ISSN:0975-9646

Congestion Minimization By Using MFMP Routing Algorithm

T.B.Sivakumar, M.Gughan Raja, B.Palanichelvam
CSE,Syed Ammal Engineering College, Ramanathapuram

Abstract - **Unlike traditional routing schemes that route all traffic along a single path, multipath routing strategies split the traffic among several paths in order to ease congestion. It has been widely recognized that multipath routing can be fundamentally more efficient than the traditional approach of routing along single paths. Yet, in contrast to the single-path routing approach, most studies in the context of multipath routing focused on heuristic methods. We formalize problem (K-Path Routing) that incorporate major requirements of multipath routing. Then, we establish the intractability of these problems a max flow multipath routing algorithm that is designed to reduce latency, provide high throughput and balance traffic load. The max flow multipath algorithm is based on a Ford-Fulkerson algorithm. It consists of determining a set of disjoint paths that are loop free with maximum flow, then splitting network traffic among those paths on a round robin fashion. Through simulation we show that our algorithm performs well than a multi shortest path in terms of computational complexity. Finally establish efficient solutions with proven performance guarantees.**

1. INTRODUCTION

CURRENT routing schemes typically focus on discovering a single “optimal” path for routing. Multipath routing is an alternative approach that distributes the traffic among several “good” paths instead of routing all traffic along a single “best” path. Multipath routing can be fundamentally more efficient than the currently used single-path routing protocols. thus, improving network utilization and providing load balancing [1]. Multipath routing algorithms that both select the routing paths and split traffic among them.

2. EXISTING SYSTEM

Multipath routing in the previous context have focused on heuristic methods. A multipath routing scheme, termed equal cost multipath (ECMP), has been proposed for balancing the load along multiple shortest paths using a simple round-robin distribution. By limiting itself to shortest paths, ECMP considerably reduces the load balancing capabilities of multipath routing; moreover, the equal partition of flows along the (shortest) paths (resulting from the round robin distribution) further limits the ability to decrease congestion through load balancing. OSPF-OMP (open shortest path first optimized multi-path) [2] allows splitting traffic among paths unevenly; however, the traffic distribution mechanism is based on a heuristic scheme that often results in an inefficient flow distribution. They focused on heuristics and did not consider the quality of the selected paths. Accordingly, investigate multipath routing adopting an accurate approach, and formulate it as an optimization problem of minimizing network congestion.

3. MODEL AND PROBLEM FORMULATION

A network is represented by a connected directed graph , $G(V,E)$ where V is the set of nodes and E is the set of links. We proceed to formulate the criterion for congestion. Given a network $G(V,E)$ and a link flow $\{f_e\}$, the value $\frac{f_e}{c_e}$ is the link congestion factor and the value $\max_{e\in E}\left\{\frac{f_e}{c_e}\right\}$ is the network congestion factor. In [4], we show that the problem of minimizing the network congestion factor is equivalent to the well-known maximum flow problem [1]. Hence, when there are no restrictions on the paths (in terms of the number of paths or the length of each path), one can find a path flow that minimizes the network congestion factor in polynomial time through a standard max-flow algorithm (using Ford and Fulkerson's algorithm).

4. MINIMIZING CONGESTION WITH K ROUTING PATHS

We investigate Problem KPR, which minimizes congestion while routing traffic along at most K different paths. we prove that Problem KPR is NP-hard in the general case but admits a (straightforward) polynomial solution when the restriction on the number of paths is larger than the number of links (i.e., $K \geq M$) .

We reduce Problem KPR to the *single-source unsplittable flow problem* that was shown to be NP hard in [4] and is defined as follows: given is a network $G(V,E)$, a capacity $c_e > 0$ for each link $e \in E$, a set of source-destination pairs $(s,t1),(s,t2),...,(s,t_k)$ associated with demands is there an assignment of traffic to paths such that for each $1 \leq i \leq k$ the demand γ_i is routed over a single path $p \in P^{(s,t_i)}$ without violating the capacity constraints?

