

VBR over VBR: the Homogeneous, Loss-free Case

Silvia Giordano Jean-Yves Le Boudec Philippe Oechslin Stephan Robert

Laboratoire de Réseaux de Communication (LRC), EPFL CH-1015 Lausanne
{giordano,leboudec,oechslin,robert}@lrc.epfl.ch

Abstract

We consider the multiplexing of several variable bit rate (VBR) connections over one variable bit rate connection where the multiplexing uses a multiplexing buffer of size B . The VBR trunk is itself a connection and has a multidimensional connection descriptor, reflecting peak and sustainable rates. Given a cost function for the VBR trunk and a connection admission control (CAC) method for the input connections, we focus on the problem of finding the VBR trunk connection descriptor that minimizes the cost function and is able to accept a given set of VBR input connections. First, we show that, under reasonable assumptions on the cost function, the optimization problem can be reduced to a simpler one. Then we consider the homogeneous, loss-free case, for which we give an explicit CAC method. In that case, we find that, for all reasonable cost functions, the optimal VBR trunk is either of the CBR type, or is truly VBR, with a burst duration equal to the burst duration of the input connections. We motivate this study by showing that the optimal peak cell rate is fixed for a given B (thus for a CBR trunk), and a VBR choice can only be an improvement. Lastly, we take as example of cost function the equivalent capacity of the VBR trunk. Those results are expected to form the basis for a general method for a connection manager at a multiplexing node in an integrated services packet network.

Keywords: CAC, VBR, Virtual Path, Virtual Trunk, ATM, Integrated Services

1 Introduction

1.1 VBR over VBR, multiplexing and virtual trunks

We consider the multiplexing of several variable bit rate (VBR) connections (called “the input connections”) over one variable bit rate connection (called “the VBR trunk”). This occurs for example with ATM when a number of VBR virtual channel connections (VCCs) are multiplexed over one virtual path connection (VPC) [1] which is also of the VBR type. Another example is the multiplexing of several IP flows

with reservations (using a protocol such as RSVP [2] or ST.II [3]) over one ATM VCC.

We are interested in such multiplexing scenarios since we believe that reducing the number of connections (or reserved flows if RSVP is used) is a key feature that will be needed in all large scale networks. This is because connection handling cost, especially network management overhead, processing and memory is not negligible and increases almost linearly with the number of connections handled at one point. One solution is to aggregate connections at all points where this is possible. Connection aggregation simplifies all aspects of connection handling, provided that it is possible to dynamically change the attributes of the multiplexed connections [4] [5]. Aggregation can take place: (1) at an ATM node performing aggregation of VCCs over a VPC; (2) at an IP router aggregating several reserved flows over one ATM connection; (3) at an IP router aggregating several reserved flows over one reserved flow (tunneling). We call *Virtual Trunk* (VT) the connection that multiplexes a number of other connections; the word “trunk” refers to the fact that those connections also have attributes of network internal links, as defined for example with P-NNI [6]. In case (1), VTs are VPCs, in case (2), VTs are VCCs, and in case (3), they are IP tunnels with reserved resources. In this paper we use mainly ATM terminology, which applies strictly to case (1) only (VT can thus be equated to VPC). Translation to cases (2) and (3) should however be straightforward. We call *multiplexer* the node that multiplexes several input connections on one output VT.

Virtual trunks have traditionally been considered as Constant Bit Rate connections, though this restriction is not mandatory. In contrast, using other traffic types has obvious benefits. In this paper, we consider VTs of the VBR type. The rationale for using VBR VTs is the following. Integrated services packet networks provide resource reservation; however, they will not allocate its peak rate to every individual con-

nection, but perform resource overbooking. At the lowest level, overbooking uses both buffering (traffic peaks are temporarily stored) and statistical multiplexing (based on the expectation that traffic peaks do not all occur at the same time). [7]. If only CBR VTs are used, then access or edge nodes that multiplex small or medium numbers of connections are not able to perform a large amount of statistical multiplexing because efficient statistical multiplexing requires a small ratio between connection rate and the VT bit rate [8]. Further, the CBR VTs have to be allocated their peak rate by intermediate nodes that multiplex them in turn, since such nodes do not have any information about the individual input connections. With CBR VTs, overbooking is thus mainly performed by burst absorption at the multiplexer. In contrast, if VBR VTs are used, then it is possible to let bursts go through the multiplexer, and count on statistical multiplexing inside the network, where the number of connections and the trunk bit rates are larger. Quantifying this statement is not simple; it requires the definition of a connection admission control (CAC) method for connections over a VBR VT; it is object of ongoing work and beyond the scope of this paper.

