Proportional Bandwidth Engineering for Service Differentiation

P. Pereira, A. Casaca

Key Words: *Service differentiation; quality of service; traffic engineering; Multiprotocol Label Switching (MPLS).*

Abstract. *This paper describes two bandwidth engineering techniques for implementing proportional service differentiation based on Multiprotocol Label Switching (MPLS) traffic engineering. Both use dynamic bandwidth allocation schemes to modify the bandwidth reserved by each traffic class according to the current network load. The first scheme uses an adaptive algorithm that qualitatively determines the required average throughput per flow for each class and moves bandwidth between classes for each path as necessary. The second scheme mathematically divides the existing bandwidth through the traffic classes for each path. The quality of service that users get with both techniques is assessed by simulation and compared with a fixed bandwidth allocation scheme.*

I. Introduction

A theorem from early networking days states that [1]: the maximum possible flow from left to right through a network is equal to the minimum value among all simple cut-sets. This theorem is also known as the max-flow min-cut theorem. In practice, it means that the maximum traffic flow a network can forward between two nodes corresponds to the capacity of the corresponding bottleneck.

When the traffic load approaches the network bottleneck capacity, packets start suffering increased delay, increased delay jitter, increased loss and decreased throughput. These parameters are called quality of service (QoS) parameters. The impact of QoS degradation depends on the particular application.

The simplest answer to provide the QoS that the user applications need is to increase the capacity of the links. This is known as overprovisioning and results in a waste of resources, especially for end-to-end QoS.

The Internet traffic can be broadly classified into two categories: streaming traffic and elastic traffic. Streaming traffic corresponds to traffic generated by applications such as real-time audio and video, while elastic traffic is generated by file transfer applications such as web browsing or FTP. The QoS requirements are fundamentally different for each case. Streaming applications require controlled end-to-end packet delays, delay jitter and packet loss. Elastic applications require reliable data transmission with an „acceptable" transfer rate.

Since the QoS requirements depend on the particular application, service differentiation is a solution to accommodate heterogeneous application requirements and user expectations, and to permit differentiated pricing of Internet service. Packets from different user applications with different QoS requirements are classified into different traffic classes and marked accordingly. Only ingress edge routers perform classification, marking and policing functions. The network core only deals with the aggregate of flows in each traffic class, giving them a common queue or scheduling behavior. The Differentiated Services architecture offers a scalable solution for providing end-to-end QoS.

Service differentiation mechanisms do not create bandwidth. They simply divide the network bottleneck bandwidth between the existing traffic classes, according to some differentiation mechanism.

Other mechanisms are commonly used to control how the existing network bandwidth is shared. Examples of such mechanisms are: admission control, traffic engineering, packet scheduling and congestion control mechanisms.

Admission control decides whether a new traffic flow can be accepted in the network, so that when there is congestion it does not get worse. Most admission control mechanisms require the use of signaling traffic. This has imposed significant constraints to customers and the network, and consequently limited its use.

Traffic engineering is concerned with the measurement, modeling, characterization and control of Internet traffic to achieve specific performance objectives. As one of the most significant functions performed by the Internet is the routing of traffic from ingress nodes to egress nodes, one of the most distinctive functions performed by Internet traffic engineering is the control and optimization of the routing function, to steer traffic through the network in the most effective way.

Packet scheduling mechanisms modify the order packets are transmitted onto outgoing links of routers. The re-ordering of packets allows giving more priority to some traffic flows, so that some QoS objectives can be achieved.

Congestion control is used to prevent the saturation of the network while ensuring some fairness in the allocation of resources. When congestion increases, elastic traffic sources decrease their transmission rates according to the congestion control mechanism to prevent congestion. On the other hand, streaming traffic usually is not elastic, suffering from packet loss under congestion.

These bandwidth sharing mechanisms usually coexist in a network, contributing to the QoS users experience.

A different differentiation model is the proportional differentiation model [2]. It is a refinement of the differentiated services model, where an adjustable and consistent differentiation between classes is provided: the QoS of one class is intended to be proportional to the QoS of some other class, whatever traffic load exists in the network. An example of a proportional differentiation requirement is: class 2 throughput = class 1 throughput × 2. A proportional mechanism continuously adapts the service rate to each class, avoiding signaling, admission control and traffic shaping. If the QoS of one class is consistently better than the QoS of another class, the service provider can charge a higher price for providing a better service.

In this paper we combine proportional differentiation with traffic engineering, proposing two proportional bandwidth management schemes that, for each possible source-destination in the network core, divide the existing bandwidth proportionally through the traffic classes, according to the current network load and the QoS required for each class. These mechanisms have the advantage of being simple, not requiring admission control, being scalable and offering a consistent differentiation. Naturally, like a disadvantage, the QoS guarantees are not as strong as when resources are reserved for each flow as for the Integrated Services architecture, but, it is well known, that this latter architecture is not scalable. Both proposed dynamic bandwidth allocation schemes were implemented with Multiprotocol Label Switching (MPLS), analyzed by simulation and compared with a fixed bandwidth allocation scheme for the QoS that users get.

The following section presents some background on bandwidth engineering. Section III describes the dynamic bandwidth allocation schemes. Section IV describes the simulation scenario and the simulation results. Finally, section V draws conclusions and raises further work topics.

II. Bandwidth Engineering

Many traffic engineering solutions are based on the adoption of the MPLS paradigm together with a routing algorithm. MPLS uses a label switching technique that consists in the addition of a label to each packet. This label indexes a forwarding table in each router, determining the outgoing link and the new label to use. The processing involved can be made much faster than looking at the destination address in the packet as in a routing process. As a disadvantage, the Label Switched Paths (LSP) must be setup using a signaling protocol.

