Self-Organizing and Adaptive Peer-to-Peer Network

Robert A. Ghanea-Hercock, Fang Wang, and Yaoru Sun

***Abstract*—In this paper, an algorithm that forms a dynamic and self-organizing network is demonstrated. The hypothesis of this work is that in order to achieve a resilient and adaptive peer-to-peer (P2P) network, each network node must proactively maintain a minimum number of edges. Specifically, low-level communication protocols are not sufficient by themselves to achieve high-service availability, especially in the case of *ad hoc* or dynamic networks with a high degree of node addition and deletion. The concept has been evaluated within a P2P agent application in which each agent has a goal to maintain a preferred number of connections to a number of service providing agents. Using this algorithm, the agents update a weight value associated with each connection, based on the perceived utility of the connection to the corresponding agent. This utility function can be a combination of several node or edge parameters, such as degree k of the target node, or frequency of the message response from the node. This weight is updated using a set of Hebbian-style learning rules, such that the network as a whole exhibits adaptive self-organizing behavior. The principal result is the finding that by limiting the connection neighborhood within the overlay topology, the resulting P2P network can be made highly resilient to targeted attacks on high-degree nodes, while maintaining search efficiency.**

***Index Terms*—Agents, peer-to-peer networks, self-organizing networks.**

I. Introduction

THE PROBLEM addressed in this paper is how to engineer a self-organized routing capacity in overlay networks and multiagent systems (MASs), specifically for networks with a high rate of node entry and exit. The goal is that the network as a whole should exhibit the ability to learn improved routes and that the adaptation mechanism is fully distributed and locally managed by each node. Ideally, we also require the resulting overlay network to be highly adaptive and resilient to both random failure and targeted attacks on key nodes. The proposed algorithm is titled, Synaptic Connection Adaptive Network (SCAN) protocol. We first review what potential mechanisms can be applied to achieve this goal. It is argued that for computational agents, the core behavior should be the maintenance of valuable social connections to other agents. This raises the question of how an agent should assign value to a social connection and how it can dynamically update the value of a connection.

Manuscript received March 15, 2005; revised January 13, 2006. This work was supported by British Telecommunications plc. This paper was recommended by Associate Editors U. Kaymak and P. Yi.

R. A. Ghanea-Hercock and F. Wang are with the British Telecommunications Research Laboratories, Ipswich IP53RE, U.K. (e-mail: robert.ghanea-hercock@bt.com; fang.wang@bt.com).

Y. Sun is with the Behaviour and Brain Science Centre, School of Psychology, University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K. (e-mail: y.sun.1@bham.ac.uk).

Digital Object Identifier 10.1109/TSMCB.2006.873211

The question being addressed is stated as follows: What are the requirements for software agents to achieve a self-organized social network? This is a useful question as the value of MAS is only fully realized in a large community of interacting agents. Even where MAS platforms provide a transport infrastructure and directory service [14], this is not a sufficient mechanism for an agent to initiate or maintain its social interactions.

A. Biological Neurons

The inspiration for the SCAN protocol stems from individual biological neurons that are capable of highly adaptive behavior [11]. In this respect, we make the implicit assumption that biological neurons are autonomous agents in their own right. Of key significance is that each neuron is actively seeking useful connections to neighboring neurons [10]. This is the principle that we are attempting to map into the MAS domain. The proposal is that a set of agents within a peer-to-peer (P2P) network architecture can achieve an efficient and adaptive communication strategy by actively seeking a minimum number of connections to other agents. The growth of a dynamic overlay network is therefore best achieved by each node running a proactive set of behaviors combined with a neural learning mechanism to reinforce high utility connections. In the experimental results, this concept is tested by assigning a local agent to each peer node. The agent implements the active behavior set and manages the edge update rules using a Hebbian reinforcement algorithm.

B. Objectives

The objective of this work is to develop adaptive strategies that enable MAS to maintain communication channels to a service or resource, and to do so under dynamic and adverse network conditions. We propose that this is a key requirement in the creation of a self-organizing agent network. The remainder of the paper presents a set of results from a simulated P2P search application that aims to demonstrate the specified abilities for resilient and adaptive agent behavior. Section II presents a short review of P2P networks and related adaptive network methods. Section III briefly covers some essential concepts from network graph theory. Section IV presents the proposed connection algorithm. Section V contains our preliminary results and analysis. Finally, our conclusion and discussion are offered in Section VI.

II. Agents and P2P Networks

The emergence of P2P networks reflects the increasing quantity and value of resources at the edge of networks. The association established between peers creates a logical connection of peers and forms a peer overlay network on top of the underlying network architecture, such as the transmission control protocol/Internet protocol (TCP/IP) layer. As P2P networks are dynamic complex systems involving distributed and continuously changing peers and resources, constructing appropriate peer connections to support efficient resource sharing is a significant challenge. In particular, if the number of edges is too small, all nodes may not be reachable in acceptable time. However, if the number of edges is too large, then any flooding or broadcast algorithm will generate a heavy bandwidth load from excessive query messages [23].

