Achieving fair and predictable service differentiation through traffic degradation policies.

Vasil Hnatyshin, Adarshpal S. Sethi

Department of Computer and Information Sciences,
University of Delaware, Newark, DE 19716

ABSTRACT

Recently a large number of approaches to service differentiation have appeared in the literature. Most of these approaches are based on the Differentiated Services (DiffServ) Architecture proposed by IETF and they have been evaluated in a number of independent studies. It is widely acknowledged that the Differentiated Services approach provides proper service differentiation in well-provisioned networks under normal traffic conditions. However DiffServ may fail to provide proper service differentiation in the presence of extreme network conditions such as congestion or in under-provisioned networks, resulting in unfair service degradation and unpredictable traffic behavior. We believe that the main reasons for the failure of service differentiation under extreme network conditions are static per-aggregate resource allocation and the lack of the specification of traffic behavior during the congestion. We propose to explicitly define traffic behavior through so called degradation policies. Degradation policies would specify how much of the network resources each traffic class could receive under different levels of congestion. In particular, the degradation policy would specify traffic behavior in terms of the performance parameters like delay, loss, or throughput. Such an approach to service differentiation allows dynamic allocation of traffic resources while maintaining fairness and predictability of the traffic behavior under all network conditions.

Keywords: Quality of Service, service differentiation, differentiated services, degradation policies

1. INTRODUCTION

Today the issue of quality of service (QoS) in the Internet attracts more and more attention. The current best effort approach to quality of service in the Internet can no longer satisfy a diverse variety of customer service requirements because of which there is a need for alternative strategies. Best Effort Service is based on the idea that all traffic is processed the same way without any promises or guarantees about timeliness or actual delivery of the traffic. Furthermore, the current Internet does not provide any differentiation among user requirements for the quality of service the traffic should receive. Such a model becomes highly unacceptable in today's more and more commercialized Internet [97]. As people become willing to pay more for the services that provide a quality of traffic delivery that satisfies their application requirements, the one-service-for-all approach of today's Internet will become obsolete.

Quality of Service requirements may be viewed as being on a multi-dimensional scale where loss, delay, throughput, or other service parameters correspond to different dimensions of that scale. For example, on the low loss, low delay end of the requirements scale are applications that exchange query-like messages and require highly reliable service with very small delays. On the opposite end of the requirements scale are applications that for example, send e-mail messages and are not concerned much with the loss of the messages or with their timely deliverance. There are many applications that lie somewhere in the middle of this requirements scale and may demand low delay but will not care for the loss or may demand low loss and not care much for the delay. For example, multimedia applications are not able to tolerate high delay or jitter but may tolerate some loss. On the other hand, FTP-like applications can tolerate delay but require minimal data loss. Clearly, there are a variety of different application requirements for data transfers over the Internet and people are willing to pay extra to satisfy them.

There are two major strategies in dealing with the problem of providing service guarantees to satisfy different application requirements. One way to approach this problem is to do nothing at all and hope that in the near future bandwidth would become extremely inexpensive due to rapid advances of the research in the area of communications media. Thus all application requirements would be satisfied without any extra effort. However, a large part of the

research community believes that despite the fact that the network resources become cheaper every year, technology will not soon achieve a state where network resources are available in abundance. It is believed that even in the foreseeable future, network resources will not keep pace with the demand imposed by the growth in numbers of users, numbers of applications, and their increasing and diverse requirements. For this reason, the second strategy attempts to provide differentiation between application requirements through introduction of additional mechanisms for traffic treatment. Among the most popular approaches to providing traffic differentiation in the Internet are Integrated Service/RSVP and Differentiated Services. We provide a brief overview of these models in the next section. Section 3 introduces the idea of the degradation policies. Section 4 provides an architectural overview of the degradation policy scheme and the structure of the service level agreement. In Section 5 we present an example of how the degradation policy model could be implemented and provide preliminary evaluation of our model through simulation. We present our conclusions in Section 6.

2. RELATED WORK OVERVIEW

Currently, the two most popular models for providing service differentiation in the Internet are Integrated Services/RSVP and Differentiated Services. The Integrated Services model is based on the idea of reserving available network resources on a per-flow basis. Thus if an application has a specific requirement for, say delay, then it reserves appropriate resources for its traffic on the path from the source to the destination. Such reservations may span one or more network domains that support Integrated Services. This would guarantee that the application would receive the service that it demands. Unfortunately, this approach has a number of shortcomings. The main problem deals with the amount of state information that must be kept at any point in the network. This amount of information is proportional to the number of flows that pass through that point. In large networks, the interior router may have to deal with hundreds of thousand of flows at a time and with per-flow reservation the router would have to deal with and maintain huge amounts of state information. This would place enormous storage and processing overhead, making such an approach not scalable in the core of the network.

The other approach to providing service differentiation is to use the Differentiated Services model [1,2]. In this scheme, each packet that is injected into the network is identified as belonging to one of several classes, and the packet is classified, policed, and marked with the appropriated Differentiated Services Code Point (DSCP), which identifies the class the packet belongs to. All of this processing is usually done only at the ingress nodes that admit traffic into the network. Interior routers process and differentiate incoming traffic based solely on the packet's DSCP. So in general, the Differentiated Services model is based on the idea of combining traffic with similar properties into a single traffic aggregate (represented by the class) and distinguishing traffic that belongs to different aggregates through a DSCP marking set in each packet header. The core routers provide a variety of services by implementing traffic forwarding mechanisms that treat incoming traffic differently based on its marking. In such an approach, the complexity of the packet processing is moved out of the network core into the boundary nodes. Furthermore, this strategy allows interior routers to be very simple and keep only a small amount of information, the size of which depends only on the number of DSCP markings. Therefore, this scheme is scalable and allows simple and fast packet processing implementation of the core routers.