Traffic engineering (TE) requirements may be met by setting up paths based on explicit route constraints or QoS constraints. With explicit routing, it is possible to use paths other than the shortest path, so that congested nodes can be avoided and traffic can be divided through different paths. Two mechanisms have been proposed for setting up paths with certain restrictions: MPLS constraint-based routing (CR) [3] and Resource Reservation Protocol Traffic Engineering (RSVP-TE) extensions for LSP tunnels [4]. As both mechanisms provide roughly the same functionality, new developments of MPLS-CR were abandoned, focusing new work in RSVP-TE [5].

Both RSVP-TE and MPLS-CR allow modifying the bandwidth and possibly other parameters of an established LSP without service interruption [6, 7]. This feature has application in dynamic network resources management, where traffic of different priorities and service classes is involved. This capability to have intelligent, dynamic network resource usage is referred to as bandwidth engineering, and it is considered a sub-set of traffic engineering capabilities [8].

Several bandwidth engineering schemes exist in the literature. Some are based on the elastic use of the bandwidth, like [8, 9]. This means that bandwidth assigned to higher priority LSPs, can be temporarily released for the amount of time in which it is not needed and put at disposal of all the other lower priority LSPs. Naturally, as soon as the higher priority LSPs require back their bandwidth, the bandwidth engineering system immediately has to satisfy that need in some way. In order to do that, a function that handles pre-emption of lower priority LSPs or, even better, that can move lower priority traffic to less-congested routes may be used. A bandwidth engineering mechanism like this reduces the need of network over provisioning, but it introduces the possibilities of LSPs being blocked. This problem is serious when the network load is above about 70% of the network capacity. Additionally, a bandwidth engineering system like this needs bandwidth requirements to be specified in service level agreements, at least for higher priority classes. This increases the complexity of the system.

Bandwidth constraint models [10, 11] specify rules for allocating bandwidth to individual traffic classes to achieve per-class traffic engineering. Preemption policies [12] are used in bandwidth constraint models as a flexible and effective control mechanism to dynamically allocate capacity among competing traffic classes with different priorities. This is especially important to deal with link failures. When a link fails, new routes are established, eventually preempting bandwidth from lower priority classes. However, these mechanisms do not account for the load in each of the individual traffic classes. In our work, we treat each end-to-end aggregate as a constant bandwidth pipe and just dynamically subdivide this bandwidth proportionally between classes according to their load and the QoS ratios specified. This allows a fairer QoS division.

If bandwidth is assigned to aggregates of flows of each class and the load in a higher priority class is higher that the load in a lower priority class, a priority inversion may occur: the QoS of flows in the higher priority classes may be worse than for the lower priority class [13].

The work in [13] uses a dynamic packet scheduler to provide proportional QoS differentiation for the different classes. However, this work only analyses TCP connections over a single link, lacking an end-toend evaluation.

The work in [14] uses a policy-based management system to adapt traffic to the current network state. The traffic is measured by bandwidth monitors. According to the current network state, some policies are enforced. The implementation of the selected policies typically leads to accepting, remarking, or dropping the multimedia traffic entering the network.

The work in [15] jointly uses dynamic bandwidth engineering, dynamic packet scheduling and measurement based admission control to enforce QoS requirements. A system like this can efficiently provide the QoS required up to the limits specified in a contract, leaving the remaining resources tc lower priority classes. Naturally, such a system is complex and it is difficult to evaluate to which limits the QoS can be assured. Additionally, when overload exists, the admission control may stop traffic from using the higher priority classes. The use of a proportional differentiation model avoids these difficulties, since a QoS relation between classes is maintained without rejecting traffic.

III. Service Diferentiation Schemes

In this section we introduce the proposed proportional bandwidth engineering algorithms.

A. Service Differentiation Model

We propose to use an Olympic service model with three traffic classes: Gold, Silver and Bronze. We propose also to use a proportional differentiation mechanism between the Gold and Silver classes to give the Gold class the best QoS. The Bronze class is a best effort class that gets a very small fraction of the existing bandwidth along with the unused bandwidth of the other classes. This nomenclature is slightly different from the Olympic service nomenclature proposed in [16].

The Gold and Silver classes could be further divided according to the specific QoS needs of the applications. For simplicity, we assumed that there are only three types of QoS requirements for the applications, corresponding to the three traffic classes. TCP applications tend to increase their transmission rate until there is a packet loss, which should not affect the transmission rate of UDP applications. Accordingly, we propose to have different classes for TCP and UDP traffic.

The use of MPLS TE was selected because it allows reserving and dynamically modifying the bandwidth in a path between a network core ingress and egress node. This characteristic can be used to configure the bandwidth, as necessary, for every traffic class in each path of the core network.

In our model, we propose to reserve bandwidth in the core through MPLS TE for each pair of source destination nodes for the Gold and Silver classes. The Bronze class has no bandwidth reserved through MPLS. All the reservations for the aggregated flows are created initially. For each pair of source-destination nodes, four MPLS reservations are made: Gold TCP, Silver TCP, Gold UDP and Silver UDP. The bandwidths are respectively *bwGoldTCP*, *bwSilverTCP*, *bwGoldUDP* and *bwSilverUDP*. The proposed dynamic bandwidth allocation schemes only modify the MPLS bandwidth reservations for each pair of source-destination nodes as necessary, according to the network load.

The network load can be determined in each ingress router by the number of active flows. New flows can generally be identified by common values in packet header fields (e.g., the 5-tuple of IP addresses, port numbers and transport protocol) and the fact that the interval between such packets is less than some timeout value. It was demonstrated in [17] that the number of active flows is not more than several hundred, even at link loads as high as 90%, and this is valid for any link capacity. This ensures the scalability of the method. Each ingress router determines the number of active flows in each class for each given egress node: *NrFlowsGoldTCP*, *NrFlowsSilverTCP*, *NrFlowsGoldUDP* and *NrFlowsSilverUDP*.

The first scheme proposed in this paper uses an adaptive algorithm that determines the required average throughput per flow for each traffic class, according to the current network load, and moves bandwidth between classes for each path in the network core as necessary.