Current P2P systems can be classified into two types, namely, 1) structured and 2) unstructured, according to the methods used to organize peer connections [15]. In structured P2P systems, peers are assigned static identifiers. Routing tables built on identity distances are distributed onto some if not all of the peers. Typical examples of structured P2P systems include P-Grid [1], Freenet [5], Pastry [19], Chord [24], and Tapestry [27]. These systems usually have predefined network topologies and resource placement schemes. Though certain heuristic information is used in Freenet for resource allocation and P-Grid for peer path maintenance, the underlying network structures of these two systems are determined. In structured P2P systems peers are well organized and resource search is relatively straightforward, but substantial knowledge and experience are required for system design, and this is always at a cost of increased maintenance to deal with changes caused by peers/resources joining and leaving. Hence, structured P2P systems target for high resource availability in a relatively persistent environment. In order to do so, they usually put extra and sometimes strict requirements (e.g., fixed data placement) on participating peers. This is sometimes difficult to achieve especially when peers come from heterogeneous and dynamic organizations or locations.

In contrast to structured systems, unstructured P2P systems usually have no global control or layout of the whole system. Peers in unstructured P2P systems manage their own associations during run time. The formation of the overlay network is therefore decentralized and self-organizing. In Gnutella [7], for example, peers are self-organized into an overlay network, which often exhibit a power-law degree distribution [18]. In Anthill [4], peers construct or remove connections to other relevant peers (called neighbors) according to instantaneous information discovered by a special kind of ant agent during a search for resources. Because unstructured P2P systems organize peers adaptively and autonomously, less prior knowledge is required, and the resulting networks can be more resilient to dynamic change. Unstructured P2P systems target for flexibility but cannot always guarantee resource availability. The search of resources based on these systems can be very inefficient. For example, the search process in Gnutella is a type of broadcasting, which often causes message flooding over the network. This inevitably restricts search flexibility. Yang and Garcia-Molina [25] improved search speed on Gnutella by dynamically controlling search steps or utilizing neighbor information in route selection. A clear advantage of P2P networks is their resilience and ability to withstand both directed attacks and highly dynamic connectivity of the overlay network [27]. The aim of this paper is to determine how to utilize such a resilient ability in support of a distributed MAS.

A significant body of work is also emerging within the MAS community that studies the merger of agents and P2P network architectures. One example is the process of group formation among P2P agents [16]. Here, they consider the decentralized formation of groups within a P2P MAS. A second thread in this field is looking at incentive mechanisms for agent-based P2P systems [26]. In this case, each peer is a software agent, and the agents cooperate to search the whole system through referrals. The work also considers the important issue of potential free-riding by looking at the behavior of the agents under two pricing mechanisms. Finally, the favorite MAS application of e-commerce has also been tested in a P2P environment [17].

This paper proposes a new approach to form weighted peer networks based on biological neuron processes. In the biological case, the connections between neurons are adjusted in real time through neural synapses. Network peers and biological neurons share the same motivation, both of which are goal driven to seek a set of social connections for task achievement. Following simple Hebbian learning rules [9], we propose that computing peers in unstructured P2P networks can achieve adaptive and efficient resource sharing by exploiting peer associations and self-organizing synaptic-type processes.

III. Related Work

The field of network theory and network dynamics has had a major impact on many disciplines. However, in most cases, the applied theory has utilized work from fixed-weight graph networks. A few papers have highlighted the point that real networks are generally more complicated with variable and dynamic weights associated with the edges between nodes [12]. This point is of particular importance if we consider the role of synaptic weights in biological neural networks. Weighted edges permit a network to exhibit and encode learned knowledge and adaptive states [11].

Previous work [21] demonstrated a mobile-agent-based method (termed SpiderLink) for adaptive route discovery in dynamic *ad hoc* networks. SpiderLink deals with the problem of supporting reliable information transfer in decentralized networks of undifferentiated mobile communication devices (i.e., *ad hoc* networks). With the corresponding architecture being both dynamic and nonhierarchical, routing should ideally depend neither on static nor on centrally updated tables. SpiderLink is designed to carry out effective message delivery in a constraining environment. However, the SpiderLink protocol, unlike the current model, assumes only binary connection strengths between nodes.

In this paper, we enable each agent to assign a floating point weight value to all of the edges to other agents or systems. In this model, an edge is defined as any channel or interaction mechanism that exists between two or more agents. Of course, the reason why weighted graphs have received less theoretical study is precisely because of the fact that they can exhibit complex dynamic behavior, including resonance, oscillation, and chaos. This factor must be considered when attempting to utilize a weighted network mechanism as advocated in this paper.