The Differentiated Services architecture consists of the following three elements:

- Per-Hop-Behavior (PHB), which specifies the treatment of the aggregated traffic and includes queuing and forwarding mechanism specifications,
- Traffic classification functions, and
- Traffic conditioning functions that include metering, marking, shaping and policing.

Traffic injected into the DS domain is subject to classification and conditioning at the boundary nodes. After incoming traffic is classified, it is assigned to a particular aggregate and is marked with the corresponding DS codepoint (DSCP). Classification and traffic conditioning rules are specified in the Service Level Agreement established between the customer and the DS domain.

One of the most important aspects of Differentiated Services is network provisioning. Without allocating adequate amount of resources in the network and then logically distributing these resources among the traffic classes, Differentiated Services model is not able to provide any service guarantees. The problem of the network provisioning still remains open; however, we will not address it in this paper. Based on the information about availability of the network resources, the DS domain can establish a Service Level Agreement (SLA) with the end-user. This allows the user traffic to join a particular aggregate and be forwarded to its destination receiving service in accordance with the

established SLA. Within the DS domain, the user traffic is treated according to the aggregate traffic specification and it receives the same treatment as all other traffic within that aggregate. Figure 1 presents an architectural view of the Differentiated Services network domain.

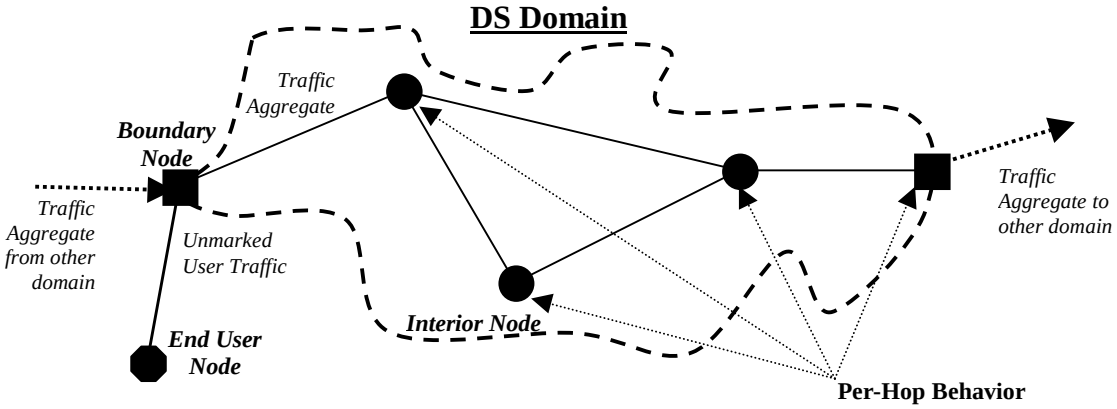

Figure 1. Architectural View of Differentiated Service network domain

The relative differentiated services model is another way to provide service differentiation in the Internet. The idea behind relative service differentiation is that network traffic is divided into a number of classes based on some performance quality. One class is considered to be better or at least not worse than another class, if its performance qualities like delay and loss are better or at least not worse than that of the other traffic class. So if we say that class A is better than class B, we mean that performance quality of class A is higher as compared to class B (or *in relation* to class B). In relative service differentiation, we determine how good one class is only relative to how good the other classes are.

The Proportional Differentiation Model proposed by Dovrolis et al [5-8] is one of the variations of the relative differentiated services. The advantages of proportional service differentiation as compared to other relative differentiation approaches are:

- ***Controllability*** -- ability to adjust how much one class is better than the other.
- ***Predictability*** -- consistency of the provided proportional quality of service even in the presence of load variation and in short timescales.

The idea behind proportional differentiation is that class performance parameters should be proportional to the differentiation parameters of the traffic classes. Let us assume that there are N classes and that $q_i(t,t+\tau)$ is the quality of service achieved by class i during some time period τ. The Proportional Differentiation model maintains a generic Quality Differentiation Parameter (QDP), c_i, for each class, and assumes that all the N classes are ordered based on that generic QDP as follows: $c_1 < c_2 ... < c_N$. Then according to the proportional differentiation model, the following equation should hold for all pairs of classes and during all time intervals $(t,t+\tau)$ for which $q_i(t,t+\tau)$ and $q_j(t,t+\tau)$ are defined:

$$\frac{q_i(t,t+\tau)}{q_j(t,t+\tau)} = \frac{c_i}{c_j} \qquad (1)$$

Dovrolis et al applied the proportional differentiation model in the context of queuing delays and packet loss rates as class performance parameters. The idea behind proportional delay differentiation is to dynamically distribute link bandwidth among the classes in such a way that proportional differentiation constraints hold. In general, when the packets enter the forwarding unit of the proportional differentiation model, they are subject to the proportional loss rate dropper, queuing, and then a proportional delay scheduler, which processes queued packets. In addition, Dovrolis et al introduced a scheme that allows providing of absolute guarantees to the end-user through the proportional differentiated services approach. In this scheme, called dynamic class selection, the user class is varied dynamically based on the network feedback.

3. DEGRADATION POLICIES

3.1 Introduction

Most of the current approaches to service differentiation are able to achieve discrimination among different types of traffic only if the network is well provisioned. However in the case of an under-provisioned network or during intervals of congestion, service differentiation could fail, resulting in unfair service degradation and having unpredictable effects on service differentiation. One of the reasons for this behavior is that most of these approaches do not precisely define how the traffic should be treated during such situations. Basically, what is missing is a specification of how the traffic should be degraded during extreme network situations.

