LINEAR CONTROL FOR DYNAMIC LINK SHARING IN MULTI-CLASS IP NETWORKS

Veselin Rakocevic [1], John Griffiths [1], Graham Cope[2]

Abstract: This paper presents Dynamic Bandwidth Partitioning, a new scheme for dynamic link sharing in a multi-class IP network. In this scheme link bandwidth is partitioned between traffic classes, so that the forwarding of each class is independent. The partitions are defined by partition parameters which are dynamic, their change following the measured level of the network performance, with the objective of maximising the end-users' satisfaction, i.e. the quality of service. The level of the network performance is measured by using *utility functions.* The important novelty of this work is in using *non-concave* utility functions for real-time traffic classes. The change of partitions follows a simple 'additive increase, additive decrease' linear control rule, and is reasonably easy to implement in the real network. This paper presents the scheme in detail, shows simulation results, and analyses the implementation of the scheme in the important area of Virtual private networks.

1. Introduction

The current Internet gives no end-to-end Quality of Service guarantees. In general, the Internet offers a single class of 'best-effort' service. Best-effort architecture has been very successful in supporting data applications, but new sophisticated real-time Internet applications (video conferencing, video on demand, distance learning, etc) require better and more reliable network performance. The problem of optimising the network control to satisfy both the issue of fairness for elastic data traffic and the issue of performance guarantees for real-time traffic is very complex. The way bandwidth is shared between concurrent flows is a major issue in Internet QoS design.

We argue that the network optimisation in the multi-class Internet is not about maximising the number of applications (or users) that can be accepted on the network, but in maximising the overall utility - the overall level of the network performance active applications generate on the network. In this work bandwidth allocation is based on the performance evaluation metric of utility. Each end-user of an Internet application receives a certain utility from the network. We use a metric called average connection utility to measure the level of the network performance. Connection utility is an approximate measurement of end-users' satisfaction with the quality of service, and is calculated using pre-defined utility functions, where a single utility function is defined for each traffic class. In the current environment of highly adaptive applications, and of the elasticity as the main feature of the traffic connections, the network performance can be efficiently evaluated by using well-defined utility functions.

The Dynamic Bandwidth Partitioning scheme (DBP) uses such an evaluation mechanism to optimize the bandwidth allocation in the multi-class environment in the IP network. Section 2 defines utility functions and connection utility, and proceeds to define the partitioning algorithm. Section 3 presents the simulation results and analysis of the comparison of the DBP scheme with other bandwidth allocation schemes.

2. Dynamic Bandwidth Partitioning

2.1. Traffic Differentiation and Utility Functions

It is very hard to precisely define utility functions. In this work we have tried to approximate the end-user's utility by defining only one utility function per traffic class. An important property of utility functions that we are defining for real-time traffic types is their non-concavity. The majority of the work considering optimisation of end-user's utility in the past [3] considered strictly concave utility functions.

[1] *Queen Mary, University of London, {v.rakocevic, j.m.griffiths} @elec.qmul.ac.uk*
[2] *Fujitsu Telecommunications Europe Ltd, Birmingham, UK, G.Cope@ftel.co.uk*

In the network model analysed in this paper, the traffic is differentiated into three traffic classes: hard real-time (brittle) traffic, stream (adaptive real-time) traffic and elastic (best-effort) traffic.

The traffic belonging to the hard real-time traffic class requires strict end-to-end performance guarantees and does not show any adaptive properties. If the network is not capable of guaranteeing the required bandwidth for a traffic flow belonging to this traffic class, the end-users' utility will be 0. That is why for hard real-time traffic class we use a very simple utility function (Fig. 1):

$$u_h(b_h) = \begin{cases} 1, \text{ if } b_h \geq b_{h\,min} \\ 0, \text{if } b_h < b_{h\,min} \end{cases} \tag{1}$$

where b_h is the allocated bandwidth, and $b_{h\,min}$ is the minimum required bandwidth.

