A Comparative Study between Max Flow Multipath, Multi Shortest Path and Single Shortest Path

Ahmed Redha Mahlous  
School of Informatics, University of Bradford, United Kingdom  
ARMahlou@bradford.ac.uk

Rod J Fretwell  
School of Informatics, University of Bradford, United Kingdom  
R.J.Fretwell@Bradford.ac.uk

Brahim Chaourar  
College of Science, Al Imam University, Riyadh, Saudi Arabia  
bchaourar@hotmail.com

Abstract  
In this paper we present a max flow multipath routing algorithm (MFMP) 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 (MSP) and a single shortest path (SSP).

Keywords: multipath, shortest path, Maximum flow, Disjoint, bandwidth, delay, and throughput.

I. INTRODUCTION

The current internet is based on a single shortest path routing which can lead to unbalanced traffic distribution due to congestion of frequently used shortest path. Some single shortest path that is used in the internet includes RIP [4].

To alleviate the problem of congestion that depends on a single shortest path, a multipath routing is proposed as an alternative to single shortest path to take advantage of network redundancy, distribute load [5], improve packet delivery reliability [9], ease congestion on a network[14],[11], improve robustness[1],[16], increase network security [17]and address QoS issues.

To prove the efficiency of our algorithm, we compared it to ECMP [8].

For the best of our knowledge the problem of max flow multipath routing has not been studied before.

The rest of this paper is organized as follows. Section 2 reviews some multipath routing protocols. Section 3 and 4 define the implementation of our algorithm and simulation environment, results and analyses are given in section 5; finally section 6 concludes the paper.

II. RELATED WORK

Several multipath schemes have been proposed in literature. In context of theoretical approach, a Multipath routing problem has been extensively studied as an optimization problem. As multi objective optimization of a network flows, using total link utilization and a bandwidth fairness as objectives [21], have given a solution by using a non linear programming. They showed that an improvement of bandwidth can be achieved by a small decline of utilization.

In [12], a multipath routing is formulated as an optimization problem of minimizing network congestion. The authors have proposed a polynomial time algorithm that approximates the optimal solution for restricting a number of paths to be chosen and their quality which is based on a length of a path. Through simulation they showed the efficiency of optimizing a congestion reduction.

In [2], a multipath scheduling algorithm called (OMS) is introduced. It splits traffic over high quality paths that have low-delay high throughput while respecting the routing weights. Through simulation and analytical method they showed an improvement of throughput of multipath TCP flows.

To improve a quality of service (QoS) routing, many technologies have been used, from routing to queuing technology. In [13], an early distributed multipath algorithm was developed, that obtains minimum average delay when traffic and network topology is not changing.
To reduce a jitter of packet delay, a class-based-queuing (CBO) [19] was developed that creates multiple queues for one outgoing interface that make specific packets get ahead of other already in queue.

For minimizing a delay, a discount shortest path algorithm (DSPA) is proposed in [6]. It takes into account the path quantity and path independence and proposes paths that are compromises of Shortest K and link disjoint algorithms.

A class-based multipath routing is proposed for a real time application [20]. By separating a UDP real-time application from other non real-time and forwarding them over a multipath, the algorithm achieved a minimum end-to-end delay.

In [15], authors proposed a distance vector algorithm named MDVA that uses a set of loop free multipath which overcome the problem of count-to-infinity found in a distributed Bellman-Ford algorithm. Through simulation and comparison with other routing protocol, they demonstrated the rapid convergence of MDVA algorithm.

The selection of good paths is also found in [18]. A widest disjoint path (wdp) algorithm is proposed. The algorithm chooses a number of paths described as a good path based on blocking probability, and then the traffic is proportioned among those paths. Through simulation and comparison with other algorithms, the wdp offered a maximum throughput and a minimum blocking probability.

Reserving resources along the paths used in routing were studied in [7]. The authors discouraged using more than two or three routes in parallel in order to preserve bandwidth along several paths.

Balancing load across network using multipath was studied in [8]. It proposes an Equal Cost Multipath (ECMP) algorithm that splits load equally through many paths using a round robin distribution.

III. MFMR IMPLEMENTATION

The Ford-Fulkerson algorithm [10] finds the maximal flow from a given source to a given destination in a network, subject to the capacities of the links.