In general it is impossible to guarantee that all the links in the network will be congestion-free without having admission control at each individual link. Since per-link or per-node admission control would be prohibitively expensive, we need to introduce some other means of dealing with network congestion. Although mechanisms like admission control at the network boundary and load-distributed routing can reduce the probability of congestion occurrence, there is no way to guarantee that congestion will never happen without explicitly controlling the incoming traffic rate on a per-link basis. Most current strategies either do not specify precisely how the traffic treatment will change in the case of unexpected network events or simply refuse service to the particular user when his or her request for QoS cannot be satisfied. We believe that this approach to handling traffic during congestion is unacceptable and that better schemes can be designed.

One solution to alleviate this problem is to specify what kind of traffic behavior each user should expect during times of congestion or other network failures. We propose to do this by specifying a degradation policy for each traffic type. This degradation policy would define how the traffic class would degrade along specific performance parameters during times of congestion. In particular, the degradation policy could specify how loss and delay that the traffic class experiences should change due to variation in the network congestion levels.

Consider an example where we have three types of traffic that have different levels of sensitivity to packet loss. The first traffic class is of major importance and cannot tolerate any loss. The second class is of lower importance and also cannot tolerate any loss. However, unlike the first traffic class, this type of traffic prefers to receive no service at all in the case when the loss rate exceeds a particular threshold, perhaps to minimize operational costs. The third class is not very sensitive to data loss and can tolerate some specified amount of loss. We could then define the following degradation policies for these three traffic classes. The first class should not experience any loss regardless of congestion. The second class should not experience any loss when the congestion is low, but it should not receive any service and experience 100% data loss during severe congestion. The third traffic class would experience an amount of loss that increases in some fashion with an increase in the level of congestion.

We believe that addition of the degradation policies can improve the overall differentiated services model and lead to predictable, fair, and scalable service differentiation in IP networks. The idea is, that based on the specified requirements, the user can select a traffic class with the corresponding degradation policy that would satisfy these requirements. We believe that such a service differentiation scheme will be fair and predictable because it is based on the degradation policy, which explicitly specifies how the traffic behavior changes due to variation in the network conditions. Scalability on the other hand, is achieved through the use of IETF's diffserv architecture.

3.2 General Approach

In order to build and evaluate the idea of degradation policy, we would like to combine and modify the ideas proposed by the IETF DiffServ WG and the general idea of proportional service differentiation proposed in [5-8]. IETF's model to service differentiation provides a very simple and scalable architecture. However, it was noted in [3] and [16] that IETF's DiffServ lacks dynamic resource allocation schemes. Proportional differentiation, on the other hand, is architecturally simple and lacks such components as traffic conditioning, policing and metering elements, which can protect the network from overloading. However, this model showed a new way to provide resource allocation, so that it guarantees fairness among the service subscribers while also achieving such features as predictability and controllability.

The idea of degradation policy is based on the observation that most of the strategies perform very well when the network is well provisioned, but suffer from unfairness and unpredictable performance degradation when congestion arises[1]. In light of this observation, we propose to define a degradation policy, which would specify how each class of traffic degrades its performance in the event of congestion or other network problems. Thus, for each class of traffic, we would define a degradation policy that would specify the amount of loss and delay experienced by packets of that class for each level of congestion. For example, a loss degradation policy could be represented in the form of a table that consists of four levels of congestion: light, moderate, heavy and extreme. For each type/class of traffic, we define an amount of loss this traffic should experience due to congestion, as shown in Table 1.

Traffic Class / Level of Congestion	Light	Moderate	Heavy	Extreme
High Importance class	0.0 %	0.0 %	2.0 %	10 %
Medium Importance class	0.0 %	0.5 %	5.0 %	50 %
Low Importance class	0.0 %	1.0 %	10.0 %	100 %

Table 1. Example of Loss Degradation Policy

It is not necessary for the degradation policy to be defined in the form of a table nor is it necessary for the congestion levels to be divided only into four discrete parts. Obviously, different types of representation of the degradation policy and the congestion levels are possible and further investigation of these issues is required.

In general, the user would first examine the set of available degradation policies and would select the one that is the most suitable for the user's needs. Then, the user traffic injected into the network is subject to conditioning and policing at the boundary of the network as in IETF's DiffServ approach. Further, the traffic is subject to the per-hop packet treatment. However, the mechanisms used to implement per-hop treatment will be different from those recommended by IETF for DiffServ PHB implementation.

Based on the level of congestion, the per-hop forwarding mechanism would be able to determine for each traffic class, which packet to schedule next for departure. In the event of congestion, the degradation policy would also be used to determine whether the packet has to be dropped or queued based on the traffic's class. Furthermore, in the event of packet loss, the degradation policy will enable the router to determine which traffic class will experience the packet loss. Implementation of the per-hop packet treatment based on the degradation policy would employ mechanisms similar to those of proportional differentiated services.

We believe that such an approach will enable us to maintain fairness among the traffic classes, while providing absolute guarantees[2]. We believe that the degradation policy model would provide predictable traffic treatment in terms of loss and delay under any network conditions.

4. ARCHITECTURE OF DEGRADATION POLICY APPROACH

4.1 General Architecture Overview

As the network architecture for the degradation policy model, we will use the architectural structure proposed in the IETF's diffserv approach. In this architecture, there are three distinct node types: user nodes, boundary or exterior nodes, and interior or core nodes. User nodes are connected only to the boundary nodes of their domain. They generate traffic and inject it into the network. The traffic generated by the user nodes should conform to the SLA established between the service provider and the end-user. In the differentiated services domain, the boundary nodes may be connected to the user nodes, to the boundary nodes of the same or a different domain, or to the interior nodes of the same domain. Interior nodes are connected to either other interior nodes or to the boundary nodes of the same DS domain. Figure 1 shows the differentiated services architecture and connectivity between the different node types.

Processing of incoming traffic by the boundary node may be divided into two stages. During the first stage of the packet processing, also called incoming traffic processing phase, the boundary node classifies, meters, marks, and

[1] We provide an example of traffic performance degradation in a congested network that supports IETF's Differentiated Services model in the Appendix.

