MARIAN: A FRAMEWORK USING MOBILE AGENTS FOR ROUTING IN AD-HOC NETWORKS

Nikos Migas, William J. Buchanan, Kevin A. McArtney
Distributed Systems and Mobile Agents, School of Computing, Napier University
10 Colinton Road, Edinburgh, EH10 4EE, UK

ABSTRACT

The purpose of MARIAN is to investigate the degree of suitability for mobile agent technology in routing, topology discovery, and automatic network reconfiguration in ad-hoc wireless networks. This research work has a technical focus and proposes to assess different models of the usage of static and mobile agents to determine the best route through ad-hoc networks. The determination of the best route is not an easy process, and requires research into metrics that identify the best path, such as network performance, processing power, node's identification, battery power, and so on. These models will be appraised in terms of performance, reconfigurability, and easy of installation. In this paper we present a framework for MARIAN, which consists of three layers: foundation layer, intermediate layer, and core layer.

KEYWORDS

Mobile agents, stationary agents, wireless ad-hoc network, routing protocol, topology discovery, automatic network reconfiguration

1. INTRODUCTION

Wireless networks emerged in 1970s and rapidly became popular in the computing industry. This is especially true within the last decade, where wireless networks have been adapted to enable mobility (Royer, E. M. and Toh C. K., 1999).

Wireless networks are grouped into two categories: infrastructure networks; and infrastructure-less networks (Royer, E. M. and Toh C. K., 1999). Infrastructure networks have fixed and wired gateways, known as access points. A mobile device within these networks connects and communicates with the required access point that is within its communications radius. Infrastructure-less networks, often called ad-hoc networks, consist of mobile devices that have no central administration, and thus form a temporary network (Frodigh M. et al, 2000). In such an environment, it may be necessary for one mobile device to seek the aid of others in forwarding data packets to their destination, due to the limited propagation range of each mobile device's wireless transmissions (Hassanein, H. and Zhou, A.., 2001).

The mobile agent paradigm is an important area of research, and has been proposed as a promising solution for distributed computing over open and heterogeneous networks (Tianhan W. et al, 2002). A mobile agent is a software program that can suspend its execution on a host computer, and transfer itself to another agent-enabled host on the network. It then resumes its execution on the new host (Jansen, W. A., 2000).

Possibly the most promising application of mobile agents is in wireless ad-hoc networks. The asynchronous nature of mobile agents is ideal for ad-hoc environments where links are extremely unstable. Braun (Braun, P., 2003) proves through a series of experiments that mobile agents can reduce network traffic and maintain load balancing in a wireless ad-hoc network when compared to traditional client-server approaches.

This research proposes a novel routing protocol based on mobile agent technology called MARIAN. A mixture of static and mobile agents are responsible for conducting a series of performance tests on mobile nodes and on the network, and gather the results, and accordingly build an optimal routing table. Our framework supports both a static and mobile approach.

2. PROPOSED RESEARCH

We propose to assess different models of the usage of static and mobile agents to determine the best route through ad-hoc networks. The idea is based on the fact that on the top of each mobile node a static agent will run in the background, monitoring available resources such as connection availability, processing power, memory capacity, cost, and so on. In addition to static agents, mobile agents will independently roam the ad-hoc network gathering performance information from static agents. Mobile agents will then cooperate by peer-to-peer communication, and according to performance criteria of mobile nodes, they will build a best path routing table. The routing table will be broadcasted to all participating hosts within the ad-hoc network.

The routing protocol will also base decisions for best routes on QoS requirements and traffic type (Ho, Y. K. and Liu, R. S., 2002). As an example, a route may be stable enough to transmit real-time multimedia traffic, while another is more appropriate for message exchange due to frequent link failures. Mobile agents are also ideal for automatic network reconfiguration. Updates in the routing protocol can be attached to mobile agents. Mobile agents can then independently roam the ad-hoc network and perform the updates where necessary, in an efficient and effective way. The reader may refer to (Migas N. et al, 2003) for an extensive description of the research question, the aims of the proposed research, and possible benefits and outcomes.

The proposed framework may work well in parallel with alternative architectures such as the global computational Grid (Phan, T. et al., 2002). For instance, static agents monitoring mobile node's available resources may inform Grid to use a part of its computational power when the node is inactive.

3. OVERALL FRAMEWORK ARCHITECTURE

The architecture of the proposed framework can be logically divided into the following three layers: Foundation, Intermediate, and Core layer (Figure 1). The foundation layer is the physical layer and consists of all mobile nodes in an ad-hoc network. The intermediate layer sits on top of the foundation layer and is divided into two categories: the static agent model, and the mobile agent model. The top layer is the Core layer and is a combination of the static and mobile agent model.