The second scheme proposed in this paper mathematically divides the existing bandwidth through the traffic classes for each possible path in the network core according to the current network load. Preliminary results of this scheme were described in [18] along with a preliminary version of the adaptive algorithm.

B. Adaptive Dynamic Bandwidth

In the first dynamic bandwidth allocation technique proposed, an adaptive algorithm divides the bandwidth between the traffic classes according to a few rules that determine when the reserved bandwidth should be moved from a traffic class to another.

First, a base bandwidth value of 30% of the available bandwidth is used for each Gold class (TCP and UDP) and 15% for each Silver class. This corresponds to a fixed bandwidth division scenario where Gold has the double of the Silver bandwidth and 10% of the bandwidth is left for best effort and signaling traffic. This fixed bandwidth scenario will be used for comparison in the simulations.

Then, the required bandwidth per flow is set to be inversely proportional to the number of existing flows. This is done independently for each class and protocol. The required bandwidth per flow is limited to the interval 25 to 200 Kbps for the Gold class. For the Silver class, the required bandwidth per flow is limited to be half the Gold required bandwidth per flow and to be within the interval 10 to 100 Kbps. These limits are a characteristic of the classes. They were chosen for the applications to be used in the simulations. The model could be extended with additional classes with different throughput limits, as necessary.

The total bandwidth required for a class is the number of existing flows in that class multiplied by the required bandwidth per flow of the class. The purpose of these calculations is to offer the Gold flows the double of the bandwidth as for the Silver flows, while still ensuring a minimum bandwidth per flow.

If the total bandwidth required for a class is higher than what is currently reserved with MPLS, the dynamic algorithm searches in all the links of the path if it is possible to increase the reserve, considering the flows of the different paths sharing common links. If possible, the reserve for the path is increased. If not possible, the algorithm will try to decrease the reserves of other paths sharing common links. The Silver class is always the first one to be decreased, even if the required bandwidth per flow for the Silver class is not respected. The Gold class is only decreased if the required bandwidth per flow allows it. If no decrease is possible, the bandwidth remains unchanged, which usually happens with high congestion.

The adaptive algorithm allows any path to be reserved for a given source-destination pair. However, for managing the bandwidth of a path, the algorithm has to know the reservations of other paths sharing common links. This requires a global knowledge of the reservations. A decentralized version of the algorithm can be implemented if for the same source-destination pair all classes share the same path and bandwidth is just moved from a class to another in the same path.

C. Mathematical Dynamic Bandwidth

In the Mathematical Dynamic Bandwidth technique, all the classes and protocols for the same source-destination pair use the same path. The existing bandwidth is mathematically divided among them. The idea is that the bandwidth available per flow, for the Gold class should be proportionally higher than the bandwidth available per flow for the Silver class. The bandwidth available per flow is related to the QoS users get.

The corresponding expressions are:

$$\begin{cases} bwGoldTCP + bwSilverTCP + bwGoldUDP + bwSilverUDP = TOTALbw \\ \dfrac{bwGoldTCP}{NrFlowsGoldTCP} = \alpha \times \dfrac{bwSilverTCP}{NrFlowsSilverTCP} \\ \dfrac{bwGoldUDP}{NrFlowsGoldUDP} = \beta \times \dfrac{bwSilverUDP}{NrFlowsSilverUDP} \\ \dfrac{bwGoldTCP}{NrFlowsGoldTCP} = \gamma \times \dfrac{bwGoldUDP}{NrFlowsGoldUDP} \end{cases}$$

The first expression says that the total available bandwidth in a path has to be divided through the 4 existing aggregated flows, one for each combination of the existing classes with the different application protocols. The bandwidth to be used for the source-destination path is given by the value *TOTALbw*. There are also fixed fractions of the bandwidth that are reserved for signaling and best effort traffic, both excluded from the above expressions.

The second expression states that the bandwidth per Gold TCP flow should be α times the bandwidth per Silver TCP flow. The bandwidth per flow is determined by dividing the total bandwidth of the aggregated flow (e.g. *bwGoldTCP*) by the number of flows existing in the aggregated flow (e.g. *NrFlowsGoldTCP*). If the TCP applications continuously generate traffic, this expression results in the throughput QoS parameter for each Gold TCP flow to be α times greater than for the Silver TCP class. A factor of α=2 will be chosen for the simulations, as an example of QoS proportionality between the Gold and Silver classes, so that the Gold QoS is double of the Silver QoS. In practice, it should depend on the relation of the prices charged to clients in the different classes.

The third expression is similar, but for the UDP protocol: the bandwidth per Gold UDP flow should be β times the bandwidth per Silver UDP flow. Again, a value of β=2 will be chosen for the simulations, so that Gold UDP flows get the double of the bandwidth of Silver flows. This usually results in a better QoS (lower delays, jitter and loss) for the Gold class.

The fourth expression states that the bandwidth to each Gold TCP flow should be γ times the bandwidth of each Gold UDP flow. In the simulations, a value of γ will be used, so that only the class influences the QoS, not the protocol chosen. Naturally, this can be changed to favor UDP over TCP, for instance.

Solving these equations, we get the bandwidth allocation for each class, corresponding to the reservations to be made in the network, as:

$$\begin{cases} bwSilverUDP = \dfrac{TOTALbw \times NrFlowsSilverUDP}{\gamma.\beta.NrFlowsGoldTCP + \dfrac{\gamma.\beta}{\alpha} NrFlowsSilverTCP + \beta.NrFlowsGoldUDP + NrFlowsSilverUDP} \\ bwSilverTCP = \dfrac{\gamma.\beta}{\alpha} \times bwSilverUDP \times \dfrac{NrFlowsSilverTCP}{NrFlowsSilverUDP} \\ bwGoldUDP = \beta \times bwSilverUDP \times \dfrac{NrFlowsGoldUDP}{NrFlowsSilverUDP} \\ bwGoldTCP = \alpha \times bwSilverTCP \times \dfrac{NrFlowsGoldTCP}{NrFlowsSilverTCP} \end{cases}$$

The mathematical algorithm modifies the bandwidth reserved for each source-destination path, according to these expressions.