In the rest of the paper we consider only VBR VTs and also simply refer to them as “VBR Trunks”. We consider also only input connections of the VBR type (which includes CBR but leaves aside ABR or UBR connection types). As explained in detail later, we focus on the problem of how to define the VBR trunk parameters in order to admit VBR input connections, while minimizing the cost of the VBR trunk.

A virtual trunk is considered as a connection by the network supporting it, and as a trunk by the connections it supports. Two sets of parameters are associated to virtual trunks: *connection descriptor* and *trunk state*.

- **Connection descriptor** is composed by the traffic and class parameters that describe the traffic characteristic of the VT when it is considered as one single connection. It is used by the supporting network to accept VTs.
- **Trunk state** (also called metrics) is the set of trunk state parameters reflects the static and dynamic characteristics of the VT. It is used for accepting connections on the virtual trunk.

In this paper, the connection descriptor for VBR VTs (and for the VBR input connections) consists in the sustainable bit rate (Mbit/s), the burst tolerance (s), the peak bit rate (Mbit/s), and the cell delay variation (s) [9], herein referred with the tuple (m, τ, R, CDV) (from Section 3 we neglect CDV).

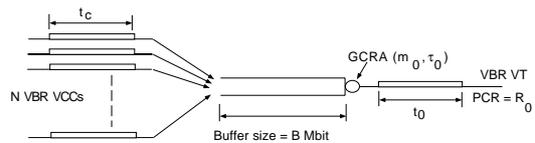


Figure 1: Reference Configuration

The trunk state depends on the CAC method used to accept input connections on the virtual trunk (this method is called here “VT-CAC”). In Section 3, we give a VT-CAC for the homogeneous, loss-free case, based on fluid models. VT-CACs for heterogeneous cases, and for supporting statistical multiplexing with losses, is the object of ongoing research not documented here.

1.2 The Problem Studied in This Paper

The reference configuration used in this paper is shown on Figure 1. A multiplexer, fed with a number of input connections of the VBR type, multiplexes them into one VBR connection (the VBR trunk), using a buffer of size B . There is no explicit assumption so far on the service discipline for the buffer, but we assume that the buffer output is regulated so that the resulting traffic conforms to $GCRRA(1/R_0, CDV_0)$ and $GCRRA(1/m_0, \tau_0 + CDV_0)$ [10].

The connection descriptor is multidimensional. For a given mix of input connections, there are several parameter sets that can carry them. This problem already exists for CBR virtual trunks, where several values of (peak rate, cell delay variation tolerance) are possible [11]. Here, we neglect cell delay variation tolerance issues and focus on supporting burst tolerance. At one end of the spectrum, it is possible to give a large value to the sustainable rate of the VBR trunk, at the limit, make it a CBR trunk; at the opposite end, a small sustainable cell rate, with a large burst tolerance is also possible. Lastly, the peak cell rate of the VBR trunk also influences all other parameters. Deciding among all these possibilities requires an additional criterion, like minimizing a cost objective. In our reference model, the cost objective is given by a function of the VBR trunk connection descriptors only. Given a cost function for the VBR trunk and a connection admission control (called VT-CAC) method for the input connections multiplexed over the VBR trunk, our problem is to find the VBR trunk connection descriptor that minimizes the cost function and is able to accept a given set of input connections.

We assume that the multiplexer accepts incoming connection requests in real time, and is able to change its own VT connection descriptor dynamically (by

negotiation with the network supporting the virtual trunk). An example of dynamic virtual path connections that uses ABT/DT [12] for reserving resources in the CBR VT case is defined in [5]. As a result, we require that the computation of the optimal VBR trunk connection descriptor be simple enough to be performed in real time.