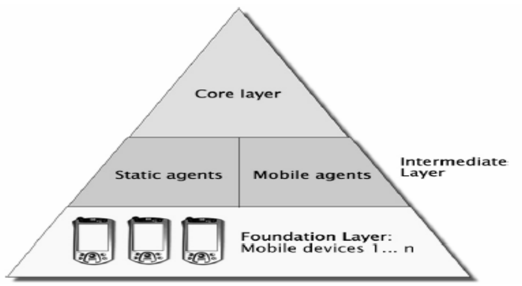

Figure 1. Layered Model

3.1 Foundation Layer

Each mobile node continuously runs a static agent (Nourani, C. F., 1998), which is responsible for two things: the calculation of mobile node's performance capabilities; and providing appropriate levels of security. Initially, the static agent conducts a series of tests and builds up a performance table. This table is maintained and regularly updated. Key factors for these tests are network performance, processing power, and node's identification. The static agent acts as an interface between routing agents and the local database, and thus provides security against tampering. Therefore, third party cannot access the local database directly since information is only available through the static agent.

In order to measure network performance it is required to conduct tests on node's connectivity with other nodes (Metz, C., 2003), network latency (Gummadi K. P., 2002), and available bandwidth. In the case of processing power, tests are required on memory capabilities, clock speed, battery life, and so on. In order to acquire identification measurements it is required to conduct tests on the CPU's identification and type, operating system and group identification. When the tests successfully completed, a database is then built and stored locally in the mobile node. This database information can then be passed to routing agents, who try to build an optimal path routing table for all possible destinations according to QoS (Quality of Service) requirements (Yen, C. H., 2002, Liu, J. et al, 2002) and traffic type (Ho, Y. K. and Liu, R. S., 2002).

Performance tests are scheduled to run repeatedly in a periodical or on-demand fashion. This is important due to the unstable nature of ad-hoc environments. A good example is links often die due to mobility and thus other optimal routes need to be discovered. However, tests should be designed in such that the mobile node's processing power is not wasted (Jones, C.E. et al, 2001). An alternative way is to monitor mobile node's behaviour and produce a set of patterns. Using these patterns, tests can become active when prediction shows that mobile node will be idle (Curran, K. and Parr, G. 2002, Su, W. et al, 2001).

Each mobile node will have Java Virtual Machine (JVM) for portable device (Sun Microsystems, 2003a) installed. This will allow an agent server to run on top, which will be responsible for the creation, management, and communication of static and mobile agents. This is depicted in Figure 2.

Since Java is a machine-independent language (Sun Microsystems, 2003b), there is no restriction on the type of portable device or on the underlying operating system. The Agent Server that will run on top of the Java Virtual Machine (JVM) has to be written in Java language and be optimal in order to consume a minimum of processing power. Most of the Agent Servers available today (Silva, A. et al, 2001) are written in Java where new versions are now being developed especially for portable device (IKV++ Technologies).

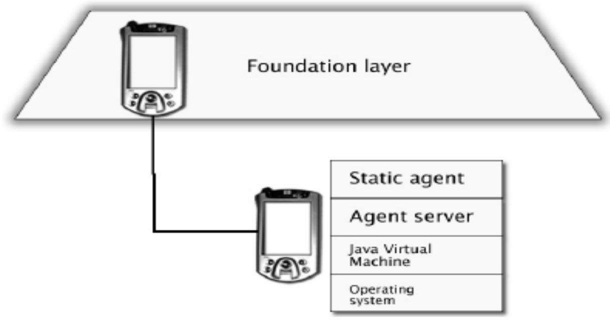

Figure 2. Required applications for a mobile node

3.2 Intermediate Layer

This layer is responsible for routing, topology discovery, and automatic network reconfiguration within a sub-network of a wireless ad-hoc environment. This process can be achieved with a static or mobile approach. According to the application, a static approach may have more advantages than a mobile approach and vice versa. Therefore, we use both models in our framework. Generally, mobile agent approaches show greater interest due to the facts that: calculations are decentralised and thus the load is distributed; and that they are fault tolerant compared to static approaches (Wu, C. et al, 2001). On the other hand, the process of routing becomes more difficult to design and implement as we move from static to mobile approaches.