The second traffic class is the stream traffic. Traffic belonging to this class results from audio and video applications and requires the network to provide a *minimum* level of bandwidth. While active in the network stream traffic applications can adapt [4] to the level of resources the network is allocating them. The rate of the stream traffic flow changes between the minimum required rate $b_{s\,min}$, and the peak rate $b_{s\,max}$. Nevertheless, admission control for this traffic class is necessary, and therefore the optimal number of active traffic flows on the link should be finite. The utility function that can approximate such behaviour is (Fig 2):

$$u_s(b_s) = 1 - e^{-a_{s2}\frac{b_s^2}{a_{s1}+b_s}} \tag{2}$$

where b_s is the allocated bandwidth. The expression in (2) comes from the work of Shenker and Breslau [5]. They used utility functions to analyse the problem of admission control in communication networks. The parameter a_{s1} is easily calculated after the value for $b_{s\,min}$ is known, and the parameter a_{s2} is the shaping parameter. We can see from Fig. 2 that at low bandwidth values the function is convex. The most important feature of this utility function is its *non-concavity*, which makes it different from the utility function for the elastic traffic.

The third traffic class is the elastic traffic class. Elastic traffic flows are established for the transfer of digital documents (files, pictures), and only have loose response time requirements. There is no is no minimum bandwidth requirement and no admission control for this type of traffic. The optimal number of active flows is infinite. The utility function that models the elastic traffic should be always concave, but not linear. The function we propose in this paper is (Fig. 3):

$$u_e(b_e) = 1 - e^{-\frac{a_e b_e}{b_{e\,max}}} \tag{3}$$

where b_e is the allocated bandwidth, and b_{emax} denotes the peak rate for the elastic flow (in the case of the best-effort scheme, $b_{emax} = B$. The parameter a_e is the shaping parameter.

The defined utility functions are used to calculate the average connection utility. Connection utility is the approximation of the network performance connection received while in the network. It is calculated for each flow when the flow terminates. The connection utility v_i of a flow belonging to class i that in time t is allocated $b_i(t)$ of the bandwidth in the network can be approximated with an integral

$$v_i = \frac{1}{T_{dur}} \int_0^{T_{dur}} u_i[b_i(t)]dt \tag{4}$$

where T_{dur} is the duration of the flow.

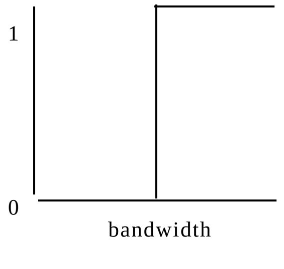

Figure 1 Utility function for the hard real-time

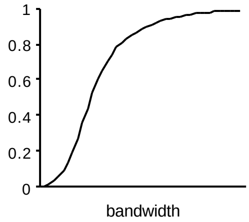

Figure 2 Utility function for the stream traffic

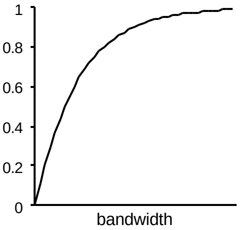

Figure 3 Utility function for the elastic traffic

2.2. *Bandwidth Allocation*

There are two major directions in bandwidth allocation – bandwidth sharing and bandwidth partitioning. In bandwidth sharing traffic can either be differentiated in traffic classes which have different priorities, like in the Differentiated Services architecture, or all traffic flows can have the same priority, like in the best-effort data network. Bandwidth is then allocated to ensure some sort of fairness of allocation to all active traffic flows. Bandwidth partitioning presents an opposite model – the available bandwidth is partitioned between different traffic classes, so that the forwarding of each class is independent.

Let us consider a network link of capacity C. Complete bandwidth partitioning scheme [6] partitions that capacity into N sublinks, where a single traffic class can use each sublink or the sublinks can be used by a traffic aggregation belonging to a single company or virtual private network. This paper presents the analysis of the case when bandwidth is partitioned between traffic classes. In any case, the partitions are defined by partitioning parameters, α_i, where i^{th} sublink uses the capacity $C_i = \alpha_i C$.

If partitioning parameters are fixed, and performance measurement method is known, it is possible [6] to calculate optimal partitioning parameters which maximize the level of the network performance. We propose a new scheme, in which the partitions would be *dynamic*. Partitioning parameters α_i in our scheme change according to a simple linear control rule in order to maximise the overall utility generated in the network. The incoming traffic flows go through the admission control, where the real-time traffic (belonging to the hard real-time or to the stream traffic class) is not accepted on the link if there is not sufficient capacity available to satisfy the minimal requirement of the flow. If the flow is rejected, that generates the negative utility for the rejected flow and thus decreases the average utility. If the admission was successful, the flow is served on the link, where each of the traffic classes has a specified portion of the link capacity available.

