CARPATHIAN J. MATH.
21 (2005), No. 1 - 2, 51 - 59

Modern routing techniques for future QoS enabled networks

SANDA DRAGOŞ and RADU DRAGOŞ

ABSTRACT. Modern technologies such as *mobile agents* are "intelligent" solutions that can be used to achieve efficient QoS routing in the Internet. We present in this article the key architectural features of this paradigm from the routing perspective. The evolution of a mobile agent population dispatched to find optimal routes is also analyzed. Our experimental results show that a high performance level could be achieved with the cost of generating too much control traffic. We present a solution that overcomes this scalability issue while still providing optimal results.

1. INTRODUCTION

Best-effort routing protocols, such as RIP and OSPF, are based on shortest path algorithms. Such strategies usually lead to an inefficient use of network resources as shortest paths become congested while other longer feasible paths remain under-utilized. Therefore, *Quality of Service routing* strategies have been developed to identify efficient paths that can satisfy given QoS constraints [6]. Computing feasible paths greatly depends on how the state information is collected, distributed and processed. Thus, authors of [6] divide such routing strategies into three main classes: *source routing*, *distributed routing* and *hierarchical routing*.

Source routing is the most popular method to implement QoS routing. It has the advantage of simplicity gained by transforming a distributed problem into a centralized one and therefore avoiding all problems of distributed systems. However such a routing strategy does not scale to large networks due to the large amount of state information that has to be dispatched to all network nodes.

QoS distributed routing protocols are difficult to implement because it is more complex to compute a QoS path in a distributed manner. This may involve a communication overhead which will make the system less scalable or produce inconsistencies which may lead to paths which contain loops [6]. Thus, the distributed routing strategies are more useful for best-effort routing.

The preferred solution for a scalable QoS routing are hierarchical routing strategies. Hierarchical routing scales very well due to aggregation mechanisms which allow that only partial global state information to be advertised outside a domain. Source routing algorithms are used at each hierarchical level to find feasible paths based on the aggregated state. Hence, hierarchical routing retains many of the advantages of source routing. Moreover, the routing computation is shared by many nodes, as in distributed routing. However, the aggregation of the network

Received: 04.02.2005; In revised form: 20.10.2005
2000 *Mathematics Subject Classification:* 68M12, 68M14.
Key words and phrases: *Hierarchical routing, Quality of Service (QoS) routing, mobile agents, MPLS.*

state is typically accompanied by an additive inaccuracy which has a significant negative impact on QoS routing [15].

Some of the researchers, including the authors, believe that modern techniques such as *mobile agents* are solutions for future routing by being able to create efficient distributed systems.

2. MOBILE AGENTS

The mobile agent technology originally emerged from advances made in distributed systems research [5]. All started by sending executable programs between clients and servers, followed by methods of remote dispatch of script programs or batch jobs. Mobile agents are considered to be an extension of such techniques [7].

A mobile (intelligent) agent is defined as [7, 14] a computational entity which acts on behalf of others in an autonomous fashion. They can behave *proactive* and / or *reactive*, while having attributes of learning, co-operation and mobility.

The *software environment* in which the agent exists is also known as *mobile agent environment* and it is defined as [14] a system distributed over a network of heterogeneous computers which has the primary task to provide an environment in which mobile agents can execute.

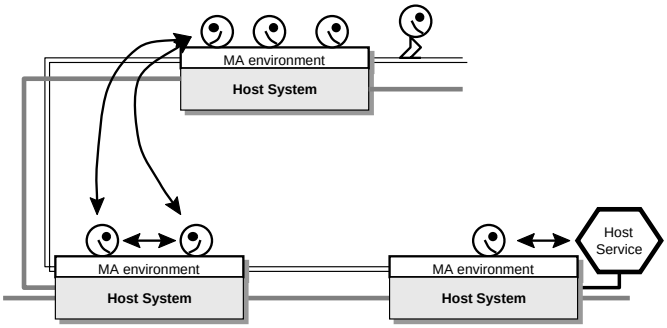


FIGURE 1. Basic mobile agent architecture

Fig. 1 explains very well the place and role of a mobile agent environment. Here, the smiling faces are mobile agents that travel between mobile agent environments, which are built on top of host systems. Communication between mobile agents (local and remote) is represented by bi-directional arrows. Communication can also take place between a mobile agent and a host service.

