International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012

DOI: 10.7321/jscse.v2.n6.4

# Proposing a Load Balancing Algorithm with the Help of an Endpoint Admission Control Algorithm to Improve Traffic Engineering

*[1]Zahra Vali, [2]Massoud Reza Hashemi, +[3]Neda Moghim
[1,] *Isfahan University of Technology, Zahra.vali@gmail.com*
[2,] *Isfahan University of Technology, hashemim@cc.iut.ac.ir*
[3,] *University of Isfahan, n.moghim@eng.ui.ac.ir*
+ Affiliation Modified on Oct 9, 2012

***Abstract.*** **The focus of this paper is to achieve a dynamic load balancing algorithm with the ability of guaranteeing the end-to-end quality of service (QoS) for a variety of service classes. The proposed algorithm consists of an explicit endpoint admission control (EEAC) mechanism, multiple path algorithm (MPA) as a multipath routing protocol and an adaptive load balancing algorithm. EEAC algorithm is composed of two phases: probing phase and data transfer phase. Information in the probing phase of EEAC algorithm such as buffer length and arrival traffic rate for each class of service is used to obtain a good estimation of network congestion state for efficient load balancing among multiple paths. The simulation results show that the proposed algorithm increases the utilization of network resources and also decreases the end-to-end delay of each path.**

**Keywords**: *multi-path routing, load balancing, end-point admission control, QoS*

## 1. Introduction

In the past, Internet was a communication network used for requests such as file transferring without any QoS requirements. However Next Generation Networks (NGN) will be packet networks able to provide communication services with high bandwidth requirements and guaranteed QoS [1]. The goal of Next Generation Networks is to provide a variety of service classes with different QoS guarantees, preserving the compatibility with the current IP networks. Despite of providing different QoS requirements, how to efficiently use all network resources and make a lower end-to-end delay for each traffic flow still need more consideration.

In order to manage the QoS requirements of the traffics and utilize network resources, one way is to use traffic engineering (TE) approaches. Optimized routing is a key mechanism in traffic engineering that improves network performance through finding efficient routes. Researches about engineered routing approaches are classified into eight groups according to four distinct criteria [2]:

1) Inradomain TE vs. interdomain TE according to traffic optimization viewpoint
2) MPLS-based TE vs. IP-based TE according to routing methods
3) Offline TE vs. online TE according to availability of traffic request or timescale of operation
4) Unicast TE vs. multicast TE according to the number of ingress-egress pairs.

According to the above classification, the proposed algorithm of this article is classified as an online intradomain IP-based unicast TE approach. The main objective of the proposed algorithm is to provide different QoS requirement while achieving higher network utilization by using an appropriate routing mechanism. Compared to single path routing, multipath routing algorithms cooperating with load balancing algorithms, provide higher bandwidth efficiency and also lower congestion probability in the network. However, how to guarantee the required end-to-end delay and simultaneously provide different service classes among multiple paths is still an open research area.

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

In order to guarantee the end-to-end QoS requirements for all service classes in the proposed algorithm, a call admission control algorithm helps differentiated service (DiffServ) to accept traffic demands and simultaneously a load balancing algorithm divides the accepted traffic among multiple paths for higher network utilization. The importance of the combination of endpoint admission control algorithm and the proposed load balancing technique is that, the end point admission control provides the required information of network states dynamically for load balancing algorithm.

The rest of this paper is organized as follows. A brief review of different call admission control mechanisms and also a classification of load balancing algorithms are explained in the remainder of this section. The proposed algorithm is introduced in section 2. Simulation scenarios and network topology is stated in section 3. Finally the paper is concluded in section 4.

## 1.1. Call admission control algorithms

A call admission control algorithm manages the arrival of new traffic streams into the network by checking the availability of resources according to the QoS requirements of the new streams. If the available resources are sufficient in the network, new traffic streams will be accepted. Call admission control algorithms can be classified with respect to some parameters such as the place where the admission control decision is made. According to this point of view, two groups of admission control mechanisms exist: centralized and distributed ones.

Centralized admission control algorithms are based on the usage of bandwidth brokers in the center of a DiffServ domain. Although centralized admission control mechanisms make a simple way of creating and changing control operations, they suffer from some noticeable disadvantages, for instance, the connections around the bandwidth broker are excessively busy and the computational load of the bandwidth broker is significantly high. Due to these problems, network operators attracted to perform distributed call admission control algorithms like end point admission control algorithms. In the endpoint admission control algorithm, when a user has a request, the related endpoint uses a measurement process to obtain an estimation of network congestion. In this method, it is not necessary that all routers use signaling messages to perform resource reservation. If routers cooperate and insert their congestion information in the probing phase, it will remove the errors caused by sampling of a short period of network dynamism.

Compared to DiffServ architecture, explicit endpoint admission control algorithm provides a higher granularity in the quality of service levels through the use of service vectors [3]. In this method, each traffic flow is allowed to choose different service classes at different routers of the path. Suppose that a path contains $m$ intermediate routers and each router supports $n$ classes of service. Then the variety of quality of service levels is in the order of $n^m$ while it is in the order of $n$ in DiffServ networks.