In Section 2, we define the optimization problem formally, and show how it can be simplified by identifying a subset that necessarily contains all possible optimal solutions. This is true under reasonable assumptions for the cost function. In order to further progress in the solution, we need a VT-CAC method; in Section 3, as a starting point, we propose such a method for the simple case where all input connections are identical, and there are no losses. We study the properties of this VT-CAC and apply the results of Section 2. We obtain that the optimal VBR trunk in that case is either a CBR trunk, or a VBR trunk with burst duration equal to that of the input connections. We also obtain a simple relation (Eq. 5) that relates the total buffer size at the multiplexer, the burst tolerance of the input connections, of the VBR trunk, and the gain obtained by having a VT sustainable cell rate higher than that of the aggregate input. In Section 4, we complete the study in the case where the cost function is the equivalent capacity [13] of the VBR trunk (considered as one connection). The equivalent capacity is one cost function that reflects the cost of the VBR trunk to the supporting network. We also give in Section 4.3 a complete example illustrating the various aspects of the method presented in this paper. Section 5 summarizes the findings of this paper and outlines further applications and directions. The lengthy proofs not directly needed for the comprehension of the paper can be found in [14].

2 Reduction of the Optimization Problem

Having given the motivation for multiplexing a set of VBR connections on a single VBR connection we can now define in more detail the problem we investigate, and then perform a first reduction. Whenever a connection is added or removed from a VT trunk the trunk state z changes and the connection descriptors y may have to be modified to adapt to this change.

We assume that the VT-CAC for the trunk under consideration can be expressed by means of a real valued function $F(y, z)$, which is non-negative if the trunk with descriptors y can accept the traffic described by the trunk state z , and negative otherwise. We denote by $c(y)$ given the cost function which gives the cost of a connection with descriptors y . The prob-

lem described in the previous section can be formalized as follows. Given trunk state z we want to find among all connection descriptors y for which $F(y, z) \geq 0$, the connection descriptor y_{opt} which minimizes $c(y)$, if it exists. Define the *feasible region* $\mathcal{FR}(z)$ for every trunk state z as:

$$\mathcal{FR}(z) = \{\text{connection descriptors } y : F(y, z) \geq 0\}$$

We can now express our problem as follows:

$$\text{find } y_{opt} \in \mathcal{FR}(z) : \forall y' \in \mathcal{FR}(z) c(y') \geq c(y_{opt}) \quad (1)$$

It is convenient to use the *partial order* on the set of connection descriptors defined by:

$$y=(m, \tau, R, \text{CDV}) \preceq y'=(m', \tau', R', \text{CDV}') \text{ iff } \begin{cases} m \leq m' \\ \tau \leq \tau' \\ R \leq R' \\ \text{CDV} \leq \text{CDV}' \end{cases}$$

We now make the following (common sense) assumptions on the VT-CAC and cost functions, that will allow us to show that y_{opt} can be found in a set much smaller than $\mathcal{FR}(z)$.

Assumptions

1. $c(y)$ is non-decreasing with respect to all components of y , namely, if $y \preceq y'$ then $c(y) \leq c(y')$;
2. the VT-CAC function $F(y, z)$ is continuous with respect to y ;
3. the set of connection descriptors y for which $F(y, z)$ and $c(y)$ are defined is a closed, convex subset of $(\mathcal{R}^+)^d$ and it contains $y_0 \preceq y \forall y \in \mathcal{FR}(z)$ for which $F(y_0, z) \leq 0$.

Now we can proceed with the reduction of the set of values for y where the optimum is found, if it exists. Let us assume that there exists y_{opt} that solves the optimization problem 1. Assume that $F(y_{opt}, z) > 0$. Consider the set of $\alpha \in [0, 1]$ such that $F(y_0 + \alpha(y_{opt} - y_0), z) \geq 0$. By the assumptions on F , this set contains the value 1 since y_{opt} is in the feasible region and, by the assumptions on F , it is non-empty, closed, and thus compact. Therefore, it has a minimum value, call it a . If $F(y_0 + a(y_{opt} - y_0), z) > 0$, then necessarily $a > 0$ and by the continuity of $F(\cdot, z)$, we can find some a' such that $0 < a' < a$ and $F(y_0 + a'(y_{opt} - y_0), z) > 0$; we have a contradiction. Therefore $F(y_0 + a(y_{opt} - y_0), z) = 0$. Now by the non-decreasing property of c , we have that $c(y_0 + a(y_{opt} - y_0)) = c(y_{opt})$. We have thus shown so far that, if the optimum exists, then it is certainly reached at a point y with $F(y, z) = 0$.