A. *Learning in Adaptive Networks*

A related area of research has proposed the use of recurrent neural networks (RNNs) within each node to perform smart packet routing [6]. In this case, the goal has been to adapt routes based on user-defined quality of service (QoS) metrics for packet loss or delay [28]. This is implemented through the use of a smart packet route discovery protocol and a recurrent random neural network for each outgoing link from a router. In contrast, SCAN associates a separate weight with each edge in the overlay network, and the nodes themselves act as the integration function of a distributed neural array. This is discussed in detail in the following section.

IV. SCAN Algorithm

Using a dynamically updated edge weight between nodes offers a solution to the problem of managing the search efficiency and growth of an *ad hoc* or overlay network. There are several aims behind this idea. First, we would like to reduce the level of messaging traffic in the overlay network as this can be a significant overhead especially in unstructured P2P systems [18], [23]. Secondly, the aim is to enable the network to learn useful routes to resources. Basically, edges to nodes with low utility resources are pruned over time; hence, each node only maintains edges to nodes that consistently return useful results. The resulting overlay network is continuously adjusted by following a set of extended Hebbian rules. In the following discussion, every node in the network is implicitly associated with a single neuron that resides on that node. Specifically, the learning process includes the following:

1) creation of new connections [growth phase];
2) elimination of existing connections [pruning phase];
3) changes in the connection strengths [adaptation phase].

Suppose w_{ij} is the strength of an association, or virtual connection, from peer i to peer j. It will be adjusted by the following rules: frequency rule; feedback rule; decay rule; new peer rule; and dynamic growth rule.

1) Frequency Rule: This rule is deployed when a neuron (associated with each node) is fired in response to the firing of another neuron. In SCAN, this happens when a peer j provides required resources to a requesting peer i. The strength of the connection from peer i to j is then updated according to (1) and (2), which are given as

$$w_{ij} = w_{ij} + \Delta w_{ij} \tag{1}$$

where

$$\Delta w_{ij} = \gamma \cdot x_i \cdot x_j \tag{2}$$

where γ is the learning rate and x_i is the output of peer i, which sends a request for particular resources. It has a value of 1 from time t when the request is sent to time $t + T$ and 0 otherwise. x_j is the output of peer j replying to the request sent. If peer j provides search results within a time period T, its output is 1 from time t' when peer j sends the results to peer i to time $t' + T$. Otherwise, x_j has a value of 0.

The learning rate is defined as

$$\gamma = \begin{cases} q \cdot \left(1 - \frac{\tau}{T+1}\right) \cdot \sigma & \tau \leq T \\ 0 & \tau > T \end{cases} \tag{3}$$

where τ is the time spent searching for a resource, $q \in [-1, 1]$ indicates the value or quality of the resources provided, which is examined by the receiving peer i, T is a constant indicating the maximum time period allowed for the search, and σ is a learning parameter.

During a search, if peer j cannot find a suitable result regarding a request, it will forward the request to one or more peers connected to it. If no solution is found in time period T or a maximum number of forwarding steps F, the search is terminated. If desired, the failure information can be sent back to the original requesting peer, but this is not essential. If a failure message is sent to the original requesting peer i, peer i may do nothing (i.e., no weight update) or update the connection strengths via feedback, by assigning a negative value (e.g., -1) to q and a zero value to Δw_{ij}. If a solution is found within a finite time, the connection from the original requesting peer i to peer k providing the resource will be strengthened in accordance with (1) and (2) (replacing j with k). A new connection will be constructed from peer i to k if they have no link before and the initial strength of the connection is given as

$$w_{ik} = \Delta w_{ik}. \tag{4}$$

A connection will be eliminated if its connection strength decays below a threshold value ε (typically $\varepsilon = 0.001$), as detailed by the connection removal rule.

2) Feedback Rule: When a resource search involves more than one peer, all the peers contributing to the search will update their connections involved after the original requesting peer evaluates the resources provided. The evaluation information works as feedback to reinforce the strengths of the connections involved. Equation (5) describes the feedback rule as

$$w_{kl} = w_{kl} + \eta_f \cdot \Delta w_{ij} \tag{5}$$

where η_f is a parameter to fine tune the feedback rule, k is any peer involved in the search except peer j, which provides the resources, and l is the peer to which the search request was forwarded by peer k. If the feedback rule implies adjustment of a weight for an association that does not currently exist, a new entry is added by following (4). In addition to the above adjustment rule, SCAN possesses a decay rule to simulate decaying connectivity over time.

3) Decay Rule: In the absence of any stimulus (i.e., lack of contributions during resource sharing), the strength of a connection from one peer to another will decay over time.

The decay rate can be simulated as the leakage current of a capacitance, which is expressed as

$$w_{ij}(s+1) = w_{ij}(s) \cdot \exp\left(-\frac{s}{\eta_d}\right) \tag{6}$$

where s is the time measured from the point at which the last update in accordance with (1), (5), or (6) occurred.

