Dynamic Partitioning of Link Bandwidth in IP/MPLS Networks

Veselin Rakocevic, John M Griffiths
Dept. of Electronic Engineering,
Queen Mary, University of London, UK
Mile End road, London E1 4NS
Email: {v.rakocevic, j.m.griffiths} @elec.qmw.ac.uk

Graham Cope
Fujitsu Telecommunication Europe Ltd, Birmingham, UK
Email: G.Cope@ftel.co.uk

Abstract: **Bandwidth allocation in the future multiservice global communication IP network presents a very interesting research issue. This paper presents the strategy of Dynamic Partitioning of link bandwidth in IP network. In the Dynamic Partitioning scheme the bandwidth of each link in the network is partitioned into two fractions, one for the low-priority data traffic, and one for the high-priority stream (real-time) traffic. The partitioning is defined by the partitioning parameter, which changes according to the traffic profile and intensity. An algorithm for the change of partitioning parameter is presented. The evaluation of the scheme is done based on a new metric, the *connection utility*, which is the measurement of the average end-user utility. Based on this measurement, the dynamic partitioning scheme is compared to several other bandwidth allocation schemes. Simulation results on a single link network model show the advantage of the dynamic link partitioning. Furthermore, the paper discusses the use of Multiprotocol Label Switching (MPLS) architecture for the implementation of the dynamic partitioning scheme.**

I. INTRODUCTION

The current Internet gives no end-to-end Quality of Service guarantees. It offers a single class of 'best-effort' service. Best-effort architecture has been very successful in supporting data applications, but new sophisticated audio-video Internet applications (video conferencing, distance learning, etc) require better and more reliable network performance. The Internet needs architectural improvements, which can bring the sophisticated adaptive real-time applications to the end-users with a guaranteed level of quality of service.

Multiprotocol Label Switching (MPLS) [1] technique raised a very large interest in recent time. MPLS was originally designed as a way of improving the forwarding speed of routers but is now emerging as a crucial standard technology that offers new capabilities for large-scale IP networks. The essence of MPLS is the generation of a short fixed-length *label* that acts as a shorthand representation of an IP packet's header. Labels are distributed between intermediate routers in a MPLS network by using the Label Distribution Protocol (LDP) [6]. MPLS is particularly interesting because it works as a powerful tool for traffic engineering. In MPLS-capable networks, traffic flows that belong to same traffic class and traverse the same path in the network are aggregated and transported via Label Switched Paths (LSP). In MPLS it is possible to reroute, modify or explicitly define the LSP, which provides large working space for traffic engineering. MPLS-capable network is able to guarantee certain levels of performance to end-users of real-time applications.

The way bandwidth is shared between concurrent flows is a major issue in Internet QoS design. It is interesting to note that new real-time Internet applications are not built to expect circuit-switched service. Instead, they are designed to adapt to the currently available bandwidth [8]. Although the advances in the adaptability of real-time applications have been tremendous, there are likely to be limitations to this adaptability. Nevertheless, we can say that the majority of traffic in Internet has some sort of 'elastic' behaviour. Real-time applications changed and became more adaptive, but they still require certain minimum level of 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 therefore very complex. The majority of the bandwidth sharing schemes presented in the literature [2] deal with the problem of bandwidth allocation in a single-service environment, when the entire Internet traffic is treated as data transfer. The objective is generally to use all available bandwidth while trying to achieve some fairness in the way the bandwidth is shared between different traffic flows.

This paper presents dynamic partitioning of link bandwidth, a new bandwidth allocation scheme. We are analysing the multiservice Internet environment, considering two broad traffic classes. The first traffic class is the *stream* (real-time) traffic class. Stream traffic flows result from audio and video applications and require the network to provide network performance guarantees (end-to-end delay, bandwidth). Stream traffic is considered to be the high priority traffic. The second 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.

The details of the scheme are presented in section 3 of the paper. The additional contribution of this work is in the evaluation mechanism. Instead of relying on traditional evaluation parameter of average bandwidth utilisation and fairness, we introduce a new metric, which is called *connection utility*. Connection utility presents the evaluation of the network performance measured by relative utility of the end-user of the traffic connection. It is almost impossible to precisely define utility functions, since the perceived utility is closely connected with each user's personal preferences. In this work we have tried to approximate the end-user's utility by defining only one utility function per traffic class. The new metric and the analysis of the utility function are presented in section 2. Section 4 analyses the simulation results done on a single-link network model. The dynamic bandwidth partitioning scheme is compared to three other bandwidth allocation schemes: complete partitioning, best-effort, and trunk reservation.

