# USING ROUTING AGENTS FOR IMPROVING THE QUALITY OF SERVICE IN GENERAL PURPOSE NETWORKS

Mohammad Mottaghi[1] and Masoud Daneshtalab[2]
[1]*Dept. of Electrical and Computer Engineering, University of Tehran, Tehran, Iran*
[2]*Dept. of Information Technology, Turku University, Turku, Finland*

Keywords: Agent, Context-aware, Collaboration, Emergence, Routing, Network capacity, Utilization, QoS.

Abstract: In this paper, we have proposed a new routing method for general purpose networks. Each router in this model is a context-aware agent and therefore the whole network of routers forms a community (society) of context-aware agents which are ever learning and adapting the routing network. During routing each agent keeps learning and specializing in routing some specific traffic type (e.g. intermittent audio packets) and makes its neighboring routes optimum for this type of traffic whereas other agents are experts in other traffic types. All agents are aware from their colleagues' expertise and when a new traffic type enters the network all agents try to collaboratively recognize (or detect) its type and route it according to their past experience (which is already learned). If the traffic type is unknown to all agents, one of them tries to learn how to route it such that its QoS constraints are met better. The idea behind the proposed model is to temporarily modify some routes whenever QoS constraints cannot be met in the network. Simulation results show that the proposed model can improve QoS of the network by 12%.

## 1 INTRODUCTION

With the advent of the Internet many applications emerged which wanted to communicate through the Internet. Few years after the Internet became popular various traffic types were injected to the Internet and now there are more than 120 homogeneous and 170 heterogeneous traffic types passing through the Internet (Dally, 2001). The most common communication scenario in today Internet begins with a *client* requesting a *server* for some data. After receiving the request, the server prepares the requested data and sends it to the client. This is the simplest and of course the most common practice in data communication over the Internet and there are more than 1 billion of such data transactions per second in today Internet (Dally, 2001). Among all different traffic types passing through a general purpose network (not necessarily the Internet) there are some special traffic types which have some constraints (often time limits). A very good example for this type of traffic is audio and video streams. In addition to these data streams which can be lossy, there are some other data streams which have to be lossless and no packet can be dropped during routing (Ni, 1993), (Ghiu, 2000), (Shin, 1996), (Kermani, 1979), and (Glass, 1992). For these traffic types some metrics are defined which are known as QoS (Quality of Service) constraints. A long list can be written for QoS constraints since the most common constraints are *delay* and *jitter*.

Now we are faced with a new problem where some packets belong to a data stream which has some QoS constraints while the constraints should be met. A very simple solution to the problem is to dedicate a well qualified route for these traffic types. While it is very expensive and impractical, each QoS constraint raises new requirements. A better and more reasonable solution is to establish a relatively well qualified and general purpose network and to use this single network for communication of all traffic types. This is what happens in many general purpose networks including the Internet.

To meet QoS constraints using a general purpose network, there should be some ways to drop or postpone the packets having no constraints or are of lower service quality. The common practice towards this goal is to use priority queues. Although this solution can meet many QoS constraints, it does not utilize network resources (e.g. bandwidth) optimally. This is due the fact that each router routes independently from other routers (non-neighboring routers) and also does not take into account the

Mottaghi M. and Daneshtalab M..
USING ROUTING AGENTS FOR IMPROVING THE QUALITY OF SERVICE IN GENERAL PURPOSE NETWORKS.
DOI: 10.5220/0003434906090613
In *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems* (SAAES-2011), pages 609-613
ISBN: 978-989-8425-48-5
Copyright © 2011 SCITEPRESS (Science and Technology Publications, Lda.)

