November 2002

Improved utilisation in IP networks using multiple path routing

D. Platt
University of Wollongong, dplatt@uow.edu.au

Improved utilisation in IP networks using multiple path routing

Abstract

This paper considers routing in an IP network running a routing protocol such as OSPF It takes as a benchmark a routing scheme. which computes a single lowest cost path between every source/destination pair. It proposes two more schemes which each generate two routes for each source/destination pair. The second scheme uses the present lowest cost route and the previous lowest cost route, different from the present one. The third scheme uses the present lowest cost route and re-computes a route by removing all the links used in the first route and using the Dijkstra algorithm. A number of tests are carried out on simulated networks to determine the throughput and stability of each scheme. It is found that the third scheme produces the best results. followed by the second and then the first. It is also found that the benefit of these schemes is only worthwhile when the number of nodes in the network is low.

Keywords

telecommunication network routing, telecommunication traffic, transport protocols

Disciplines

Physical Sciences and Mathematics

Publication Details

This paper originally appeared as: Platt, D, Improved utilisation in IP networks using multiple path routing, The 8th International Conference on Communication Systems, 25-28 November 2002, vol 1, 617-622.

IMPROVED UTILISATION IN IP NETWORKS USING MULTIPLE PATH ROUTING

Don Platt

University of Wollongong, Northfields Avenue, Wollongong, NSW, Australia, Email: d.platt@uow.edu.au

Abstract - This paper considers routing in an IP network running a routing protocol such as OSPF. It takes as a benchmark a routing scheme, which computes a single lowest cost path between every source/destination pair. It proposes two more schemes which each generate two routes for each source/destination pair. The second scheme uses the present lowest cost route and the previous lowest cost route, different from the present one. The third scheme uses the present lowest cost route and re-computes a route by removing all the links used in the first route and using the Dijkstra algorithm.

A number of tests are carried out on simulated networks to determine the throughput and stability of each scheme. It is found that the third scheme produces the best results, followed by the second and then the first. It is also found that the benefit of these schemes is only worthwhile when the number of nodes in the network is low.

Keywords - IP routing, multipath, OSPF, traffic engineering

I. INTRODUCTION

Within an autonomous system, interior gateway protocols, such as OSPF, use link state algorithms, such as the Dijkstra algorithm, to determine routing paths through a network. Although OSPF allows multiple paths, the usual implementation makes use of only one path between a source/destination pair. In the literature, a number of writers have sought to improve network performance by using multiple paths. In references [1-3] the intention is to reduce delay and in [4] the improvement in reliability is cited.

This paper will examine the use of two paths between each source/destination pair, and will compare the performance of two such schemes with the standard single path scheme in utilisation of network bandwidth.

II. LOAD SHARING IN IP NETWORKS

In connectionless networks such as IP, the mechanism, which allows us to ensure that all the offered load is transported, as far as possible, is the periodic re-writing of the routing tables. We will assume that the routing tables for the nodes in the network are re-written at distinct times. The time step we will use is the time between re-writing the routing table of one node and re-writing the routing table of the next node.

When the routes for an edge node (the source node) are to be written, it is assumed that the information available is the recent average traffic through each link, the traffic from the source node to all other nodes, and those existing routes which start from the source node. It is a simple matter to re-calculate the traffic that would exist if no traffic entered the source node from outside the network. This removes the effect of the existing set of routes, and allows the calculation to proceed with all possible routes receiving equal consideration. The square of the utilisation is computed as the cost function for each link (see below), and the Dijkstra algorithm is used to determine the set of lowest cost routes from the source node to every other node.

If the traffic entering at the source node was incremental, then it could be guaranteed that the new set of routes would be optimal. However, if these flows are a considerable component of the flow down any of the links, this is no longer guaranteed. As the flows rise, they increase the costs of the links, and raise the cost of the routes. At some point, another route may become an equal cost alternative, and beyond this point, the optimal solution would be to divide any increment in the traffic between the two. It is not feasible to do this in practice, since it requires knowledge of all the flows originating at the source node, and these are still to arrive at the time the calculation is carried out.