2.1. **Mobile Agents in Network Routing.**

A number of research groups studying alternative solutions for solving networking issues, including routing, have introduced new mechanisms based on the mobile agent paradigm [8, 9, 11, 16, 20].

Most of the mobile agent routing proposals were based on the *swarm intelligence* concept which was inspired by the behavior of biological swarms (e.g. ants, bees). This was considered a good model for totally decentralized, yet robust and

adaptive routing algorithms. The key element of this new concept is the *emergent behavior* or *emergent intelligence* where a complex and often intelligent behavior is gained through interactions of thousands of autonomous swarm members, with simple primitives. The main principle behind these interactions is called *stigmergy*, or communication through environment. This term was introduced by Grasse in [13] to describe indirect communication among individuals through modifications induced in their environment. For instance, real ants were able to find shortest paths using only the pheromone trail deposited by other ants.

In the routing models proposed to use swarm intelligence, the swarm members are mobile agents. Advantages gained by using such models are [4, 9, 16, 19]:

"Strong mobility" [2]: Mobile agents encapsulate code (intelligence) and data (states). Each agent runs independently of all others and by navigating the network they build solutions and modify the problem by using the collected information.

"Stigmergy" [9, 16, 20]: Mobile agents have the ability of indirect communication by leaving information on the nodes they have visited. Other agents visiting those nodes can access such information. It is argued that stigmergy is a robust and adaptive mechanism for information sharing. The term was first defined by Grass in [13] and describes how real ants find shortest paths by using the pheromone trail deposited by other ants.

"Emergent behavior" [16]: A complex behavior may be accomplished using simple interactions of autonomous agents, with simple primitives.

Robustness and fault tolerance [4, 14, 16, 18, 19]: Robust and fault tolerant distributed systems are easier to be built by using mobile agents as they have the ability to dynamically react to unfavorable situations. Therefore, the loss of network nodes or links will not affect the overall distributed system, leading to a graceful, scalable degradation by dispatching mobile agents from those specific nodes or links somewhere else or by terminating them.

Adaptability [14, 16, 18, 19]: Mobile agents react autonomously to chan- ges in their environment. They can change, die or reproduce according to the network changes. The population of mobile agents can increase / decrease according to the size of the network; they react to the loss of network nodes or links by terminating or dispatching themselves somewhere else in the network; they can extend their capabilities on-the-fly, by downloading required code off the network; they can change over time, new usage contexts and models can be accommodated. Moreover, propagation of changes in the network is sometimes faster than by using traditional methods (e.g. the Bellman-Ford algorithm) [3].

Efficiency [4, 7, 14, 20]: Mobile agents consume fewer network resources as they move the computation to the data rather than the data to the computation. Code is often smaller than data it processes. Therefore, transferring mobile agents to the source of data creates *less traffic* than transferring the data. Moreover, computing data in a distributed manner is less computational intensive than computing it all-together. *Space saving* is another efficiency issue where mobile agents have an advantage because a

mobile agent resides only on one node at a time. Moreover, mobile agents are able to duplicate and distribute themselves in the network, therefore each agent will execute on different machines in parallel. This means that mobile agents are also *time efficient*.

Modularity [16, 19]: Different mobile agents solve different parts of the problem obtaining intermediate results which are used to obtain the final, global solution.

Autonomy [4, 14, 16, 19]: Once released into the network, the mobile agent can operate asynchronously and independent of the sending program. They will autonomously decide which locations they will visit and what instructions they will perform.