This algorithm allows a very easy implementation. Each ingress node operates independently, by monitoring the load (number of flows) in each class and changing the bandwidth division through the classes for each destination when necessary.

Naturally, each modification of the bandwidth needs some signaling traffic to modify the bandwidth reservations. As such, the reservations should be modified as little as possible. However, it is necessary to modify the reservations only when the load proportions in the different classes change significantly.

An analysis of the above equations shows that the error in the proportional differentiation factor between classes (α, β and γ) is approximately equal to the error difference in the number of flows of the protocols of that class. This means that if we want to keep the error in the proportionality between classes below, say, 5%, we need to modify the reservations whenever the number of flows changes by about 5% relative to the time of the last modification in the reservations. This change could be, for instance, a 3% increase in the number of TCP flows and simultaneously a 2% decrease in the number of UDP flows.

IV. Simulation

A. Network Configuration

The network topology used for the simulations is shown in *figure 1*. The network has a core containing 12 nodes with at least 5 link-disjoint paths between each pair of nodes. This allows for some redundancy, so that there are several alternative paths between each source-destination pair that can be used by traffic engineering. Additionally, there are 3 ingress routers and 3 egress routers where traffic sources and traffic sinks are placed.The core links were configured with 1Mbit/s capacity. Although rather low for current networks, this capacity value allows an easy evaluation of the bottleneck use without burdening the simulations. The core links were configured with a 2 ms delay. This delay corresponds to a typical delay between cities within a country. Since we only wish to study the behavior of the core, the access links were configured with a sufficient large capacity.

The network topology is described and discussed in [19]. The core should have some redundancy, so that there are alternate paths for traffic engineering. The Steiglitz method [20] was used to generate a 5connected core network. This method begins by numbering the nodes at random. Then a link deficit of 5 is assigned to each node. This is the number of links still needed at that node. The method then proceeds by adding links, one at a time, until the deficit at each node is zero or less. At each step, the link to be added is between the node with the highest deficit and the nearest node with the second highest deficit. In case of a tie, the lowest numbered node is chosen. Some perturbation heuristics may be needed in the end to satisfy the connectivity constraint. For the topology of *figure 1*, no perturbation heuristic was needed: the network has 5 link-disjoint paths between each pair of core nodes.

Three types of traffic sources were used simultaneously: long lived FTP/TCP sources, constant bit rate (CBR) sources over UDP and Pareto On/Off sources over UDP. Sources are randomly generated with equal proportions of the three traffic types. Therefore, there might be paths more congested than others in some simulations.

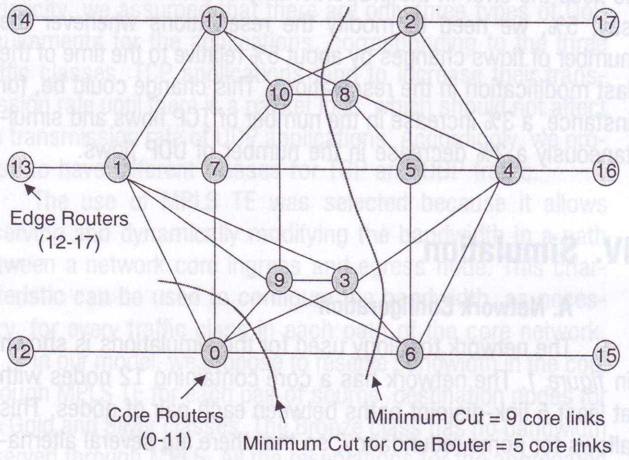


Figure 1. Network topology used in the simulations

Paths are reserved in the core through MPLS TE for each pair of source-destination nodes. The Bronze class has no paths reserved through MPLS. For Gold and Silver classes, there are 2 classes × 2 protocols × 3 source nodes × 3 destination nodes = 36 aggregated flows mapped onto MPLS reservations. All these aggregated flows are created initially. The dynamic bandwidth allocation schemes only modify the MPLS bandwidth reservations for each path as necessary.

The minimum cut from left to right for the network is drawn in *figure 1*, corresponding to 9 links and a bottleneck of 9 Mbit/s of traffic. Each ingress node has 5 links to the core, so the maximum traffic an ingress node could generate, if there was no more traffic, would correspond to the capacity of 5 links. Since there are 9 pairs of source-destination nodes, the traffic that can be transmitted along each source-destination pair is 1 Mbit/s. This capacity is divided by the different aggregated flows as a result of the bandwidth engineering algorithms.

Since the order in which the MPLS paths between sources and destinations are created influences the paths chosen, two orders are used.

The first possibility is to create first the UDP paths for Gold and Silver classes and then the TCP paths for Gold and Silver classes. This will enable that several paths with different source-destination pairs share links in the core. This possibility was used in the adaptive algorithm.

The second possibility is for all source-destination pairs, no matter what their class or protocol, to use the same path. This will enable a better division of the bandwidth between the different classes. This possibility was used in the mathematical algorithm.

The first possibility results in a slightly better delay and jitter for UDP flows, as they use the shortest paths. As streaming traffic usually uses UDP and requires better QoS than elastic traffic, using the shortest paths for UDP flows is a good idea. Otherwise, there are no significant differences in the QoS for the two possibilities.

All simulations were made in the NS2 network simulator [21] with the MPLS for NS2 extensions (MNS) v2.0. These extensions use MPLS-CR for signaling, not the more recent RSVP-TE.

B. Fixed Bandwidth

The fixed bandwidth scenario is used only as a term of comparison in this paper. In this scenario, the bandwidth has a constant division through the traffic classes as follows: 30% for Gold UDP, 30% for Gold TCP, 15% for Silver UDP, 15% for Silver TCP, 10% for Bronze and signaling. These values were chosen so that at least 10% of the bandwidth is available for the best effort traffic, the Gold class has the double of the bandwidth of the Silver class and the UDP and TCP classes have the same bandwidth available. Naturally, these values should depend on the services sold by the Internet Service Provider and the expected traffic pattern. But, in this scenario, they are fixed, whatever the actual traffic pattern is, while in the following scenarios, they change according to the algorithms proposed.