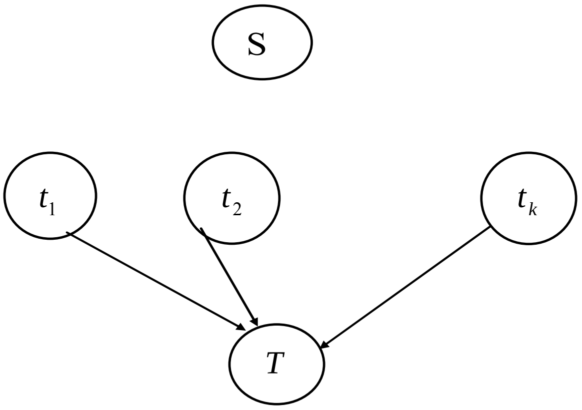

Fig.1. Reducing the single-source unsplittable flow problem into Problem KPR

The single source unsplittable flow problem is transformed to Problem KPR as follows (see Fig1). Add an “aggregated” target T . Then, for each k add a link $t_i \rightarrow T$ with a capacity γ_i .

Thus, Problem KPR is NP-hard.

In the general case, it admits a polynomial solution when the restriction on the number of paths is larger than the number of links (i.e., $K \geq M$). It is possible to obtain a flow that minimizes the network congestion factor with a single execution of a max-flow algorithm. Moreover, using the Max flow multipath routing [3], it is possible to transform in polynomial time *every* link flow representation into a path flow representation that admits at most M routing paths. Therefore, with a single execution of a max-flow algorithm followed by a single execution of the Max flow multipath routing algorithm, it is possible to solve Problem KPR in polynomial time in the case $K \geq M$.

4.1 Ford-Fulkerson Algorithm

Suppose *G (V, E)* is a finite directed graph and every edge *(u, v)* has a capacity *c(u, v)* (a non-negative real number). Further assume two vertices, the source *s* and the sink *t*, have been distinguished. A cut is a split of the nodes into two sets *S* and *T*, such that *s* is in *S* and *t* is in *T*. Hence there are possible cuts in a graph. The capacity of a cut *(S, T)* is:

The sum of the capacity of all the edges crossing the cut, from the region *S* to the region *T*.

The following three conditions are equivalent:

- *f* is a maximum flow in *G*
- The residual network *Gf* contains no augmenting paths.
- | *f* | = *c(S, T)* for some *cut (S, T).*

An augmenting path is an alternating sequence of vertices and edges of the form *s, e1, v1, e2, v2, …, ek, t* in which no vertex is epeated and no forward edge is saturated and no backward edge is free.

4.2 MFMR Implementation

Implementation of our Max flow multipath routing (MFMP) algorithm has been done in two steps:

Step 1: We determined paths that maximize the flow.

Step 2: We distribute flows through a set of paths in a round robin fashion.

To find a multipath that can be used in splitting traffic we used a breadth-first search (BFS) that finds the shortest augmenting path from the source to the sink. We basically: find a shortest path from the source to the sink and compute the minimum capacity of an edge (that could be a forward or a backward edge) along the path - the path capacity. Then, for each edge along the path we reduce its capacity and increase the capacity of the reversed edge with the path capacity.

5- SIMULATION ENVIRONMENT

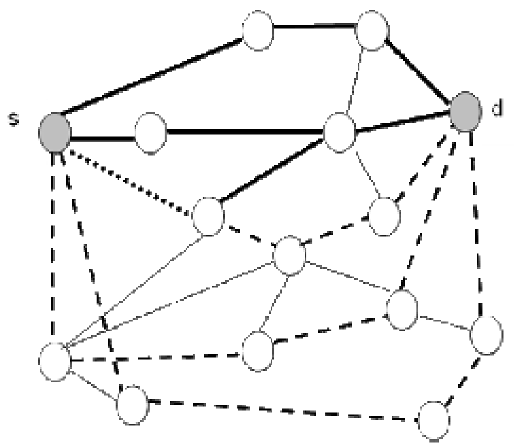

Fig 2 - Topology

For test environment, the network topology of Figure 2 is used in our study. It consists of 13 nodes (routers), one bursty source and one sink (destination) and 25 bidirectional links, of which 20 Wide Area Network (WAN) an links with a capacity of 1Mbps and a delay of 5ms, and 5 Local Area Network (LAN) links with a capacity of 100Mbps and a delay of 0.01 ms. We assume all links have equal cost of 1.