[2] Absolute guarantees under any network conditions can be provided only to high importance classes, while classes of low importance will experience predictable service degradation.

possibly polices incoming traffic as shown in Figure 2. These actions are based on the SLA agreement established between the end-user and the DS domain. All the incoming end-user traffic is always subject to these actions. The incoming traffic processing phase can also deal with the traffic that arrives from other DS domains. In this case, the traffic treatment is based on the bilateral agreement between these neighboring domains.

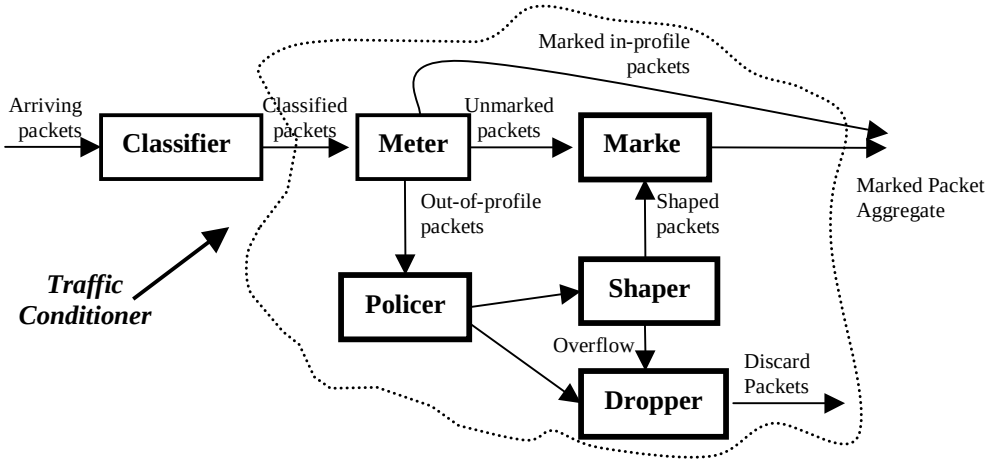

Figure 2. Incoming traffic processing phase

During the second stage of the packet processing, also called outgoing traffic processing phase, the boundary nodes perform per-hop actions like buffering, dropping, and scheduling of the packets. The second stage actions could also be referred to as the packet forwarding or per-hop behavior. The outgoing traffic processing phase usually determines traffic treatment based solely on the packet's DSCP marking. The DSCP value is set in the IP Type of Service (TOS) field by the boundary nodes or in some cases by the end-user. Interior nodes are not required to carry out the actions of the first processing stage (e.g. metering and traffic classifications). However, in a DS-capable network, interior nodes must perform second stage actions on all the incoming traffic. In general, all traffic that travels through the DS domain is subject to the actions of the outgoing traffic processing phase at every node of this domain. Figure 3 shows an overall view of a differentiated services capable node.

4.2 Service Level Agreement (SLA)

In order for the ISP to provide service guarantees to the end-user, it should also define a set of requirements regarding the user traffic behavior. These requirements consist mostly of the rules that define the type of traffic the user can send in order to receive desired quality of service. The traffic specification consists of information about the maximum rate limit at which the user can inject traffic into the domain, the maximum burst size, and possibly other traffic characteristics. The degradation policy proposed by us, can be specified as a part of service guarantees contained in the SLA. Thus the SLA specifies what type of traffic will be accepted into the domain and how this traffic will be treated inside the network.

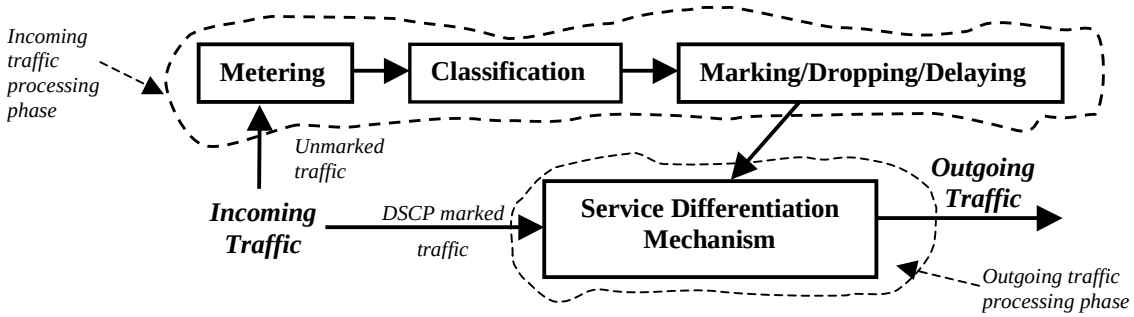

Figure 3. Overall view of the DS-capable node

In our scheme, we use an SLA that consists of two parts: one that defines the user profile and the other that defines the ISP guarantees including traffic degradation policies. The user profile part would include information like: user identification information (e.g. user address), traffic identification information, the type of service, the average traffic rate, the maximum burst size, conforming traffic treatment, and non-conforming traffic treatment (e.g. marking, dropping, or shaping).

User address information will be used to distinguish user traffic at the boundary nodes. Traffic identification information is needed in the cases when the same user runs applications that have different QoS requirements and thus will be mapped into different traffic classes. Such fields in the packet header as IP protocol, TCP/UDP port numbers, or others could be used for identifying traffic from different applications. The average traffic rate and the maximum burst size information are needed to limit the incoming rate of the user traffic. If the user sends its traffic according to the SLA requirements then it is deemed conforming and is marked with the corresponding DSCP value specified in the conforming traffic treatment part of the SLA. If the incoming user traffic violates the SLA (e.g. arrives faster than the rate specified in the agreement), then it is deemed to be non-conforming and is punished according to the non-conforming traffic treatment field of the SLA. Non-conforming traffic treatment includes such punishing actions like dropping the packet, assigning it to a type of service of lower importance, or delaying the packet until it becomes conformant. When non-conforming traffic is re-marked, one should take care not to cause reordering of the packets in the same flow, if packet order needs to be preserved.