The decay rule can also be simplified as a linearly regressive function as shown in (7), which is defined as

$$w_{ij}(s+1) = w_{ij}(s) \cdot \eta_d \tag{7}$$

where η_d is a constant between [0,1]. A strength w_{ij} can be considered as 0 if w_{ij} is below the threshold value ε (e.g., 0.001). The connection from peer i to j will be removed from the system if its weight falls below the value ε.

4) New Peer Rule: This rule is applied when a new peer joins the network. In order to join the network, the peer should make connections to at least one existing peer. The new peer may use prespecified information on available peers to allow the connection to be bootstrapped by the system [7]. The new connections between the new peer i and an existing peer j are given a default value if the utility of these two peers are unknown to each other, that is,

$$w_{ij} = \eta_0 \text{ and } w_{ji} = \eta_0 \tag{8}$$

where η_0 is a predefined constant. The strengths are then updated via the other reinforcement rules through resource sharing.

5) Dynamic Growth Rule: The principal connection rule that is applied to increase the resilience of the resulting network is for each node to periodically build a new link to neighboring nodes, until it has acquired a specified maximum number of links ($k_{\max}$). Each link is made using a preferential attachment rule, i.e., based on the degree of connectivity of the target node. In addition, the utility of a new node can be defined according to a combined function of the target nodes' number of connections and resource level. This use of a multicriteria connection rule is similar to the process adopted in related work on self-organizing P2P networks [13]. In the case of a random node failure, the SCAN algorithm adapts and the network grows new connections in the vicinity of the damaged area. This is due to the nodes that lost links now having free channels to establish new connections and hence actively attempt to grow new links to available nodes.

The addition of this rule is a key process in enabling the resulting network to adapt to the frequent arrival and removal of nodes. However, there are two problems with the use of the dynamic growth rule that need to be addressed. 1) An excessive numbers of links can reduce the efficiency of the network and incur resource costs. 2) In the case of a targeted attack, the network can be heavily damaged if the growth rule uses any form of preferential attachment [2]. This is often the case if the network is trying to optimize some parameter, e.g., efficiency, or number of hops to a resource [22]. In the following section, we present a potential solution to this problem based on a network topology constraint model.

A. Growth Constraints

In order to address the problem associated with the brittle nature of a preferentially attached network and the destructive impact of a targeted attack, a constraint is added to the dynamic growth rule. This is based on a constraint using a neighborhood topology metric (e.g., space or social tags [8]). In this work, we use a two-dimensional (2-D) simulation model with a simple Cartesian distance constraint between nodes. Whatever constraint metric is used the effect is to limit the percentage of available nodes that any given node can connect to at each time step. The impact on the resulting P2P network topology, and its dynamic behavior under attack is of interest and is discussed in the following section.

V. Simulation Experiments and Results

This section describes the experimental system used to evaluate these concepts. The work is based on a P2P simulation built on the Repast multiagent simulation platform [http://repast.sourceforge.net/]. This provides a pure Java simulation environment with such features as run-time model manipulation via graphical user interface (GUI) widgets, logging, and a range of scheduling methods. It also facilitates replication of work by offering a common simulation platform. The model we constructed uses a 2-D bounded space with each agent represented by a single peer node in the space. The initial conditions are as listed as follows:

1) five peer nodes that are located randomly across the simulation space;
2) maximum number of edges per node $K_{\max} = 30$;
3) number of hops per message (TTL) $= 4$;
4) update rule based on the frequency of positive Ack messages per 50 simulated time units;
5) a uniform and random distribution of searchable resources across the nodes.

At each time step, each agent is scheduled to run the rules defined in Section IV. After a minimum of 5000 iterations, the resulting network is sampled to determine the topology or required statistical parameter; such as average path length or edge degree distribution. The constraint rule is implemented as a continuous parameter d in the range 0.0 : 1.0, which specifies the simulated range over which a peer node can initialize new connections.

A. Search Within Weighted Edge Graphs

The first set of experiments was aimed at testing the hypothesis that using weighted edges with simple Hebbian learning can increase the efficiency of search in dynamic networks.

The message efficiency e_f was defined as the percentage of positive responses received by a node as a function of the number of transmitted search requests. Fig. 1 illustrates the impact on e_f with fixed and adaptive weights. The use of adaptive weights increases the steady state value for e_f by a factor of 2.

B. Network Resilience

One of the key objectives of the SCAN algorithm is to increase the resilience and adaptability of an interagent P2P network. Using a constrained attachment version of SCAN, as defined in Section V, the resulting network proves far more resistant to a simulated targeted attack.

