MTU Based Dynamic Routing Using Mobile Agents

Karthik Balasubramanian
bkarthik_au@yahoo.com

Karthik Subramanian
s_karthik_au@yahoo.com

Prathap Shanmugasundaram
prathapsundaram@yahoo.com

School of Computer Science and Engineering
Anna University, India.

Abstract

In large inter-networks like the Internet a packet travels through a number of networks with different MTUs. The difference between packet size and the MTU of a network results in reduced performance. When the packet size is greater than the MTU fragmentation occurs, resulting in increased error rate. On the other hand smaller packet size reduces the throughput of the network. We propose a routing algorithm that addresses the above problem by considering the MTU in routing. We use an agent model to reduce the additional overhead of communicating MTU information.

1. Introduction

Routing is an important aspect of network communication which affects the performance of any network, since other characteristics of the network like throughput, reliability and congestion depends directly on it. When a packet travels from a source to destination, it has to pass through a number of networks with varying characteristics. The small pieces into which the datagram is divided to traverse a network with a MTU smaller than its size are called fragments. The fragments are reassembled only at the destination. Fragmentation leads to inefficiency. Even if the physical networks encountered after the point of fragmentation have a larger MTU capability, only small fragments traverse them, thereby not efficiently using the available bandwidth. Even if a single fragment is lost or arrives late at the destination, the datagram cannot be reassembled and the entire datagram has to be discarded increasing the retransmission rates.

The efficiency of routing can be increased by taking into consideration the MTUs of the various networks. Using traditional message passing techniques for communicating MTU sizes incurs high overhead for large networks. An agent based routing strategy has been devised to reduce the overhead.

The paper is organized as follows Section 2 gives the rationale behind the use of agents. Sections 3, 4 and 5 outline the methodology and the related issues.

2. Why Agents?

Intelligent mobile agents are programs than migrate among the constituent nodes of the environment, take decisions based on the present situation and perform intelligent actions without the explicit control of the user.

Mobile agents have the following advantage over message passing techniques: (1) Agents can move easily across the network. (2) Agents are small in size. Hence the cost associated with hosting and transporting an agent, is minimal. (3) An agent is able to cooperate with other agents in order to perform complex or dynamic tasks. Agents may read from and write to a shared block of memory on each node, and can use this facility both to coordinate with other agents executing on that node and to leave information behind for subsequent visitors. (4) An agent is able to identify and use resources specific to any node on which it finds itself.

3. Routing

The routing algorithm used is similar to *Antnet* model given by Amain and Mikler [1]. There are two types of agents: *forward agents* which travel from source to the destination, gathering information along its path and *backward agents* which return to the source updating the state of the intermediate routers based on the information gathered. In any router we associate two counters with each entry in the routing table: A *Goodness Counter (GC)* which gives the goodness of the path between the source and the destination and a *Limit Counter (LC)* which is used to limit the agent population.

Packets of different sizes will arrive at the router for a particular destination. This may result in different optimal paths for different packet sizes. To accommodate this we modify the routing table to contain multiple entries for a single destination. Since the routing table size is limited we use the following algorithm to optimally store the different paths.

When a packet arrives at a destination the following steps are taken

1. Check the routing table for an entry which matches the packet's destination and its size
2. If no entry exists, go to 5
3. Send the packet to the router indicated by the router.
4. Check if the goodness counter is below the threshold. If so create a new forward agent and send it to the packet's destination. Stop.
5. Check if an entry exists with the packet size greater than the current packet size. If so use that information for routing. Goto 7
6. Use any entry which matches the destination address for routing.
7. Create new forward agents and send them through all possible paths to the destination. Create a new entry in the routing table if the number of entries for that destination is less than a upper limit and set its limit value to maximum.

4. Limiting the Agent Population

The Limit Counter is used to limit the number of agents in the network and to reduce the network bandwidth required for routing. Whenever an agent spawns new agents at a router the value of the limit counter corresponding to that destination is set to the maximum value. The counter decays with time. When an agent has to find a path between the router and the destination it checks to see if the limit counter is less than the *cloning threshold*. If so the agents spawns many agents and sends them along all routers that can be reached from the current router. Otherwise it takes a random path from the current router to the destination. This reduces the agent population. Agent limiting however does not affect the efficiency of the algorithm as all the necessary information is made available at the destination, by the agents spawned earlier at the router.