## 1.2. Load balancing mechanisms

TE approaches are classified into two groups [4]: time-dependant and state-dependant. State-dependant algorithms are concerned with relatively fast network state changes while time-dependant algorithms are used for utilization of network resources in response to long timescale traffic changes. Load balancing among multiple paths belongs to state-dependant TE approaches.

Basic requirements of different load balancing mechanisms are stated in [5]. Since load balancing algorithms are performed per-packet or per-flow, they should be simple and they should avoid keeping a lot of network states, in order to keep the network overhead in a reasonable level. Load balancing mechanisms can be performed dependent or independent of network states. Equal cost multipath protocol (ECMP) is a technique for independent load balancing [6]. In this protocol if multiple equal-cost paths exist for a destination, traffic demands are shared equally among these paths. Some static traffic splitting methods have been introduced in [7, 8] as follows:

- Dividing traffic in a per packet round robin manner
- Diving destination prefix among available next hops in the forwarding entries

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

- Dividing traffic according to a hash function applied to the header fields of the data packet

Although traffic demands are divided into multiple paths with statistic load balancing approaches, the possibility of congestion still exists since no adaptation to the network load conditions is applied. Therefore state dependant algorithms which perform traffic splitting according to the network state are more efficient to reduce the congestion probability in the network. State dependant mechanisms consist of four basic steps [9]:

1) Choosing multiple loop-free paths between the source-destination pairs
2) Collecting and advertizing information regarding the network load condition
3) Finding an appropriate algorithm to calculate the traffic ratio flowing in each path
4) Distributing traffic according to the algorithm of step 3

Before load balancing, appropriate multiple paths are picked out by using a multipath routing algorithm in step 1. A variety of multipath routing algorithms are proposed in [10]. In order to collect and advertize the information related to network congestion, probing packets are sent from the source to the destination. Based on the information collected by the probing packets, in step 3 the algorithm calculates the proper traffic rate for each of the available path.

Step2, step3 and step4 form the process of general load balancing. A general load balancing system consists of a traffic splitter, an allocation table and a number of outgoing links [11, 12, 13, 14]. First, the incoming traffic is divided into M parts according to a basic criterion at the ingress router. M indicates the smallest amount of traffic which can be transferred into different paths. Afterwards, traffic will be shifted from M different parts to the available IP or LSP paths with respect to the allocation table. In many researches, table-based hashing is used as a traffic splitter [11, 12, 13, 14, 15, 16, 17]. The difference between different load balancing systems is how the incoming traffic is distributed into multiple outgoing links according to the allocation table. In general, different allocation functions can be classified into two groups: optimization-based functions and heuristic functions.

Although optimization-based functions are more accurate than heuristic functions, they impose a high complexity to the network elements. For this reason, these functions are not applicable as a scalable solution in large networks. In addition, complexity increases the computational power and consequently a high computational overhead is imposed to the network. Slow convergence is another problem associated with the optimization-based allocation functions.

Generally, heuristic functions use the information of probing packets and simple equations to perform load balancing. In contrast with optimization-based functions, heuristic functions are more applicable, more scalable and less expensive to be implemented in large networks.

A heuristic algorithm was proposed in [16] which is based on the measurement of average one-way delay by sending probing packets to the destination. In order to calculate the average one-way delay, the time at which a probing packet is sent to the destination and the time at which a probing packet is received at the destination are recorded and the difference between them is averaged and then it is used as a value of one way delay. Although this method belongs to state -dependant algorithms, the network is not involved in the estimation of the splitting parameter and the result of the measurement is just related to an instant sampling of the network congestion state. Furthermore, delay parameter cannot present the real network congestion. Suppose that the speed of network links is not the same. Packets take more time transferring on low speed link than high speed links. Besides, low speed links are usually less congested. As a result, dividing traffic according to delay parameter, leads to transferring more traffic on high speed links and consequently more congestion occurs.

In other similar methods [15, 17], both packet loss and end-to-end delay are used for congestion measurement. A series of probing packets are sent through the network to estimate packet loss. Although this method makes a better estimation of network congestion, periodically sending a series of probing packet may create an extra traffic and causes more congestion. Also, load balancing is according to a short period of time rather than real network dynamism.

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

## 2. Proposed algorithm

The proposed algorithm is a combination of three algorithms: load balancing algorithm, explicit endpoint admission control algorithm and MPA algorithm. Endpoint admission control algorithm consists of two phases: probing phase and data transfer phase. Whenever a user requests a new connection, a new probing phase is started by sending probing packets into multiple paths. Probing phase is repeated every T seconds to adapt the network dynamism with new users' requests by the senders.