II. CONNECTION UTILITY

Each end user of an Internet application receives a certain *utility* (quality of service level) from the network application he is using. The user derives the utility from that application's performance in the network (e.g. the picture quality for video, the sound quality for audio application, etc.). We approximate end-user's utility by using *utility functions*. For simplicity reasons, we assume that QoS requirement of a connection is expressed using a single bandwidth parameter.

We consider two utility functions. For elastic traffic, the utility function is continuously concave, but not linear. Marginal utility of extra bandwidth is larger when the bandwidth is small. In the area of high bandwidth, adding extra bandwidth does not improve utility as much as when bandwidth is small. A function that can model this is (see Fig. 1):

$$\pi_e(c) = 1 - e^{-\frac{a_1 c}{c^+}} \quad (1)$$

c^+ denotes the peak rate for the elastic flow. For stream traffic we use the utility function (see Fig. 2):

$$\pi_s(c) = 1 - e^{-a_2 \frac{c^2}{a_3 + c}} \quad (2)$$

The expression in (2) comes from the work of Shenker and Breslau [4]. They used utility functions to analyse the problem of admission control in IP networks. When it comes to stream traffic, due to human perceptual factors, minimal levels of bandwidth are not very useful, so that at low bandwidths the marginal utility of additional bandwidth is fairly small. It is very important to underline the non-concavity of the utility function used for the stream traffic class.

We argue that the evaluation parameter for bandwidth allocation schemes in the multiservice environment cannot be just the number of accepted traffic connections and the average bandwidth utilisation. In the multiservice Internet stream traffic flows require performance guarantees, i.e. they require minimal level of allocated bandwidth. In order to compare efficiently different bandwidth allocation schemes in the new environment, a new metric is introduced

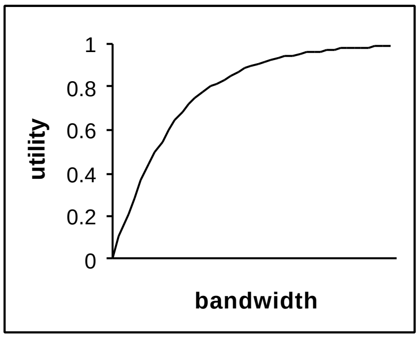

Fig.1. Utility function of elastic (best-effort) traffic flows

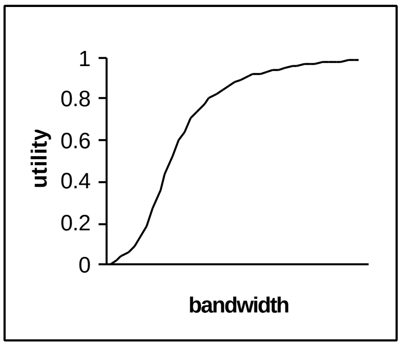

Fig.2. Utility function of stream traffic flows

The new metric is called *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 $i \in \{s, e\}$ that in time t is allocated capacity $c_i(t)$ in the network can be approximated with an integral

$$v_i = \frac{1}{T_{dur}} \int_0^{T_{dur}} \pi_i[c_i(t)]dt \quad (3)$$

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

III. DYNAMIC PARTITIONING

Let us consider a network with set of links L, where each link l has link capacity C_l. The dynamic partitioning scheme is defined by the set of partitioning parameters α_l. We assume that each link is *virtually* partitioned, so that elastic traffic uses $\alpha_l C_l$ of the link capacity, and stream traffic uses the remaining $(1-\alpha_l)C_l$. The partitioning parameter a_l changes with the number of active traffic connections on the link.

In the route calculation procedure only portions of links that are available to the traffic class of the connection come under consideration. After admission, connections use precomputed paths to reach their destination. This provides assurances that there will be no interaction between different traffic classes, i.e. sudden changes in burstiness of the elastic traffic will not affect the performance of the stream traffic.

For the purpose of the efficiency evaluation of the new scheme, a network simulator has been designed. Results presented in this paper describe the case of a very simple single-link network model. Traffic flows of both types are transported from the source to the destination. The experiments were made for different load environments, with the percentage of blocked stream connections under 5%, and with load of elastic and stream traffic being approximately equal.

