that only carry the information with no sharing in the stand-in community (see section 3.4),
- community stand-in agents, sharing the information updates with other members of the stand-in agent community (see also section 3.4).

To guarantee the uniformity of results, we have used the same negotiation protocols and work-flow for the interaction between the acting agents and their environment. Both the requests in villages and goods in ports are generated from unique pseudorandom sequence used for all measurements.The only aspect that differentiates the scenarios is the mode of information transmission between requesting villages and goods providers in the ports.

4.2 Measuring Accessibility

On Figure 4, we can identify three major states of the community from the accessibility point of view, as defined in section 2. At first, before the communication radius reaches 60, static community members are isolated and information is not transmitted (see first row of Table 1), but only carried by moving entities. In this state, path accessibility is not significantly different from link accessibility. Therefore, probability of relaying is almost negligible.

Then, with increasing communication radius, larger connected components do start to appear, covering several static and mobile entities and allowing the use of relaying over these portions of the graph. This phase appears around the percolation threshold, that can be observed above radius of 80, corresponding with link accessibility of 0.2. This state is characterized by important variability of connected components. Due to the dynamic nature of our system, these components are relatively short-lived, resulting in a high variability of the system,

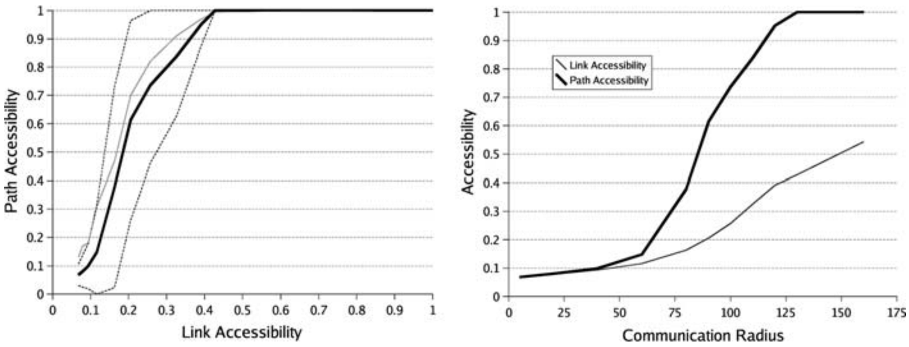


Fig. 4. *Left:* The dependency of probability of existence path between two agents and link accessibility in our test scenario. The dot lines show average deviation of values. The gray thin line shows theoretical value for random graph with 10 nodes (see Fig. 1). *Right:* The dependency of link and path accessibility on communication radius

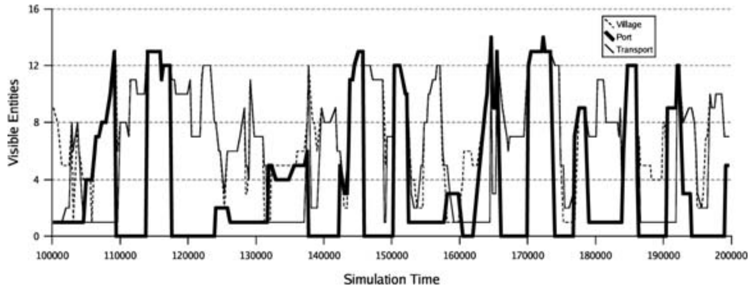


Fig. 5. Number of visible entities for different types of entities in our scenario, for communication radius of 80, near percolation threshold

as we can see on Figure 5. Path accessibility in the community may be described by relation (see second row of Table 1).

In the last state, when communication radius is above 120 and link accessibility reaches 0.4, the entities become almost completely connected. This state of the community is described by relation (see third row of Table 1). System properties does not change when we further increase the radius and link accessibility.

The dynamic nature of our network near percolation threshold is clearly visible on the following graph (Figure 5), where we present the number of locations visible from one randomly chosen entity of each location type over a period of time. As we are near the percolation threshold, in the state described by second row of Table 1, we can observe the appearance of relatively large, but short lived, accessible components.