Flexibility [7, 20]: Mobile agents can change over time, new usage contexts and models can be accommodated. Moreover, multiple network management strategies can coexist and co-evolve.

Distributivity [14]: Mobile agents are inherently distributed in nature and hence are offering a natural view of a distributed system. Moreover, a more realistic and up-to-date network image can be obtained by applying the computational operations to data directly where is physically located.

Parallelism [16]: Mobile agents are able to duplicate and dispatch themselves within the network and therefore agents will execute on different machines in parallel. For a large amount of data a single computer might not be able to process all data within a given time frame.

Extensibility: Mobile agents can extend their capabilities on-the-fly, by downloading required code off the network. They can carry a minimal functionality, which can be enhanced depending on future requirements [4].

AntNet [9] is one of the pioneering approaches for routing with mobile agents. It is inspired by the observation of ant colony behavior that cooperate to achieve the common goal of finding food [13]. The agents are modelled to emulate the ants pheromone technique for finding the shortest path. Here, individual agents leave traces of their presence in network nodes so that other agents can determine the probabilities of choosing specific paths to follow, until they all converge at a similar result (the shortest path). This work served as an inspiration for other research efforts [16, 17, 20]. Moreover, authors of [23] extend the ant-based routing by adopting the solution of multiple colonies of ants to search for optimal paths in order to mitigate the stagnation problem within ant colony optimization algorithms.

Mobile agent *cooperative schemes* were also introduced for routing purposes [17, 20, 26]. Authors of [20] proved that, within dynamic networks, a population of mobile, cooperating software agents are able to build and maintain routing tables which remain reasonably accurate even as the topology of the network changes over time.

Mathematical models for determining the behavior of mobile agents in network routing were also built. Some of them analyze the growing and jumping

Stagnation occurs when a network reaches its convergence (or equilibrium state), which may lead to congestion on the optimal paths that are already found and reduces dramatically the probability of selecting other optimal paths.

behavior of an agent based routing algorithm [24], while others are concerned with the population distribution and probability of success of mobile agents [21]. Their results provide guidelines for future designs of agent-based routing systems.

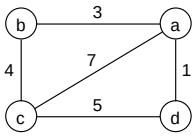
In [10] we proposed a new hierarchical routing protocol, called *Macro-routing* entirely based on mobile agents. Currently we are testing an extension of this protocol that allows finding feasible paths based on multiple constraints.

2.2. **Routing using WAVE.** The Wave technology [22] is based on parallel spreading of recursive program code (or *waves*) in computer networks, accompanied by dynamic creation of virtual Knowledge Networks (KNs). Such networks can persist and reflect any declarative or procedural information. Moreover, they may become active and capable of self-evolution, self-organization and self-recovery. Other *waves* can navigate, control and modify KNs. All these actions are performed without a central memory or a centralized control. Another important Wave feature is that routines such as synchronization, message passing and garbage collection are implemented within the Wave Interpreter which resides on physical nodes rather than being implemented within mobile agents as with other mobile agent technologies. This and its syntax make Wave code very compact, perhaps 20 to 50 times shorter that equivalent programs written in C, C++ or Java [25].

We used this system for routing purposes within Multiprotocol Label Switching (MPLS) networks. The use of Wave in MPLS networks for routing purposes has already been advocated in [12] to discover multi-point to point trees. Here, we are monitoring the evolution of a mobile agent population while finding paths between different nodes.

For our tests, we used a Solaris-Unix implementation of the Wave Interpreter, written in C, which is available for public download on the Internet [1]. We also used *Georgia Tech Internetwork Topology Models* (GT-ITM) (described in [27]) for generating random topologies.

Using such topologies, we created WAVE virtual networks by writing code similar to that in Fig. 2. We used various topologies (with different number of nodes and different numbers of links).