From the different simulations performed, the most interesting case is the one where the number of Gold class sources increases with a fixed number of Silver sources (6 were used) of each type (FTP/TCP, CBR/UDP, Pareto/UDP) and there is a large number of Bronze sources of each type (60 were used). The throughput results for the three different types of traffic sources are shown in *figures 2, 3* and *4*, respectively. Several simulation runs were made with different random number seeds and the same number of traffic sources. The results presented correspond to the average of the different simulations. For the CBR traffic, sources transmitting at 100 Kbit/s were used. The throughput shown for TCP flows is the data rate arriving at the destination in bit/s, while for UDP flows the throughput is the percentage of the traffic sent by the source that arrived to the destination.

As the load in the Gold classes increases, the same fixed bandwidth is divided through more flows, resulting in less throughput per flow for the Gold classes. For the Silver classes, the throughput is almost constant, independently of the load in the Gold class.

The results show an undesirable priority inversion between the throughput for the Gold and Silver classes for high loads: the QoS for the Gold class is worse than for the Silver class for high loads. The dynamic bandwidth algorithms address this problem.

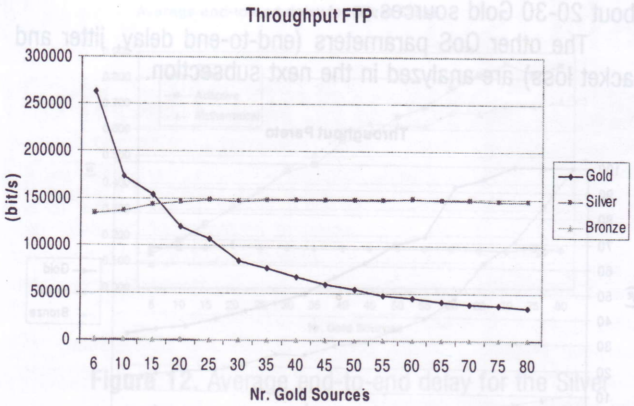


Figure 2. FTP throughput per flow for the fixed bandwidth scenario

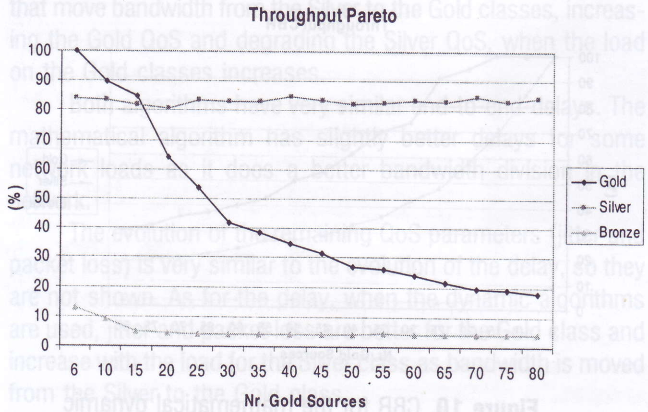


Figure 3. Pareto throughput per flow for the fixed bandwidth scenario

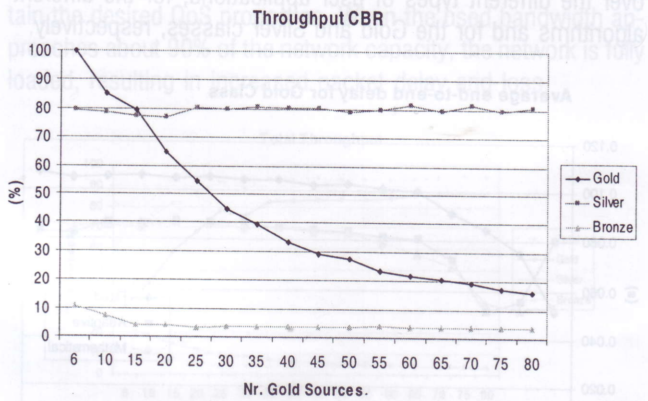


Figure 4. CBR throughput per flow the for the fixed bandwidth scenario

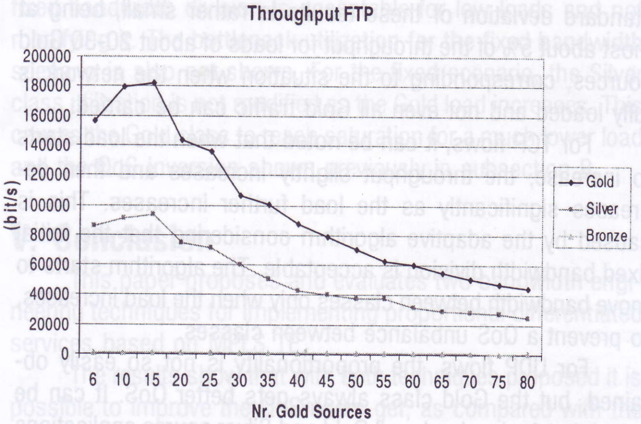


Figure 5. FTP throughput for the adaptive dynamic bandwidth technique

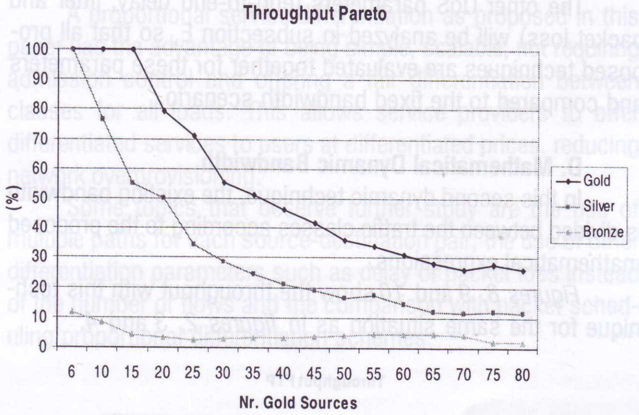


Figure 6. Pareto throughput for the adaptive dynamic bandwidth technique

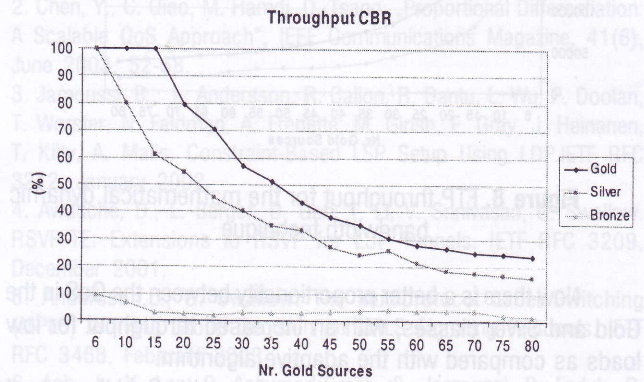


Figure 7. CBR throughput for the adaptive dynamic bandwidth technique

C. Adaptive Dynamic Bandwidth

In the first dynamic technique, an adaptive algorithm moves bandwidth between the traffic classes, as necessary.

Figures 5, 6 and 7 show the throughput with this technique for the same situation as in *figures 2, 3* and *4*.

Now there is no priority inversion for the throughput of the different types of applications, as the load increases. The relation between the Gold throughput and the Silver throughput is on the average 1.7 for UDP traffic (excluding low loads) and 1.9 for TCP traffic, not far from the configured value of 2. The standard deviation of these results is rather small, being at most about 5% of the throughput for loads of about 20-30 Gold sources, corresponding to the situation when the network is fully loaded and not even all Gold traffic can be carried.