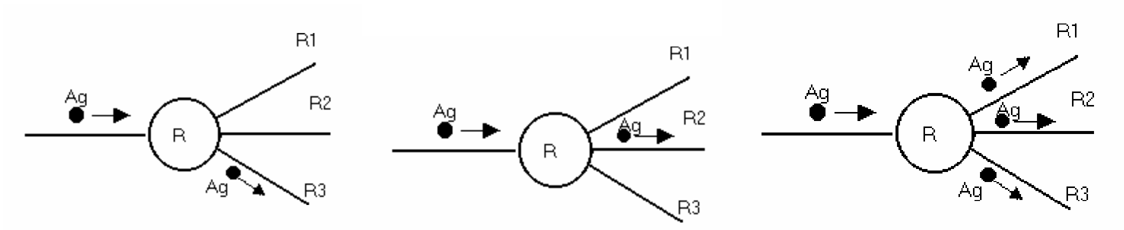
An agent has a *Time to Live (TTL)* field which specifies the life time of the agent. If the agent cannot reach its destination before its lifetime expires the agent dies.

5. Methodology

5.1 Intermediate Router

An agent arriving at an intermediate router performs the following operations

1. The agent checks the presence of other agents.
2. If a forward agent discovers a backward agent or vice versa and if the forward agent finds its destination and the current router as intermediate routers in the optimal path that is being traversed by the backward agent, the forward agent gets the route from the current router to its destination from the backward agent's optimal path and follows the same to reach the destination
3. If the value of the goodness counter for the destination and the packet size for which forward agent searching the path is greater than the threshold then the agent takes the next hop specified in that entry of the routing table. (Fig 1a)



GC>threshold | GC<threshold LC>=threshold | GC<threshold LC<threshold

(a) (b) (c)

Ag – Agent
R – Intermediate Router
LC – Limit counter
R1, R2, R3 – Routers connected to R
GC – goodness counter

Fig 1: Possible cases when an agent arrives at an intermediate router and finds an entry for {destination, packet-size} with next hop router as R3.

4. If the routing table has an entry for the destination and the packet size and the goodness counter value is less than the threshold then
   i. If the limit counter's value is greater than or equal to the threshold the agent proceeds in the path specified in the routing table.(Fig 1b)
   ii. If the limit counter's value is less than the threshold, set the limit counter value to the maximum and the agent clones itself and proceeds in all possible paths from that router. It also sets the LC to maximum value. (Fig 1c)
5. If the routing table has an entry for the destination but not of the same packet size the agent is looking for, then the agent creates a new entry for that packet size replacing an entry with the lowest limit counter value, sets the limit counter to maximum, clones itself and seeks all possible paths to the destination.
6. If the agent is a backward agent then, as the agent is traversing in the optimal path it updates the next hop for destination and the packet size in the routing table. If the entry for that packet size does not exist, the agent creates a new entry for that packet size replacing an entry with the lowest limit counter value, recording the next hop for the {destination ,packet size} in the routing table.

The goodness counter and the limit counter values are decremented at fixed time intervals. The interval depends on the network stability.

5.2 Destination Router

When an agent arrives at a destination the following take place

1. If the agent is the first from a particular source, a counter is initialized to zero and it is incremented with time.
2. Copy the information in the agent to the system and dispose the agent.
3. When the counter for any source reaches a threshold the optimal path from that source to the destination is calculated from the information obtained from the agents by using graph theory concepts.

4. Create a new mobile agent and send it along the optimal path from the destination to the source. The agent updates all routing tables along its path by setting the GC value depending on the goodness of the route.

6. Finding the optimal path

The following parameters are used for finding the optimal path

Amount of fragmentation:

Higher the number of fragments higher the error rate. Hence fragmentation has to be kept minimal.

Point of fragmentation:

A path with larger fragmentation at the end may be better than a path with relatively less fragmentation in the beginning.

Bandwidth utilization:

Bandwidth losses due to small packets traveling through networks with larger MTU must be minimized.