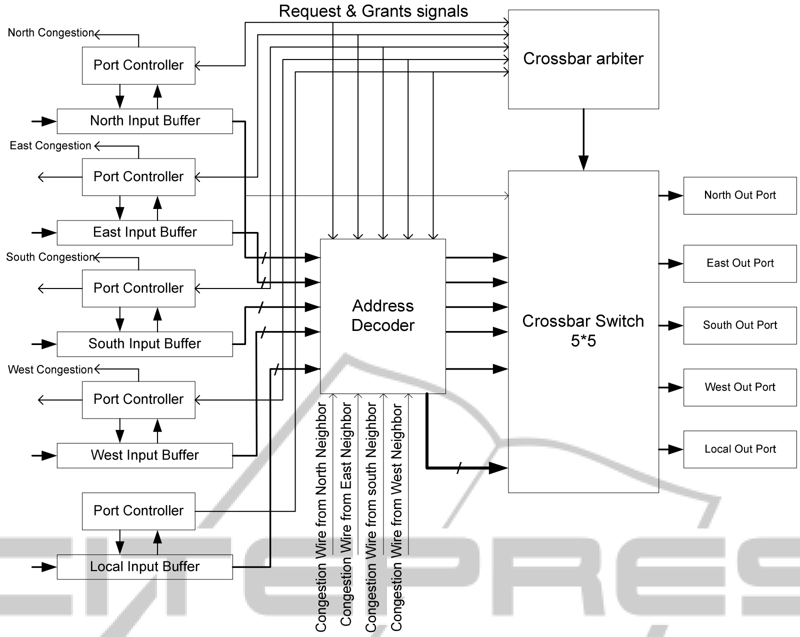


Figure 1: The proposed structure.

SCIENCE AND TECHNOLOGY PUBLICATIONS

entire network capacity. In fact, in the conventional routing method, each router routes individually regardless of what other routers are doing and where in the network some extra bandwidth exists.

In this paper, we have proposed a new routing method which using context-aware routing agents to switch between minimal and non-minimal routes. This enables each router (agent) to alter the routes such that all network capacity is utilized. This is in contrast to the conventional technique in which even if routers are context-aware, the entire network capacity cannot be utilized.

# 2 THE PROPOSED ROUTING TECHNIQUE

The proposed routing technique attempts to utilize all network resources such that the packet loss becomes less and higher quality services become feasible.

The drastic difference between the proposed and the conventional routing techniques is that in the proposed technique each router uses both minimal and non-minimal paths based on the entire network information and other routers' experience. In contrast, the conventional routers just use minimal paths to route packets in the network.

## 2.1 Context Awareness

If we know the attributes of the traffic (like inter-arrival times, acceptable delay, etc.) we can decide whether or not a dedicated route is needed. When a new traffic type is injected to the network the receiving agent tries to recognize its type. It measures some attributes of the incoming data stream. If it could not recognize the traffic type it asks a neighboring agent (router) to recognize the traffic type. If agents could not recognize the traffic type it is considered as a new type and the receiving agent (at one of network edges) is responsible for learning how to treat it. The learning process is based on reinforcement learning methods (Sutton, 2002). When a new traffic type comes to the network the routing system does not perform necessarily well and it takes some time for an agent to learn how to treat this type of traffic. The learning process is even not always convergent and some controlling criteria should be applied to avoid the system from oscillation. These criteria (heuristics) are often not mathematically proven; we used the number of hops that a packet visits as the criterion.

## 2.2 Emergence

After the traffic type is recognized the *expert* agent is found too. The expert agent is the agent who knows how to treat the coming traffic. It evaluates the entire network capacity and determines whether or not the existing routes can accommodate the newly coming traffic. If no accommodation problem exists then the new traffic flows through the network without any modification in the network. If any shortcoming was found, the expert agent looks for agents who can provide more bandwidth. If it was successful in finding any helping agent, the two agents issue *reconfiguration* commands to reconfigure the network such that new QoS constraints are met. Otherwise newly coming packet are all dropped, since the network can not serve them.

If the expert agent realizes that a reconfiguration in the network is required and if it is successful in convincing other agents to reconfigure the network, a new route emerges which can accommodate the newly coming traffic.

