

# **An Efficient End-to-End QoS Algorithm Using a New End-Point Admission Control in DiffServ Networks**

Neda Moghim\*, Seyyed Mostafa Safavi\*, Masoud Reza Hashemi\*\*

\*Department of Electrical Engineering, Amirkabir University of Technology

\*\*Department of Electrical and Computer Engineering, Isfahan University of Technology

## **Abstract**

*DiffServ is assumed as a scalable solution for service differentiation in the internet. It applies differentiated behaviors on data packets for different quality levels on a per hop basis (PHB). However, it does not have any control on the traffic entering the network and the paths which are traversed by data packets. Therefore, if no complementary mechanism is used with DiffServ, the end-to-end service accuracy and scalability requirements will not be achieved jointly in a service environment.*

*In this paper a distributed end-point admission control algorithm is proposed to control the amount of traffic entering the network and the data packets path.*

*In this algorithm, the input traffic will be efficiently managed while an acceptable end-to-end quality of service is achieved using different local PHBs in different DiffServ nodes.*

## **Keywords:**

*DiffServ, Quality of Service, Traffic Engineering, Call Admission Control*

# **An Efficient End-to-End QoS Algorithm Using a New End-Point Admission Control in DiffServ Networks**

Neda Moghim\*, Seyyed Mostafa Safavi\*, Masoud Reza Hashemi\*\*

\*Department of Electrical Engineering, Amirkabir University of Technology

\*\*Department of Electrical and Computer Engineering, Isfahan University of Technology

## **Abstract**

*DiffServ is assumed as a scalable solution for service differentiation in the internet. It applies differentiated behaviors on data packets for different quality levels on a per hop basis (PHB). However, it does not have any control on the traffic entering the network and the paths which are traversed by data packets. Therefore, if no complementary mechanism is used with DiffServ, the end-to-end service accuracy and scalability requirements will not be achieved jointly in a service environment.*

*In this paper a distributed end-point admission control algorithm is proposed to control the amount of traffic entering the network and the data packets path.*

*In this algorithm, the input traffic will be efficiently managed while an acceptable end-to-end quality of service is achieved using different local PHBs in different DiffServ nodes.*

### **Keywords:**

*DiffServ, Quality of Service, Traffic Engineering, Call Admission Control*

## **1. Introduction**

Originally, the Internet was designed for connectivity of best effort traffic. As Internet is expected to support various types of services, such as voice, video, and etc., it becomes necessary to provide better quality of service (QoS). Existing packet-based network lacks an efficient mechanism for end-to-end QoS [1]. In order to extend the Internet service model so that QoS can be better supported, differentiated services (DiffServ) have been proposed.

Differentiated services (DiffServ) [2] are designed for the scalability of the Internet. The nodes in DiffServ do not need to keep any flow state information. Traffics with similar requirements are classified into the same class, and then each node provides class-based differentiated service by applying different per hop behaviors. However, DiffServ has a basic problem: It works almost predictable only in under-load condition but this condition cannot be guaranteed in a large network with stochastic traffic. DiffServ guarantees maximum delay bound only when

network resources are significantly under loaded. As DiffServ model does not have any control on the amount of traffic entering the network and the paths traversed by data packets. DiffServ can not resolve the problem of congestion without the help of complementary mechanisms [3]. RFC 2998 [4] points out that it is required to integrate DiffServ network Services into an end-to-end service delivery model. If there is a Mechanism in the network that can control data packet path as well as traffic entering the network, end-to-end quality of service will be guaranteed. A lot of research work in the field of routing algorithms and call admission control algorithms are conducted to better engineer the traffic and manage the resources.

The rest of the paper is organized as follows. In the next section, call admission control algorithms are briefly discussed. In section 3, our proposed scheme and its advantages are introduced. The proposed scheme is evaluated via a simulation study presented in section 4. The concluding remarks are provided in Section 5.

