Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Communications 28 (2005) 829–836

computer communications

www.elsevier.com/locate/comcom

Multiple path routing algorithm for IP networks

Miguel Rios*, Vladimir Marianov, Andres Avagliano

Department of Electrical Engineering, Universidad Catolica de Chile, Vicuna Mackenna 4860, Casilla 306, Correo 22, Santiago 6904411, Chile

Received 16 June 2003; revised 4 November 2004; accepted 24 November 2004
Available online 20 December 2004

Abstract

Internet routing protocols, such as Open Shortest Path First (OSPF), compute a shortest path tree from each node to other nodes in the network, using link-state information. Such protocols do not consider the queueing situation at a given node. An alternate path of higher cost may be more convenient to use than the optimum path when a long queue is present at the node. This paper proposes a new Multiple Path Routing Algorithm (MPRA), which uses dynamic shortest path tree mechanisms, load balancing among alternative paths to destination, and path feasibility analysis to avoid network loops. A better-cost model is also introduced. Computer simulations show that MPRA measured parameters (throughput, packet delay, etc.) improve over OSPF by amounts close to 30% for unbalanced networks, and close to 20% for balanced networks.

Keywords: Dynamic routing; Load balancing; Network routing

1. Introduction

In today's IP networks, routing protocols are responsible for building a path that will carry a data packet to its destination [1]. Each router in the network has to send the packet to its next hop, independently from what other routers are doing at the time, on rules based only on its own knowledge base. These routing tables are built based on topological and traffic information, send or captured from information being received from other routers. This paper focuses on finding improvements to the Open Shortest Path First (OSPF) routing protocol [7,9]. OSPF is a link-state based routing protocol, which finds the shortest (minimum cost) path for a data packet through the network [10], using the cost associated to each link. In most cases, however, there are alternative paths to the destination that, even if they are not optimum, have available capacity that could be used to improve the total network throughput.

This paper proposes a new cost model for evaluating the network links and a new dynamic algorithm for the re-routing of packets, that implements data load balancing, by way of using alternate paths to a given destination. Furthermore, we analyse the feasibility of sending a packet for each link using a shortest path compatible criterion. The new algorithm is called Multiple Path Routing Algorithm or MPRA.

The paper is organised as follows: in Section 2 a short background on the routing subject is given. Section 3 discusses some previous work, which is used to build the new algorithm. Section 4 explains the cost model used in the algorithm. Section 5 gives implementation details of the MPRA algorithm. Section 6 presents computer simulation results showing the improvements that MPRA offers with respect to OSPF. Finally Section 7 presents the conclusions of this work.

2. Background

In packet switching networks, such as the Internet, a packet goes to its destination through several intermediate nodes. When a given packet reaches a router, a routing protocol determines which output link should be used for sending the data to the next hop.

Each link between two routers has an associate cost, which can be used to calculate the best path to use to get to the packet destination from a given router. Usually these

* Corresponding author. Tel.: +56 23544291; fax: +56 25522563.
E-mail address: mrios@ing.puc.cl (M. Rios).

0140-3664/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2004.11.014

costs, independently of how they are calculated, are considered static through time, and each router provider establishes its own way of calculating them depending, at least, on the link available bandwidth [2].

But, in the real world, costs are not fixed as assumed but dynamic, depending not only on the physical characteristics of each link, but also on the load or other variables of the network. This is the scenario, where this paper proposes a path cost model, allowing a better discrimination on the packet next hop to follow.

On the other hand, when the optimum output link is congested, the marginal cost of transmitting the packet through that link is very high, due to the queueing nature of a congested link operating near its capacity. In this situation, sending the packet through an alternate path can be very attractive in terms of the total cost. The proposed MPRA algorithm uses this fact, in such a way that each router will distribute the traffic load among several output links (with available capacity) that can take the load to their destination. One of the advantages of the proposed algorithm is that it is based on the same algorithm (Shortest Path First or SPF) that is used by the OSPF protocol.

Finally, the next hop output link has to be chosen carefully since, the same as in link-state based routing protocols like OSPF, the chosen path must guarantee it will not produce undesirable loops.

3. Related work

There have been several previous efforts to improve the network routing algorithms, both in time and space. This paper follows the study done by Narvaez [3,4], which proposed several improvements in different stages of the routing process. One of the most important proposals is changing the current static SPF algorithms (such as Bellman-Ford, Dijkstra and D'Esopo-Pape [5]) into dynamic algorithms. On the other hand, Narvaez proposed a routing protocol limiting the amount of information to be transmitted through the network and a routing algorithm that search for multiple paths to a destination.