(a) physical topology

```
CR(@#a.Fa=A.7#b.\
     4#c.Fc=A.5#d.2#Fa.1#Fc)
```

(b) WAVE code for creating the virtual Knowledge Network

FIGURE 2. The Wave implementation for creating a four-nodes network

For those topologies we performed path searches using the algorithm from Fig. 3. Then we counted the number of waves which pass through each link of the network and we computed a mean value for each topology.

```
Falg=RP(OS(Fcost==0,A~Fborders).\
      A~Fpath.Fcost<Frequired.Fpath&A.Fcost+L.♯).\
Fborders=ARG1.Frequired=ARG2.Fmanaging=ARG3.\
@♯Fborders.Fcost=0.\
Falg.\
Fpath&A.Fcost+L.@♯Fmanaging.Npath&Fpath.Ncost&Fcost
```

FIGURE 3. The Wave implementation .

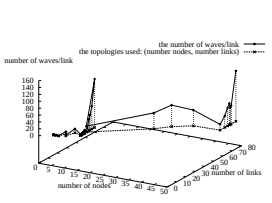
The results from Fig. 4(a) show how the mean value of the number of waves / link may vary depending on the number of nodes and / or the number of links within a network.

We could see from our experiment that the increase in the number of waves / link is mostly influenced by the degree of connectivity. As you can see in Fig. 4(a), for the same number of nodes, when the number of links increases, the number of waves / link increases as well.

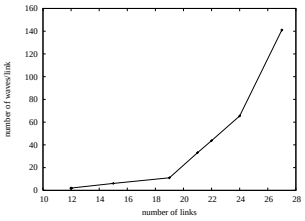
A mechanism is clearly required to limit the number of waves generated. We considered the effect of limiting the *lifespan* of waves, i.e. the maximum number of nodes through which they can propagate. This can be limited by adding a "time to live (TTL)" field as a supplementary constraint to each wave.

To investigate this refinement we considered a 12 node network where the number of links was varied from 11 (giving the most scarcely connected network) to 65 (giving a fully connected network). The GT-ITM package was used to generate each topology.

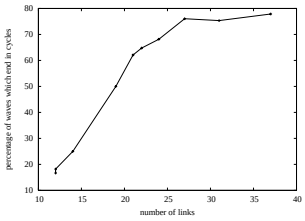
Since each wave has an unlimited lifespan, as in Fig. 4(b), the number of waves / link increases rapidly as the connectivity increases. However, Fig. 4(c) shows that the number of waves which end in a cycle (and which thus do not contribute to route discovery) is very high in such cases. By limiting the wave *lifespan*, we reduce the load induced by wave traffic on each link, as shown in Fig. 5(a) for a 19-link network. Fig. 5(b) shows that the proportion of waves ending in cycles is also reduced.



(a) Number of *waves* / link in different topologies having different numbers of nodes and different connectivity degrees



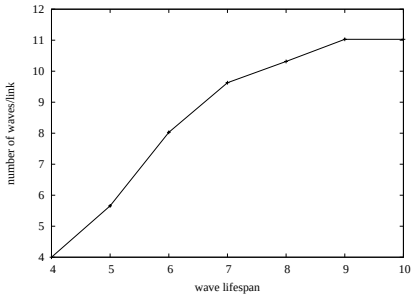
(b) Number of *waves* / link in a 12 node network and different number of links



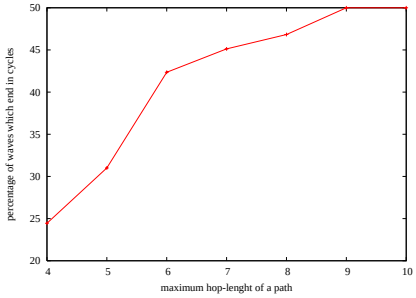
(c) The percentage of *waves* which end in cycles relative to the total number of *waves*