In the Figure 5, we may also note that the transport is accessible from significantly more locations than static elements. As this holds for all transports in

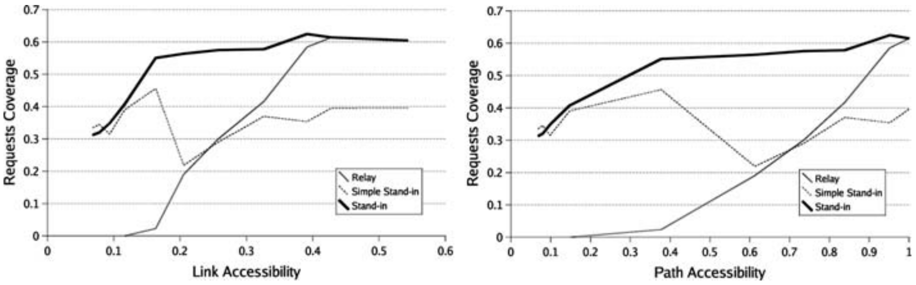


Fig. 6. The average requests coverage of three presented inaccessibility solutions and different accessibility settings

our scenario, a parallel with scale-free networks [16] arises. In these networks, a small number of nodes called hubs has significantly more connections with others than the rest, while in the random networks most nodes have the same number of adjacent edges. In this respect, transport platforms with stand-ins on them serve as hubs of our system, spreading the information as they roam through the map. In our future experiments, we will examine this possibility and test the hypothesis on a larger agent community.

4.3 Comparing the Solutions

After having determined the extent of inaccessibility in our system, we will study the effects inaccessibility has on the system performance. The system performance is given by a number of goods successfully delivered to villages. Zero value means complete failure, when no goods were transported, while 1 implies that all orders were completed.

On the following graph (see Figure 6), we can observe the relationship between path accessibility and overall system performance for each of three solutions. Here we present the average requests coverage for different solution of inaccessibility. Results do follow the accessibility state partitioning from the previous paragraph. We can see that relay agents start to be reasonably useful when the link accessibility reaches 0.2, in the middle of the transition phase, well corresponding to the percolation threshold. Performance of isolated, non communicating stand-in remains constant. This is easy to understand, as these agents communicate only locally. They present an optimal solution for disconnected networks, as they require only a small number of messages to function.

On the other hand, performance of interacting community of stand-ins is more than a mere supremum of both previous methods. This is allowed by the dynamic nature of the system, where the stand-ins on mobile entities carry the up-to-date information through the system and spread it in small local communities, but relatively often. Thanks to this approach, the efficiency of system with these stand-ins approaches the optimum level with path accessibility of 0.4, instead of 0.9 for relay agents.

5 Conclusions and Future Work

In our experiments, we have proved that the theory describing the behavior of random graphs can serve as a basis for formalization and measure of the inaccessibility within multi-agent systems. In the future, we will extend our experiments to verify the hypothesis that the scale-free approach can be used to precise description of our system around percolation threshold. Moreover, we have provided several solutions, including new concept of stand-in agent, for inaccessibility and experimentally determined their boundaries of applicability.

As we have illustrated above, stand-in agents provide more than a viable alternative to message relaying in environments with low link accessibility or high cost of communication. They allow efficient coordination and collaboration in communities with low and transient accessibility and they match the performance of relaying in connected communities. However, the implementation of the stand-in agents for a given domain is not trivial and its use in larger communities of agents requires some additional tuning of two principal methods they use – swarming of the stand-in agents and knowledge distribution/synchronization. Currently implemented version of stand-ins is appropriate for environments with low and moderate accessibility, due to the fact the number of messages used for knowledge updates grows rapidly with increasing accessibility and the size of the domain. To extend its operational use for environments with the accessibility beyond the "transition phase", stand-ins shall be aware of the typical information flows in their neighborhood and better target their information updates, as mentioned in section 3.5.