Some other studies related to multiple path routing include the Flow Deviation Algorithm (FD) [16] and the Bertsekas-Gallager Algorithm (BG) [6]. Both algorithms minimise the average delay through the packet path. These algorithms, after several iterations, tend to a state with a shortest delay on average. One of the problems of these algorithms is that they use information from the entire network at all times, so the routing information, about the link congestion on the network, circulating among nodes is very large and contributes to the network congestion. Both FD and BG were designed to work in virtual circuit networks, which calculate the routing in a centralised way, heavily incrementing the amount of information (overhead) and delay to deliver the routing information to all the routers in the network.

4. Cost model

In this paper, we first find a cost model that is useful for characterising an output link. The model has to be matched to the basic traffic characteristics of a channel, besides reasonably representing the physical limitations of the channel. A linear approximation was finally used to reduce the computation time in routers, using only those iterations that are strictly necessary, and without losing any useful information. After finding this approximate model, a basic criterion is proposed to perform the load balancing among the output links.

Finally, some priority logic is used to select among candidate output links for the next hop because, at the time of balancing the load, there are several alternate paths and, generally (because of congestion), the optimum output link will not necessarily be used, given the shortest distance criterion used by OSPF. These chosen paths should be feasible and should not produce loops in the network.

4.1. Queueing model

Since a transmission line K times as fast will accommodate K times as many packets/s at K times smaller average delay per packet[1], costs in the Internet are generally assigned as being inversely proportional to the channel bandwidth [7], that is:

$$C = \frac{\mathrm{BW}_{\mathrm{ref}}}{\mathrm{BW}} \tag{1}$$

where C is the channel cost, $\mathrm{BW}_{\mathrm{ref}}$ is the reference bandwidth of network channel [2,8], and BW is the real link bandwidth. It can also be observed that the cost is related to both the waiting and service time of the packet being transmitted.

Other cost representations may include the delay cost. For instance, when dealing with satellite links, the following formula has been suggested:

$$C = \left(K_1 \times \frac{\mathrm{BW}_{\mathrm{ref}}}{\mathrm{BW}}\right) + (K_2 \times \mathrm{Delay}) \tag{2}$$

where K_1 and K_2 are weighing coefficients. In Eqs. (1) and (2), cost is considered fixed over the network operation time. The cost can only be changed when a given link fails, or a new link is added to the network, or any topological change in the network is produced.

In the real world, costs are not fixed but depend on the queueing model being used. Generally speaking, the output link cost can be related to the total time (T) the packet stays at a node, which is the sum of the service (or transmission) time (T_S) and the queueing waiting time (T_W) or queueing delay. This total time has been modelled as a M/M/1 Markov Chain [6,11,12], where data packet arrivals are

[1] See pp. 170 of Ref. [17].

assumed independent with exponential interarrival time distributions, according to Burke's theorem.

If λ is the average packet arrival rate, and $T_S = 1/\mu$ is the average packet transmission rate, then by Little's Theorem the node throughput ρ is given by:

$$\rho = \lambda T_S = \lambda/\mu \tag{3}$$

It follows then that the total time the packet stays at a node (both waiting and in service) is given by [6]:

$$T = T_S + T_W = 1/\mu \times 1/(1-\rho) = T_S/(1-\rho) \tag{4}$$

where T_S, as it was mentioned before, is related to the fixed cost C_F through:

$$C_F = \mathrm{BW}_{\mathrm{ref}} \times T_S = K \times T_S \tag{5}$$

Now, if several output links exist at a node, these links can be seen as independent queues. An example of three output link queues is shown in Fig. 1.

When the output link is used at a small capacity (say $\rho < 0.5$ in Fig. 1), the cost versus ρ is almost constant. This value is called *fixed cost* or C_F, which depends on the physical characteristics of the link. On the other hand, when the amount of information being carried approaches the capacity of the channel, the cost raises due to the congestion related to the traffic load of the channel. This cost will be called *variable cost* or C_V.