The network model consists, therefore, of a single link, with capacity C. Traffic flows come as Poisson stream. Depending on the traffic class, traffic flows have exponentially distributed holding times (for stream type of traffic) or exponentially distributed sizes (for elastic type of traffic) with mean values $1/\mu_s$ and f_e, respectively. We denote the number of active stream and elastic flows on the link with n_s and n_e respectively. Each connection requires a certain minimum bandwidth c_s^- and c_e^-, and maximum bandwidth c_s^+ and c_e^+. Traffic flows belonging to the same traffic type get allocated some amount of bandwidth, c_s and c_e for stream and elastic traffic, respectively. Values that were used in the simulation are presented in Table 1.

The Dynamic Partitioning scheme has admission control for stream traffic flows only. An incoming stream flow is accepted on the link if there is space in the part of the link bandwidth that is reserved for stream traffic:

$$\textit{if } \left(n_s c_s^- \le (1-a)C\right) \textit{ then accept the new stream flow}$$
$$\textit{else reject it}$$

There is no admission control for elastic traffic. The link bandwidth is partitioned into two portions, $C = C_e + C_s$. The elastic portion is $C_e = \alpha C$ and the stream portion is $C_s = (1-\alpha)C$. Flows get the simple fair share of the available capacity, $c_e = C_e/n_e$ and $c_s = C_s/n_s$.

Parameters a_1, a_2, and a_3 define the shape of the functions. Values for a_1 and a_2 are chosen so that $\pi_i\left(c_i^+\right) \ge 0.99, i \in \{e, s\}$. Value for a_3 is chosen so that optimal minimal capacity for stream traffic is $c_s^- = 1$Mbps (see [4] for this calculation). With the data from Table 1 it is fairly easy to calculate these values, $a_1 \approx 4.61, a_2 \approx 0.62, a_3 \approx 2.29$.

Table 1 Values used in the simulation

C (Mbit/s)	**100**
a_{min}	**0.3**
$step_a$	**0.05**
t (for trunk resv.)	**0.7**
c_s^- (Mbit/s)	**5**
c_s^+ (Mbit/s)	**10**
$1/m_s$ (seconds)	**6**
c_e^- (Mbit/s)	**0**
c_e^+ (Mbit/s)	**100**
f_e (Mbits)	**20**

The partitioning parameter is calculated in the following way. Firstly reservations are made for the stream traffic flows that require reservations. The remaining capacity is then partitioned into two parts, with the goal of getting maximal utility from the traffic flows that are active on the link. The partitioning parameter is then increased or decreased by small fixed value depending on whether the newly calculated utility is larger or smaller then the previous one. The aggregated traffic utility on the link at each instant is:

$$V = V_{resv} + V_{part} \tag{4}$$

The utility from the reservations is fixed, $V_{resv} = n_s p(c_s^-)$, and the utility of the rest of the traffic depends on the partitioning parameter a.

$$V_{part} = n_s \Delta p_s(\Delta c_s) + n_e p_e(c_e) \tag{5}$$

where $\Delta c_s = \dfrac{(1-\alpha)\left(C - n_s c_s^-\right)}{n_s}$,

$$c_e = \frac{\alpha\left(C - n_s c_s^-\right)}{n_e}, \quad \Delta\pi_s(\Delta c_s) = \pi_s(c_s) - \pi_s\left(c_s^-\right)$$

The objective is to find the partitioning parameter that can give the highest traffic utility V_{part}. We introduce a correction parameter $corr \in \{-1, +1\}$ which shows the last direction of change for the partitioning parameter. Calculation is very simple. In time $t+1$:

- calculate $V_{part}(t+1)$
- if $\left(V_{part}(t+1) < V_{part}(t)\right)$ change $corr$
- $\alpha = \alpha + step_\alpha * corr$
- value for α has to be in the interval $\alpha_{\min} \le \alpha \le 1$

The idea is that the change of the partitioning parameter should always increase the existing utility. This technique for calculating the partitioning parameter is feasible, since partitioning changes only for the value of $step_\alpha$ in each iteration.