2.3. *Partitioning Algorithm*

The way bandwidth is allocated between active traffic flows determines the generated utility. The optimisation problem can be defined as follows: Find α_i, $i = 1,...,N$ to maximise $\sum_i n_i u_i(b_i)$, where $b_i = \alpha_i C / n_i$, and n_i is the number of active flows of class i. In our network model, the utility can be calculated from

$$U(t) = \omega_h u_h(b_h) + \omega_s u_s(b_s) + \omega_e u_e(b_e) \quad (5)$$

where ω_h, ω_s, and ω_e are the scaling factors, and b_h, b_s and b_e are bandwidth levels for the three traffic classes at time t. The scaling factors are introduced to show that the defined traffic classes should not be treated with the same priority. Without the scaling factors, the generated utility will be determined by the number of the active flows from each of the classes. However, we argue that prioritisation is necessary in the multi-class IP environment. It is far more complicated to serve a customer that uses sophisticated video connection, then the one doing a simple file transfer. In the simulation, the following values have been chosen: $\omega_h = 10$, $\omega_s = 3$, $\omega_e = 0.5$.

The idea for the dynamic change in partitioning parameters in the DBP scheme is that every time the utility decreases for a certain value, a change in the partitioning parameters happens, in the direction which increases the overall utility.

Let us define the *normalised* utility $NU(t)$:

$$NU(t) = \frac{\sum w_i u_i(b_i)}{\sum w_i} \quad (6)$$

If the normalised utility $NU(t)$ between two time instances, $t-\tau$ and t, decreases for more then a certain specified amount δ, a change of the partitioning parameters happens. The direction of the change in defined by the change of the utilities for each traffic class, $\Delta u_i(t) = u_i(t) - u_i(t-\tau)$. The parameter with the largest decrease in the utility is increased, while all other parameters are decreased. The linear control algorithm is used to calculate the new value for the partitioning parameters. The value of the partitioning parameter α_i at time t is then,

$$\alpha_{\mathrm{i}}(t) = \alpha_i(t-\tau) + f\left[NU(t), NU(t-\tau)\right] \qquad (7)$$

A linear control algorithm is successfully used in the TCP congestion control mechanism. TCP uses additive increase, multiplicative decrease rule [7], with transmission rates being increased until the network signals the loss of packets. Then, the transmission rates are decreased by multiplying the current transmission rate with some constant, usually 0.5. We use a similar approach to design the partitioning algorithm. The function $f\left[NU(t), NU(t-\tau)\right]$ contains the information about the change in the utility. Based on that change, it generates the appropriate change for the partitioning parameters. If we introduce a change indicator, $\xi \in \{0,1\}$, and an indicator for the direction of change, $\theta_{\mathrm{i}}(t) \in \{0,1\}$, the partitioning algorithm can be defined as follows:

$$\alpha_i(t) = \alpha_i(t-\tau) + \xi\left[\varepsilon_{inc}\theta_i(t) - \varepsilon_{dec}(1-\theta_i(t))\right] \qquad (8)$$

The only constraint is that the partitioning parameters need to be within the interval $a_i \in \{a_{i\,min}, 1\}$, where $a_{i\,min}$ defines the part of the capacity that is reserved for the traffic class i. In our model, $a_{h\,min}$ and $a_{s\,min}$ define the reserved bandwidth for real-time flows. Also, there is always a level of link bandwidth $\alpha_{e\min}C$ reserved for the elastic traffic.

ε_{inc} and ε_{dec} are additive parameters, for increase and decrease respectively, and their values are very important. Equation (8) clearly shows that our algorithm follows the additive increase, additive decrease control rule. The main idea is to always perform the change that will increase the utility of the active traffic flows.