Relating, as before, variable costs with the queue waiting time T_W, and assuming all packets through the node request the same service type, then

$$C_V = K \times N_q \times T_S \tag{6}$$

where K is a constant with the same functionality as $\mathrm{BW}_{\mathrm{ref}}$ and $N_q = \lambda T_W$ is the number of packets in queue waiting to be transmitted. Then the total cost for the node output link is given by

$$C_L = C_F + C_V \tag{7}$$

$$C_L = K \times T_S + K \times N_q \times T_S = K \times T_S \times (N_q + 1) \tag{8}$$

where the first term explains the physical characteristics of the link and the second the link traffic load.

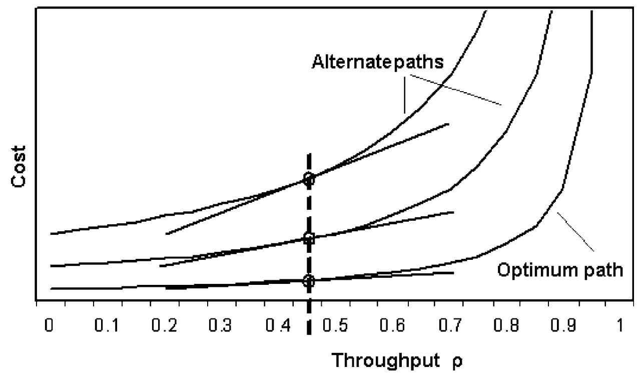

Fig. 1. Link throughput curves for a three-output link node.

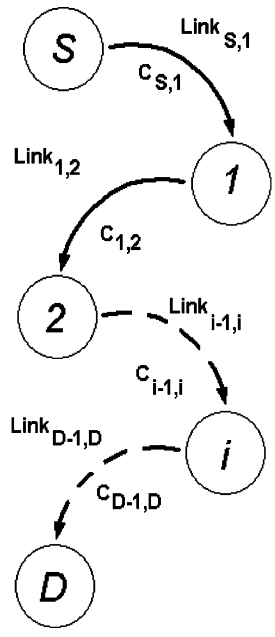

Fig. 2. Path example.

4.2. Proposed cost model

From the previous discussion, the new cost model is based in the following assumptions:

1. Burke's Theorem reasonable represents the link transmission capacity[2].
2. The output link cost can be modelled as the sum of a fixed cost and a variable cost, where the fixed cost is determined by the physical structure of the links in the path and is calculated using the current SPF algorithm, and the variable cost is determined by the node output link cost and its utilisation at a given time.
3. The variable cost represents the output link traffic load and is a good estimation of the complete path load to the destination.

In Fig. 2 it can be seen, according to the assumptions, that if SPF delivers the shortest path between a source S and a destination D, then the path cost is the sum of every link cost in that path, which would be given by:

$$C_T = C_{S,1} + C_{1,2} + \dots + C_{i-1,i} + \dots + C_{D-1,D}$$
$$= \sum_{j=S}^{D-1} C_{j,j+1} = C_V + C_F \tag{9}$$

From Eq. (8) we observe that:

$$C_L = K \times T_S \times (N_q + 1) = C_{\mathrm{queue}} + C_{S,1} \tag{10}$$

[2] From Kleinrock's assumptions, see pp. 147–160 of [11] and pp. 314–329 of [12].

By the third assumption, C_L is the cost of the next hop from the source node and represents a good estimation of the complete path load to the destination. Then the total cost of the path not including the first hop, $C_{REM,}$ is given by:

$$C_{REM} = \sum_{j=S}^{D-1} C_{j,j+1} - C_{S,1} \tag{11}$$

The total cost of the path is then given by

$$C_T = C_L + C_{REM}$$

$$C_T = (C_{queue} + C_{S,1}) + \left(\sum_{j=S}^{D-1} C_{j,j+1} - C_{S,1} \right) \tag{12}$$

$$C_T = C_{queue} + \sum_{j=S}^{D-1} C_{j,j+1} = N_q \times C_{S,1} + \sum_{j=S}^{D-1} C_{j,j+1}$$

Finally the cost model for the complete path is:

$$C_T = K \times N_q \times T_S + \sum_{j=S}^{D-1} C_{j,j+1} \tag{13}$$

where $C_{j,j+1}$ is the link cost used in the OSPF protocol.

This equation shows that a part of the cost depends on the OSPF costs (typically assigned by the users) and the other part depends on the traffic load from the.

4.3. Next feasible hops

The routing algorithm through alternate paths has to comply with the following:

- The alternate path should be the second best to the optimal one and without loops.
- The decision on the path to be followed by the packet at a given node should be based only on local information available at that node.
- The congestion state of each output link has to be taken into account, as well as the cost of the path to the destination.