We further reduce the set of possible solutions by considering *non dominated* points in $\mathcal{FR}(z)$. We say that $y \in Y$ is non-dominated in the set Y if

$\forall y' \in Y, y' \preceq y \Rightarrow y' = y$. Let us assume again that there exists y_{opt} that solves the optimization problem (1). Consider the set \mathcal{E} of connection descriptors y' that are feasible and dominate y , namely, $\mathcal{E} = \{y' \in \mathcal{FR}(z) : y' \preceq y\}$. By the non-decreasing property of c , all points in this set are also optimal. We now proceed with showing that at least one point in this set is non-dominated in $\mathcal{FR}(z)$. This set is closed, and by the third assumption, \mathcal{E} is non-empty and compact. Therefore, there exists at least one point y_1 in \mathcal{E} that minimizes the first coordinate. Call $p_i(y)$ the i^{th} coordinate of y . Call \mathcal{E}_1 the set of all y' in $\mathcal{FR}(z)$ that dominate y_1 . Obviously, $\mathcal{E}_1 \subset \mathcal{E}$ and $p_1(y) = p_1(y_1)$ for all $y \in \mathcal{E}_1$. By applying the same procedure recursively we build a sequence of decreasing sets \mathcal{E}_k and points y_k , such that $p_1(y) = p_1(y_k), \dots, p_k(y) = p_k(y_k)$. Ultimately, when k equals d , we have $\mathcal{E}_d = \{y_d\}$ and thus y_d is non-dominated in $\mathcal{FR}(z)$, and realizes the optimum for c .

In summary, we define the set $\mathcal{S}(z)$ (for Solution space) by:

$$\mathcal{S}(z) = \{y : F(y, z) = 0 \text{ and } \forall y' \in \mathcal{FR}(z) : y' \preceq y \Rightarrow y' = y\}$$

We have shown that if there exists a solution to problem 1, then it is in the solution space $\mathcal{S}(z)$. Our problem (Eq. 1) can thus be reformulated in the following way:

$$\text{find } y_{opt} \in \mathcal{S}(z) : \forall y' \in \mathcal{S}(z) : c(y') \geq c(y_{opt}) \quad (2)$$

This simplification is independent of the cost function, provided that the common sense assumptions are satisfied. Under this form it is, in general, easier to find y_{opt} . The solution space is a limited subset of the feasible region and it depends on a smaller number of variables since at least one can be expressed as function of the others from $F(y, z) = 0$. The condition that the elements of $\mathcal{S}(z)$ be non-dominated in $\mathcal{FR}(z)$ further restricts the solution space.

3 Homogeneous, Loss-less VT-CAC: General Results

Here we apply the reduction of the preceding section to the homogeneous case, namely when all input connections are identical. We give an explicit function F for that case, based on a loss-less (or worst case) CAC. We assume the worst case traffic of one input connection as the pattern consisting of a burst at the maximum rate for the maximum allowed time, followed by a silent period (ON/OFF). We know that this is not the general worst case [15], [16], and [17], but in the homogeneous case it requires the same amount of resources than the effective worst case, and it is easier to study.

This section clearly represents only a first step towards the resolution of the general case, however, it is complex enough to be worth investigating in detail.

First, we give an algorithm for VT-CAC, then we apply the results of Section 2.

3.1 VT-CAC function for the Homogeneous, Lossless Case: requiredBuf

In this Section we present a deterministic CAC function to decide the acceptance of VBR traffic, regulated by a shaping buffer with a fixed buffer size B , under no cell loss. We assume that buffers are large compared to the size of the cells, such that we can ignore the Cell Delay Variation Tolerance. We also assume that the traffic is homogeneous, meaning that all the input connections multiplexed on the VBR trunk have the same connection attributes: m, τ and R . The number of input connections is indicated by N . The VT attributes are thus defined by:

- Trunk state: $z = (N, m, \tau, R)$

- Connection descriptor: $y = (m_0, \tau_0, R_0)$.