## **2. Call Admission Control**

Admission control comprises a set of actions required at the service establishment phase to check whether a service request is admitted or rejected. A new service should be admitted only when the requested QoS can be satisfied without causing any QoS degradation to the already established services.

### *Centralized admission control algorithms:*

The initial proposal was to use a centralized agent, called Bandwidth Broker (BB) [5], to manage the resources within each DiffServ domain and make local admission control decisions. There may be some scalability problems in the centralized approach if the BB has to process lots of requests per second and also the links around the BB will become very congested when the overhead of the signaling messages is high.

### *Distributed admission control algorithms:*

These algorithms perform admission control through signaling at each ingress node of the network. This method categorizes into three methods, parameter-based admission

control, measurement-based admission control and end-point admission control.

Parameter-based admission control method has scalability problem, In addition if the traffic parameters do not reflect the actual behavior of the sources, the performance of this scheme can be very low. Another problem is that this method needs to police the traffic and when statistical traffic descriptors are provided this task will be very hard [6].

Measurement-Based Admission Control (MBAC) avoids these problems by shifting the task of traffic specification from the application to the network by the means of measurement. Relying on measurements alone for admission control raises a number of problems, such as the estimation errors due to network dynamics when the measurement period is short and memory related issues when the measurement period is long [7].

In end-point admission control algorithms and in its probing phase, the decision is made according to the probing performed in a limited time. In this way only a sample of the random process that characterizes the network conditions along the traffic path, will be obtained. Therefore, the probing process without the cooperation of the network may result in incorrect admission decision.

In addition to the above, setup delays may be high and, furthermore, simultaneous probing by many sources can lead to a problem known as thrashing [8].

### 3. Proposed Algorithm

Distributed admission control algorithms decide according to the global information they gather from the network, but as mentioned in the previous section, each category of distributed algorithms has its own shortcomings too.

The use of end-point admission control will help us to correctly and efficiently manage the traffic if the intermediate routers co-operate with the algorithm.

The intermediate routers should co-operate with the algorithm and write the QoS information of their service classes in the probing packets according to which the destination node makes its decision. We make use of this network co-operative endpoint admission control scheme to propose an efficient algorithm to better use the network resources and to reach higher network utilization.

Since only a few classes of service are provided in the DiffServ networks, the current DiffServ model can simply provides a coarse end-to-end QoS granularity, which may result in low resource utilization and Some network resources will be wasted by providing the better than required QoS guarantee. If an application is mapped to the EF service and some network routers can not provide this level of service, the application may prefer receiving a lower level of service in these routers compared to being blocked because despite this degradation of service in some

network spots the required end-to-end quality of service may still be met.

On the other hand, if this local service degradation results in a lower end-to-end quality of service, it may still be acceptable for some applications, especially considering that this degradation will be temporary. Therefore, we can use the network resources more efficiently if we make use of this temporary degradation of service levels in the congested network spots.

The proposed algorithm is an endpoint admission control algorithm which includes two phases at the endpoint side. In the probing phase, the source sends a request to the network. The edge router processes the request and sends out some probe packets in which the required service class is mentioned. The probing packets use the EF class service in the network. Each intermediate router after the reception of a probing packet announces the possibility of providing the required service class in the probing packet. If the required service class is not available, the lower service class availability will be checked and announced.

Finally after the reception of probing packets by the destination, it will process the information and will choose a *service vector* [9] including the intermediate routers and their service classes. The information will be sent back to the source and a path will be set up using this service vector between the source and the destination. Data forwarding phase will begin and the data packets will be sent in the network along this path.

It is assumed that the network dynamics have not changed considerably when the data forwarding phase begins.

It is noticeable that the probing phase needs to be repeated every T seconds to reallocate the service classes according to the new network conditions. The probe repetition will be triggered by the source node. Therefore, there will not be necessary for the edge router to maintain any information about the ongoing flows.