MPA algorithm is used to route traffic flows into multiple paths. MPA is implemented with minor changes in the OSPF framework [18]. In the probing phase, probing packets which contain the requested service class will be sent to the paths found by the MPA such that one probing packet is sent into one single path. As the probing packet passes the routers along the path, each router checks the availability of the requested service class. If the requested service class is not available, lower service classes will be checked and finally the available service class in the router will be indentified and its information will be written in the probing packet. However, QoS degradation in a router may not cause the end-to-end QoS degradation, since it is temporary and in the next probing phases it may increase to the requested service level. Even if the end-to-end QoS decreases, still it is acceptable for the user because the user may prefer to be served rather than to be totally discarded [19].

When the probing packets reach the destination, the information attached to them will be analyzed. By comparing the analyzed information of the probing packets, two, or in general more, paths are selected among the probed paths. By choosing the two most proper paths, two service vectors are selected. Each vector contains the intermediate routers with their available service classes. Then the destination sends two acknowledgment packets to the sender on the selected paths. Any number of paths that satisfy the QoS requirement criteria can be chosen among all available paths at the destination. It was shown that routing on two paths is significantly different from single path routing but routing on more than two paths is not very different from routing on two paths [16]. Therefore two paths are selected at the destination to simplify the algorithm. In the implementation of this algorithm the following assumptions are considered:

-The end to end performance of the network is not much different during probing phase and after that.

-Although probing phase is repeated every $T$ seconds with one probing packet in each path, it is assumed that the probing traffic is not that effective in the overall network performance compared to the integrated flows that pass through the core of the network.

### 2.1. Evaluation of service classes

When a probing packet passes through the path, the availability of each service class at each router is determined and indicated in a two bits field in the probing packet. Three service classes are considered in each router: EF, AF, and BE. Although any scheduling algorithm can be used for the three service classes, weighted fair queuing (WFQ) is used to guarantee delay bounds and fair allocation of network resources for all three service classes.

In order to avoid oscillation which can be caused by bursty traffics or arrival of new flows or departure of existing flows, exponential moving average (EMA) equations are used to average the incoming data rate and buffer length in each router for all classes [3].

$$\bar{r}_{sj}(t) = \left(1 - e^{\frac{-\tau_{sj}(t)}{k}}\right) \times \frac{l_{sj}(t)}{\tau_{sj}(t)} + e^{\frac{-\tau_{sj}(t)}{k}} \times \bar{r}_{sj,old}(t) \quad (1)$$

$$\bar{L}_{sj}(t) = \left(1 - e^{\frac{-\tau_{sj}(t)}{k}}\right) \times L_{sj}(t) + e^{\frac{-\tau_{sj}(t)}{k}} \times \bar{L}_{sj,old}(t) \quad (2)$$

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

$s_j$ : service class
$\bar{r}_{sj}(t)$ : The average arrival data rate
$l_{sj}(t)$ : The length of the received packet at time *t*
$\tau_{sj}(t)$ : The period between the current time *t* and the reception time of the previous packet
$\bar{r}_{sj,old}(t)$ : The last updated average data rate
$k$ : a constant value (0.01)
$\bar{L}_{sj}(t)$ : The average of buffer length
$\bar{L}_{sj,old}(t)$ : The last updated average buffer length
$L_{sj}(t)$ : Buffer length at time *t*

Since the measurement of $\bar{r}_{sj}(t)$ and $\bar{L}_{sj}(t)$ are done periodically with the cooperation of routers along the path, the obtained performance of service classes are the reflection of real network state over the time. In contrast, sending a probing chain from the source to the destination shows the measurement results in a small time interval [15, 16, 17].

*k* is a constant that controls the behavior of a bursty traffic. The calculation of the acceptable bounds of *k* in a similar algorithm is introduced in [18]. If *k* is larger than the higher allowable threshold then the convergence speed of averaging equations is slow and it cannot show the behavior of the real traffic. On the other hand, if *k* is smaller than the lower allowable threshold then the effect of current observations is more than the effect of previous observations; therefore averaging equations cannot filter out the transient effects of bursty traffic.

After calculation of average equations, incoming traffic rate is compared to the maximum allowable bandwidth allocated to the requested service class in WFQ. If the arrival rate is lower than the allocated rate, then the traffic will be accepted otherwise the lower service classes are considered.

## 2.2. Calculating the load balancing parameter

As a probing packet traverses a path, the required information of each router is attached to the probing packet. In addition to the information about the availability of the service classes, the required parameters of the load balancing algorithm can be acquired from each router.

In previous researches a variety of parameters such as end-to-end delay, packet loss probability, maximum allowable hop count and link capacity have been used for traffic splitting. End-to-end delay or packet loss probability alone cannot provide a good estimation of real network congestion state [17]. However buffer length can be used to estimate the congestion state of all paths instead of both end-to-end delay and packet loss probability. As a router becomes more crowded, buffer length increases until congestion occurs. When the buffer space increases, it takes longer for a packet to be served. On the other hand, the buffer space calculation does not impose high computational overhead to the network. Remaining buffer space for each class of service is calculated as follows.

$$B_{sj} = L_{sj} - \bar{L}_{sj}(t) \quad (3)$$

$L_{sj}$ : The maximum buffer length of service class $s_j$

As the probing packet goes through the path, the remaining buffer spaces of all classes in each router are obtained and then they are compared to the values of previous router. If the current values are lower than the previous values, they are replaced with the previous values. Therefore, bottleneck of the network can be detected as a limiting factor for sending traffic through the network.