For simplicity, we assume that there is no link failure during simulation. The Max flow algorithm routes multipath traffic at the packet level routing. Routing at the flow level could be carried out by means of hash function, but is not considered here. Paths are determined in the initialization phase and are stored in a routing table. MSP algorithm uses Djikstra’s algorithm to determine paths, MFMP uses Ford Fulkerson algorithm. The solid links in represents paths used by MSP, the dotted lines are the extra paths used by a MFMP algorithm, and the small dotted line link is the link that is used by both MFMP and MSP. Flows arriving into the network are assumed to require one unit of bandwidth. Simulated traffic corresponds to a TCP packets. Flows arrive at a source node according to a Poisson process. The MSP uses only the shortest path from source to destination. The MFMP use only paths that maximize the flow without restricting their number as long as they offer maximum flow. For traffic load we used a bursty source that emulates TCP traffic. The transmission details are produced according to the distribution presented in Table 1. Burst Time Truncnormal distribution Sleep Time Exponential distribution Inter-Arrival Time Poisson distribution Table 1.

Burst Time	Truncnormal distribution
Sleep Time	Exponential distribution
Inter-Arrival Time	Poisson distribution

Table 1

TCP packets are of size equal to 512 bytes that correspond to a VoIP packet [6]. We run our simulation under different traffic scenarios varied from heavy traffic load (long inter arrival time) to light traffic (short inter arrival time) load by varying the inter arrival of packets, which correspond to _ parameter in the Poisson formula:
P {interarrival time > t} =e −λt.
Duration of generating packets is about truncnormal (0.2s, 0.2s), then the bursty source goes for sleeping for exponential (0.5s), then start another burst. Queue capacity for each outgoing link is about 100 packets. Each router in the topology uses MFMR at first to determine path(s) to each other node in the topology, then it stores the results in the routing table, then we run MSP. The simulation was run for 60 simulated seconds. The metrics of interest evaluated are

Packet delivery Ratio percentage: The number of packets received by the destination divided by the number of packets originated by the source.

Mean End to End delay: Includes all possible delay incurred to packets from the time the source attempts to send a packet to the time the packet arrives at the destination.

Packet loss percentage: Is the percentage of the number of packets lost.

The performance of MFMP routing protocol is compared using simulation with that of MSP. The simulations are carried out using OMNET simulator [5] on the network topology shown on Figure 2. A router can be seen from inside as shown in Figure3

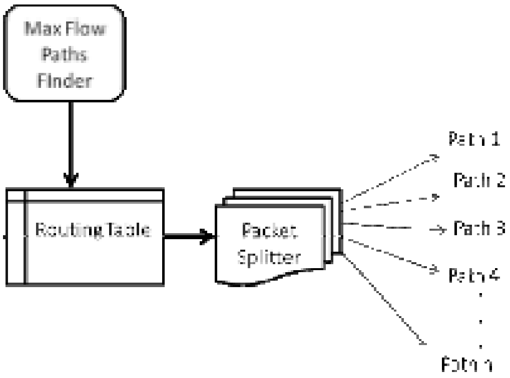

Fig 3 – Router model.

6- SIMULATION RESULTS AND ANALYSIS

In order to check the efficiency of our algorithm, we run a simulation under TCP traffic; MFMP and MSP use the same simulation parameters. As we can see clearly from a graph of mean end-to-end delay on Figure 4,
TCP Traffic:

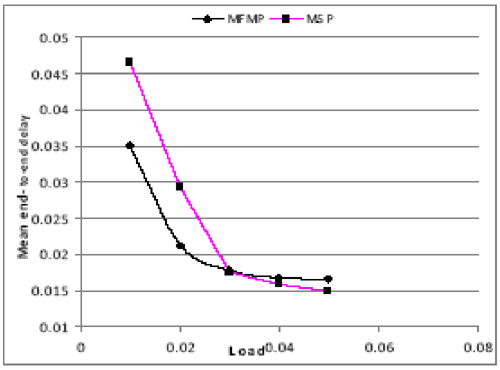

Fig 4. Mean End – to – End Delay.