## 2.3 Collaboration

The whole routing in the proposed technique is collaborative; *collaboration* among agents can be seen in three different situations. First, when a traffic type is to be recognized, agents send *recognition* messages to their neighbors. Second, while looking for more bandwidth the expert agent requests its neighbors to provide it with more bandwidth. Finally, when the expert and the helping agent want to reconfigure the network they issue *reconfiguration* commands to their neighbors.

## 2.4 Router's Hardware Core

In this section, implementation details of the cited router are described. As shown in figure 1, each input port includes an input buffer and a controller which are used for temporary storage of packets. The controller of each input port is responsible for two tasks:

1. Receiving packets from the link and requesting the crossbar arbiter for packet injection grant. Also, controlling the buffer status including empty and full states.
2. Calculating the sign of the rate at which the buffer is getting full: positive rate indicates that the buffer is getting full and negative rate indicates that the buffer is getting empty. This sign is compared with the buffer status to activate *congestion* signal. Each input port has a congestion signal through which it informs its adjacent router whether it is congested or not. Therefore, the owner of an input port may be considered a hotspot for its adjacent router and the adjacent router tries to remove this problem.

Another part of this router is the crossbar which establishes a connection from an input port to an output port. Since a crossbar can serve only a single port at a time, it uses an arbitration for the crossbar to arbiter among the input ports which are not empty. Afterward, the crossbar is dedicated to the granted input port.

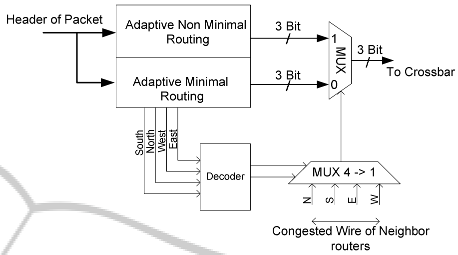


Figure 2: Address decoder module.

Address decoder is another part of the router taking the header of the packet stored in the input buffer. It determines the routes according to both minimal and non-minimal path adaptive routing algorithms. The routing algorithm is based on HAMUM (Daneshtalab, 2011) which is an adaptive minimal and non-minimal routing algorithm. Then, the address decoder selects a route which is less congested, i.e. if the route resulted from the minimal path adaptive routing algorithm is congested, the router does not take this route and instead it uses the result of the non-minimal path adaptive algorithm.

In order to know if the input buffer status is congested or not, we should measure the rate at which the data come into the buffer. To do so the simple module depicted in Figure 3 is used. As shown in Figure 3, since routers are locally synchronous and globally asynchronous, no timing source is used for reading from and writing to the buffer. Therefore, to obtain the rate at which the buffer is getting full, the number of filled buffer cells at each rising edge of router's internal clock ($N_{new}$) is compared with that of the previous rising edge ($N_{old}$), i.e. $N_{new}>N_{old}$ shows that the buffer is getting full while $N_{new}>N_{old}$ indicates that the buffer is getting empty.

Buffers of this router have two special signals *W_Empty* and *W_Full*. When the number of empty cells of the buffer is smaller than a threshold value (75%), *W_Full* (Warning Full) is activated warning that most buffer cells are full. Congestion status, which utilizes the *W_Full* signal and the buffer filling rate, is used to inform adjacent routers regarding the input port congestion condition. When a packet reaches the input port waits until other previously arrived packets leave the input buffer.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

When it is the packet's turn to be routed and the crossbar arbiter allows the packet to be sent to one of the output ports, packet header is delivered to the address decoder to find an efficient non-congested route to the destination. Thereby, both minimal and non-minimal path adaptive routing algorithms are simultaneously loaded to specify two minimal and non-minimal routes. Afterward, the result of minimal path adaptive algorithm of HAMUM checks the corresponding congestion signal coming from the neighbor input port. If the input port of the corresponding neighbor router is not congested the packet is sent to that output port. Otherwise, the non-minimal route is selected.