The scenario is a targeted attack against a network grown using the dynamic growth rule. (For a targeted attack, we

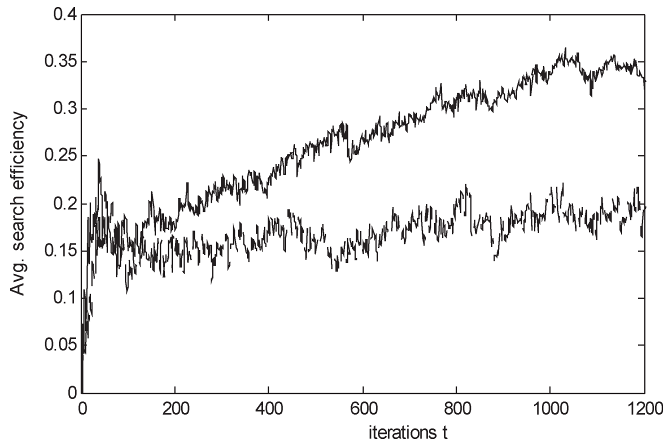


Fig. 1. Effect of using weighted edges (solid curve) on message efficiency compared to a network grown with fixed edge weights (all $w = 1.0$, dashed curve) (under targeted attack conditions with ∼500 nodes).

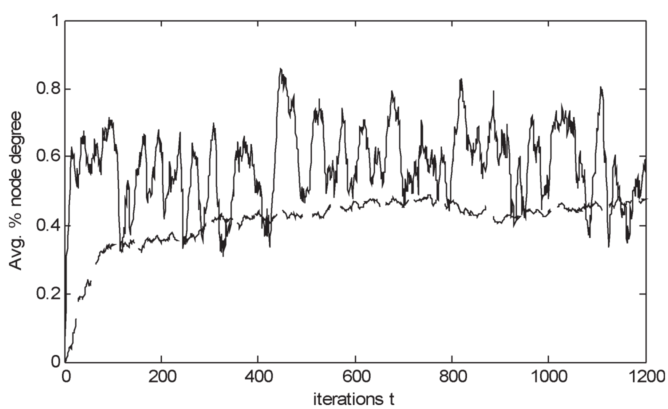


Fig. 2. SCAN average connectivity (A_c) with a targeted attack process in operation and the effect of a spatial constraint $d = 1.00$ (upper curve) and $d = 0.25$ (lower curve) of the maximum node separation ($K_{\max} = 30$, growth by preferential attachment, and average of 500 nodes).

assume that the attacker has global knowledge of the network and probabilistically selects the node with the highest edge degree at random intervals and deletes that node, i.e., a worst case scenario).

In Fig. 2, the effect of constraining the range d is shown, where each node can communicate from $d = 0.2$ to $d = 1.0$. In the case where each node is allowed to connect to any node in the space $(d = 1.0)$, the system rapidly achieves a high degree of connectivity. However, under a repeated targeted attack, the resulting connectivity collapses to oscillate chaotically between ∼30% and 80%. In this mode, the giant component of the network also becomes disconnected into a number of small subnets. In contrast with a constraint range ∼0.25, the average connectivity is stable at a value around 40% of $K_{\max}$. More importantly, the network giant component still contains $> 90\%$ of the peer nodes.

The most interesting effect of the attack process is shown in Fig. 3 where the impact on search efficiency is pronounced. Although the attack has reduced A_c for all values of d, the resulting network exhibits far greater stability and search efficiency when the constraint protocol is used. In effect, what this constraint rule produces is an emergent soft-celled overlay structure on the network. Hence, when a node with a

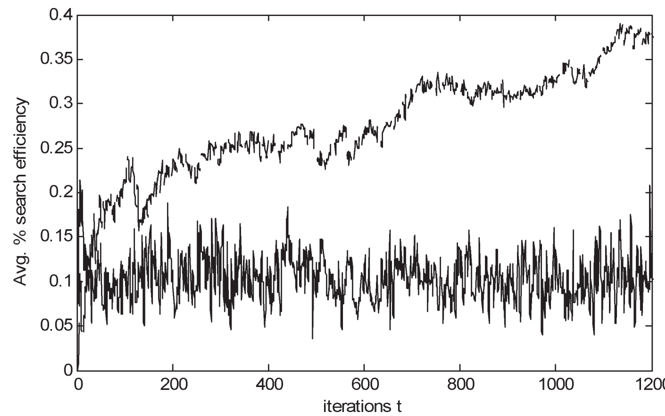


Fig. 3. Impact of the constraint rule on message search efficiency, with a targeted attack process in operation and a spatial constraint $d = 1.00$ (lower curve) and $d = 0.25$ (upper curve) of the maximum node spatial separation. ($K_{\max} = 30$, growth by preferential attachment, and average of 500 nodes).

large number of edges is deleted, the number of nodes that it affects is constrained to a smaller percentage of the total node population. Similar work on P2P *ad hoc* networks by Saffre *et al.* [22] has demonstrated that limiting the connection range of nodes in a dynamic P2P class network also has a minimal impact on the key properties of average path length and network diameter. Interestingly, they also find a threshold effect when the connectivity range is limited to approximately 20%. The implication for this work is that using any restricted virtual connection neighborhood (which could be tag based) to increase the resilience of the network can have minimal cost on the desired global network topology parameters and efficiency of search processes.