We assume MPLS as environment in which the dynamic partitioning scheme can be implemented. All stream traffic flows are transported via Label Switched Paths (LSPs). In the simplest network model we consider, there is no need for route computation for LSPs, but it is necessary to assume that traffic is transported via LSPs. Since partitioning changes in time, it is necessary to have the mechanism for dynamic renegotiations of the amount of bandwidth allocated to each traffic flow (i.e. of the size of LSPs). A recent IETF Internet draft [5] defines the mechanisms for dynamic resizing of LSPs. To modify the reserved bandwidth on a LSP, the ingress router sends a new LABEL_REQUEST message [6]. In that message, intermediate routers are informed about the necessary changes. The changes are encoded in the traffic parameter TLV (LDP [6] uses a Type-Length-Value (TLV) encoding scheme to encode the information carried in LDP messages). The intermediate router then reserves only the difference of bandwidth requirements, in order to avoid the double booking of bandwidth. These propositions show that MPLS is fully capable of supporting dynamic partitioning scheme. The only question that remains is whether MPLS label set-up process is quick enough to react on frequent requests for changes in the LSP size.

IV. RESULTS AND ANALYSIS

The simulation compared the dynamic partitioning scheme with three other bandwidth allocation schemes: best-effort, trunk reservation and complete partitioning. The schemes differ in the admission control and the bandwidth allocation procedure. The *Complete partitioning* scheme differs from dynamic partitioning scheme only in the fact that partitioning parameter is fixed, and defined to be $\alpha = \alpha_{\min} = 0.3$. *Best-effort* scheme has no admission control at all, all incoming flows are accepted on the link, where they are allocated the same amount of capacity. Because of the absence of the admission control, there are no guarantees that capacity allocated to a stream flow c_s will be greater than the required minimum, c_s^-. In the *Trunk Reservation* scheme an incoming elastic traffic flow is accepted if there is space in the tC portion of capacity, where t is the trunk parameter. For stream traffic, an incoming flow is accepted if there is space on the link, i.e. if $n_s c_s + n_e c_e < C$. Accepted flows are allocated the equal share of the link capacity.

If we consider average link bandwidth utilisation as only performance measurement, there is no apparent advantage of the partitioning schemes. (see Fig. 3). The advantage of best-effort and trunk reservation schemes is obvious, because these schemes do not limit the capacity for elastic traffic.

However, in this work we are comparing different resource allocation schemes on the basis of the connection utility. The dynamic partitioning scheme is optimising the partitioning parameter in order to generate the highest average connection value. Fig. 4 shows the comparison of the average connection utility. Partitioning strategies perform better, giving higher connection utility. It is not easy to quantify the margin in the average connection utility. We can get close to quantifying it by saying that the end-users will be approximately 10-15% *"happier"* if the partitioning schemes were used. Furthermore, we can see that the dynamic partitioning performs better then the complete partitioning.

The results on Fig. 4 are better understood after looking at Fig. 5 and Fig. 6, which present the comparison of the average connection utility for two traffic classes individually. A very interesting result here is that the dynamic partitioning scheme generates the *lowest* connection utility for the stream traffic. This may seem as a surprising result, since dynamic partitioning was introduced in order to 'protect' the stream traffic. The dynamic partitioning allocates the capacity in order to generate the highest overall connection utility. This means that sometimes stream traffic flows will suffer, while the overall utility increases. The 'suffering' of stream flows, however, is very small, especially when compared with the benefit that is generated for the elastic traffic flows (Fig. 6).

There are two important factors that influence these results. The first is the *traffic profile*, because we have chosen approximately equal traffic loads for both traffic types, with stream traffic load being slightly larger. However, if we increase the elastic traffic load, we would still generate higher connection utility in the dynamic partitioning scheme. Elastic traffic in the dynamic partitioning scheme is limited, not allowed to interfere with the stream traffic. The best-effort allocation schemes will not be able to limit the influence of the elastic traffic on the stream traffic, and the connection utility for the stream traffic will drop substantially, especially after the point when the allocated capacity reaches the 'knee' point on the stream utility function (see Fig. 2). Therefore, the traffic profile used for results in this paper was deliberately chosen to show that even if we have smaller elastic traffic load, the dynamic partitioning could still generate higher overall utility.

The second important factor is that in the partitioning algorithm presented here, there is no notion of prioritisation. Both utility functions are scaled to 1, and stream and elastic traffic flows are therefore treated as completely equal in the partitioning algorithm.