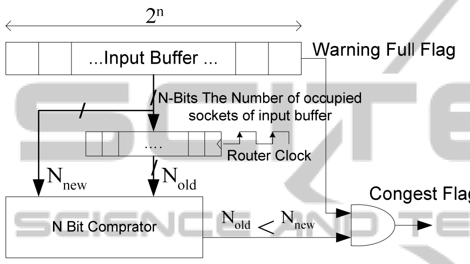


Figure 3: Congestion detection module.

## 3 SIMULATION RESULTS

To evaluate the proposed scheme and to compare it with existing routing schemes, we have simulated our proposed scheme with the network simulator (NS-2, 2006). The agents were implemented in C (.m files) and simulated using MATLAB (Matlab, 2006). The network core in our simulation has 150 routers through which 80 different traffic types were flowing (a total of 150000000 packets per second). For traffic generation we have used NS packages developed by Ulrich Fiedler for http traffic type (http traffic generator, 2006) and video streams (TES video traffic generator, 2006). For other traffic categories we used NS package BonnTraffic (scenario based traffic generator, 2006) which is a scenario based traffic generator. We also generated a traffic pattern with high QoS constraints, using MATLAB, and injected to the network.

Results are depicted in Figure 4 to Figure 6. As shown in Figure 4 the proposed technique has less delay in comparison with that of priority queues, but as compared to FIFO it has longer delay. This is because in FIFO there are more packet loss than in the proposed technique and these packets (since dropped) are not taken into account in delay calculation. In Figure 5 the proposed routing technique is compared with other conventional routing techniques in terms of network utilization. As expected, the proposed technique has better network utilization. The best network utilization of the conventional techniques belongs to WFQ (weighted fair queuing) which is only 30%; network utilization of our technique is 72% which is by 42% better than WFQ. This is because in the conventional techniques no route is altered for QoS purposes, whereas in the proposed technique the routers can temporarily form new highways (collections of routes) by changing the direction of some hop-to-hop routes. In fact in the proposed technique new routes emerge.

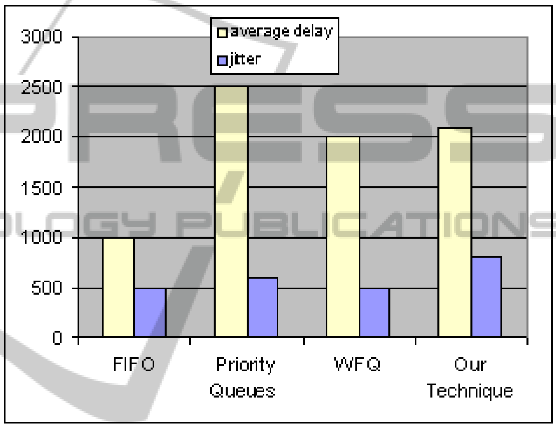


Figure 4: Delay and Jitter comparison among 4 routing techniques: FIFO, Priority Queues, WFQ and the proposed technique.

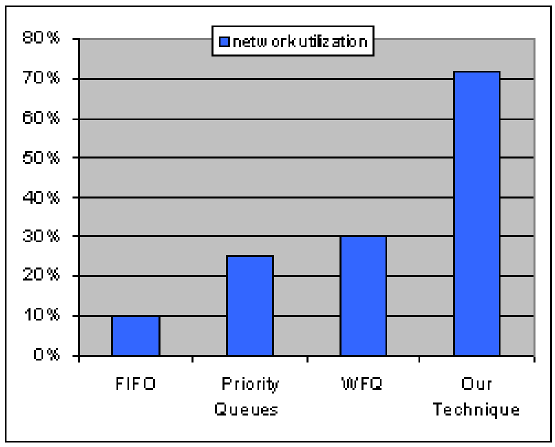


Figure 5: Network utilization of 4 techniques: FIFO, Priority Queues, WFQ and the proposed technique.