## 2.3. Decision function at the destination

How to select the desired two paths among all available paths depends on the QoS parameter. Maximum end-to-end delay of the path is chosen as a criterion to select the most proper paths. In

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

view of the fact that, user has agreed with service level degradation, two paths with lower maximum end-to-end delay are selected even if they experience larger delay than the required end-to-end delay. The maximum end-to-end delay of service class $s_j$ is calculated according to the following equation:

$$D_{sj} = \frac{L_{sj}}{R_{sj}} + \frac{L}{R} \qquad (4)$$

$R$ : Output link capacity
$L$: Maximum packet length
$R_{sj}$: Guaranteed service rate of service class $s_j$

With respect to the type of service class and the maximum calculated delay of each class, the maximum end-to-end delay of the path is calculated as follows:

$$Max.End-to-End\ Delay = D_{EF} \times EF_{No} + D_{AF} \times AF_{No} + D_{BE} \times BE_{No} \quad (5)$$

$EF_{No}$: Number of routers with EF class along the path
$AF_{No}$: Number of routers with AF class along the path
$BE_{No}$: Number of routers with BE class along the path

It is assumed that the buffer length of similar service classes at different routers are the same, while the buffer lengths of different service classes at each router is different [3].

## 2.4. Data transfer phase

If two ack (acknowledgement) packets arrive in the predefined period, the sender will perform load balancing algorithm on the two paths. If just one ack packet arrives, all traffic will be sent through a single path and no load balancing will be performed.

After obtaining the information of each probing packet, remaining buffer space of the three service classes of the most crowded router in the path are added together and the incoming traffic will be shared on the two paths according to equation (6). Traffic rate of each path is in the direct proportion to the above summation. The proposed load balancing algorithm can be applied to more than two paths.

$$Sum(i) = B_{EF}(i) + B_{AF}(i) + B_{BE}(i) \quad (6)$$

$$Rate\ Path\ (i) = \frac{Sum(i)}{\sum_i Sum\ (i)} \quad (7)$$

$Sum(i)$: Summation of the remaining buffer space of three service classes in the most crowded router of path $i$

## 3. Simulation result

In this section, the performance of the proposed algorithm is evaluated through modeling and simulation using OPNET. The network topology used throughout this study consists of seven routers and all the network links' capacity is assumed to be 4.5 Mbps. Two scenarios are considered for the evaluation: in the first scenario all procedures of the proposed algorithm are implemented except load balancing, and the accepted traffic is sent through one single path. In the second scenario all procedures of the proposed algorithm are implemented and the accepted traffic flows are divided between two paths.

Data packets are 500 bytes long and the length of probing packets are 50 bytes. Probing phase is repeated every 0.5 seconds by sender. The maximum buffer length of EF, AF and BE classes are respectively 24, 53 and 374 packets. The normalized WFQ weights for EF, AF and BE classes are respectively 0.22, 0.33 and 0.44. Theses weights are the proportion of guaranteed rate of each

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

service class to the output link capacity. Considering the assumed maximum buffer length, the maximum delay of each service class will be: $D_{EF}$ = 0.097 sec, $D_{AF}$ = 0.142 sec, $D_{BE}$ = 0.748 sec.

### 3.1. Multipath routing with one cross traffic

In this experiment as depicted in figure 1 two types of traffic are considered: engineered traffic and cross traffic. The proposed algorithm is applied completely on the engineered traffic. The average arrival data rate and the average buffer length of service classes are updated in the routers after the reception of either probing or data packets. On the other hand, no control mechanism is imposed on the cross traffic and one class of service is used constantly in all routers along the path. The effect of cross traffic is to increase the average arrival data rate and average buffer length for the requested service class in all the routers in its path. Figure 1 shows the path of the cross traffic. Main traffic and cross traffic use self similar sources with the following characteristics:

**Main Traffic:**
*Traffic Generation Parameters:*
Start Time (Second): Constant (5)
ON State Time (Second): Pareto (40, 1.4)
OFF State Time (Second): Pareto (1, 1.4)
*Packet Generation Arguments:*
Inter-arrival Time (Second): Exponential (0.001)
Packet Size (Byte): Constant (500)

**Cross Traffic:**
*Traffic Generation Parameters:*
Start Time (Second): Constant (5.1)
ON State Time (Second): Pareto (40, 1.4)
OFF State Time (Second): Pareto (1, 1.4)
*Packet Generation Arguments:*
Inter-arrival Time (Second): Exponential (0.008)
Packet Size (Byte): Constant (500)