This paper uses as a starting base the Narvaez Algorithm [3,4], called MPA, to select the next feasible hop. This algorithm uses an efficient data structure plus the basic SPF algorithm, checking and registering on each iteration the necessary information about alternate paths to the destination. This algorithm was chosen since the objective is to improve the re-routing system using only a more efficient local and dynamic decision. The data structure used by the MPA algorithm is as follows:

	Next hop	Link cost	Distance
Shortest distance from node S to node X	Hop $S \to A$	$w(S,A)$	$d_A(S,X)$
	Hop $S \to B$	$w(S,B)$	$d_B(S,X)$
	Hop $S \to C$	$w(S,C)$	$d_C(S,X)$
	Hop $S \to D$	$w(S,D)$	$d_D(S,X)$
$d_{opt}(S,X)$	…	…	…

For each node X, the first input in the data structure contains the shortest distance $d_{opt}(S,X)$ found so far. This information is computed using the Dijkstra algorithm in OSPF. Furthermore, the cost $w(S,p)$ for every output link connecting the node S to node p is also maintained in the structure, as well as the shortest distance to the destination through that node p, that is $d_p(S,X)$. When the algorithm ends, $d_p(S,X)$ should be larger or equal than the sum of $w(S,p)$ and the shortest distance $d(p,X)$ from p to X.

4.4. Feasibility test

To check which are the next feasible hops without loops, the following test is performed.

$$\text{If } d_p(S,X) < (d_{opt}(S,X) + w(S,p)), \text{ then } p \text{ is a feasible next hop.} \tag{14}$$

The test ensures the distance to the destination decreases monotonically, hence a packet will never return to a node twice.

The path without loops policy is then as follows: if router S sends a packet with destination R_2 to its next hop R_1, then:

$$d(R_1, R_2) < d(S, R_2) \tag{15}$$

4.5. Next hop choosing

Similarly to OSPF, the objective is to find the *shortest path to destination*. To that purpose the costs are compared and the next hop of minimum cost will be chosen. Fig. 3 shows an example of how this is done.

In Fig. 3 for a given throughput U, the curve with lower cost corresponds to the shortest path to the destination. Of course if U is high (as in congested systems) the cost will tend to infinity, even in the optimum path. The costs when $U=0$, correspond to the OSPF design criterion.

The traffic load balancing should be done precisely when the throughput U of the optimum path reaches the fixed cost of the next alternate path (point b in Fig. 3, where the curve cost represents the alternate paths in terms of costs).

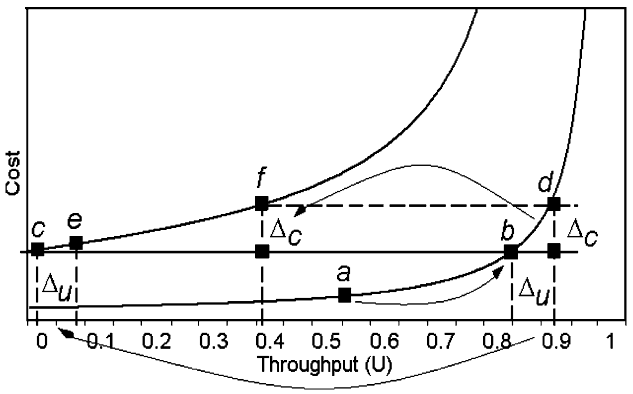

Fig. 3. Throughput curves showing the load balancing process.

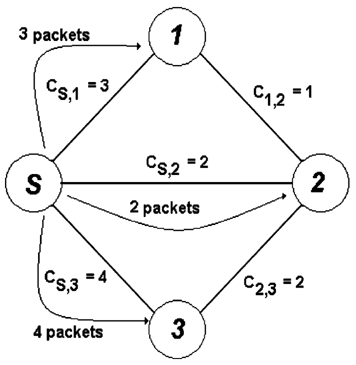

Fig. 4. Four node network example showing costs and packet load.

In Fig. 3, moving from point b to d implies a throughput difference Δu with a cost difference of Δc. The option at this time is placing Δu in the alternate path, i.e. moving from point c to point e with same Δu, or moving from point c to point f with same Δc. In this paper we adopted the first approach, that is, to move the difference Δu to other alternate path. It is clearly seen from Fig. 3 that point e has a lower cost than point d and so provides a better solution.

The balancing process can be shown, as an example, for the network of Fig. 4.