#### Probing Packet Marking Scheme:

We assume that three classes of service are provisioned in an edge or core router. In a DiffServ network, there must be performance bounds for service classes at each router. Weighted fair queuing (WFQ) is a scheduling policy that can guarantee the delay bound and fair resource allocation to each service class [10]. In this paper, we assume that WFQ is used as the scheduling algorithm among the different service classes at each router. The average packet arrival rate [11] and the average queue length [12] will be updated after the reception of both data and probing packets using the following relations:

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

$S_j$  : represents packet service class

$\tau_{sj}(t)$  : Represents the interval between the current time and the arrival time of the previous received packet

$\bar{L}_{S_j,old}(t)$ : Represents the most recently updated average queue length before t.

$L_{S_j}(t)$ : Represents the queue occupancy at time t.

$k$  is a constant that is assumed equal to .01 in our simulation.

$$r_{S_j}(t) = (1 - e^{-\frac{\tau_{S_j}(t)}{k}}) \times \frac{L_{S_j}(t)}{\tau_{S_j}(t)} + e^{-\frac{\tau_{S_j}(t)}{k}} \times r_{S_j,old}(t)$$

$r_{S_j,old}(t)$ : Represents the most recently updated average rate before t.

$l_{S_j}(t)$ : Represents the received packet length at time t.

Three services, EF, AF, and BE, are provisioned at each router and the outgoing link bandwidth is divided among these 3 classes. Each service class will get its own share of network bandwidth according to the weights assigned to them in WFQ scheme.

If the last updated average packet arrival rate of the requested service class ( $r_{S_j}(t)$ ) is lower than the service class share of network bandwidth at the time of probe reception, the availability of the service class will be announced in the probing packet.

#### Decision making at the destination node:

Finally the destination node will process all of the probing packets and calculate the length and maximum end-to-end delay of the paths which are traversed by the probing packets.

For simplicity, the QoS performance metric which is considered in this paper is end-to-end delay. Therefore, the destination will select the shortest path if its maximum end-to-end delay is lower than the required amount. Otherwise, the selected path will be the path which meets the delay requirement. If no such path exists and if the application allows, the path with minimum end-to-end delay will be selected.

#### Maximum e2e delay calculation:

Regarding the WFQ performance, if we denote the output link capacity by  $R$  and the maximum buffer lengths of service classes by  $L_{EF}, L_{AF}, L_{BE}$ , then the corresponding delay bound can be represented by:

$$D_{S_j} = \left( \frac{L_{S_j}}{R_{S_j}} \right) + \left( \frac{L}{R} \right)$$

Where:

$$S_j \in \{EF, AF, BE\}$$

$L$ : maximum packet length

$R_{S_j}$ : guaranteed service rate of  $S_j$

According to the information which is obtained from the probe packets we will have:

EF No. = the counted number of EF classes in the probing packet

AF No. = the counted number of AF classes in the probing packet

BE No. = the counted number of BE classes in the probing packet

The max. end-to-end delay = max EF delay \* EF No. + max AF delay \* AF No. + max. BE delay \* BE No.

## 4. Simulation and Performance Evaluation

In the following, we evaluate the performance of our proposed scheme through modeling and simulation. The network topology used throughout this study is shown in Fig. 1. Specifically, the network consists of seven core routers connected by links of 4.5 Mb/s.

Three services, EF, AF, and BE, are provisioned at each router. The normalized weight of EF, AF, and BE services are assumed .25, .25, and .5, respectively.

Data packets are 500 bytes long. The probing period of each flow is 0.5 s and the length of probing packets is 50 bytes. Two QoS scenarios are considered and evaluated in this study:

-Scenario I: There is the possibility of using lower service levels when the requested service class is not available.

-Scenario II: In each router the users can only use the requested service level.