Consider an initial state where a number of mobile devices are dispersed in a rather small geographical region such as a department of a university building. Suppose that users want to connect to each other and communicate (Figure 3a). In the figure, the solid squares denote the mobile nodes, and their position in the diagram denotes their geographical placement in the building. The first step of the routing protocol will be to classify nearby mobile nodes into clusters (Cheng, Y., 1992). These clusters represent the sub-networks of an ad-hoc network (Figure 3b). In this case, the protocol has grouped mobile nodes in three clusters: sub-network A, sub-network B, and sub-network C. Node i belongs to Sub-network A and B, as it is situated in the interception of A and B ($A \cap B$), node ii to $A \cap C$, node iii to $B \cap C$, and node iv to $A \cap B \cap C$. Thus, nodes i, ii, iii, and iv can be used to route traffic between different sub-networks and thus these are named gateway nodes in this framework.

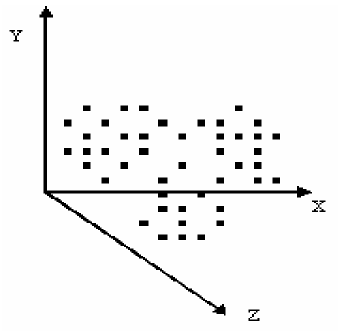

Figure 3a. Representation in 3 dimensions of mobile nodes

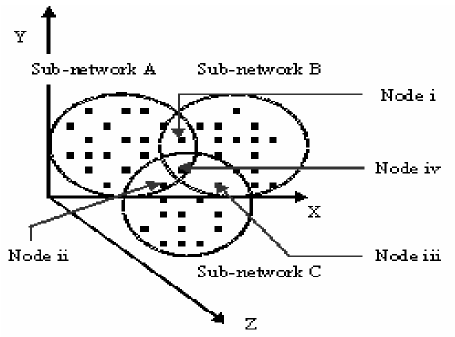

Figure 3b. Mobile nodes grouped in clusters according to geographical criteria

In this example, one gateway node is used for each pair of sub-networks. In real-life applications, more than one gateway nodes may be identified. For this an election process will take place (Zhang, J. et al., 2002). The most powerful node, such as in terms of processing power, or memory capacity, will then be used as a gateway and the others used as backup gateways. In this way, if the gateway node fails at some point due to mobility or due to some other reason, the backup gateway will take responsibility for routing. In the next phase, the protocol will determine domain nodes. Each sub-network will have a single domain node, which will be responsible for maintaining an optimal path routing table for that sub-network. As above, backup domain nodes will be used in case the domain node fails. An election process will choose the domain node, according to some criteria, such as available processing power.

In order to demonstrate the above we use the following example. Consider the case where mobile node α wants to transmit to mobile node δ (Figure 4a & 4b). There is a number of routes that node α can use to transmit data to node δ. For the sake of the example, we consider two of these routes: the solid line, and the dashed line. Suppose that the solid line has minimum latency and the widest bandwidth, where dashed line has maximum latency and the lowest bandwidth. A simple and rather inefficient technique is to count the hops from α to δ for each possible route and choose the one with the least hops (Perkins, C. E. and Royer, E. M., 1999). The proposed framework uses a more sophisticated technique that is based on the nature of traffic and performance characteristics of available routes. For example, if node α wants to transmit real-time multimedia traffic to node δ, then the solid line will be used. If node α wants to send e-mail data to node δ, then the dashed lines will be used because of the asynchronous nature of messaging.

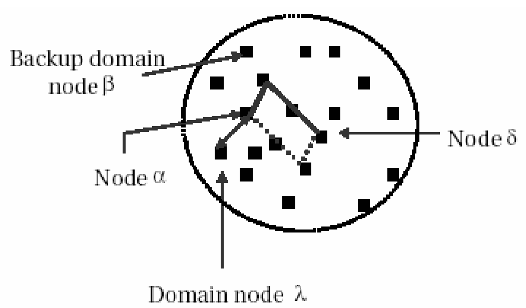

Figure 4a. Domain node routing agent using static approach

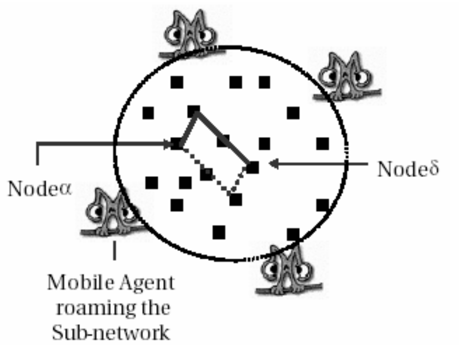

Figure 4b. Domain node routing agent using mobile approach

In case of the static agent approach, the domain node is responsible for periodically determining the optimal path routing table for the sub-network that it belongs to. An obvious drawback is that the domain node will have to perform complex computations in order to derive the optimal routing table, thus this approach could downgrade node's performance or even make it unavailable. In addition, this technique will not perform too well where nodes are continually moving. This is due to the fact that domain nodes would frequently change the sub-network they belong to, and therefore elections should take place very often in order to decide which node will become the new domain node.