Table 1 shows the data structure for the example with next hops to node 1 from node S, calculated by the MPA algorithm ($S\rightarrow 1$, $S\rightarrow 2$ y $S\rightarrow 3$).

OSPF would use hop $S\rightarrow 1$ as *next hop* to node 1, with cost 3. OSPF also would take hop $S\rightarrow 2$, since it has the same total cost 3 to node 1. Let assume the number of packets at a given time in the queues is given as in Table 2. Then it follows the costs of the different paths to node 1 would appear as those shown in Table 3.

It follows that the shortest lowest cost path to node 1, in this scenario, should use $S\rightarrow 2$ as next hop.

Fig. 5 shows the three alternate paths to node 1, in terms of the number of packets in queue. It follows that the best hop is $S\rightarrow 2$.

Table 1
Cost example

	Next hop	$W(S,P)$	$d_p(S,X)$
Shortest distance from router S to node 1	Hop $S\rightarrow 1$	3	3
	Hop $S\rightarrow 2$	2	3
	Hop $S\rightarrow 3$	4	7
$d_{\mathrm{opt}}(S,1)$	Hop $S\rightarrow 1$	3	3

Table 2
Packets in queue example

Link	No. of packets in the queue
Hop $S\rightarrow 1$	3
Hop $S\rightarrow 2$	2
Hop $S\rightarrow 3$	4

Table 3
Routing table construction example

Link hop $S\rightarrow X$	No. of packets (N)	Path	Path cost $C_{\mathrm{T}}=NC_{S,X}+\sum_{j\in \mathrm{PATH}} C_{j,j+1}$
Hop $S\rightarrow 1$	3	$S\rightarrow 1$	12
Hop $S\rightarrow 2$	2	$S\rightarrow 2\rightarrow 1$	7
Hop $S\rightarrow 3$	4	$S\rightarrow 3\rightarrow 2\rightarrow 1$	23

5. Mathematical formulation of MPRA

As mentioned before, MPRA is comprised of the MPA algorithm plus the proposed cost model plus the load-balancing algorithm. The model is shown in Fig. 6, compared with OSPF.

The MPRA operation depends on the information delivered by the MPA algorithm, stored at each node in the MPA routing table, as it was explained in Section 4.3.

The mathematical formulation of the model is very simple, being as follows:

We have shown the total cost of a given path from a given source node to be

$$C_{\mathrm{T}}=K\times N_{\mathrm{q}}\times T_{\mathrm{S}}+\sum_{j=S}^{D-1} C_{j,j+1} \qquad (16)$$

We point out that all the necessary information to calculate this cost is known at the node, from the MPA routing table and the knowledge of the queue length of the output link. Hence the MPRA algorithm running at the source node should select, from all the available alternate paths to the destination node, that path k of minimum total cost $C_{T,k}$, that is:

Minimize $C_{T,k}$
s.a.

$$C_{T,k}=K\times N_{q,k}\times T_{S,k}+K\times\sum_{j\in \mathrm{MPA}_k} Ts_j \qquad (17)$$

With $k=1,\ldots,n$ and $K>0$

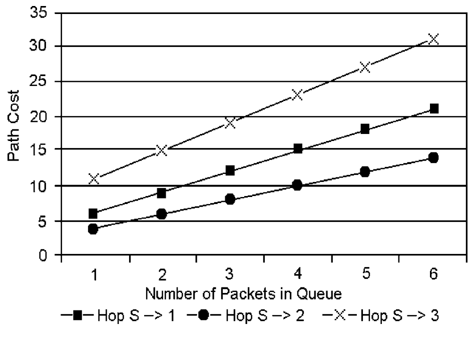

Fig. 5. Cost versus packet queue.

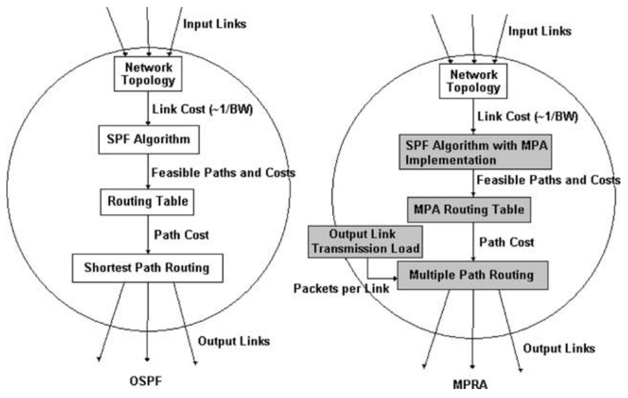

Fig. 6. Node model for MPRA.