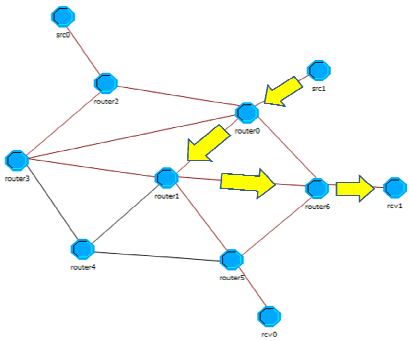


Figure 1. Applied topology with one cross traffic

Cross traffic with a constant bit rate of 0.5 Mbps passes through router 0, router 1 and router 6. The main traffic is distributed through the whole network with the rate of 4 Mbps. Therefore, service rate of some routers is sometimes equal to the arrival traffic rate. As a result, buffer length of those routers increases compared to the case that there is no cross traffic in the network. Simulation results are shown in figure 2 and figure 3. Network throughputs in the two scenarios are almost the same since buffer lengths do not reach to their maximum amount; however the growth of buffer lengths causes a higher end-to-end delay in the first scenario compared to the second scenario because in the second scenario, load balancing algorithm routes traffic on less crowded path, so the

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

time at which packets should be waited to be served in the queues, will be less than the first scenario. By increasing the cross traffic rate, routers' average buffer lengths increase and the throughput improvement achieved through using the second scenario is more noticeable. The rate of the cross traffic is increased up to 0.8 Mbps and the obtained simulation results are shown in figure 4 and figure 5.