The service name specified in the SLA is a domain-unique name and identifies a particular type of service provided within that domain. The traffic degradation policy is a part of the service definition and specifies how the quality of service would degrade during the extreme network conditions such as congestion. This information could be kept in a form of a table that specifies delay, loss, jitter, and throughput values for the different levels of congestion. The traffic degradation policy for a particular type of traffic also could be specified as a set of functions for each performance degradation parameter. These functions would return the value of the particular degradation parameter (e.g. loss, delay, etc.) based on the current network situation (e.g. level of congestion or failure). This information should be used during the congestion by the routers to make decisions regarding which traffic should be discarded first or how the traffic should be scheduled for departure. The probability of service degradation can also be specified as a part of the traffic degradation policy. This information would provide data about how often the user may experience a certain level of service degradation due to network failure, traffic congestion or other factors.

5. EVALUATION OF THE DEGRADATION POLICY

To evaluate the degradation policy model, we have designed an OPNET [18] simulation model for the differentiated services architecture with the degradation policy extension. In our simulation, we built degradation policy in the form of a table that has four levels of congestion: low, medium, heavy, and extreme. Such degradation policy was defined for each traffic class. For simplicity, we defined congestion levels through the buffer occupancy. For each end-user, we have established a service level agreement, which contains the following information:

- End-user IP address
- Target Rate
- Actions toward conforming traffic
- Actions toward non-conforming traffic

We implemented both incoming and outgoing traffic processing phases as described in the previous section: incoming traffic processing phase deals with the unmarked user traffic that arrives at the router, while the outgoing traffic processing phase provides service differentiation to the marked traffic that is scheduled for departure on a particular outgoing link. Boundary nodes classify the arriving traffic based on the IP address and process it based on the established SLA. We used the Time Sliding Window (TSW) mechanism [4,9] for estimation of the traffic arrival rate in order to determine if the traffic is conformant to its SLA or not. During the congestion, the outgoing phase discards traffic using the Proportional Loss Rate Dropper (PLR) [8]. In order to realize degradation policies we modified the PLR to use the values from the loss degradation table for a particular level of congestion instead of loss differentiation parameters.

The goal of our simulation is to show that the degradation policy model could be easily implemented through a simple modification of already existing queuing and scheduling mechanisms. Also we want to show that the degradation policy provides additional flexibility, enabling a network administrator to define a variety of services. In our experiments, we used three traffic classes with different loss requirements. The first traffic class has a requirement that it

should experience no loss unless there is extreme congestion, in which case all of its traffic should be dropped. The second traffic class should not experience more than 10% of traffic loss in the worst case, while the third traffic class requires performance similar to best-effort service. Table 2 presents the loss requirements for each traffic class through their corresponding loss degradation policies.

Traffic Class	Low Congestion	Medium Congestion	Heavy Congestion	Extreme Congestion
First Class	0% loss	0% loss	0% loss	100% loss
Second Class	0% loss	0% loss	5% loss	10% loss
Third Class (BE)	0% loss	5% loss	20% loss	50% loss

Table 2. Loss Degradation Policy

In our simulation, we use a simple topology as shown in Figure 4. In this simulation scenario, all the traffic from the client nodes travels to the single server. The traffic treatment is based on the degradation policies and applied on the bottleneck link between the router and the server. All links in this topology have a capacity of 64 Kbps.

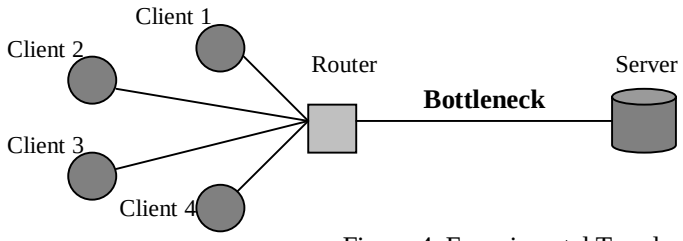

Figure 4. Experimental Topology

The router processes incoming traffic based on the user SLA. This includes metering incoming traffic from each user and marking it with the corresponding DSCP value. The router processes outgoing traffic based on its DSCP value, which identifies the corresponding degradation policy for a particular traffic class. Each of the user nodes generates one type of traffic. The service level agreements for each user are shown in Table 3. In this simulation we used buffer occupancy for determining the level of congestion. Table 4 shows definitions of congestion levels based on the buffer occupancy. The drop probability values for each level of congestion were selected in such a way that buffer occupancy and therefore the congestion level would change only due to a significant change in the aggregated traffic rate.

IP Address	User Name	Traffic Class	Target Rate	Conforming Traffic Treatment	Non-Conforming Traffic Treatment
192.0.1.2	Client 1	Second	23 Kbps	Mark as Second class	Drop
192.0.2.2	Client 2	Best-Effort	23 Kbps	Mark as Best-Effort class	Drop
192.0.3.2	Client 3	Best-Effort	23 Kbps	Mark as Best-Effort class	Drop
192.0.4.2	Client 4	First	23 Kbps	Mark as First class	Drop

Table 3. Service Level Agreement

Congestion Level	Congestion Index	Buffer Occupancy (%)	Drop Probability
Low Congestion	0	[0, 30]	0%
Medium Congestion	1	(30, 50]	0%
Heavy Congestion	2	(50, 90]	10%
Extreme Congestion	3	(90, 100]	10%

Table 4. Definition of Congestion Levels

The executed simulation used the following scenario:

- At time 0 clients 1, 2 and 4 start to transmit traffic at the rate of 23 Kbps.

- At time 200 client 3 joins in and starts sending traffic at the rate of 23 Kbps.