The model is calculated for every packet destination. The pseudocode that formally describes the MPRA algorithm is as follows:

```
Hop_routing_choose_hop(packet) {
    IF (packet_arrived(packet)){
        Minimum_cost = Infinity;
        d = packet_destination(packet);
        IF (packet_is_at_destination(d)) {
            END;
        } ELSE {
            hop = read_MPA_table(d);
            WHILE (hop) {
                cost = read_route_cost_MPA_table(hop, d);
                hop_cost =
                read_hop_cost_MPA_table(hop,d);
                number_of_packet =
                read_number_of_packet_in_queue(hop);
                total_cost = hop_cost*number_of_pack-
                ets + cost;
                IF (total_cost < minimum_cost) {
                    Minimum_cost = total_cost;
                    Next_hop = hop;
                }
                hop = read_MPA_table (d);
            }
        }
    }
    RETURN (next_hop);
```

Table 4
Simulated networks

No. of nodes	No. of node pairs communicating	No. of networks
Unbalanced networks (different bandwidth on links)		
1–10	1–10	100
11–20	1–20	100
25	10	10
30	10	10
Balanced networks (same bandwidth on all links)		
1–15	1–10	150

Table 5
Balanced network results

Parameter	OSPF	OSPF with MPRA	% Variation
Throughput	387.35	468.08	20.84
Delay	56.94	47.8	−16.05
Data load	0.27	0.33	22.22
Packets dropped	240.69	212.31	−11.79
Max. delay/pack	438,339.39	454,502.88	3.7
Ave. occupied buff. spc.	4637.96	3575.66	−22.9

It is observed in the pseudocode, within the while loop, the implementation of Eq. (17). As it was pointed out in Fig. 3, MPRA performs exactly like OSPF, while moving following the optimum cost versus path curve up until point *b* is reached, where an alternative path to the destination exist but with lower incremental cost. Of course alternate paths are also carrying traffic to other destinations, so MPRA has to look through all feasible alternate paths in order to find if there exist one with lower cost.

5.1. Discussion

MPRA is aimed to obtain a good but not exact routing solution, because of its use of local information only. This was chosen that way so as to ease the MPRA implementation by using all the information OSPF is using.

6. Computer experience

The MPRA algorithm was compared with the OSPF protocol through computer simulation. The Random Topology Generator and Graphical Previewer (RTG) [13] was adapted and used to perform the network topology simulation. Furthermore, the MaRS routing testbed [14,15] was used to test networks using the OSPF protocol and then, with the appropriate modifications, to test the MPRA algorithm. Several network were simulated, as shown in Table 4.

Every network was tested 10 times to ensure the results confidence. In all 3800 networks were tested.

Table 6
Unbalanced network results

Parameter	OSPF	OSPF with MPRA	% Variation
Throughput	193.85	245.46	26.63
Delay	166.60	133.63	−19.79
Data load	0.23	0.31	33.82
Packets dropped	790.68	444.30	−43.81
Max. delay/pack	1,973,619.98	1,682,846.54	−35.08
Ave. occupied buff. spc.	5125.53	4385.75	−14.73

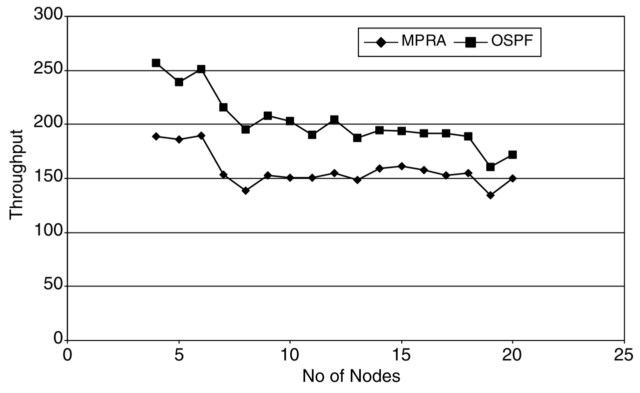

Fig. 7. Throughput versus N of nodes.

The parameters measured in the simulations were the following:

(a) *Throughput*, the number of successfully received data bytes through a time interval over the time interval considered.
(b) *Delay*, average delay for all the successfully received data packets.
(c) *Data load*, as a fraction of the network capacity.
(d) *Packets dropped*, discarded by the nodes.
(e) *Maximum Delay per Packet*, in the time interval.
(f) *Average occupied buffer space*, on the network queues.