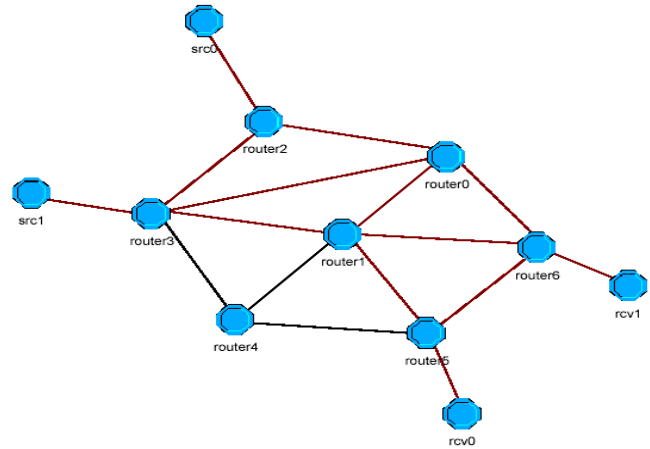


Fig.1. Simulated Network Topology

The predefined service classes' delay bounds in the routers are:

Max EF Delay = .128 Sec., Max AF Delay = .284 Sec.,

Max BE Delay = 10 \* (EF+AF delays)/2 = 2 Sec.

The performance of the network will be evaluated when two following applications appear in the network.

1-Src0 to Rcv0 with EF service class (desired end-to-end delay = 650ms)

2-Src1 to Rcv1 with AF service class (desired end-to-end delay = 850 ms)

The applications agree to receive a degraded level of service temporarily.

The simulation time is 1000 seconds. The simulation random seed and also the simulation time are selected such

that the 95% confidence interval of the results is within 2-4% of the average value.

The network performance is evaluated regarding the number of data packets received at the destination and the probe packet acceptance.

Throughout this study, we use traffic sources with **self similar** distributions. Self similar sources are produced by on-off sources that have the following characteristics.

**On-State-Time (Pareto Distribution):**

**Location: 40** , **Shape: 1.4**

**Packet inter arrival time: exponential (.001)**

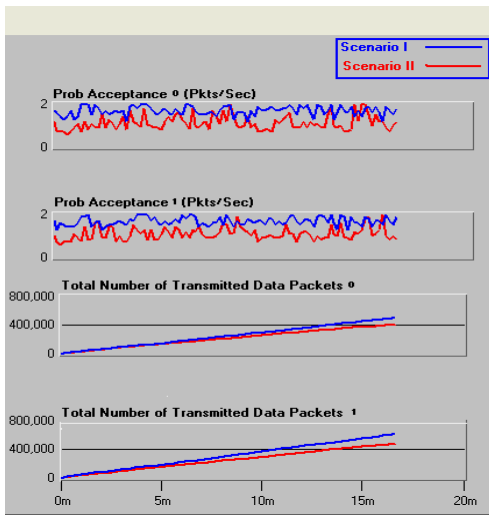
**Off-State-Time (Pareto Distribution):**

**Location: 1** , **Shape: 1.4**

The simulation results are shown in Figures 2 and 3.

As seen in the results, our proposed algorithm leads to an improvement in the percentage of call admissions and also the number of delivered data packets to the destination compared to the second scenario.

In the second scenario there is a management on the amount of packets entering the network and also on the routes which is traversed by the packets through admission control. But the difference of these two scenarios is on the number of service levels which are allowed to be used by the routers.



**Figure (2)**  
Number of accepted probing Pkts & Number of Transmitted data Pkts versus simulation time

Although, the end-to-end delay increases, it is still in the acceptable range. The end-to-end delay increase can be controlled by choosing suitable decision making rules at the destination for the applications that do not want to receive lower quality levels even temporarily.

The performance evaluation and choosing of a suitable rule in such situations are the next steps of our research work.