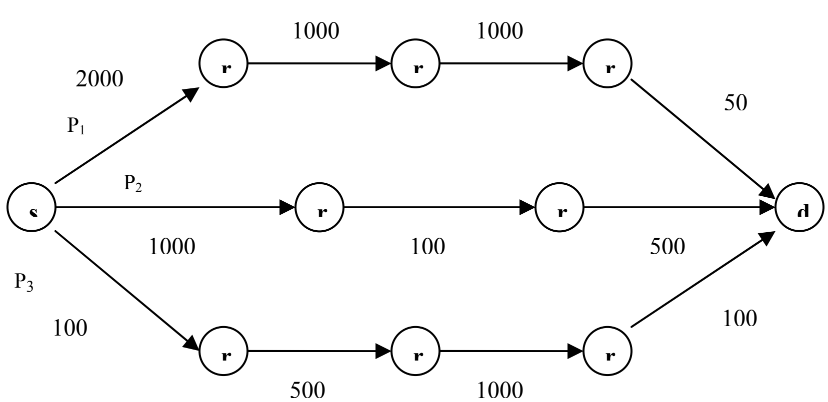
The following function is used to evaluate the goodness of a path. The lower the value of the function, the better is the path.

$$F(p_k) = \sum_{i=1}^{n} (\alpha\, (f_i * (1-e_i)) + b_i(1-\alpha))$$

$$f_i = \lceil PS/SMTU_i \rceil$$
$$b_i = (MTUi - SMTUi)/SMTU_i$$

n – number of networks in the path p_k,
PS – Packet size
MTU_i – MTU of the i^{th} network
e_i – error rate in the i^{th} network
$SMTU_i$ – smallest MTU in 1^{st} to i^{th} network.
α – decision parameter

The value of α gives the tradeoff between maximizing throughput and minimizing fragmentation. Lower α value gives higher weightage to fragmentation, and vice versa.



F - number of fragments created in the network from the original packet
r – router s – source d – destination → - network

Fig 2 A sample network with MTU in number of octets

Packet size = 1000 octets, $\alpha = 0.5$
Assuming a zero error rate for all the networks we have
$F(p_1) = (.5+.5)+(.5+0)+(.5+0)+(10+0) = 12$
$F(p_2) = (.5+0)+(5+0)+(5+2) = 12.5$
$F(p_3) = (5+0)+(5+2)+(5+4.5)+(5+0) = 26.5$

The calculations show that path p1 is the best path for a packet size of 1000, as it strikes a good balance between number of fragments and bandwidth utilization. The path p1 has lesser fragmentation than path p3 and better bandwidth utilization than path p2 (though p2 has lesser number of hops). Path p3 has poor bandwidth utilization and large fragmentation.

7. Conclusion

In this paper we presented a routing algorithm, which effectively addresses the problems associated with the disparity between packet size and the MTU of the network through which it has to traverse. We used an agent based approach to reduce the bandwidth used for routing. A methodology for finding the optimal path, based on the tradeoff between amount of fragmentation and bandwidth utilization was also presented. Fixing the value of the decision parameter of a network requires further study.

References

[1] K.Amin, J.Mayes and A.Mikler, *"Agent Based Distance Vector Routing"*. Proceedings of the Third International Workshop,MATA 2001,Montreal, Canada, August 2001.

[2] Kaizar A.Amin and Armin R.Mikler, *"Towards Resource Efficient And Scalable Routing : An Agent Based Approach"*

[3] Gianni Di Caro and Marco Dorigo, *"An adaptive multi-agent routing algorithm inspired by ants behavior"*

[4] F.P. Kelly, "*Network routing",* Philosophical Transactions of the Royal Society, 1991

[5] Boyan J. A. & Littman M. L. 1994. *"Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach*". In Proc. of NIPS-6, San Francisco, CA: Morgan Kaufmann, 671–678.

[6] Shankar A.U., Alaettinoglu C., Dussa-Zieger K., & Matta I. 1992. Performance Comparison of Routing Protocols under Dynamic and Static File Transfer Connections. *ACM SIGCOMM Comp. Comm. Review* 22(5): 39–52.

[7] Steenstrup M. E. (ed.) 1995. *Routing in Communications Networks*. Englewood Cliffs, NJ: Prentice-Hall.

[8] Stone P. & Veloso M. 1996. Multiagent Systems: A Survey from a Machine Learning Perspective. *Tech. Rep. CMU-CS-97-193*, Carnegie Mellon University, PA.