Once the new partitioning parameter is calculated, all active traffic flows have to update their transmission rates. The calculation of the partitioning parameter as presented here assumed a centralised intelligence which does the calculation. The interesting thing is that, since all users have to change their transmission rates, the Dynamic Bandwidth Partitioning scheme is realised by a simple change in the rates of all active flows. The information about the overall traffic value can be sent from the network to the individual users, where individual users would need only the information about the change in the partitioning parameter.

3. Simulation Results and Analysis

In order to evaluate the efficiency of the DBP scheme, a special simulator has been developed. Simulation is done on the call level, with the traffic loads approximately the same for all three traffic classes. We observe average connection utility as a measurement of the performance of the scheme.

The results on Figures 4-7 show the comparison in the average connection utility between the DBP scheme and two other schemes. The first one is the *best-effort* scheme, in which there is no admission control, and all flows that are active on the link receive the same share of the link bandwidth. This gives much higher utilisation of the bandwidth, in terms of the amount of bandwidth used at each moment. However, Figure 2 clearly shows that the *average connection utility is greater* for the DBP scheme. Reasons for this come from the fact that the best-effort scheme does not provide any bandwidth guarantees for the hard real-time traffic. At lower traffic loads, there is a large amount of bandwidth available, and no partitioning mechanism is really necessary for the active traffic flows to receive maximal utility. At higher traffic loads, the great majority of hard real-time traffic flows does not receive the required bandwidth while active in the network, which makes these flows unusable to the users and so their utility becomes 0. This is shown on figures 5-7, which show the comparison of the average connection utility for each individual class. Even though the DBP scheme performs worse for the stream and elastic traffic, the difference for these classes is not as significant as the difference for the hard real-time traffic. The DBP scheme performs rather well for the hard real-time traffic, with only a small fraction (up to 5%) of the incoming traffic being rejected at the ingress point. This result clearly proves the point we argued previously, that in the multi-class environment network performance must not be only about accepting as much traffic as possible, but more about satisfying the needs and requirements of the flows.

The second scheme we used for the comparison is the *fixed partitioning.* In this scheme, the bandwidth is also partitioned, but now partitioning parameters are fixed, $a_h = a_s = a_e = 0.33$. It is obvious from the figures 4-7 that the fixed partitioning scheme is not adaptive, performing poorly for the hard real-time traffic. While in the area of small traffic load the dynamics is not that important, as shown on figure 4, as the load increases the *dynamic* partitioning scheme performs much better then the *fixed* partitioning.

There are a number of other interesting experiments that can be performed on a defined model. The implementation of such a model in the analysis of the bandwidth management in the environment of virtual private networks (VPN) is rather obvious. If we define VPN as the traffic aggregation with the same identification, Dynamic Bandwidth Partitioning can be used to partition the available bandwidth between a number of such aggregates. Naturally, a hierarchical DBP scheme can be defined here, with further partitioning between different traffic classes within each of the VPNs.