FIGURE 4. The unrestricted evolution of *waves*

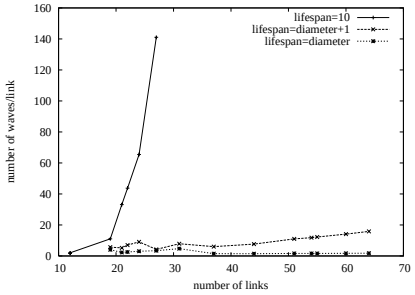
It remains to determine the optimal choice of lifespan, since too low a value will result in suboptimal routing (or, in the extreme, a failure to find any path to the destination) while too high a value is excessively burdensome for the network. Figs. 5(c) and 5(d) show the results for a variety of networks. The impact effectiveness of the lifespan mechanism in reducing the wave overhead is clear. However, the best choice of the value of this parameter requires further research. With such suitable choices, the algorithm will still find the optimal path in the majority of cases.



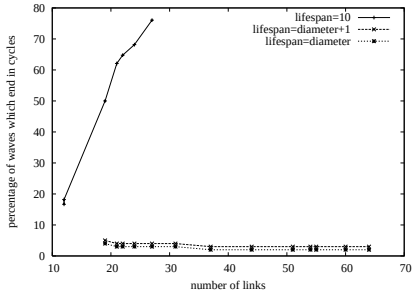
(a) Number of *waves* / link in a 12 node - 19 link network with increasing the path hop-length restriction from diameter to 10



(b) The percentage of *waves* which end in cycles relative to the total number of waves in a 12 node - 19 link network with increasing the path hop-length restriction from diameter to 10



(c) Number of *waves* / link in different network topologies with increasing the path hop-length restriction from diameter to 10



(d) The percentage of *waves* which end in cycles relative to the total number of *waves* in different network topologies with increasing the path hop-length restriction from diameter to 10

FIGURE 5. The evolution of *waves* under the *lifespan* restriction

3. CONCLUSIONS

There are many proposals for using mobile agents in QoS routing. We have presented some of them here. The reason they are not implemented and deployed yet is because there are still issues like security and authentication to be addressed.

Another important issue with mobile agents is that the search for a path is performed in a flood-based manner. This assures finding all available paths, even considering multiple QoS constraints. The cost of this level of performance is that a large number of mobile agents (implemented as *waves* in our approach) traverse the network. However, the level of wave traffic can be restricted without significantly impairing the algorithm performance by restricting the *lifespan* of the waves. Future work will investigate algorithms for the optimal choice of the lifespan parameter.

REFERENCES

[1] *The wave page.* http://www-zorn.ira.uka.de/wave/wave.html.

[2] J. BAUMANN, *Mobility in the Mobile-Agent-System Mole*, in 3rd CaberNet Plenary Workshop, January 1997.

[3] D. BERTSEKAS AND R. GALLAGER, *Data Networks*, Prentice Hall, 1992.

[4] A. BIESZCZAD, B. PAGUREK, AND T. WHITE, *Mobile Agents for Network Management*, IEEE Communications Surveys, 1 (1998).

[5] L. CARDELLI, *A Language with Distributed Scope*, in 22nd ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages, San Francisco, California, United States, 1995, pp. 286–297.

[6] S. CHEN AND K. NAHRSTEDT, *An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions*, IEEE Network Magazine, 12 (1998), pp. 64–79.

[7] D. CHESS, C. HARRISON, AND A. KERSHENBAUM, *Mobile Agents: Are They a Good Idea?*, Tech. Report RC 19887 (December 21, 1994 - Declassified March 16, 1995), IBM, Yorktown Heights, New York, 1994.

[8] M. COLLIER, *The Netlets Architecture: A Network Architecture for the Next Millennium*, tech. report, Dublin City University, Dublin, Ireland.

[9] G. DI CARO AND M. DORIGO, *Mobile Agents for Adaptive Routing*, in 31st Hawaii International Conference on System Science (HICSS), Big Island of Hawaii, January 1998.

[10] S. DRAGOS AND M. COLLIER, *Macro-routing: a new hierarchical routing protocol for MLPS networks*, in IEEE Global Telecommunications Conference (GLOBECOM'04), November-December 2004.