Both balanced (that is, all links of the network have the same cost) and unbalanced networks were studied, the average results for all the networks are shown in Tables 5 and 6.

It is apparent from the results that OSPF works better with the MPRA algorithm especially with unbalanced networks, where an improvement of around 30% can be achieved. Even with balanced networks a 15–20% improvement can be reached in all parameters but the maximum delay per packet.

The following figures show other results, considering the number of nodes and number of established connections.

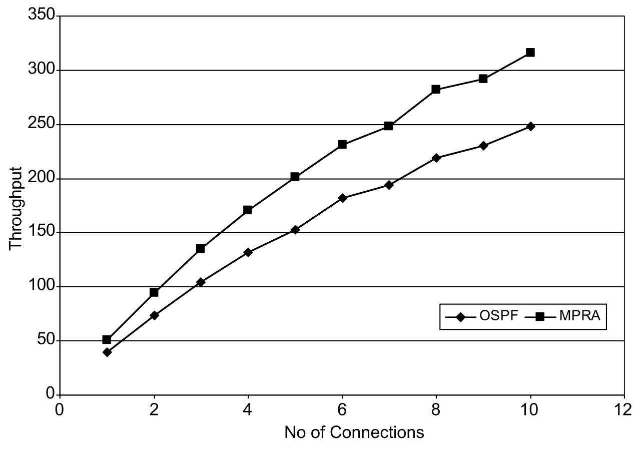

Fig. 8. Throughput over number of connections.

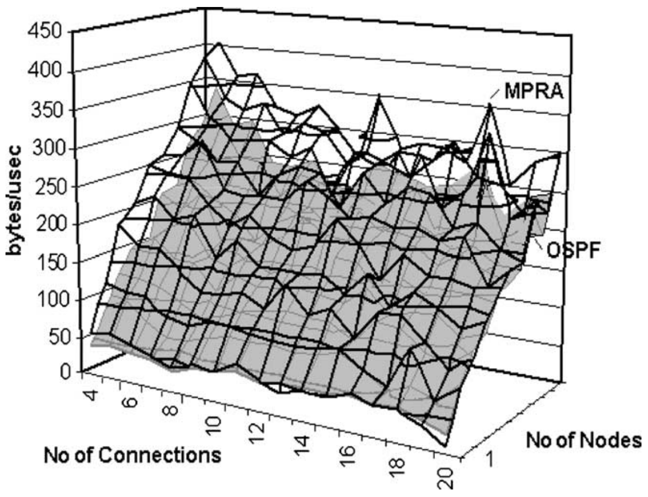

Fig. 9. Dependence of throughput on connections and nodes.

Fig. 7 shows that MPRA improvement over OSPF is almost independent on the number of nodes. Fig. 8 shows that MPRA is more efficient than OSPF when the number of connections increases.

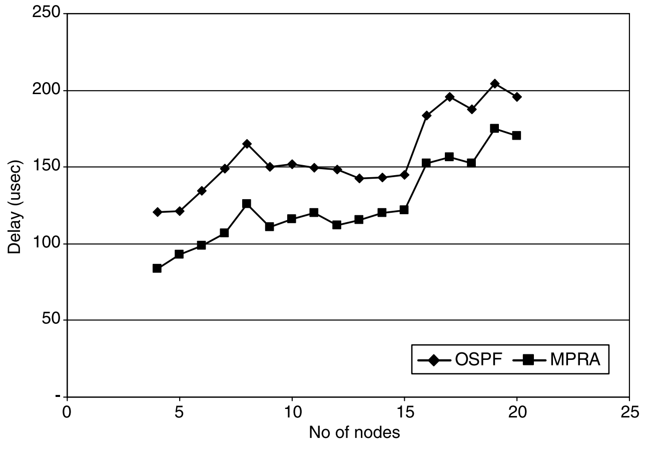

Fig. 10. Packet delay versus number of nodes.