The end-to-end delay increases because:

When lower service classes are not allowed to be used in the connection setup time, the busy service classes will use the unused bandwidth share of idle service classes (due to WFQ) and the packets will be transmitted more quickly But the number of data packets that can enter the network will

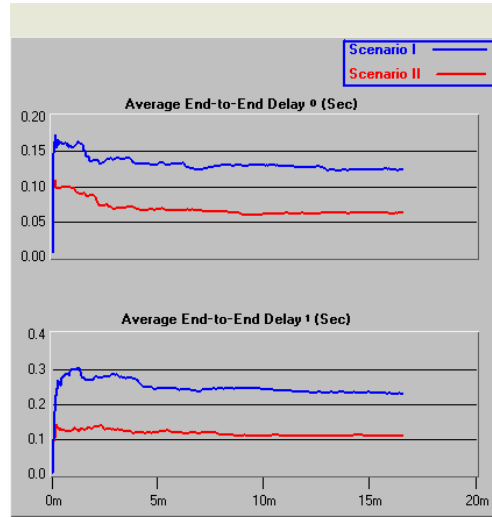
decrease due to the initial call rejection. In the next step of the simulation, a new connection (src2 -> rcv2) is added to the network while the first two connections act as background traffic in the network.

We change the rate of these first two sources to vary the load of network and see the effect of the network load on the third connection performance.

The third connection is a self similar connection with a stochastic variable as its arrival time which has the following defined parameters:

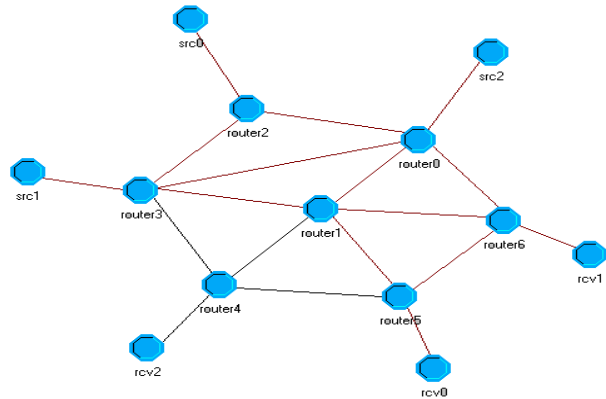
*Arrival time (exponential distribution):*

*Mean: 100 sec.*



**Figure (3)**  
Data packets average end-to-end delay Versus simulation time

The depicted throughput graph is based on the average of the throughputs obtained from different values of connection arrival time. The new connection source generates a self similar traffic with average data packet inter-arrival time of .003 and data packet size of 500 bytes that requests for receiving EF service. The average throughput and the average end-to-end delay of the third



**Figure (4)**

connection are depicted versus the average data packet inter-arrival time of the first two connections that provide a

range of 0.8-80 Mbps load in the network. As it is seen in Figure 5, the effect of our proposed scheme on the throughput is going to be negligible when the network load is low or extremely high but in other load conditions, the proposed algorithm causes a noticeable increase in the total number of packets received at the destination (throughput). When the network load is low, there are sufficient available resources in the network. Therefore, the data packets can get the requested service level and there will not be any need to downgrade the service level in the routers.

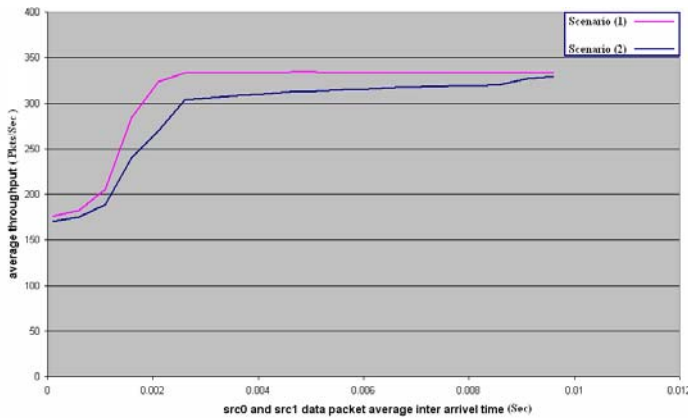


Figure (5)

**The average throughput versus the data packet average inter arrival time**

When the network load is extremely high, all of the network resources have been used and all of the new requests will be rejected. Therefore the proposed algorithm will not be effective.