For TCP flows, it can be noted that when the load starts to increase, the throughput slightly increases and then decreases significantly as the load further increases. This is caused by the adaptive algorithm considering that the initial fixed bandwidth division is acceptable. The algorithm starts to move bandwidth between classes only when the load increases, to prevent a QoS unbalance between classes.

For UDP flows, the proportionality is not so easily obtained, but the Gold class always gets better QoS. It can be noted that for low loads, all Gold and Silver source applications can transmit all their packets, so the throughput is similar for both classes. When the load increases, the Silver class is much more affected than the Gold class as bandwidth is moved from the Silver to the Gold class.

The other QoS parameters (end-to-end delay, jitter and packet loss) will be analyzed in subsection E, so that all proposed techniques are evaluated together for these parameters and compared to the fixed bandwidth scenario.

D. Mathematical Dynamic Bandwidth

In this second dynamic technique, the existing bandwidth is divided between the traffic classes according to the proposed mathematical expressions.

Figures 8, 9 and *10* show the throughput with this technique for the same situation as in *figures 2, 3* and *4*.

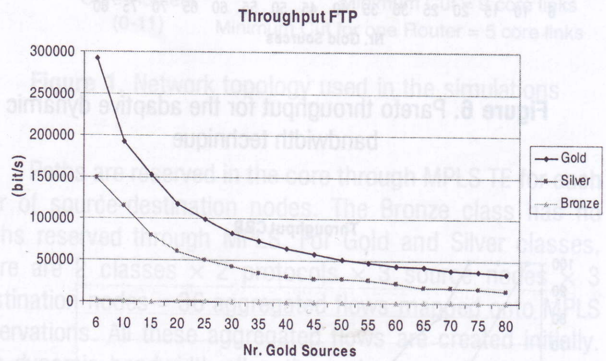


Figure 8. FTP throughput for the mathematical dynamic bandwidth technique

Now there is a better proportionality between the QoS in the Gold and Silver classes, with an increased throughput for low loads as compared with the adaptive algorithm.

For the TCP flows, the relation between the Gold throughput and the Silver throughput is 2.0 on the average.

Again, for UDP flows, for low loads, all Gold and Silver source applications can transmit all their packets, so the throughput is similar for both classes. When the load increases, the Silver class is much more affected than the Gold class. The relation between the Gold throughput and the Silver throughput is 1.9 on the average (excluding low loads).

Again, the standard deviation of these results is rather small, being at most about 5% of the throughput for loads of about 20-30 Gold sources.

The other QoS parameters (end-to-end delay, jitter and packet loss) are analyzed in the next subsection.

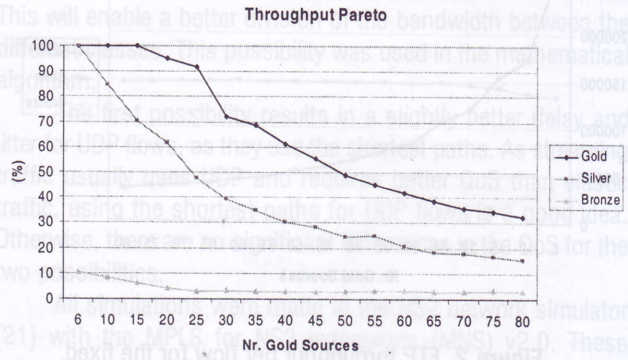


Figure 9. Pareto throughput for the mathematical dynamic bandwidth technique

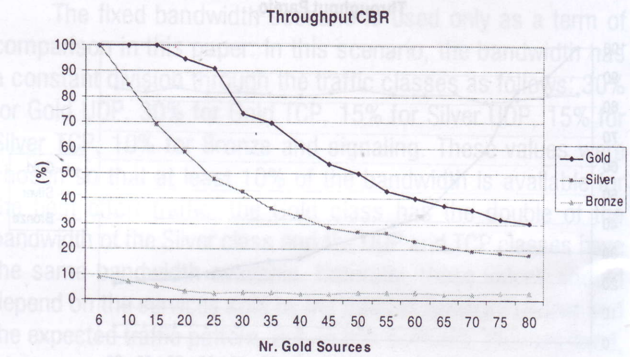


Figure 10. CBR for the mathematical dynamic bandwidth technique

E. QoS Comparison

Figures 11 and *12* show the end-to-end delay averaged over the different types of user applications, for the different algorithms and for the Gold and Silver classes, respectively.