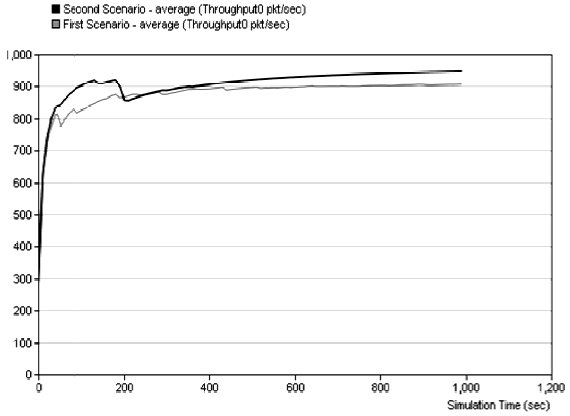


**Figure 2.** Throughput of main traffic with one cross traffic (cross traffic rate=0.5 Mbps)

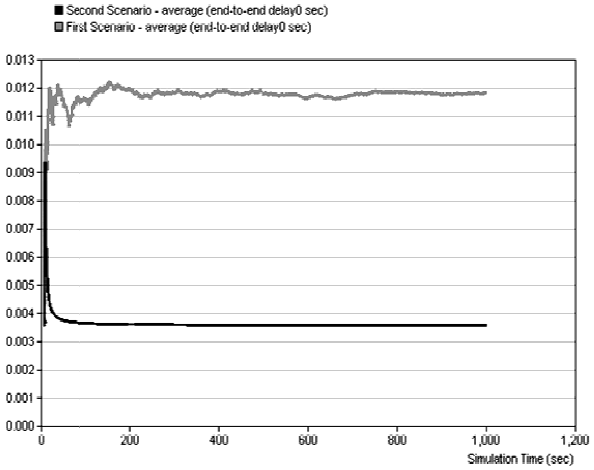


**Figure 3.** End-to-end delay of main traffic with one cross traffic (cross traffic rate=0.5 Mbps)

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

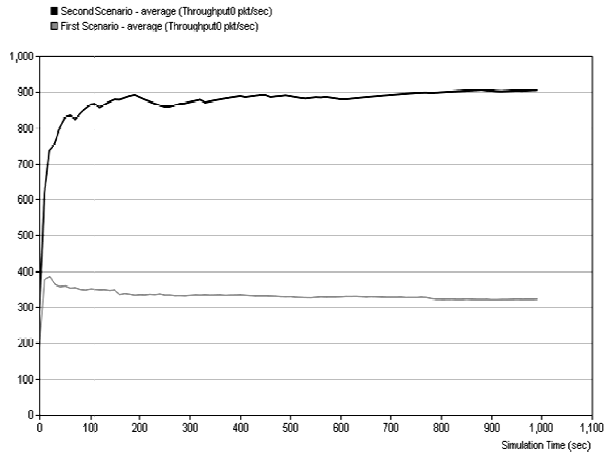


**Figure 4.** Throughput of main traffic with one cross traffic (cross traffic rate= 0.8 Mbps)

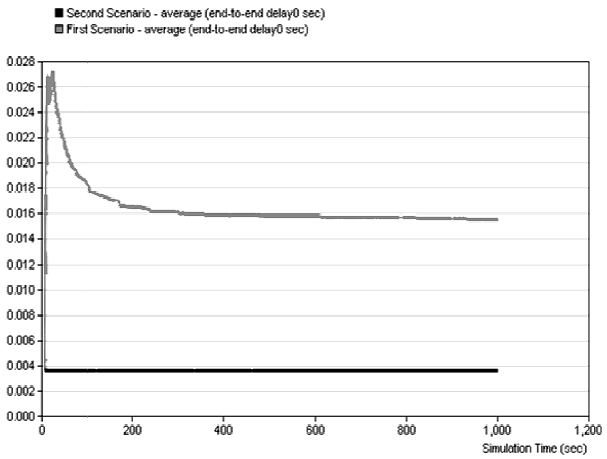


**Figure 5.** End-to-end delay of main traffic with one cross traffic (cross traffic rate= 0.8 Mbps)

## 3.2. Multipath routing with two cross traffics

In this part, three pairs of source-destination exist in the network shown in figure 6 and figure 7, one engineered traffic and two cross traffics. The proposed algorithm is completely implemented on Src0-Rcv0 as the main traffic and Src1-Rcv1 and Src2-Rcv2 act as cross traffics. The paths traversed by cross traffics in both scenarios are shown in figure 6 and figure 7. Cross traffics are divided equally and constantly on two paths in the second scenario but they are sent through one single path in the first scenario. Both main and cross traffic are assumed to be self similar.

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

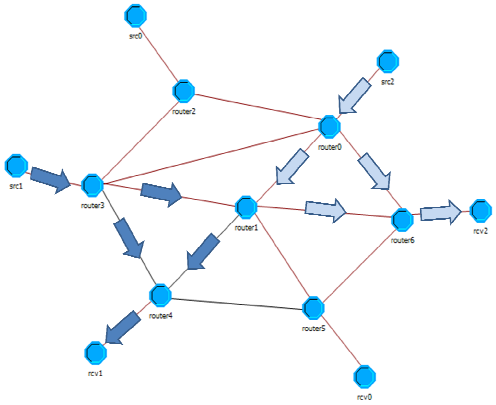


**Figure 6.** Routes of cross traffics in the second scenario

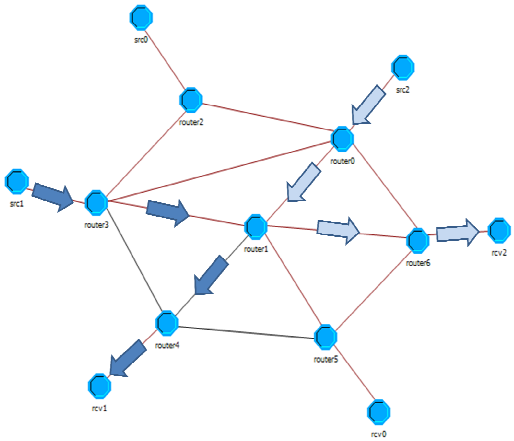


**Figure 7.** Routes of cross traffics in the first scenario

**Main Traffic:**
*Traffic Generation Parameters:*
Start Time (Second): Constant (5)
ON State Time (Second): Pareto (40, 1.4)
OFF State Time (Second): Pareto (1, 1.4)
*Packet Generation Arguments:*
Inter-arrival Time (Second): Exponential (0.001)
Packet Size (Byte): Constant (500)
**Cross Traffic (Src1-Rcv1):**
*Traffic Generation Parameters:*
Start Time (Second): Constant (5.1)
ON State Time (Second): Pareto (40, 1.4)
OFF State Time (Second): Pareto (1, 1.4)
*Packet Generation Arguments:*
Inter-arrival Time (Second): Exponential (0.05)
Packet Size (Byte): Constant (500)
**Cross Traffic (Src2-Rcv2):**

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

*Traffic Generation Parameters:*
Start Time (Second): Constant (5.1)
ON State Time (Second): Pareto (40, 1.4)
OFF State Time (Second): Pareto (1, 1.4)
*Packet Generation Arguments:*
Inter-arrival Time (Second): Exponential (0.005)
Packet Size (Byte): Constant (500)

Simulation results are shown in figure 8 and figure 9. By comparing the results of this part of simulation with the results of the previous one, adding the second cross traffic causes the throughput of the first scenario to decrease more, while throughput of the second scenario does not change significantly. By obtaining the required information from the probing phase and performing load balancing algorithm on the selected paths, the proposed algorithm adapts itself to the extra loads and distributes the incoming traffic with respect to the network congestion state.