The mobile agent approach maybe more appropriate for applications where mobile nodes move constantly in arbitrary directions. An example of such an application is the FleetNet project (Mauve, M. et al, 2001), which is associated with a mobile network, and more specifically in ad-hoc radio networks for inter-vehicle communication. In such an environment, the topology of the ad-hoc network will change shape and position, making the static agent approach inefficient.

3.3 Core Layer

The core layer is responsible for routing, topology discovery, and automatic network reconfiguration between sub-networks in the entire wireless ad-hoc network. A core agent gathers optimal routing tables maintained by domain agents (static approach) or mobile agents (mobile approach) and produces an overall routing table that can be used to route traffic from a node in sub-network x to a node in sub-network y. The following diagram is an example of five sub-networks forming an ad-hoc network and one core agent (Figure 5). As the network gets larger, the core agent gets larger as well.

Figure 5 could also represent a sub-network of a larger ad-hoc network. An ad-hoc network can consist of tens of such sub-networks with inner sub-networks of lower level (Figure 6). Thus we may have more than one core agents. As with gateway and domain agents, core agents have a secondary backup agent that maintains mirror storage with all routing information in case of primary failure. In this framework, we use a bottom-up technique, where we start from a mobile host and end up with the entire ad-hoc network consisting of a number of sub-networks with multiple levels. Core agents can communicate and exchange optimal path routing information.

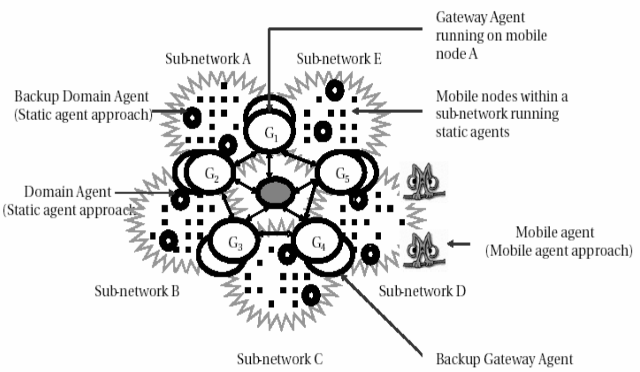

Figure 5. Core agents are used to route between sub-networks

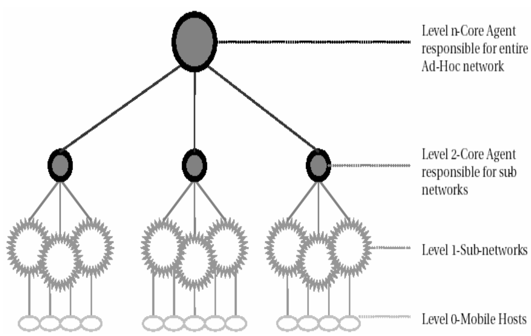

Figure 6. Multiple levels-core agent hierarchy

4. CONCLUSIONS

This research paper has provided an introduction on wireless ad-hoc networks and mobile agent paradigm. It has also presented the proposed framework for this research, which may also cooperate with alternative architectures such as the emerging Grid. The framework uses static and mobile agents for routing, topology discovery, and automatic network reconfiguration in ad-hoc networks. As we showed above, the architecture can be abstracted as a pyramid divided into three logical layers. The bottom layer represents the physical layer of the environment. The middle layer makes use of static and mobile agent approaches according to the application domain. This layer is responsible for routing, topology discovery, and automatic network reconfiguration within a sub-network. The top layer makes use of a mixture of static and mobile agents and is responsible for routing, topology discovery, and automatic network reconfiguration for the entire ad-hoc network. We have also demonstrated that the proposed scheme utilises a multilevel clustering of nodes making possible to apply the framework to a large ad-hoc network.

REFERENCES

Braun, P., 2003. The Migration Process for Mobile Agents, Implementation, Classification, and Optimization. *PhD thesis, Fiedrich Schiller University Jena*. http://swt.informatik.uni-jena.de/mit_braun.html.