Given the plummeting prices of hardware and many emerging low cost platforms designed specifically to be embedded with the environment to provide the measurements [17], we will often face the situations when the communication ability will be a limiting factor of such systems, due to the limited battery power and constraints on their emitters. In such cases, the sole cost of communication would prohibit the use of advanced negotiation or auctioning techniques between the agents residing on different nodes of the system. Stand-ins, created by all interested agents and deployed on an agreed node that provides sufficient computational resources and where the negotiation takes place, can be a solution to this problem too.

Acknowledgement

The research described in this chapter has been supported in parts by the EOARD/AFRL projects number FA8655-04-1-3044, FA-8655-02-M-4057 and ONR grant N00014-03-1-0292.

References

1. Woo, A., Tong, T., Culler, D.: Taming the underlying challenges of reliable multihop routing in sensor networks. In: Proceedings of the first international conference on Embedded networked sensor systems, ACM Press (2003) 14–27

2. Pěchouček, M., Dobíšek, M., Lažanský, J., Mařík, V.: Inaccessibility in multi-agent systems. In: Proceedings of International Conference on Intelligent Agent Technology. (2003) 182–188
3. Bollobas, B.: Random Graphs. 2nd edn. Cambridge University Press (2001)
4. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74** (2002) 47–97
5. Grimmett, G.: Percolation. Springer-Verlag, New York (1989)
6. A-Globe: A-Globe Agent Platform. `http://agents.felk.cvut.cz/aglobe` (2004)
7. Elias Leontiadis, Vassilios V. Dimakopoulos, E.P.: Cache updates in a peer-to-peer network of mobile agents. In: Proceedings of the Fourth IEEE International Conference on Peer-to-Peer Computing, IEEE Computer Society (2004) 10–17
8. Stallings, W.: Data and computer communications (5th ed.). Prentice-Hall, Inc. (1997)
9. Sycara, K.: Multi-agent infrastructure, agent discovery , middle agents for web services and interoperation. In: Mutli-agents systems and applications, Springer-Verlag New York, Inc. (2001) 17–49
10. Pěchouček, M., Mařík, V., Štěpánková, O.: Role of acquaintance models in agent-based production planning systems. In Klusch, M., Kerschberg, L., eds.: Cooperative Infromation Agents IV - LNAI No. 1860, Heidelberg, Springer Verlag (2000) 179–190
11. Cao, W., Bian, C.G., Hartvigsen, G.: Achieving efficient cooperation in a multi-agent system: The twin-base modeling. In Kandzia, P., Klusch, M., eds.: Cooperative Information Agents. Number 1202 in LNAI, Springer-Verlag, Heidelberg (1997) 210–221
12. Thomas, V., Bourjot, C., Chevrier, V., Desor, D.: Hamelin : A model for collective adaptation based on internal stimuli. In Schaal, S., Ijspeert, A., Billard, A., Vijayakumar, S., Hallam, J., Meyer, J.A., eds.: From animal to animats 8 - Eighth International Conference on the Simulation of Adaptive Behaviour 2004 - SAB'04, Los Angeles, USA. (2004) 425–434
13. Simonin, O., Ferber, J.: Modeling self satisfaction and altruism to handle action selection and reactive cooperation. In: Proceedings Supplement SAB 2000, The Sixth International Conference on the Simulation of Adaptative Behavior, From Animals to Animats 6, Paris, France (2000) 314–323
14. Bedau, M.A., Packard, N.H.: Evolution of evolvability via adaptation of mutation rates. Biosystems **69** (2003) 143–162
15. Sierhuis, M., Bradshaw, J., Acquisiti, A., van Hoof, R., R., J., Uszok, A.: Human-agent teamworks and adjustable autonomy in practice. In: Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space: i-SAIRAS - NARA, Japan (2003)
16. Barabi, A.L., Albert, R.: Emergence of scaling in random networks. Science **286** (1999) 509–512
17. Estrin, D., Culler, D., Pister, K., Sukhatme, G.: Connecting the physical world with pervasive networks. IEEE Pervasive Computing **1** (2002) 59–69