Based on the congestion level at each particular time instance, the router drops the packet with the probability that corresponds to that congestion level as defined in Table 4. If the packet has to be dropped, the router selects the traffic class that should experience the loss using degradation policies implemented using the PLR dropper. Figure 5 shows the amount of traffic loss experienced by each of the traffic classes, and Figure 6 shows how the congestion level was changing during the simulation.

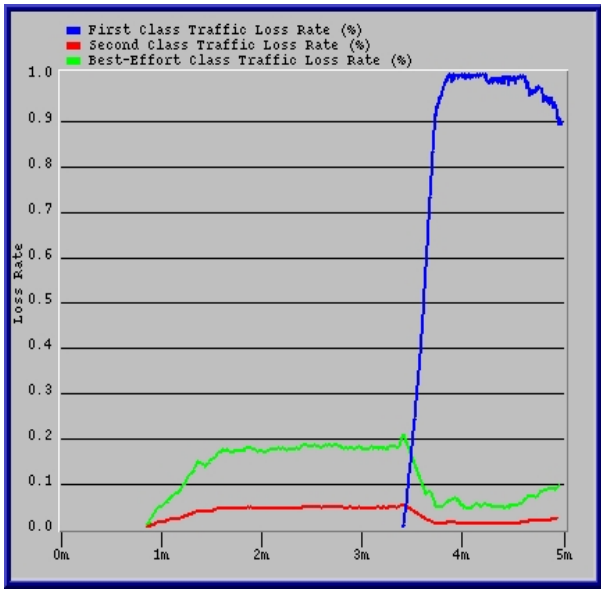

Figure 5. Traffic loss distribution using the degradation policy approach

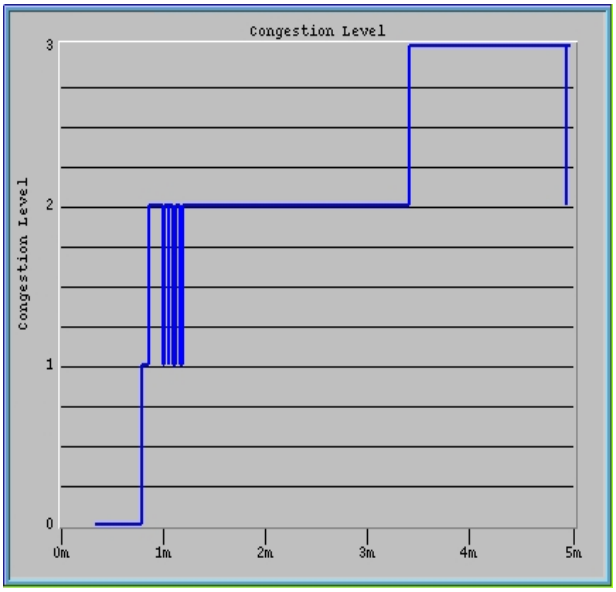

Figure 6. Congestion Levels

As the results show, each traffic class experiences data loss according to its degradation policy. In particular, the first traffic class does not experience any loss when the congestion is not severe, while during that time period the other traffic classes distribute the data loss among them according to their degradation policies. However, when the congestion increases, all the packets that belong to the first traffic class are dropped, while the other traffic classes continue to experience data loss according to their degradation policies. This scenario also shows that the second traffic class achieves absolute loss guarantees; however, when the link is not well-provisioned, this may not be the case and the degradation policy guarantees could be violated. Also, the definition of the congestion used in this set of experiments cannot be used because even if the overall arrival rate remains the same, a change in the arrival rate distribution among the traffic classes may cause degradation policy guarantees to be violated.

6. CONCLUSIONS

In this paper, we have introduced a new approach for service differentiation based on the observation that, during periods of congestion, all the traffic that passes through a congested node should experience different levels of degradation of their quality of service. So in order to differentiate among different traffic classes we proposed to introduce a set of rules that define how traffic classes would degrade their performance during the congestion. We believe that our scheme has a number of advantages. In particular, the degradation policy model provides greater flexibility to a network administrator in defining a wide range of traffic classes based on the user requirements in terms of loss and delay. Also our approach is simple, scalable, and easy to extend. It enables providing absolute guarantees to high importance traffic classes by providing controlled degradation of performance to the traffic classes of lower importance. Furthermore, our scheme will enable users to determine the quality of service their traffic receives based on the current congestion level. However, our proposal introduced only a general framework for the degradation policy model and further study of the problems and issues of the proposed framework is required. In particular, the degradation

policy model cannot be realized without finding solutions to such problems as defining congestion levels and determining the conditions under which the degradation policies are feasible.