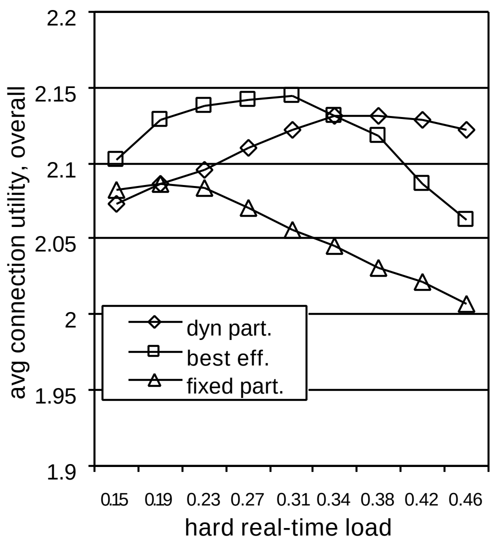

Figure 4 Comparison of the average connection utility, all classes

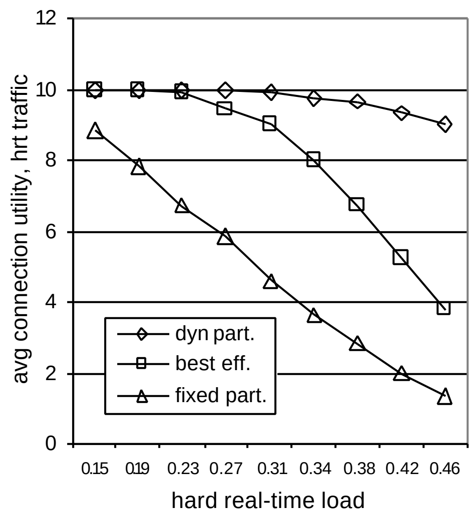

Figure 5 Comparison of the average connection utility, hard real-time traffic

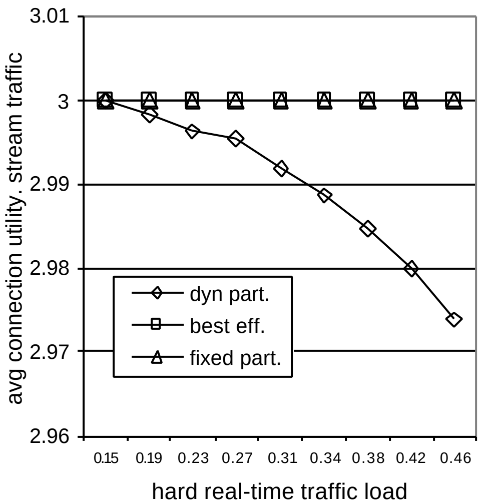

Figure 6 Comparison of the average connection utility, stream traffic only

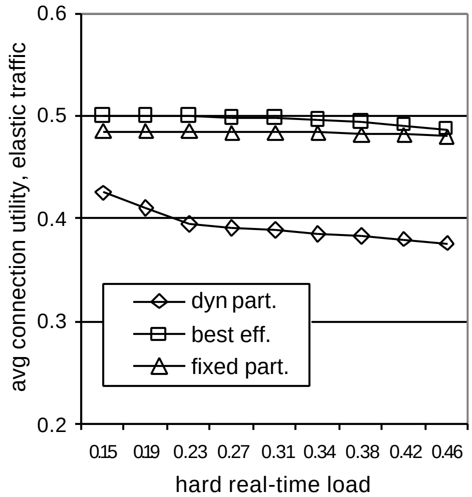

Figure 7 Comparison of the average connection utility, elastic traffic only

4. Conclusion and Future Work

The most important conclusion from the experiments presented in this paper is that the way we evaluate the performance of the network is crucial in the future Internet that needs to be oriented towards providing quality of service guarantees. This work presents a bandwidth allocation scheme based on optimising the network in terms of the end-user satisfaction, i.e. in terms of quality of service. The scheme is fairly easy to implement in the real network, considering there is only a limited number of parameters that can be translated into the weights in the scheduling mechanism in the intermediate routers.

The model created in this paper can be extended, making space for a large number of different experiments. Especially interesting work can be done in the implementation of the hierarchical DBP scheme, in which the link capacity would be partitioned between different virtual private networks, each serving a number of different traffic classes.

5. References

[1] L.Massoulie, J.Roberts, "*Bandwidth Sharing: Objectives and Algorithms*", IEEE INFOCOM 99, New York, USA

[2] V.Rakocevic, J.Griffiths, G.Cope, "*Dynamic Partitioning of Link Bandwidth in IP/MPLS Networks*", accepted for IEEE International Conference on Communications 2001, Helsinki, Finland, June 2001

[3] FP Kelly, AK Maulloo and DKH Tan, "*Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability*", Journal of Operational Research Society 49 (1998) 237-252

[4] B.Vandalore, W.Feng, R.Jain, S.Fahmy, "*A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia*", OSU Technical Report, OSU-CISRC-5/99-TR-14, available through http://www.cis.ohio-state.edu/¬jain/papers.html

[5] L.Breslau, S.Shenker, "*Best-Effort versus Reservations: A Simple Comparative Analysis*", ACM Computer Communications Review, vol. 28, pp. 131-143, September 1998

[6] Keith W. Ross, "*Multiservice Loss Models for Broadband Telecommunication Networks*", Springer, 1995

[7] Dah-Ming Chiu and Raj Jain, "*Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks*", Journal of Computer Networks and ISDN, vol.17, No.1, June 1989, pp. 1-14