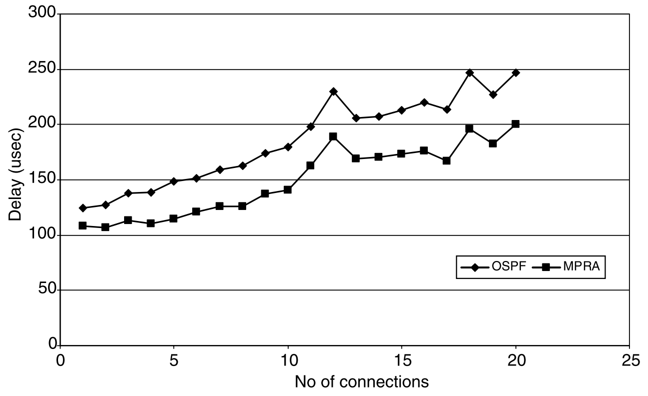

Fig. 11. Packet delay versus number of connections.

Fig. 9 shows the throughput related to the combination of nodes and connections. It is observed that in all cases MPRA works better than OSPF.

Figs. 10 and 11 show the dependence of the packet delay on the number of nodes and connections, respectively. Again MPRA provides lower delays than OSPF.

7. Conclusions

A new Multiple Path Routing Algorithm (MPRA) has been presented, which improves on the OSPF routing algorithm used in the Internet. The improvements are based on balancing the traffic load among one or several available alternate paths to the destination, and avoiding network loops.

The MPRA algorithm uses local node information to dynamically adjust to changes in the network. A cost model has been proposed that serves to efficiently choose among alternate paths.

The MPRA algorithm in simulation conditions provides throughput improvements on the order of 30% for unbalanced networks and of 20% for balanced networks.

For future work, the development of more exact models to evaluate the total path route to the destination should be addressed. MPRA uses only a cost model for the next hop output link and do not consider possible congestion problems that may present at other nodes in the network.

The MPRA routing algorithm is a simple and efficient method, which looks for a local optimum solution, and provides with better results when compared with the OSPF routing algorithm.

Acknowledgements

This research was supported in part by funds from Fondecyt (project 1040577).

References

[1] Internetworking Technology Handbook: Internet Protocols (IP). Cisco Systems, Inc., 2002.

[2] Internetworking Technology Handbook: Open Shortest Path First (OSPF). Cisco Systems, Inc., 2002.

[3] P. Narváez, K. Siu, H. Tzeng, New dynamic algorithms for shortest path tree computation, IEEE/ACM Transactions on Networking (TON) 8 (6) (2000) 734–746.

[4] P. Narváez, K. Siu, H. Tzeng, New dynamic SPT algorithm based on a ball-and-string model, IEEE/ACM Transactions on Networking (TON) 9 (6) (2001) 706–718.

[5] C. Wai-Kai, Theory of Nets: Flows in Networks, Wiley, New York, 1990.

[6] A. Kershenbaum, Telecommunications Network Design Algorithms, McGraw-Hill International, New York, 1993.

[7] J. Moy, Open Shortest Path First IGP (OSPF). http://www.ietf.org/html.charters/ospf-charter.html.

[8] Microsoft Corporation. Archives of OSPF@DISCUSS.MICROSOFT.COM. Mailing list, 1997.

[9] J. Moy, OSPF version 2. Ascend Communications, Inc. RFC 2328, April 1998.

[10] E. Dijkstra, A note two problems in connection with graphs, Numerical Mathematics 1 (1959) 269–271.

[11] L. Kleinrock, Queueing systems, Theory, vol. 1, Wiley Interscience, New York, 1970.

[12] L. Kleinrock, Queueing systems, Computer Applications, vol. 2, Wiley Interscience, New York, 1970.

[13] L. Wei, Random Topology Generator (RTG). University of Southern California, Los Angeles, CA. Available at: http://netweb.usc.edu/daniel/research/sims/topology/.

[14] C. Alaettinoglu, A. Udaya Shankar, K. Dussa-Zieger, I. Matta, Design and implementation of MaRS: a routing testbed. Technical Report, CS-TR-2964, Department of Computer Science, University of Maryland, Sept 1992, available at http://citeseer.nj.nec.com/alaettinoglu92design.html.

[15] A.U. Shankar, C. Alaettinoglu, K. Dussa-Zieger, I. Matta, Performance comparison of routing protocols under dynamic and static file transfer connections, ACM SIGCOMM Computer Communication Review 22 (5) (1992) 39–52.

[16] L.J. LeBlanc, R. Reddoch, J. Chifflet, P. Mahey, Performance of the flow-deviation algorithm for telecommunication routing with non-standard assumptions, Informs 1995; TD28.1.

[17] D. Bertsekas, R. Gallager, Data Networks, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1992.