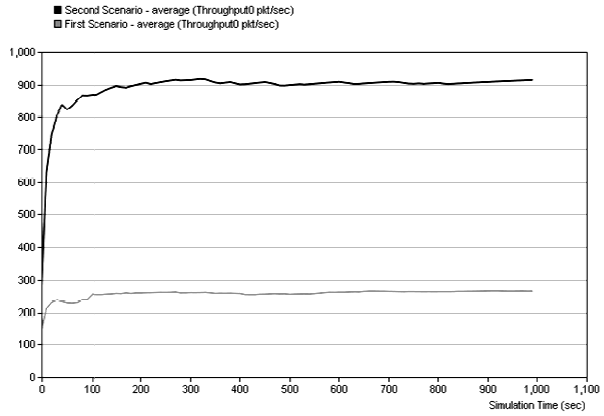


**Figure 8.** Throughput of main traffic with two cross traffics

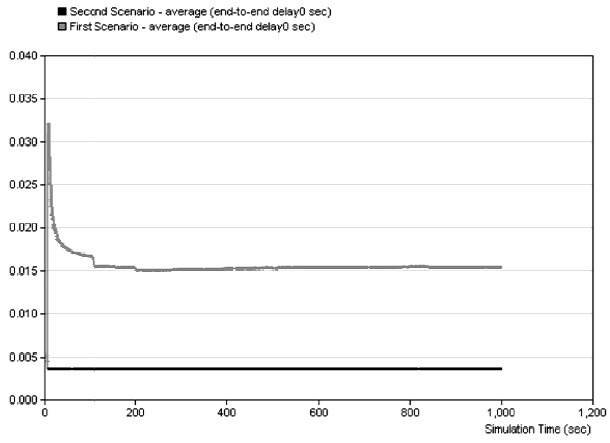


**Figure 9.** End-to-end delay of main traffic with two cross traffics

## 3.3. QoS degradation effect on network throughput

In this part, the effect of QoS degradation in the probing phase is evaluated on the network parameters. Despite the previous scenarios, one extra scenario is added to the simulation. The third

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

scenario is the same as the first scenario and all traffics will be sent through one single path except that in the probing phase traffic flow is accepted only if there are sufficient resources for the requested service class and lower service classes will not be checked. In this case network topology is the same as figure1 and there is only one cross traffic in the network which passes through one constant path.

**Main Traffic:**

*Traffic Generation Parameters:*
Start Time (Second): Constant (5)
ON State Time (Second): Pareto (40, 1.4)
OFF State Time (Second): Pareto (1, 1.4)
*Packet Generation Arguments:*
Inter-arrival Time (Second): Exponential (0.001)
Packet Size (Byte): Constant (500)

**Cross Traffic:**

*Traffic Generation Parameters:*
Start Time (Second): Constant (5.1)
ON State Time (Second): Pareto (40, 1.4)
OFF State Time (Second): Pareto (1, 1.4)
*Packet Generation Arguments:*
Inter-arrival Time (Second): Exponential (0.005)
Packet Size (Byte): Constant (500)

Simulation results are shown in figure 10 and figure 11. As it is shown in the three scenarios the proposed algorithm, which benefits from the cooperation of multipath routing and QoS degradation, results in better throughput and lower end-to-end delay compared to other scenarios. Also the first scenario has higher throughput and higher end-to-end delay compared to the third scenario. The reason is that in the first scenario when available resources are not sufficient for the requested service class, lower classes will be considered and if it is possible, lower service class will be offered to the users. Contrary to the third scenario, WFQ algorithm assigns the remainder bandwidth of lower classes to the higher ones. On the other side, lower classes have higher loss probability and higher end-to-end delay as opposed to higher service classes, therefore the first scenario has higher throughput and higher end-to-end delay compared to the third scenario.

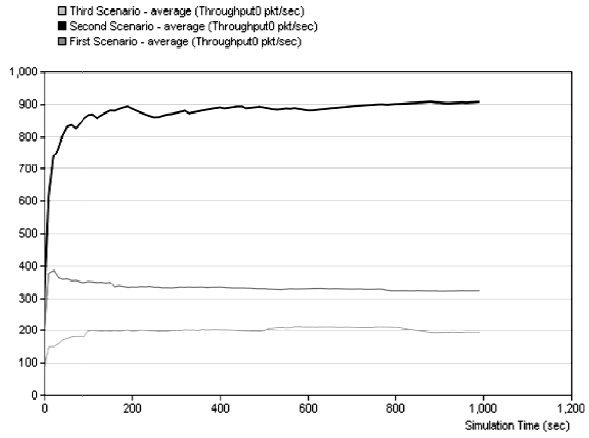


**Figure 10.** Throughput in three scenarios

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

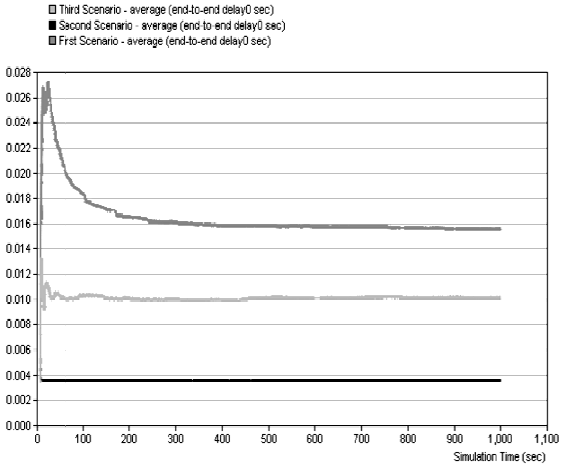


**Figure 11.** End-to-end delay in three scenarios

Despite the mentioned advantages of the proposed algorithm, it can be improved by removing some of its limitations. The proposed network topology, number of cross traffics and their paths through the network, the arrival rate of main traffic and cross traffics, number of ingress-egress pairs and the number of paths performing load balancing are all selective parameters. Although it is predicted that changing these parameters lead to the same results, doing more simulation can improve the accuracy of the results.