[11] S. GONZALEZ-VALENZUELA, *QoS-Routing for MPLS Networks through Mobile Processing*, master thesis, University of British Columbia, Faculty of Graduate Studies, Department of Electrical and Computer Engineering, January 2002.

[12] S. GONZALEZ-VALENZUELA AND V. LEUNG, *QoS Routing for MPLS Networks Employing Mobile Agents*, IEEE Network Magazine, 16 (2002), pp. 16–21.

[13] P. GRASS, *La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. La theorie de la stigmergie: Essai d'interpretation des termites constructeurs*, Insectes Sociaux, (1959), pp. 41–81.

[14] S. GREEN, L. HURST, B. NANGLE, P. CUNNINGHAM, F. SOMERS, AND R. EVANS, *Software Agents: A Review*, Tech. Report TCS-CS-1997-06, Trinity College, Dublin, Ireland, 1997.

[15] R. GUERIN AND A. ORDA, *QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms*, in IEEE Conference on Computer Communications, April 1997, pp. 75–83.

[16] I. KASSABALIDIS, M. EL-SHARKAWI, R. M. II, P. ARABSHAHI, AND A. GRAY, *Swarm Intelligence for Routing in Communication Networks*, in IEEE Global Telecommunications Conference (GLOBECOM'01), San Antonio, Texas, November 2001.

[17] K. H. KRAMER, N. MINAR, AND P. MAES, *Tutorial: Mobile Software Agents for Dynamic Routing*, Mobile Computing and Communications Review, 3 (1999), pp. 12–16.

[18] D. B. LANGE AND M. OSHIMA, *Seven Good Reasons for Mobile Agents*, Communications of the ACM, (1999), pp. 88–89.

[19] S. LIPPERTS AND B. KRELLER, *Mobile Agents in Telecommunications Networks - A Simulative Approach to Load Balancing*, in 5th International Conference on Information Systems, Analysis and Synthesis (ISAS'99), 1999, pp. 231–238.

[20] N. MINAR, K. H. KRAMER, AND P. MAES, *Cooperating Mobile Agents for Dynamic Network Routing*, Springer-Verlag, 1999, ch. 12.

[21] W. QU AND H. SHEN, *Some Analysis on Mobile-Agent Based Network Routing* , in International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'04), Hong Kong, SAR, China, May 2004, pp. 12–17.

[22] P. SAPATY, *Mobile Processing in distributed and Open Environments*, Wiley, 2000.

[23] K. M. SIM AND W. H. SUN, *Multiple Ant-Colony Optimization for Network Routing*, in 1st IEEE International Symposium on Cyber Worlds (CW'02), 2002.

[24] J. SUM, H. SHEN, C. SING LEUNG, AND G. YOUNG, *Analysis on a Mobile Agent-Based Algorithm for Network Routing and Management*, IEEE Transactions on Parallel and Distributed Systems, (2003), pp. 193–202.

[25] S. VUONG AND I. IVANOV, *Mobile Intelligent Agent Systems: WAVE vs. JAVA*, in 1st Annual Conference of Emerging Technologies and Applications in Communications, May 1996.

[26] J. S. WONG AND A. R. MIKLER, *Intelligent mobile agents in large distributed autonomous cooperative systems*, Journal of Systems and Software, 47 (1999), pp. 75–87.

[27] E. ZEGURA, K. CALVERT, AND S. BHATTACHARJEE, *How to model an internetwork*, in IEEE Infocom, vol. 2, 1996, pp. 594–602.

SWITCHING AND SYSTEMS LABORATORY
SCHOOL OF ELECTRONIC ENGINEERING
DUBLIN CITY UNIVERSITY
GLASNEVIN, DUBLIN 9, IRELAND
E-mail address: dragoss@eeng.dcu.ie
E-mail address: dragosr@eeng.dcu.ie