The routing problem could be stated in traffic engineering terms, as follows. "Given the network bandwidths and connections, and the offered traffic at any time, how should that traffic be distributed through the network in order to achieve the maximum possible total transmission of traffic?" Typically, the routing solutions in use today provide a single route from one edge device to another. But the optimal solution will usually be to distribute the traffic between each pair over several routes. The computation of the optimal solution will normally be found to be extremely complicated and to scale very poorly. However, this paper will explore the benefits of using two good routes for each source/destination pair.

Multiple paths allow more traffic between two nodes. If the offered traffic between all pairs of nodes is more or less uniform, then there may be little to be gained, because the routing tables will distribute the traffic fairly evenly over the

available links. But if the offered traffic between two particular nodes is great, while traffic between all others is small, then this traffic could be routed down two parallel paths and perhaps twice as much traffic could be accepted between this pair of nodes.

The routing tables will be generated using the Dijkstra algorithm. It is necessary to assign a cost to every link in the network, and the cost used here is the square of the utilisation of each link. The traffic flowing from node i to node j, down link(i,j) is t_{ij}, and the bandwidth of the link is b_{ij}. The contribution of link(i,j), in the direction i→j, to the total cost is

$$c_{ij} = \frac{t_{ij}^2}{b_{ij}^2}.$$

Since traffic may be different in the two directions of a single link, it follows that in general, the two costs, c_{ij} and c_{ji} are not equal.

In a sense, the use of the quadratic cost function is arbitrary. However, the second power has the advantage that it increases faster as the traffic approaches the bandwidth in a link (convex function). This is a desirable characteristic since it will keep the system away from saturation under a wide set of circumstances. It is well recognised that a convex function of utilisation such as this serves well as the link cost function [3, 5].

III. MULTIPLE ROUTES

It is proposed that two routes be generated between every source/destination pair. Traffic would then be shared between them. The question of the method used to compute the second route is difficult.

In this paper, we will follow the logic that the most recent lowest cost path, different from the new lowest cost path, will still be a good path. It is already known, and so requires no further computation. Under normal conditions, we can reasonably expect that the offered load will change only relatively slowly, and so the former "best" path will still be a good path. The scheme which uses only a single route will be called "scheme 1" and the scheme which uses the two routes described here will be called "scheme 2".

It is quite likely that the greatest advantage of routing with two paths occurs when the offered load for a few source/destination pairs is much higher than the average. Then we will be able to pass perhaps twice as much traffic because we have two parallel paths. If this is the case, then we do not want the second route to contain any of the links used in the first route. The two routes will be limited by the bandwidth of the links that make them up, and so the traffic down one route would be limited by the link with the lowest bandwidth. The total flow between the pair would be limited by the sum of these values for the two routes. But if the two routes pass through the same link, then the total traffic will be limited by the bandwidth of that single link. So the total flow is likely to be lower.

It is a fairly simple matter to calculate the second path in this case. The links used in the first case are removed, and the Dijkstra algorithm is run again. This need only be done for the one or two highest flows entering at each node. Under these conditions, the scaling properties are similar to those of the Dijkstra algorithm. In this paper, this process is carried out for only the highest flow, and this scheme is called "scheme 3".

It still remains to determine the proportion of traffic that would be routed down the lowest cost route. It seems sensible to suggest that the proportion should be over 50%, and of course, it must be no more than 100%. The appendix provides the calculation of this ratio.

IV. NETWORK MODEL AND ROUTING CONDITIONS

Two networks have been used in the simulation experiments. One network has six nodes and ten links. The second network has 20 nodes and 31 links. All links were given 0.622 Gbps capacity. The simulation was written in Matlab. The model of a link allows all offered traffic to pass, until link utilisation reaches 100%. Beyond that point, the link carries its full capacity, but no more.