The end-to-end delay increases using the proposed scheme and the amount of increase becomes very high as we get closer to the highly loaded condition.

So we conclude from the results that our proposed algorithm will work well if the network is not in the low load or very high load conditions.

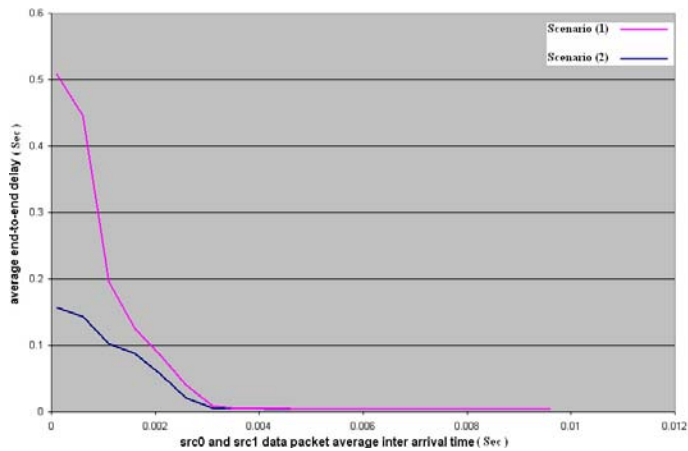


Figure (6)

**The average e2e delay versus the data packet average inter arrival time**

## 5. Conclusion

In this paper a distributed end-point admission control algorithm is proposed to control the amount of traffic entering the network as well as the data packets path. In this algorithm, the input traffic will be efficiently managed while an acceptable end-to-end quality of service is achieved. Users are allowed to temporarily use lower service levels in the routers if the required service class is not available.

In this way, we can increase the data throughput and also the percentage of call admission. The drawback of this method is the increase of end-to-end delay but this increase can be limited.

As the simulation results showed, the call admission percentage and the number of transmitted data increased through this algorithm. Therefore, we can improve the network utilization.

## 6. References

- [1] ITU-T Rec. Y.2001, "General Overview of NGN", 2004
- [2] S. Blake, D. Black, M. Calson, E. Davies, Z. Wang and W. Weiss, "An architecture for differentiated Services", RFC 475, Dec. 1998.
- [3] G. Huston, "Next Steps for the IP QoS Architecture", RFC2990.
- [4] Y. Bernet et al. "A Framework for Integrated Services Operation Over DiffServ Networks", RFC 2998, November 2000.
- [5] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture For the Internet," *Internet RFC 2638*, July 1999.
- [6] S. Georgoulas, P. Trimintzios, G. Parlou and K. H. Ho, "Heterogeneous Real-time Traffic Admission Control in Differentiated Service Domains", IEEE GLOBECOM 2005
- [7] M. Grossglauser and D. Tse, "A Time-Scale Decomposition Approach to Measurement-Based Admission Control", IEEE/ACM Transactions on Networking, August 2003.
- [8] G. Bianchi, N. Blefari-Melazzi, "Admission Control over Assured Forwarding PHBs: a Way to Provide Service Accuracy in a DiffServ Framework" IEEE Globecom'01, pp.25-29, Nov.2001
- [9] 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 JSAC, Vol .23, No.2, Feb.2005
- [10] A. K. Parekh and R. G. Gallager, "A generalized Processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344-357, Jun. 1993
- [11] S. Floyd and V. Jacobson, "Random Early Detection for Congestion Avoidance", IEEE/ACM Trans. Netw., vol. 1, pp. 397-413, Jul. 1993.
- [12] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: A scalable architecture to approximate fair bandwidth allocations in high-speed networks", IEEE/ACM Trans. Netw., vol. 11, No. 1, pp. 33-46, Feb.2003.