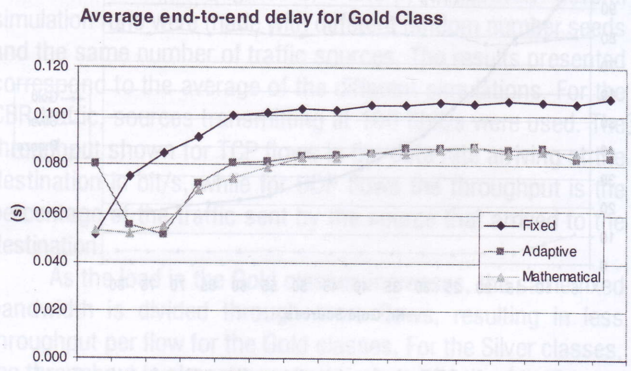


Figure 11. Average end-to-end delay for the Gold class

The end-to-end delay for the Gold classes with the dynamic algorithms has a similar evolution as for the fixed bandwidth scenario, but with smaller values, resulting in better QoS.

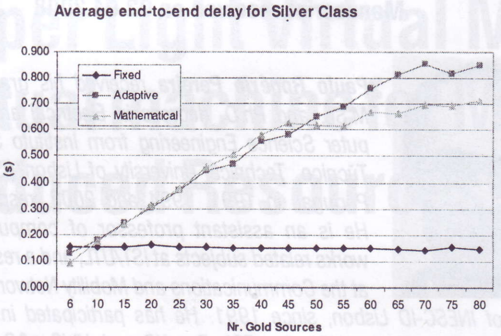


Figure 12. Average end-to-end delay for the Silver class

On the other hand, the end-toend delay for the Silver classes with the dynamic algorithms increases as the load increases, while it is almost fixed in the fixed bandwidth scenario.

These evolutions are caused by the dynamic algorithms that move bandwidth from the Silver to the Gold classes, increasing the Gold QoS and degrading the Silver QoS, when the load on the Gold classes increases.

Both algorithms have very similar end-to-end-delays. The mathematical algorithm has slightly better delays for some network loads as it does a better bandwidth division in the network.

The evolution of the remaining QoS parameters (jitter and packet loss) is very similar to the evolution of the delay, so they are not shown. As for the delay, when the dynamic algorithms are used, jitter and packet loss are better for the Gold class and increase with the load for the Silver class as bandwidth is moved from the Silver to the Gold class.

Figure 13 shows the used bandwidth at the network bottleneck as the load increases for the mathematical algorithm. This figure shows that as the Gold load increases, the mathematical algorithm increases the available bandwidth to the Gold class, decreasing the bandwidth available to the Silver class to maintain the desired QoS proportion. When the used bandwidth approaches about 90% of the network capacity, the network is fully loaded, resulting in increased packet delay and loss.

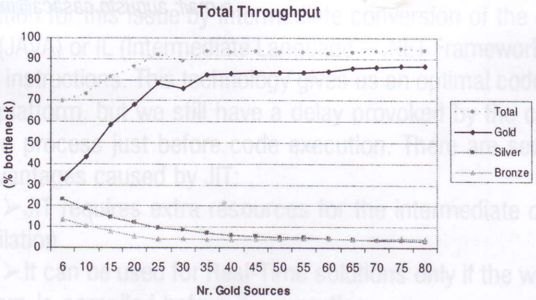


Figure 13. Bottleneek throughput for the mathematical dynamic bandwidth technique

The network bottleneck utilization for the adaptive algorithm is similar to the utilization shown in *figure 13* for the mathematical algorithm and, for this reason, not shown. The exception is for low loads, where it is slightly inferior. This is caused by the adaptive algorithm considering that the initial fixed bandwidth division is acceptable for low loads and not modifying it. The bottleneck utilization for the fixed bandwidth scenario is also not shown. For the fixed scenario, the Silver class utilization is not modified as the Gold load increases. This causes the Gold class to reach saturation for a much lower load and the QoS inversion shown previously in subsection B.

V. Conclusion

This paper proposes and evaluates two bandwidth engineering techniques for implementing proportional differentiated services based on MPLS TE.

The results show that with both techniques proposed it is possible to improve the QoS users get, as compared with the situation of fixed bandwidth division.

The results show that the mathematical technique makes a slightly more uniform division of the bandwidth, according to the number of existing flows, resulting in a better overall QoS and better proportional differentiation of the throughput.

A proportional service differentiation as proposed in this paper has the advantage of being simple, scalable, not requiring admission control and offering a fair differentiation between classes for all loads. This allows service providers to offer differentiated services to users at differentiated prices, reducing network overprovisioning.

Some topics that deserve further study are the use of multiple paths for each source-destination pair, the use of other differentiation parameters such as delay or packet loss instead of the number of flows and the comparison with packet scheduling proportional differentiation schemes.

References

1. Elias, P., A. Feinstein, C. Shannon. A Note on the Maximum Flow through a Network.- *IEEE Transactions on Information Theory*, 2(4), Dec 1956, 117-119.

2. Chen, Y., C. Qiao, M. Hamdi, D. Tsang, „Proportional Differentiation: A Scalable QoS Approach", IEEE Communications Magazine, 41(6), June 2003, 52-58.

3. Jamoussi, B., L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, A. Malis. Constraint-Based LSP Setup Using LDP.IETF RFC 3212, January 2002.

4. Awduche, D., L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. IETF RFC 3209, December 2001.

5. Andersson,L., G. Swallow. The Multiprotocol Label Switching (MPLS) Working Group Decision on MPLS Signaling Protocols. IETF RFC 3468, February 2003.