In one time step, the existing traffic is calculated, one node is selected and the traffic originating at that node is removed. This gives the network traffic generated by all the other nodes and the link costs are calculated, based on this traffic. The routes are now computed for this node. We are now ready to consider the traffic offered at the next time step. If there are N routers in the network, then every router will have received one new routing table in the course of N time steps.

The model assumes that the time steps are long enough for the traffic to settle down into a new pattern and that the new traffic flows down each link can be measured and communicated to the relevant nodes. The sequential writing of routing tables allows the system the best chance of reaching minimum total cost.

The three different routing schemes, described above, were used. Two different load patterns were used, and the tests were carried out on the six node network. The first pattern

established the offered traffic of four source/destination pairs only. This pattern was designed to determine the likely performance of the network when the loading conditions are uneven.

In the second pattern, equal traffic was offered between every source/destination pair. While this is unlikely to be reproduced in practice, it gives an indication of the performance of the network under more or less uniform loading.

The results show total traffic vs offered traffic. They were generated by allowing the offered traffic to increase uniformly at each time step and then computing the sum of all traffic flowing over the network. Before the results were taken, the offered traffic was allowed to vary randomly for 500 time steps. This allowed the routing algorithms opportunity to generate more than one path between any two nodes, and so there was a good chance that at least two alternate paths had been calculated by the time the results were taken. However, this was not guaranteed, particularly in a case where two nodes were linked by a single link, and any alternate route involved several links. Since this is also likely to be the case in practice, it was not regarded as a deficiency in the method.

Since the stability and integrity of the system are of interest, two other tests were carried out in simulation. The first was a large scale rise and drop in traffic load. The second was the disappearance and return of a node, while the network was under fairly heavy load.

V. RESULTS

Figure 1 shows the results when traffic is offered to only four source/destination pairs in the six-node network. The offered traffic rises linearly with time steps, and is kept equal in all cases.

It is clear that the best performance is offered by routing scheme 3. This scheme continues to pass all the traffic offered to it, until the offered traffic reaches 3.5 Mb/s. The other two schemes start to reduce the traffic flow at about 2.5 Mb/s. After this saturation effect has begun, routing scheme 2 is able to pass more traffic than the one which uses only a single route for each source/destination pair.

Figure 2 shows the results for the same network when the offered traffic between all source/destination pairs is equal.

Again, scheme 1 is able to pass the least amount of traffic. Schemes 2 and 3 offer similar performances. Saturation starts at a slightly higher level of offered traffic (about 6Gb/s vs about 5Gb/s) and these schemes are able to pass significantly higher total traffic when congestion becomes severe.

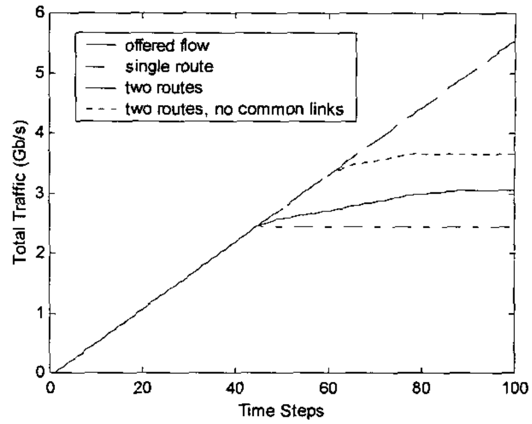

Figure 1. Traffic Offered to Four Source/Destination Pairs

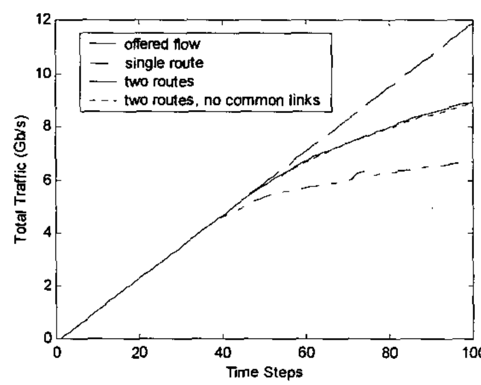

Figure 2. Traffic Offered Uniformly

Because of the random nature of the selection of the alternative route, the simulation results are not exactly repeatable. However, the results presented here are a reasonable average. Over a number of simulations, it has been found that scheme 3 passes a slightly higher level of traffic. However, the effect is not great.