As depicted in Figure 6, the proposed techniques cause fewer packet losses than that of FIFO, Priority queues and WFQ. This emanates from context-

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

awareness and collaboration among agents. In the conventional routing techniques when QoS constraints cannot be met, the packets are simply dropped whereas in the proposed technique, agents collaboratively try to modify the routes such that new QoS constraints can be met. The efficiency of the context-awareness along with the collaboration of agents result in the emergence of new efficient routes helping the entire routing system drop fewer packets. Despite having longer *average* delay, the proposed routing technique has a higher throughput and has lower packet loss (Figure 6).

# 4 CONCLUSIONS

We have proposed a new agent-based routing method for general purpose networks. Each router decide to route the packets to either the minimal path or non-minimal path based on the presented learning technique among agents (routers). All agents are aware from their colleagues' status and when a new traffic type enters the network all agents try to collaboratively detect its type and route packets according to their past experience. Simulation results show that the proposed model can improve QoS of the network by 12%.

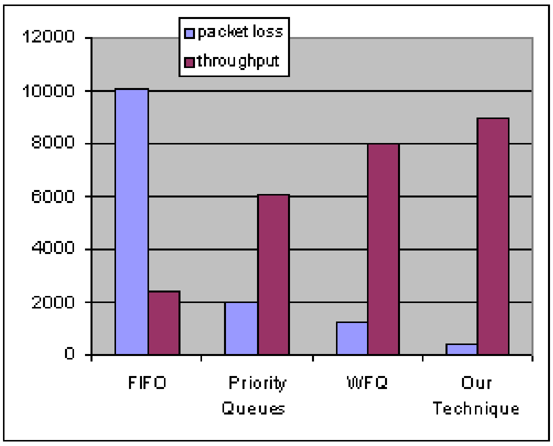


Figure 6: Throughput and Packet Loss of our technique in comparison with those of FIFO, Priority Queues and WFQ.

# REFERENCES

Dally, W. J., Towles. B., 2001. Internet traffic diversity analysis. *In Proc. DAC, pages 684.689*.

Ni. L. M., McKinley, P. K., 1993. A survey of wormhole routing techniques in direct networks. In *IEEE Tran. on Computers, 26:62.76*.

Chiu, G., 2000. The Odd-Even Turn Model for Adaptive Routing. In *IEEE Tran. On Parallel and Distributed System, Vol.11, NO. 7*.

Shin, K. G., Daniel, S. W., 1996. Analysis and implementation of hybrid switching. In *IEEE Tran. on Computers, 45(6):684.692*.

Kermani, P. Kleinrock, L., 1979. Virtual cut-through: a new computer communication switching technique. In *Computer Networks, volume 3, pages 267.286*.

Glass, C. J., Ni, L. M., 1992. The Turn Model for Adaptive Routing. In *Proc. Symp. Computer Architecture, pp. 278-287*.

NS-2, network simulator 2, retrieved on 2006-06-25, http://nsnam.isi.edu/nsnam/index.php/

Matlab, 2006. http://www.mathworks.com/ products/matlab/.

Http traffic generator, NS-2 package, retrieved on 2006-06-22, http://www.tik.ee.ethz.ch/~fiedler/provisioning.

TES video traffic generator - NS-2 package, retrieved on 2006-06-22, http://nsnam.isi.edu/ nsnam/index.php/Contributed_Code

Scenario based traffic generator, NS-2 package, retrieved on 2006-06-22, http://web.informatik.uni-bonn.de/IV/BoMoNet/

Sutton, R. S., Barto, A. G., 2002. Reinforcement Learning: An Introduction. *MIT Press, Cambridge*.

Blaha, M. R., Premerlani, W., 2004. Object- and Agent-Oriented Modeling and Design. *McGrawHill.*

Daneshtalab, M., Ebrahimi, M., Xu, T. C., Liljeberg, P., Tenhunen, H., 2011. A Generic Adaptive path-based routing method for MPSoCs. In *Journal of Systems Architecture (JSA-elsevier), Vol. 57, No. 1, pp. 109-120.*