This algorithm was implemented in one single DiffServ domain. How to extend it for inter domain usage is still needs more research. On the other hand, passing periodic probing packets through the network, limits the scalability of the proposed algorithm. Considering the effect of probing packets' overhead on simulation results is another research issue. In addition, an appropriate packet reordering mechanism can be added to the algorithm to make it more applicable for TCP streams. Implementing this algorithm on MPLS networks and wireless networks and considering the required changes may lead to a new research issues in NGN networks.

## 4. Conclusion

In this paper, a new adaptive load balancing mechanism was proposed with the help of an explicit end point admission control algorithm. The admission control algorithm consists of two phases: probing phase and data transfer phase. Probing packets are sent through the network in the probing phase and the average buffer lengths and data rates will be updated in each router. The availability of the required service class is checked in the probing phase. If there are enough resources, the requested service class will be offered. Otherwise, lower service classes are considered. At the destination, two paths with the least end-to-end delay are selected. The information of the probing phase regarding buffer lengths was used to split the incoming traffic according to the network congestion state of each path. Multipath routing and load balancing on multiple paths resulted in higher network throughput, better network utilization and lower end-to-end delay. The heuristic equation of load balancing algorithm is simple and scalable in large networks. Additionally, buffer length parameter played a key role in splitting traffic efficiently on multiple paths.

## References

[1] ITU-T Rec., General Overview of NGN, 2001 (12/2004).

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.2,No.6, 2012
Published online: Jun 17, 2012
DOI: 10.7321/jscse.v2.n6.4

[2] N. Wang, K.H. Ho, G. Pavlou, M. Howarth, "An Overview of Routing Optimization for Internet Traffic Engineering", IEEE Communications Surveys & Tutorials, vol. 10, no. 1, pp. 36-56, 2008.
[3] J. Yang, J. Ye, S. Papavassiliou, N. Ansari, "A Flexible and Distributed Architecture for Adaptive End-to-End QoS Provisioning in Next-Generation Networks", IEEE Journal On Selected Areas In Communications, vol. 23, no. 2, 2005.
[4] D. Avduche, A. Chiu, A. Elwalid, I. Vidjaja, X. Xiao, "Overview and Principles of Internet Traffic Engineering", IETF Internet Draft, 2002.
[5] Z. Cao, Z. Wang, E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing", IEEE INFOCOM, Israel, 2000.
[6] J. Moy, "OSPF version 2", RFC2328, 1988.
[7] C. Villamizar, "OSPF Optimized Multipath (OSPF-OMP)", Internet Draft, 1999.
[8] D. Thaler, C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, 2000.
[9] A. Takacs, A. Csaszar, R. Szabo, T. Henk, "Generic Multipath Routing Concept for Dynamic Traffic Engineering", IEEE Communication Letters, vol. 10, no. 2, pp. 126-138, 2006.
[10] H.S. Palakurthi, "Study of Multipath Routing for QoS Provisioning", 2001.
[11] S. Kandula, D. Katabi, S. Sinha, A. Berger, "Dynamic Load Balancing Without Packet Reordering", ACM SIGCOMM Computer Communication Review, 2007.
[12] X. Hesselbach, R. Fabregat, C. Kolias, "The Impact Over the Packets Sequence at the Output Interface in Load Balancing Strategies", Optical Networks, 2006.
[13] H. Abrahamsson, B. Ahlgren, J. Alonso, A. Andersson, P. Kreuger, "A Multi Path Routing Algorithm for IP Networks Based on Flow Optimization", Lecture Notes in Computer Science, pp. 135-144, 2002.
[14] A. Elwalid, C. Jin, S. Low, I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering", IEEE INFOCOM, vol. 3, pp. 1300-1309, 2001.
[15] G. Yuan, Y. Chen, Y. Wei, S. Nie, "A Distributable Traffic-Based MPLS Dynamic Load Balancing Scheme", Asia-Pacific Conference on Communications, Western Australia, 2005.
[16] D. Gao, Y. Shu, S. Liu, O.W.W. Yang, "Delay-Based Adaptive Load Balancing in MPLS Networks", IEEE International Conference on Communications, vol. 2, pp. 1184-1188, 2002.
[17] X. He, H. Tang, M. Zhu, Q. Chu, "Flow-Level Based Adaptive Load Balancing in MPLS Networks", China, 2009.
[18] P. Narvaez, K.Y. Siu, "Efficient Algorithms for Multi-Path Link State Routing", ISCOM'99, Taiwan, 1999.
[19] N. Moghim, S.M. Safavi, M.R. Hashemi, "Evaluation of a new end-to-end quality of service algorithm in differentiated service networks", IET Communications, vol. 4, pp. 1687-1695, 2010.