The graph in Figure 3 shows the response to large step changes in the offered traffic. The network is in a steady state, with the offered traffic between all pairs of nodes set at 0.149 Gb/s. At the 20th time step, the offered load between four pairs

of nodes is increased to 0.8 Gb/s, and is returned to its original value at the 60th time step.

Before the load step, the network is able to carry all the offered traffic, under all three routing schemes. After the step up, the responses, in all three cases, take about 10 time steps to settle down into a new steady state, although scheme 3 still shows some oscillation. After the load steps down again, all three schemes return to their original steady state in one time step. As we might expect, scheme 3 is able to carry the most traffic, followed by scheme 2.

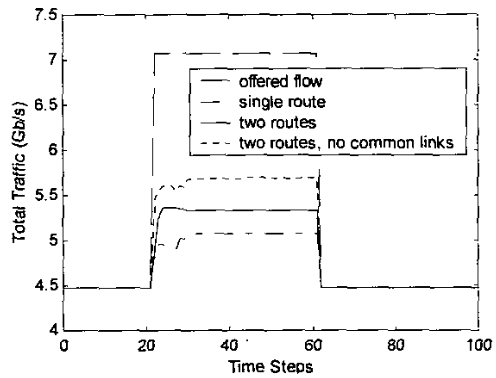

Figure 3. Response to Step Changes in Offered Load

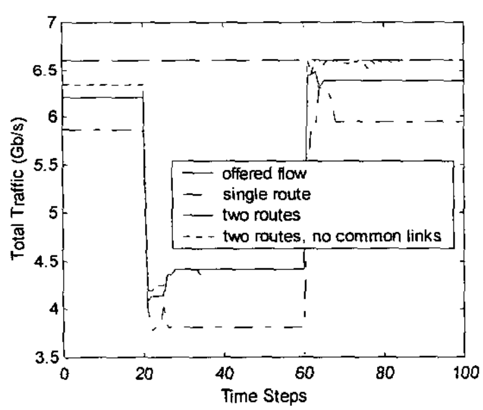

Figure 4. Loss and Restoration of One Node

The situation was simulated in which one node became non-functional, while the network was operating at near maximum total load. The result is shown in Figure 4, which shows the network in a steady state for 20 time steps. The offered load is uniform between each source/destination pair. Under each routing scheme, the network is able to carry nearly the full offered load. At time step 20, one node becomes non-functional, but the offered load remains the same. At time step 60, the node is restored.

In all cases, the system remains stable, although it requires up to about ten time steps to settle down. This is not surprising, since six time steps are necessary for the recalculation of the routing tables for all nodes in the network. It is interesting that the traffic is restored to a level that is a little greater than it was before the interruption. This is true, even for the single route scheme. This illustrates the point that none of these schemes guarantee an optimal solution for the whole system, although, in practice, the solutions generated are quite similar in performance.

It is interesting to see how the schemes perform when applied to a larger network, and the uneven load test is applied to a network with 20 nodes and 31 links. It is to be expected that schemes 2 and 3 will allow more traffic than scheme 1 when the offered load is very uneven (see Figure 5).