Cheng, Y., 1992. Clustering with competing self-organizing maps. *In IJCNN International Joint Conference on Neural Networks*. Vol. 4, pp 785-790.

Curran, K. and Parr, G., 2002. A framework for the transmission of streaming media to mobile devices. *International Journal of Network Management*. Vol. 12, No. 1, pp 41-59.

Frodigh M. et al, 2000. Wireless ad hoc networking: The art of networking without a network. *Ericsson Review*, No. 4, pp 248-263.

Gummadi K. P., 2002. King: estimating latency between arbitrary internet end hosts. *Proceedings of the second ACM SIGCOMM Workshop on Internet measurement workshop*. Marseille, France, pp 5-18.

Hassanein, H. and Zhou, A., 2001. Routing with load balancing in wireless Ad hoc networks. *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, Rome, Italy, pp 89-96.

Ho, Y. K. and Liu, R. S., 2002. A novel routing protocol for supporting QoS for ad hoc mobile wireless networks. *Wireless Personal Communications*. Vol. 22, No. 3, pp 359-385.

Jansen, W. A., 2000. Countermeasures for mobile agent security. *In Computer Communications, Special Issue on Advances in Research and Application of Network Security*. Vol.23, No.17, pp.1667-1676.

Jones, C. E. et al, 2001. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*. Vol. 7, No. 4, pp. 343-358.

Liu, J. et al, 2002. A Unified Framework for Resource Discovery and QoS-Aware Provider Selection in Ad-hoc Networks. *ACM Mobile Computing and Communications Review*. Vol. 6, No. 1, pp 13-21.

Mauve, M. et al, 2001. A survey on position-based routing in mobile ad hoc networks. *IEEE Network*. Vol. 15, No. 6, pp 30-39.

Metz, C., 2003. The latest in virtual private networks: part I. *IEEE Internet Computing*. Vol. 7, No. 1, pp 87-91.

Migas N. et al, 2003. Mobile Agents for Routing, Topology Discovery, and Automatic Network Reconfiguration in Ad-Hoc Networks. *10th IEEE International Conference and Workshop on the Engineering of Computer Based Systems*. Huntsville, USA, pp 200-206.

Nourani, C. F., 1998. Software agents and intelligent object fusion. *Software Engineering Notes*. Vol. 23, No. 1, pp 98.

Perkins, C. E. and Royer, E. M., 1999. Ad-hoc on-demand distance vector routing. *Proceedings of the second IEEE Workshop on Mobile Computing Systems and Applications*. New Orleans, USA, pp 90-100.

Phan, T. et al., 2002. Challenge:: integrating mobile wireless devices into the computational grid. Proceedings of the eighth annual international conference on Mobile computing and networking. Atlanta, USA, pp 271-278.

Royer, E. M. and Toh C. K., 1999. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Communications Magazine*, Vol. 6, No. 2, pp 46-55.

Silva, A. et al, 2001. Towards a reference model for surveying mobile agent systems. *Autonomous Agents and Multi-Agent Systems*. Vol. 4, No. 3, pp 187-231.

Su, W. et al, 2001. Mobility prediction and routing in ad-hoc wireless networks. *International Journal of Network Management*. Vol. 11, No. 1 pp 3-30.

Tianhan W. et al, 2002. Integrity protection for code-on-demand mobile agents in e-commerce. *Journal of Systems and Software*. Vol. 60, No. 3, pp 211-221.

Wu, C. et al, 2001. Configurable mobile agent and its fault tolerance mechanism. *International Conference on Computer Networks and Mobile Computing*, Beijing, China, pp 380-389.

Yen, C. H., 2002. A geographically pinned QoS routing scheme for mobile ad hoc networks. *In proceedings of 10th IEEE International Conference on Networks (IEEE ICON'02)*. Piscataway, USA, pp 329-334.

Zhang, J. et al., 2002. Task-oriented self-organization of ad hoc sensor systems. In Proceedings of the first IEEE International Conference on Sensors. Orlando, USA.

WEB SITES: (Last visited: 17-03-2003)

IKV++ Technologies, 2003. Grasshopper 2 The Agent Platform. http://www.grasshopper.de/.

Sun Microsystems, 2003a. Java™ 2 Platform, Micro Edition. http://java.sun.com/j2me/j2me-ds.pdf.

Sun Microsystems, 2003b. The Java™ 2 Platform, Standard Edition. http://java.sun.com/j2se/.