C. Graph Topology

We also investigated the topology of the resulting network under a range of attack and random deletion scenarios. Inasmuch as the algorithm is using a preferential attachment rule for local connections and a random long-range connection process, we would expect the node degree distribution to be approximately scale free with a power law distribution [2]. However, the degree distribution as shown in Fig. 4 appears to be closer to an exponential distribution. This reflects previous results where a network is grown from a large number of existing nodes rather than growth from a single seed node [20]. It is also in good agreement with work from studies of physical and real social networks that exhibit complex topologies, i.e., neither simply scale free nor exponential. In particular, such studies consider both the age and cost of vertices during growth [3]. Because the SCAN algorithm is using a weighted edge structure, then the results indicated in Fig. 4 conform to that expected for this class of networks.

It is useful to compare this result with the measured topology of the Gnutella network from 2001 [18]. Of interest is that the reported distribution for the Gnutella network, after it had undergone substantial evolutionary development and refinement of the protocol algorithms, resulted in a similar distribution to that in Fig. 4. In that paper, the author proposes that "… the more uniform connectivity distribution preserves the network capability to deal with random node failures while

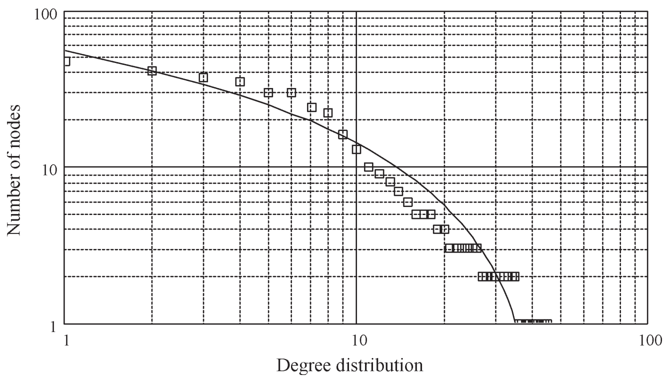


Fig. 4. Node degree distribution (log-log scale) resulting from the SCAN algorithm, with an exponential function fit (500 nodes, random attrition, and $K_{\max} = 30$).

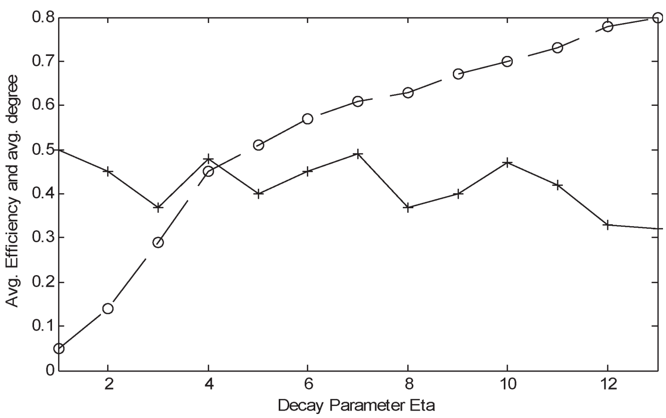


Fig. 5. Impact of edge decay parameter η_d on the average search efficiency (solid line) and average node degree (dashed line) (starting with 200 nodes in a random node removal scenario).

reducing the network dependence on highly connected, easy to single out nodes" [18].This indicates that the SCAN algorithm may offer a similar degree of resilient behavior in a deployed environment.

D. Impact of the Edge Decay and Reinforcement Parameters

The effect of the edge decay rate on the search efficiency and average node connectivity was also measured as shown in Fig. 5 (in which nodes are randomly selected for deletion.)

In Figs. 5 and 6, a higher value of η_d corresponds to a slower decay rate of the edges in the network. In Fig. 5, we see that the average degree of the nodes increases as the edge decay rate is reduced and saturates at around 80%, which equals 32 edges per node. The average search efficiency has an average figure of ~40% over the active range of η_d (i.e., for values of $\eta_d < 1.0$; for values of $\eta_d > 1.0$, the network connectivity has typically saturated.)

This result indicates that the dominant effect on the search efficiency is the rapid emergence of hubs with a high degree, rather than the adaptive behavior of the SCAN algorithm. However, for low values of η_d, the high decay rate creates a very dynamic network in which state the algorithm does help sustain the search efficiency.