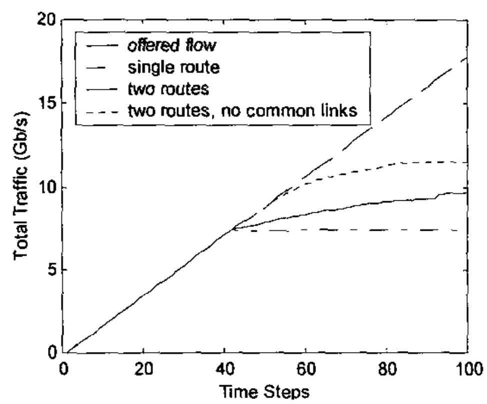

Figure 5. Traffic Offered to 12 Source/Destination Pairs in 20 Node Network

Clearly, in the case of scheme 1, each path has reached the bandwidth limit of all links (0.622 Gb/s) at the same point, and no more traffic can be passed. Scheme 3 ensures that the traffic is separated as much as possible, and is able to carry about 30% more traffic before running into limits imposed by

bandwidth. Scheme 2 offers performance between the other two.

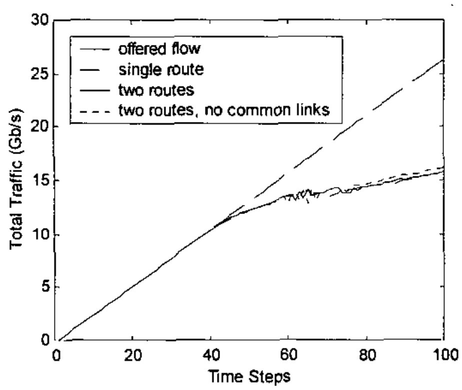

Figure 6. Twenty Node Network, Uniform Load.

In Figure 6 we see the result of an offered load that is uniform between each source/destination pair in the 20 node network, but steadily increasing. We see that there is little difference in the performance of the network under the three different schemes. Saturation of the network begins at an offered load of about 12 Gb/s

VI. COMMENTS AND CONCLUSIONS

This paper has examined the problem of routing in connectionless networks, using only the square of the most recent value of utilisation as the cost function for a link. Each router re-writes its routing tables at distinct times, with knowledge of the traffic on each link of the network. The smallest unit of time, the time step, is the time between the re-writes of successive routers. It has been assumed that the burstiness and buffer control phenomena can be ignored in this time scale, that a link can be fully modeled by its bandwidth, and flow can be specified by its average value over one time step.

Three routing schemes have been investigated, two of which generated two routes for each source/destination pair. The other scheme generated only a single route.

It was found that all three schemes were stable under a variety of conditions, including sudden large changes of load and the loss and restoration of a node. These tests were carried out with the network near its maximum capacity.

A number of tests were carried out where the load was increased steadily to find at what levels the various schemes reached the limit of traffic throughput. These tests were carried out on both a six-node and a twenty-node network. Although they are a little more complex, schemes 2 and 3 offer significantly greater throughput, when the load is uneven. These schemes also offer greater throughput when the load is uniform in the case of the six-node network. All three schemes offer about the same performance in the case of uniform load in the twenty-node network.

All three schemes were well behaved when links reached saturation. In the case of scheme 1, under some conditions, saturation was hard, but the other two schemes offered a softer saturation, with higher throughput available when the load was increased. All three schemes remained stable.

The scaling properties of the three schemes are similar to those of the standard Dijkstra algorithm, and should not give any cause for concern. However, the greatest advantage of this work will be in the use of schemes 2 and 3 to make it possible to carry more traffic in small networks.

REFERENCES

1. Vutukury, S and Garcia-Luna-Aceves, J.J, 'An algorithm for multipath computation using distance-vectors with predecessor information' Proc Eighth Int Conf on Computer Communications and Networks, Oct 1999, pp 534-539

2. Vutukury, S and Garcia-Luna-Aceves, J.J, 'A distributed algorithm for multipath computation', GLOBECOM, Dec 1999, vol 3, pp 1689-1693

3. Vutukury, S and Garcia-Luna-Aceves, J.J, 'A traffic engineering approach based on minimum delay routing', Proc Ninth Int Conf on Computer Communications and Networks, Oct 2000, pp 42-47

4. Jian Pu, Manning, E. Shoja, G.C. 'Routing reliability of partially disjoint paths', Proc IEEE Pacific Rim Conf on Communications, Computers and Signal Processing, Aug 2001, vol 1, pp 79-82