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

**Step 1:** We determined paths that maximize a 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.

IV. SIMULATION ENVIRONMENT

For test environment, the network topology of Fig 1 is used in our study. It consists of 14 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 and SSP algorithm use Dijkstra’s algorithm to determine paths, MFMP uses Ford Fulkerson algorithm [10]. The solid link in Fig 1 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.

![Fig 1. Topology](image-url)
Table 1

<table>
<thead>
<tr>
<th>Burst Time</th>
<th>Truncnormal distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep Time</td>
<td>Exponential distribution</td>
</tr>
<tr>
<td>Inter-Arrival Time</td>
<td>Poisson distribution</td>
</tr>
</tbody>
</table>

The red line are the path used by SSP flows arriving into the network are assumed to require one unit of bandwidth.

Simulated traffic corresponds to a TCP and UDP 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 a flow without restricting their number as long as they offer maximum flow.

For traffic load we used a bursty source that emulates TCP and UDP traffic. The transmission details are produced according to the distribution presented in Table 1.

A TCP packets are of size equal to 512 bytes, the UDP packet are of size 188 bytes that correspond to a VoIP packets (8 byte UDP header + 20 byte IP header + 160 byte voice data) [3]. We run our simulation under different traffic scenarios varied from heavy traffic load to light traffic load by varying the inter arrival of packets (which correspond to varying the lambda parameter in a Poisson distribution : \( \lambda = 0.01..0.05 \)).

Duration of generating a packets is about trunccnormal (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 use 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 and SSP. 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 [22] on the network topology shown on Fig 1. A router can be seen from inside as shown in Fig 2.

V. SIMULATION RESULTS AND ANALYSIS

In order to check the efficiency of our algorithm, we run a simulation under two scenarios. One for TCP traffic and another for UDP traffic. MFMP, SSP and MSP use the same simulation parameters.

**Scenario 1: TCP traffic**

![Fig 3. Mean End-to-End Delay (TCP)](image)

![Fig 4. Packet (TCP) Loss %](image)
As we can see clearly from a graph of Mean end-to-end delay on Fig 3, the max flow multipath (MFMP) algorithm performs well than a multi shortest path (MSP) during heavy load traffic (corresponding to poisson distribution $\lambda = 0.01..0.03$). This performance is due to that when some paths in a multipath are congested, the max flow uses other alternative paths, that can be with same length as paths in MSP or longer, that maximize a flow. So 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 (Fig 4) and higher delivery ratio (Fig 5). However, this difference in performance decreases when the traffic became lighter (less load $\lambda = 0.03..0.05$), the MFMP and MSP have nearly the same mean end to end delay. We can see also from Fig 4 and 5 respectively, that, MFMP has low loss packets and high delivery ratio especially during heavy load, while the load decrease on the link, both, MFMP and MSP tend to be similar. 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 Fig 6, the destination in MFMP, has received more packets than MSP under heavy load. This big difference is due to the fact that the MFMP algorithm uses more paths than MSP when this latter is congested. For the SSP the difference is clear, MFMP and MSP performs well than SSP because, this later uses only one shortest path and this one has a high congestion during heavy load, which explain the high number of packet loss and high mean end to end delay.

**Scenario 2: UDP traffic**

To evaluate the performance of max flow multipath algorithm, we also simulated UDP packets as stated in section 6. We can see clearly that MFMP has high delivery ratio (Fig 8) over MSP and SSP and high number of packets delivery (Fig 10) with less loss packet percentage (Fig 9), this can be explained that MFMP benefits of having more paths than MSP, thus less congestion on its paths. However, MSP has a better mean end to end delay (Fig 7) than MFMP; this is due to the size of packets and a capacity of links. MFMP treats more packets than MSP, so while it is busy of forwarding packets over shorter and longer path, MSP forwards packets quickly using only shortest paths, thus having a slight less mean end to end delay. Despite this difference MFMP still performs better than MSP for UDP packet, because in VoIP, we can accept to have a low delay than to have loss packets. Single shortest path still performs worse than MFMP and MSP for the same reasons mentioned in TCP scenario.
VI. CONCLUSION

In this paper, we have proposed a maximum flow multipath algorithm that 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. We have shown through simulation the effectiveness of our proposed routing algorithm (MFMP) over multi shortest path (MSP) and single shortest path (SSP). As future work, we will investigate the performance of MFMP in a weighted network topology.

REFERENCES