In future, experiments will be done with the scaled utility functions, where the prioritisation will be introduced. Furthermore, it will be interesting to see the results if used bandwidth is charged with different prices per bandwidth unit to flows from different traffic classes. Then the optimal bandwidth allocation scheme would have to find the bandwidth allocation that maximises the traffic revenue.

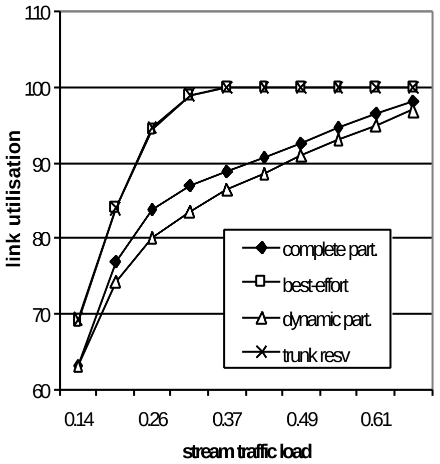

Fig.3. Comparison of the average bandwidth utilisation

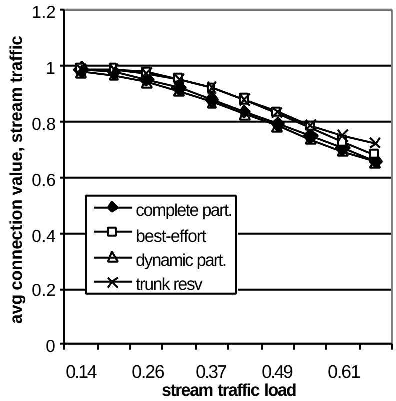

Fig.5. Average connection utility, stream traffic only

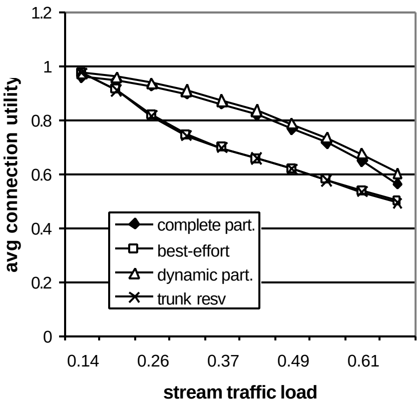

Fig.4. Comparison of the average connection utility

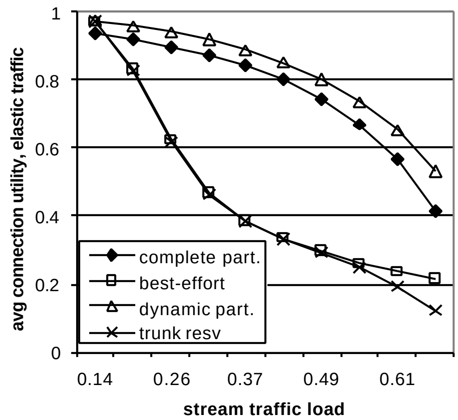

Fig.6. Average connection utility, elastic traffic only

V. CONCLUSION

This paper presented dynamic link partitioning, a new bandwidth allocation scheme for the mutliservice IP environment. In this scheme, network links are virtually partitioned in two parts, and traffic flows belonging to two traffic classes are independent. Admission control exists for the stream traffic flows only. The contribution of this work is in the new performance metric that is introduced, the connection utility, which is a time average utility derived by the end-user of a traffic flow . End-user's utility has been approximated by two different utility functions. Simulation results show the advantage of the dynamic partitioning scheme when the comparison is done based on the connection utility.

REFERENCES

[1] R.Callon, A.Viswanathan, E.Rosen, "*MPLS Architecture*", Internet Draft, August 2000

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

[3] V.Rakocevic, J.Griffiths, "*Value-based Evaluation of Strategies for Integration of Differentiated Internet Traffic Classes*", 16th UK Teletraffic Sy mposium, May 2000

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

[5] G.Ash, et. al, "*LSP Modification Using CR-LDP*", Internet Draft, February 2000

[6] B. Jamoussi, "*Constraint-based LSP Setup using LDP*", Internet Draft, July 2000

[7] P.Key, "*Optimal Control and Trunk Reservation in Loss Networks*", Probability in the Engineering and Informational Sciences, vol.4, 1990, pp. 203-242

[8] 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