5. Fortz, B. and Thorup, M. 'Internet traffic engineering by optimising OSPF wieghts', Proc INFOCOM, March 2000, vol 2, pp 519-528

Appendix

Calculation of traffic splitting ratio, a_{mn}

One route between nodes m and n passes through p links with traffic flows $g_1+a_{mn}f_{mn}$, $g_2+a_{mn}f_{mn}$, ...$g_p+a_{mn}f_{mn}$. (The flow, g_1, is equal to $t_{mk} - a_{mn}f_{mn}$ where the first link is from node m to node k, etc.) The other route flows through q links with traffic

flows $h_1+(1-a_{mn})f_{mn}$, $h_2+(1-a_{mn})f_{mn}$, … $h_q+(1-a_{mn})f_{mn}$. (The flow, h_1, is equal to $t_{ml}-(1-a_{mn})f_{mn}$ where the first link is from node m to node l, etc.)

The bandwidth in the p links of the first route are b_1, b_2, …b_p, and the bandwidths in the q links of the second route are c_1, c_2, …c_q. The contribution that these links make to the total cost function is

$$\Delta C = (g_1^2 + 2g_1a_{mn}f_{mn} + a_{mn}^2f_{mn}^2)/b_1^2 + (g_2^2 + 2g_2a_{mn}f_{mn} + a_{mn}^2f_{mn}^2)/b_2^2 + \ldots + (g_p^2 + 2g_pa_{mn}f_{mn} + a_{mn}^2f_{mn}^2)/b_p^2 + (h_1^2 + 2h_1f_{mn} - 2h_1a_{mn}f_{mn} + f_{mn}^2 - 2a_{mn}f_{mn}^2 + a_{mn}^2f_{mn}^2)/c_1^2 + (h_2^2 + 2h_2f_{mn} - 2h_2a_{mn}f_{mn} + f_{mn}^2 - 2a_{mn}f_{mn}^2 + a_{mn}^2f_{mn}^2)/c_2^2 + \ldots + (h_q^2 + 2h_qf_{mn} - 2h_qa_{mn}f_{mn} + f_{mn}^2 - 2a_{mn}f_{mn}^2 + a_{mn}^2f_{mn}^2)/c_q^2$$

We select a_{mn} by setting $d(\Delta C)/da_{mn} = 0$

That is:

$$(g_1 + a_{mn}f_{mn})/b_1^2 + (g_2 + a_{mn}f_{mn})/b_2^2 + \ldots + (g_p + a_{mn}f_{mn})/b_p^2 + (-h_1 - f_{mn} + a_{mn}f_{mn})/c_1^2 + (-h_2 - f_{mn} + a_{mn}f_{mn})/c_2^2 + \ldots + (-h_q - f_{mn} + a_{mn}f_{mn})/c_q^2 = 0$$

That is:

$$a_{mn}f_{mn}\left(\sum_1^p \frac{1}{b_n^2} + \sum_1^q \frac{1}{c_n^2}\right) = f_{mn}\sum_1^q \frac{1}{c_n^2} + \sum_n^q \frac{h_n}{c_n^2} - \sum_1^p \frac{g_n}{b_n^2}$$

Or

$$a_{mn} = \frac{f_{mn}\sum_1^q \frac{1}{c_n^2} + \sum_1^q \frac{h_n}{c_n^2} - \sum_1^p \frac{g_n}{b_n^2}}{f_{mn}\left(\sum_1^p \frac{1}{b_n^2} + \sum_1^q \frac{1}{c_n^2}\right)}$$

If all bandwidths are equal, this formula reduces to the following.

$$a_{mn} = \frac{q * f_{mn} + h_1 + h_2 + \ldots + h_q - g_1 - g_2 - \ldots - g_p}{(p+q)f_{mn}}$$

We must place limits on the value of the a_{mn} so that $0.5 < a_{mn} < 1$. If the calculations above result in something outside the range, then we must put a_{mn} at the appropriate limit.