REFERENCES

1. Yoram Bernet, James Binder, Steven Blake, Mark Carlson, Brian E. Carpenter, Elwyn Davies, Borje Ohlman, Dinesh Verma, Zheng Wang, Walter Weiss Srinivasan Keshav "A Framework for Differentiated Services", February 1999. Internet Draft: draft-ietf-diffserv-framework-02.txt
2. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. "An Architecture for Differentiated Services", December 1998. IETF RFC 2475
3. Hungkei Chow, Alberto Leon-Garcia, "A Feedback Control Extension to Differentiated Services", March 1999. Internet Draft: draft-chow-diffserv-fbctrl.txt
4. David Clark, Wenjia Fang, "Explicit Allocation of Best Effort Packet Delivery Service," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 362-373, August 1998
5. C. Dovrolis, P. Ramanathan, "A Case for Relative Differentiated Services and Proportional Differentiation Model," *IEEE Network*, October 1999
6. C. Dovrolis, D. Stilliadis, "Relative Differentiated Services in the Internet: Issues and Mechanisms," In *ACM SIGMETRICS*, May 1999
7. C. Dovrolis, D. Stilliadis, P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," In *ACM SIGCOMM*, September 1999
8. C. Dovrolis, D. Stilliadis, "Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping," In *IEEE/IFIP International Workshop on Quality of Service (IWQoS)*, June 2000
9. Wenjia Fang, Nabil Seddigh, Biswajit Nandy, "A Time Sliding Window Three Colour Marker (TSWTCM)", October 1999. Internet Draft: draft-fang-diffserv-tc-tswtcm-00.txt
10. Wu-chang Feng, Dilip D. Kandlur, Dabanjan Saha, and Kang G. Shin "Understanding and Improving TCP Performance over Networks with Minimum Rate Guarantees," IEEE/ACM Transactions on Networking, Vol. 7, No. 2, pp. 173-187, April 1999
11. Wu-chang Feng, Dilip D. Kandlur, Dabanjan Saha, and Kang G. Shin, "A Self-Configuring RED Gateway," In *Proceedings IEEE/INFOCOM*, April 1999
12. Rezende, J. F., "Assured Service Evaluation", In *Proceedings Globecom'99*, March 1999
13. Victor Firoiu Marty Borden, "A Study of Active Queue Management for Congestion Control," *In Proceedings of IEEE/INFOCOM '2000*, March 2000
14. Sally Floyd and Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397-413, August 1993
15. N. Seddigh, B. Nandy, P. Pieda, J. Hadi Salim, A. Chapman "An Experimental Study of Assured Services in DiffServ IP QoS Network," In *Proceedings of SPIE symposium on QoS Issues Related to the Internet*, November 1998
16. Nabil Seddigh, Biswajit Nandy, Peter Pieda, "Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network," In *Proceedings of GLOBECOM '99*, December 1999
17. D. Verma, "Supporting Service Level Agreements on IP Networks", Macmillan Technical Publishing, 1999. ISBN: 1-57870-146-5
18. OPNET Modeler. OPNET Technologies Inc. http://www.mil3.com

APPENDIX:
SERVICE DIFFERENTIATION IN AN UNDER-PROVISIONED NETWORK

To show how the traffic performance degrades in a congested network that supports IETF's Differentiated Services model, we conducted the following experiment using the OPNET simulator [17]. In this experiment we used three traffic classes: first class, second class, and best-effort class. We assume that the first class should receive the highest quality of service in terms of loss and delay. The quality of service experienced by the second class should be better than that of the best-effort class, but not as good as the performance received by the traffic of the first class.

In our simulation we have six user nodes injecting traffic into the Differentiated Services network. This traffic is being processed at the boundary of the network by the ingress nodes that mark arriving packets according to the SLA

specification and forward them further. Core nodes process incoming traffic based on the packet's ToS marking, while the server node processes user requests and generates reply messages. Figure 7 shows the network topology that we use in our experiment.

Ingress	Traffic Type	Target rate	Conforming traffic	Non-conforming traffic
Ingress1	*All*	*10 KBps*	*ToS = 1*	*ToS = 0*
Ingress2	*All*	*16 KBps*	*ToS = 2*	*ToS = 0*

Table 5. Ingress Node SLA

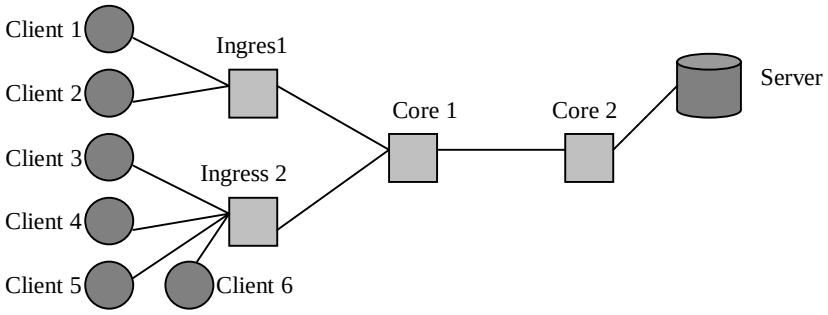

Figure 7. Network Topology

In our experiment user nodes *Client 1*and *Client 2* inject traffic into the network through the router *Ingress1*, while the rest of the user nodes inject their traffic into the network via boundary router *Ingress2*. The ingress routers treat all the incoming traffic according to the SLA shown in Table 5. For example, the router *Ingress1*, meters incoming traffic for each individual user and sets ToS byte to 1 (*Second class*) if the incoming rate is below *10 KBps*. Otherwise it marks traffic as *Best-Effort* class by setting ToS byte to 0. The core routers distinguish traffic classes via a packet's ToS value and use the Weighted Fair Queue scheduler to provide service differentiation. Table 6 provides the amount of bandwidth provisioned for each class on each link in the network as well as the details of the WFQ configuration. In our simulation each link has capacity 64Kbps and we set the link between the nodes C*ore1* and *Core2* to be a bottleneck.

Traffic Type	ToS	Provisioned Rate	WFQ weights	Queue Size (packets)
First Class	*2*	*32 Kbps*	*8*	*200*
Second Class	*1*	*20 Kbps*	*5*	*200*
Best-Effort	*0*	*12 Kbps*	*3*	*200*

Table 6. Configuration and provisioning parameters of the network core.

We ran our experiment for 400 seconds and compared the quality of service achieved by the traffic classes. We set up the simulation so that the user nodes start sending traffic according to the following schedule:

- At time 20 seconds user nodes *Client 1*,*Cleint 2*, *Client 3*, and *Client 4* start sending traffic. At this point the bottleneck link is well provisioned, so that the first and second traffic classes do not experience any performance degradation.
- At time 200 seconds user nodes *Client 5* and *Client 6* start injecting traffic as well, which results in the first traffic class being overloaded.