In Fig. 6, the initial conditions and measured decay parameter are the same as in Fig. 5, but the nodes are removed using

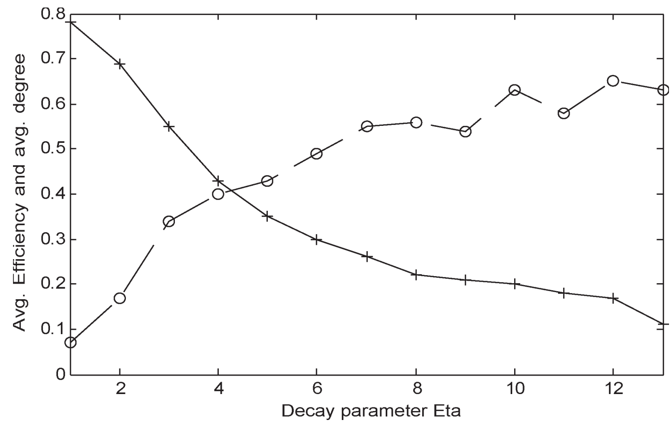


Fig. 6. Impact of edge decay parameter η_d on the average search efficiency (solid line) and average node degree (dashed line) (starting with 200 nodes in a targeted node attack scenario).

a preferential deletion rule that simulates a targeted attack. Hence, in this case, any highly connected hubs that form have a higher probability of deletion. This is reflected in the lower maximum edge degree that is achieved at ~60%. The resulting network has a flatter topology and is a more challenging search environment. In this case, the algorithm performs significantly better than the random node removal case up to a decay value of 0.4.

Further work is required to understand the interaction between the edge decay parameter η_d and the learning rate γ, which was held constant in the aforementioned experiments. The implication of these results is that the algorithm is most effective in pruning edges to nodes that offer poor search returns and hence in reducing the total network resources consumed, which is an important goal for any P2P broadcast network. However, more work is required to determine if the algorithm is effectively learning optimal search routes in the resulting overlay network.

VI. Conclusion

The SCAN algorithm is a novel approach to using weighted edges to provide an adaptive and self-organizing overlay network. The resulting network demonstrates improved search efficiency compared to current P2P messaging protocols and overlay network topologies. The use of a learning algorithm to adapt edge weights means that connections to nodes with irrelevant or poor resources are automatically removed, and edges to high-value nodes are reinforced. The set of adaptive behaviors is managed by an agent located at each peer node, and the resulting network is of particular value in the formation of adaptive communication channels for MASs. The novel contribution is in the use of a behavior-driven adaptation mechanism, i.e., the dynamic growth rule that proactively builds new connections.

Second, when adaptive edge weights are combined with the constrained dynamic growth rule, a cellular topology that preserves the short path length and network diameter achieved via a preferential attachment growth process emerges while providing a highly resilient response to targeted attacks.

Further work is required to determine the precise effects of the learning parameter and edge decay function on the SCAN algorithm. It was observed that variations in either parameter could have a significant impact on the search efficiency, depending on the state of the network at that point. At present, the algorithm is also susceptible to a distributed malicious peer attack that attempts to corrupt the learning rate applied to specific nodes and edges, and this requires investigation.

Work is also required to determine the relative merits of alternative learning algorithms that could be applied to the edge update process, e.g., genetic algorithms, genetic programming, or RNN. In particular, we need to determine how the convergence rate of the learning algorithm affects the stability of the resulting network under hostile attack conditions.

ACKNOWLEDGMENT

The authors would like to thank the BT Pervasive Computing Laboratory for the support in this research.

REFERENCES

[1] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, and M. Punceva, "P-Grid: A self-organizing structured P2P system," *ACM SIGMOD Rec.*, vol. 32, no. 3, pp. 29–33, Sep. 2003.

[2] R. Albert, K. Jeong, and A. Barabási, "Attack and error tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, Jul. 2000.

[3] L. Amaral, A. Scala, M. Barthelemy, and H. Stanley, "Classes of small-world networks," *Proc. Nat. Acad. Sci.*, vol. 97, no. 21, pp. 11149–11152, Oct. 10, 2000.

[4] O. Babaoglu, H. Meling, and A. Montresor, "Anthill: A framework for the development of agent-based peer-to-peer systems," in *Proc. 22nd Int. Conf. Distrib. Comput. Syst.*, Vienna, Austria, 2003, pp. 15–22.

[5] I. Clarke and O. Sandberg, "Freenet: A distributed anonymous information storage and retrieval system," in *Proc. ICSI Workshop Des. Issues Anonymity Unobservability*, Berkeley, CA, 2000, pp. 25–26.

[6] E. Gelenbe, R. Lent, and A. Nunez, "Self-aware networks and QoS," *Proc. IEEE*, vol. 92, no. 9, pp. 1478–1489, Sep. 2004.

[7] Gnutella. (2003). [Online] Available: http://www.gnutella.com

[8] D. Hales, "Self-organising, open and cooperative P2P societies—From tags to networks," presented at the 2nd Workshop Engineering Self-Organizing Applications, New York, 2004.

[9] D. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

[10] J. Kelso, *Dynamic Patterns, the Self-Organization of Brain and Behaviour*. Cambridge, MA: MIT Press, Apr. 1995.

[11] C. Koch and I. Segev, "The role of single neurons in information processing," *Nat. Neurosci.*, vol. 3, pp. 1171–1177, Nov. 1, 2000. Review.

[12] V. Latora and M. Marchiori, "Economic small-world behavior in weighted networks," *Eur. Phys. J., B*, vol. 32, no. 2, pp. 249–263, Mar. 2003.