6. Ash, J., Y. Lee, P. Ashwood-Smith, B. Jamoussi, D. Fedyk, D. Skalecki, L. Li. LSP Modification Using CR-LDP. IETF RFC 3214, January 2002.

7. Polk, J., S. Dhesikan. A Resource Reservation Protocol (RSVP) Extension for the Reduction of Bandwidth of a Reservation Flow. IETF RFC 4495, May 2006.

8. Bosco, A., E. Manconi, R. Sabella, L. Valentini. An Innovative Solution for Dynamic Bandwidth Engineering in IP/MPLS Networks with QoS Support.-*Kluwer Photonic Network Communications*, 7(1), January 2004, 37-42.

9. Iovanna, P., R. Sabella, P. D'Aprile. GMPLS Traffic Engineering

System to Dynamically Manage Several Classes of Services: Performance Analysis. Conference on Optical Network Design and Modeling, Feb. 7-9, 2005, 483-491.

10. Faucheur, F. Le, W. Lai. Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering. IETF RFC 3564, July 2003.

11. Lai, W. Bandwidth Constraints Models for Differentiated Services (Diffserv)-aware MPLS Traffic Engineering: Performance Evaluation. IETF RFC 4128, June 2005.

12. Goldberg, J., S. Dasgupta, J. C. de Oliveira. Bandwidth Constraint Models: A Performance Study with Preemption on Link Failures. 49th IEEE GLOBECOM, San Francisco CA, USA, November 2006.

13. Li, J.-S. and C.-S. Mao. Providing Flow-based Proportional Differentiated Services in Class-based DiffServ Routers.- *IEE Proceedings Communications*, 151(1),February 2004, 82-88.

14. Ahmed, T., R. Boutaba, A. Mehaoua. A Measurement-Based Approach for Dynamic QoS Adaptation in DiffServ Network.-*Journal of Computer Communications*, Special issue on End-to-End Quality of Service Differentiation, Elsevier Science, 28/18, Nov.2005, 2020-2033.

15. Pereira, Paulo. Service Level Agreement Enforcement for Differentiated Services. Wireless Systems and Network Architectures in Next Generation Internet: Second International Workshop of the EURO-NGI Network of Excellence, Italy, Springer-Verlag Lecture Notes in Computer Science, 3883, May 2006, 158-169.

16. Heinamen, J., F. Baker, W. Weiss, J. Wroclawski. Assured Forwarding PHB Group. IETF RFC 2597, June 1999.

17. Kortebi, A., L. Muscariello, S. Oueslati, J. Roberts, On the Scalability of Fair Queueing. ACM HotNets-III, San Diego, USA, November 2004.

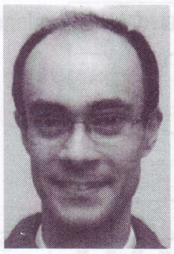
18. Neves, João, Paulo Rogerio Pereira, Augusto Casaca. Proportional Service Differentiation with MPLS. *5th IFIP/IEEE International Conference on Network Control and Engineering for QoS, Security and Mobility (Netcon'2006)*, Santiago, Chile, Springer-Verlag IFIP series, 2025 August 2006, 157-169.

19. Pereira Paulo, Pasquale Lepera, Augusto Casaca. DiffServ Traffic Management with MPLS.*5th Conference on Telecommunications*, ConfTele'2005, Tomar, Portugal, April 2005. ISBN: 972-98115-9-8.

20. Steiglitz Kenneth, Peter Weiner, D. J. Kleitman. The Design of Minimum-Cost Survivable.-Networks. *IEEE Transactions on Circuit Theory*, CT-16, November 1969, No. 4, 455-460.

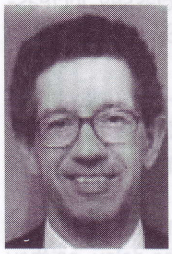
21. UCB/LBNL/VINT Network Simulator (version 2). http://www.isi.edu/nsnam/ns/.

Manuscript received on 02.04.2008



Paulo Rogério Pereira *received his graduation, M.Sc. and Ph.D. degrees in Electrical and Computer Science Engineering from Instituto Superior Tücnico, Technical University of Lisbon (IST/UTL), Portugal, in 1991, 1994 and 2003, respectively. He is an assistant professor of computer networks related subjects at IST/UTL, and a researcher at the Communications and Mobility Networks laboratory of INESC-ID Lisbon, since 1991. He has participated in the IST European projects EuroNGI, EuroFGI, EuroNF and UbiSec&Sens. His research interests include IP quality of service, network management, and wireless sensor networks.*

Contacts:
Inesc-ID/Instituto Superior Tecnico
Rua Alves Redol, 9. 1000-029 Lisboa, Portugal
e-mail: prbp@inesc.pt



Augusto Casaca *graduated in Electrical Engineering at the Instituto Superior Túcnico (IST), Technical University of Lisbon, Portugal. He got the M.Sc. degree in Digital Electronics at UMIST, Manchester, UK, and the Ph.D. in Computer Science at the University of Manchester, Manchester, UK. Presently he is Full Professor at IST and Leader of the Action Line on „Communications and Mobility Networks" at INESC-ID. He has been involved in many research and development activities in the area of Networking and has participated in several EURESCOM, RACE, ACTS and IST projects in the area of Broadband Communications and Networking. He has actively participated in standardization activities at ITU-T and ETSI, has published about 100 scientific papers in journals and conferences, and has chaired or co-chaired five IEEE and IFIP conferences on Networking. He was Chairman of IFIP Technical Committee 6 (Communication Systems) from 1998 to 2004 and is a Senior Member of IEEE. His present research interests are in the areas of Traffic Management and WSN Architecture.*

Contacts:
Inesc-ID/Instituto Superior Tecnico
Rua Alves Redol, 9. 1000-029 Lisboa, Portugal
e-mail: augusto.casaca@inesc.pt