Figure 8 shows aggregate per-class rates of the traffic that arrives from the ingress nodes at the router *Core1*. The second class always sends traffic at a rate not more than its provisioned bandwidth, while the first class at time 200 seconds has two more users joining in which results in congestion for that class. Because of the static resource allocation strategy employed by the Differentiated Services model, no class can obtain more than its share of provisioned bandwidth when no excess resources are present. This causes the first class to experience excessive loss and delay as shown in Figures 10 and 11 due to the fact that the link is under-provisioned. However, each traffic aggregate receives its provisioned rate regardless of the congestion as shown in Figure 9.

As Figures 9, 10 and 11 show, although all the aggregate traffic classes achieve their provisioned rates on the core link regardless of congestion, the end-users that subscribed to the first traffic class will not receive the quality of service they expected to receive. Even more, the amounts of loss and delay experienced by the first class are higher than those experienced by the second class.

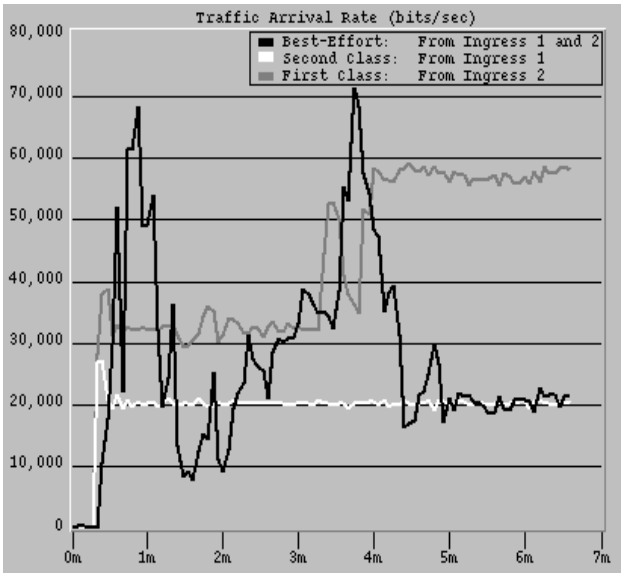

Figure 8. Arrival rate of traffic classes

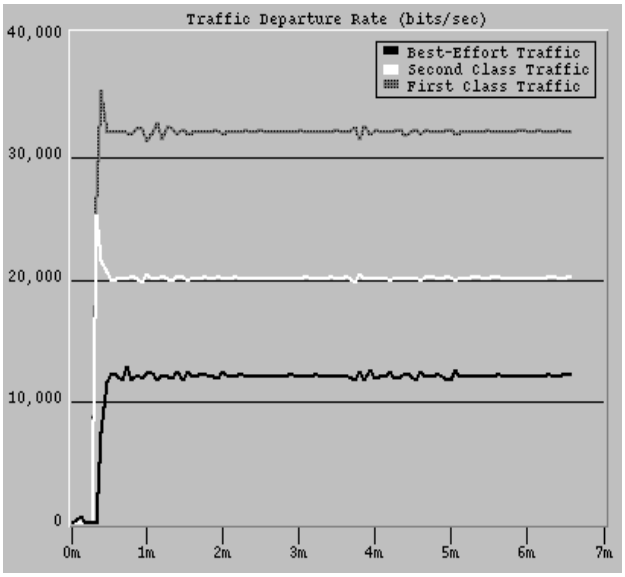

Figure 9. Achieved rate of traffic classes.

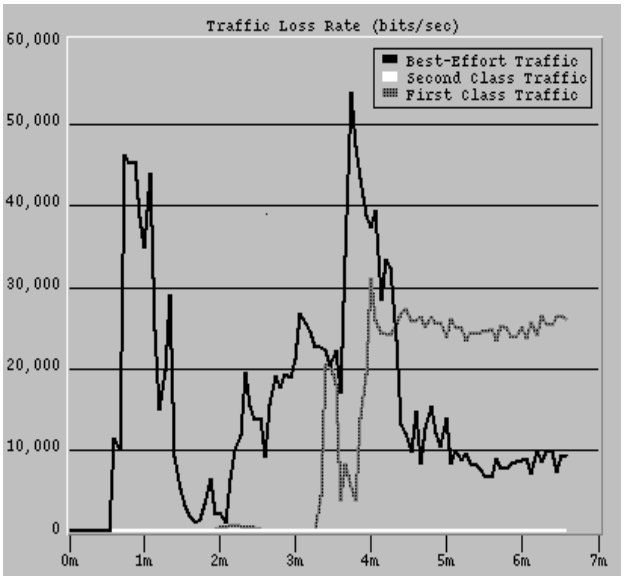

Figure 10. Loss Rates experienced by traffic classes.

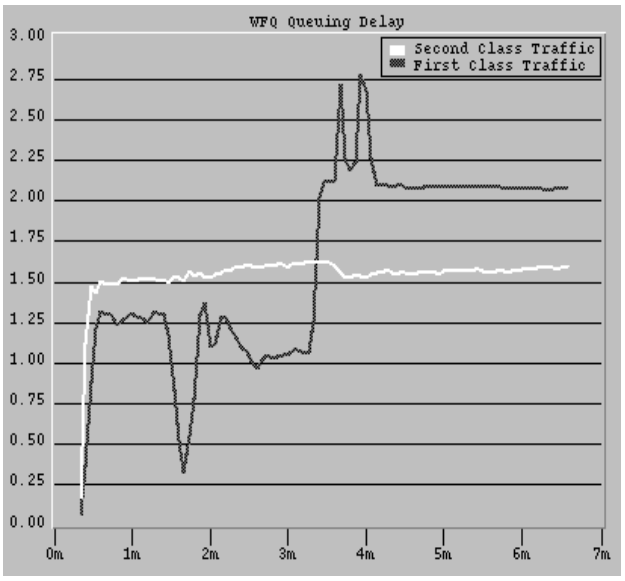

Figure 11. Delay experienced by traffic classes.