[13] J. Ledlie, J. Taylor, L. Serban, and M. Seltzer, "Self-organization in peer-to-peer systems," in *Proc. EW SIGOPS Conf.*, Bordeaux, France, 2002, pp. 125–132.

[14] M. Luck, P. McBurney, and C. Preist, *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*. Southampton, U.K.: AgentLink, 2003.

[15] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proc. 16th Int. Conf. Supercomput.*, 2002, pp. 84–95.

[16] E. Ogston, B. Overeinder, M. van Steen, and F. Brazier, "Group formation among peer-to-peer agents: Learning group characteristics," in *Proc. 2nd Int. Workshop AP2PC*, Melbourne, Australia, 2003, pp. 59–70.

[17] M. Purvis, M. Nowostawski, S. Cranefield, and M. Oliveira, "Multi-agent interaction technology for peer-to-peer computing in electronic trading environments," in *Proc. 2nd Int. Workshop AP2PC*, Melbourne, Australia, 2003, pp. 150–161.

[18] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *IEEE Internet Comput.*, vol. 6, no. 1, pp. 50–57, Jan./Feb. 2002.

[19] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," in *Proc. 18th Conf. Distrib. Syst. Platforms*, Heidelberg, Germany, 2001, pp. 329–350.

[20] F. Saffre and R. Ghanea-Hercock, "Simple laws for complex networks," *BT Technol. J.*, vol. 21, no. 2, pp. 112–119, Apr. 2003.

[21] ——, "SpiderLink, survival of the dumbest in ad-hoc networks?" in *Proc. Eur. Conf. Comput.-Support. Coop. Work*, 2001.

[22] F. Saffre, H. Jovanovic, C. Hoile, and S. Nicolas, "Scale-free topology for pervasive networks," *BT Technol. J.*, vol. 22, no. 3, pp. 200–208, Jul. 2004.

[23] P. Silvey and L. Hurwitz, "Adaptive peer-to-peer topologies to improve system performance," in *Proc. Hawaii Int. Conf. Syst. Sci.*, Big Island, HI, Jan. 2004, p. 70199.

[24] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proc. SIGCOMM*, 2002, pp. 149–160.

[25] B. Yang and H. Garcia-Molina, "Improving search in peer-to-peer networks," in *Proc. 22nd Int. Conf. Distrib. Comput. Syst.*, 2002, pp. 5–14.

[26] B. Yu and M. Singh, "Incentive mechanisms for agent-based peer-to-peer systems," in *Proc. 2nd Int. Workshop Agents Peer-to-Peer Comput.*, Melbourne, Australia, 2003, pp. 77–88.

[27] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 41–53, Jan. 2004.

[28] E. Gelenbe and R. Lent, "Power-aware *ad hoc* cognitive packet networks," *Ad Hoc Netw. J.*, vol. 2, no. 3, pp. 205–216, Jul. 2004.

Robert A. Ghanea-Hercock received the First Class B.Sc. Physics and Electronic Engineering Joint Hon's Degree from Keele University, Keele, U.K., the M.Sc. degree in real-time electronic systems from the University of Bradford, Bradford, U.K., and the Ph.D. degree in robotic systems from the University of Salford, Salford, U.K.

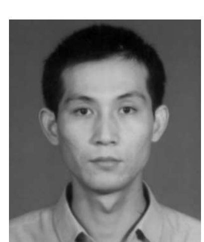
He is a Principal Research Engineer at British Telecommunications (BT) Research Laboratories, Ipswich, U.K. He is an experienced researcher and lecturer in a wide range of artificial intelligence (AI) systems, including software agents, security systems, and autonomous robots. He has several years of industrial experience as a Senior Researcher in intelligent software agents at BT. Prior to this position, he was a Lecturer at Salford University, teaching courses in AI, robotics, and knowledge-based systems. He has filed ten international patents in the field of software agents and security architectures. His publications include over 20 papers in refereed journals and international conferences. Recently, he was appointed as the first Business Research Fellow at the Santa Fe Institute, Santa Fe, NM, for his work on complex systems and is currently an Honorary Research Fellow at Imperial College, London, U.K.

Dr. Ghanea-Hercock is a Chartered Member of the British Computer Society.

Fang Wang received the M.Sc. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China, and the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K.

She is a Senior Research Scientist in the Pervasive ICT Center within British Telecommunications (BT) Research Laboratories, Ipswich, U.K. She joined BT in 2000 after previously working as an Assistant Professor in China. Her research interests include software agents, cognitive neuroscience, and distributed computing. She has published many papers in books, journals, and conferences and filed a number of patents.

Dr. Wang is a member of ACM. She has also received several technical awards.



Yaoru Sun received the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K.

He is a Research Fellow in the Behaviour and Brain Science Centre, School of Psychology, University of Birmingham, Edgbaston, Birmingham, U.K. He is an experienced researcher in machine vision and robotics, neural computing, and intelligent control.