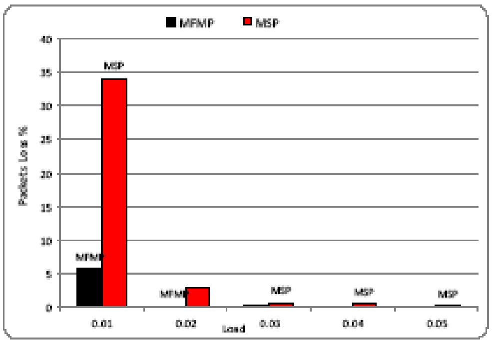

Fig 5. Packet Loss %.

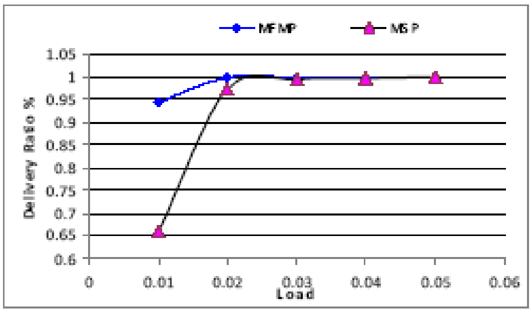

Fig 6. Delivery Ratio.

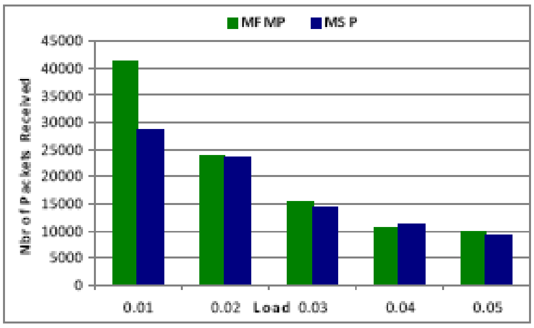

Fig 7. Nbr of Packets Received

The max flow multipath (MFMP) algorithm performs well than a multi shortest path (MSP) during heavy load traffic. MFMP is able to benefit from the number of alternative paths and, as result have less packets waiting on queue, thus less loss packet percentage (Figure 5) and higher delivery ratio (Figure 6). However, this difference in performance decreases when the traffic became lighter (less load), the MFMP and MSP have only a slight difference in the mean end-to-end-delay and in the number of packets loss as it can be seen from Figure 3 and Figure 4 respectively .This can be explained that, during light traffic there is no congestion on the network and the MSP benefits of forwarding packet through shortest paths while MFMP forward packets on both shortest and longer paths. For the number of packets received by the destination, the difference is clear. It is shown clearly from Figure 6, the destination in MFMP, has received more packets than MSP under heavy load.

CONCLUSION

Previous multipath routing schemes for congestion avoidance focused on heuristic methods. We investigated multipath routing as an optimization problem of minimizing network congestion. We have established a *polynomial time* algorithm that approximates the optimal solution by a (small) *constant* approximation factor. The solution to Problem KPR is established by restricting the flow invokes a set of successive computations of a max-flow algorithm, its distributed implementation is straightforward due to [6] that provide distributed implementations for max-flow algorithms. The distributed implementation of Algorithm RMP remains an open issue for future investigation. Finally, as discussed in [7], multipath routing offers a rich ground for research also in other contexts, such as survivability, recovery, network security, and energy efficiency. We are currently working on these issues and have obtained several results regarding survivability.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, Network Flows: Theory,Algorithm, and Applications. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[2] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," IETF RFC 2702, Sep. 1999.

[3] M. R. Garey and D. S. Johnson, Computers and Intractability. San Francisco, CA: Freeman, 1979.

[4] J. Kleinberg, "Single-source unsplittable flow," in Proc. 37th IEEE *FOCS*, 1996.

[5] www.omnetpp.org (last visited 2008).

[6] G. Malkin, "RIP version 2 protocol analysis," IETF Internet RFC 1721, November 1994.

[7] H. Suzuki and F. A. Tobagi, "Fast bandwidth reservation scheme with multi-link and multi-path routing in ATM networks," in: IEEE INFOCOM'92, Piscataway, USA,

[8] R. Banner and A. Orda, "Multipath routing algorithms for congestion minimization," Dept. Electr. Eng., Technion, Haifa, Israel, CCIT Rep. 429, (2005). [Online]. Available: http://www.ee.technion.ac.il/people/ron/Congestion.pdf.