The VT traffic is smoothed by the associated shaping buffer such that it conforms to $GCR A(1/m_0, \tau_0)$, as shown in figure 1.

We define $\text{requiredBuf}(y, z)$ as the buffer size required for accepting the input traffic on the VT with zero cell loss. Thus a connection can be accepted iff

$$B - \text{requiredBuf}(y, z) \geq 0$$

which defines the function F .

To avoid cell loss, we consider the worst case: the input connections are synchronized and send data all together at the peak cell rate until the GCR A reacts. At the beginning, the buffer is assumed to be empty. We analyze the problem from the aspect of the required buffer size, identifying six different situations. Two cases are evident:

- if $Nm > m_0$, the buffer length must be infinite, $\text{requiredBuf} = \infty$ (CASE 1)
- if $NR < m_0$, there is no need of buffer, $\text{requiredBuf} = 0$ (CASE 2).

Beyond these two cases, we examine the quantity of traffic that can be absorbed by the VT burst, and we deduce the buffer size required to buffer the remaining traffic. The burst lengths are given [10] by:

$$t_{burst} = \frac{\lfloor \tau / (T - 1/R) + 1 \rfloor}{R}$$

where $T = 1/m$. We assume that the effect of integer cells (the factor +1 in the numerator), is negligible compared to the burst size. When $NR < R_0$, the

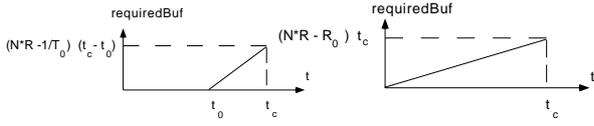


Figure 2: Buffer filling when $NR < R_0, t_0 < t_c$ (CASE 3), and when $NR > R_0, t_0 > t_c$ (CASE 6), respectively.

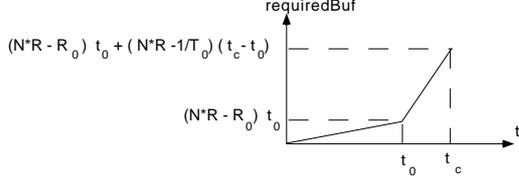


Figure 3: Buffer filling when $NR > R_0, t_0 < t_c$ (CASE 5).

burst length of the VT is considered for traffic equal to NR , because this is the maximum traffic generated by input connections. Thus, the burst length of the VT is given by:

$$t_0 = \tau_0 m_0 / (NR - m_0)$$

When $NR > R_0$, the burst length of the VT is considered for traffic equal to R_0 , because this is the maximum traffic that the VT can absorb. Thus, in this case, the burst length of the VT is given by:

$$t_0 = \tau_0 m_0 / (R_0 - m_0)$$

The burst length of the input connections is given by:

$$t_c = \tau m / (R - m)$$

Either $t_0 \geq t_c$ or $t_0 < t_c$, moreover, we have to consider $NR > R_0$ and $NR < R_0$:

- $NR < R_0, t_0 < t_c$ (CASE 3, Figure 2)

We see easily that:

$$\text{requiredBuf} = (NR - m_0)(t_c - t_0)$$

- $NR < R_0, t_0 > t_c$ (CASE 4)

$$\text{requiredBuf} = 0$$

- $NR > R_0, t_0 < t_c$ (CASE 5, Figure 3)

Figure 3 shows that:

$$\text{requiredBuf} = (NR - R_0)t_0 + (NR - m_0)(t_c - t_0)$$

- $NR > R_0, t_0 > t_c$ (CASE 6, Figure 2)

Figure 2 shows that:

$$\text{requiredBuf} = (NR - R_0)t_c$$

RequiredBuf is thus defined by the following algorithm:

Algorithm 1 : requiredBuf

if $Nm > m_0$ then requiredBuf = ∞ CASE 1

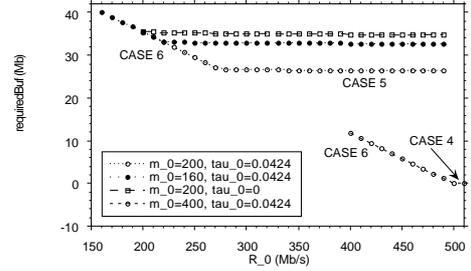


Figure 4: Evolution of requiredBuf versus R_0 in the worst case with $\tau=0.0424$ sec., $m = 3$ Mbit/s, $R=10$ Mbit/s, $N=50$

else if $NR \leq m_0$ then requiredBuf = 0 CASE 2

else if $NR \leq R_0$ then

if $t_0 < t_c$ then requiredBuf = $(NR - m_0)(t_c - t_0)$

CASE 3

else requiredBuf = 0 CASE 4

else if $t_c \geq t_0$ then

requiredBuf = $t_0(NR - R_0) + (t_c - t_0)(NR - m_0)$

CASE 5

else requiredBuf = $t_c(NR - R_0)$ CASE 6

3.2 Analysis of the RequiredBuf Function

RequiredBuf has some interesting aspects that we discuss in this section. In Figure 4, 5, and 6, we have plotted requiredBuf versus each of the three connection attributes of the VT m_0 , τ_0 , and R_0 . Analyzing the curves in Figure 4, we note that R_0 affects the buffer size only for values smaller than the rate of the input connections burst ($NR = 500$) and the burst length of the VT smaller than the the burst length of the input connections. In this case requiredBuf decreases when R_0 increases. The slope of the curve is $\partial \text{requiredBuf} / \partial R_0 = -t_c$, constant and negative for every value of m_0, τ_0 . In the other cases requiredBuf remains constant for any value of R_0 , therefore it is useless to increase R_0 . The slope of the curve is always zero.

Note that R_0 must always be larger than or equal to m_0 . When $R_0 = m_0$ or $\tau_0 = 0$, the type of the VT connection is CBR. The buffer has to absorb all the bursts from the input connections exceeding m_0 . In these cases requiredBuf only depends on m_0 .

As shown by Figure 5 and Figure 6, m_0 and τ_0 affect requiredBuf only for cases in which $t_0 \geq t_c$ because m_0 and τ_0 influence the burst length and not the rate of the VT. For this reason, after having reached the equality between the burst lengths, any increase of m_0 or τ_0 is not significant. In fact, for smaller input connections burst lengths (CASE 3 and CASE 5) $\partial \text{requiredBuf} / \partial (m_0) = -t_c - \tau_0$, and $\partial \text{requiredBuf} / \partial \tau_0 = -m_0$ in both cases. In the other

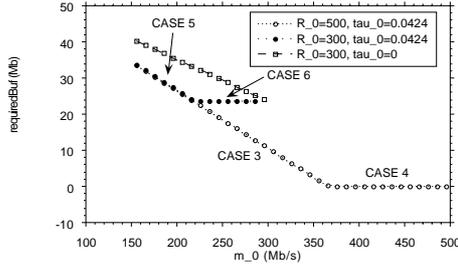


Figure 5: Evolution of requiredBuf versus m_0 in the worst case with $\tau=0.0424$ sec., $m = 3$ Mbit/s, $R=10$ Mbit/s, $N=50$

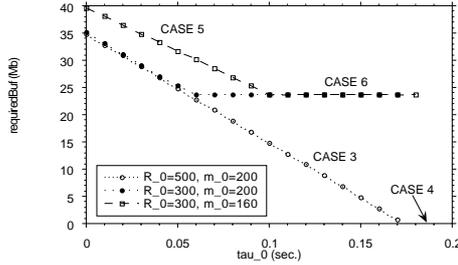


Figure 6: Evolution of requiredBuf versus τ_0 in the worst case with $\tau=0.0424$ sec., $m = 3$ Mbit/s, $R=10$ Mbit/s, $N=50$

cases, $\partial \text{requiredBuf} / \partial (m_0) = \partial \text{requiredBuf} / \partial \tau_0 = 0$.

The feasible region for a fixed buffer size B , VT connection descriptor $y = (m_0, \tau_0, R_0)$, N input connections with attributes $\{m, \tau, R\}$, thus VT trunk state $z = (N, m, \tau, R)$, is given by:

$$\mathcal{FR}(z) = \{y : B - \text{requiredBuf}(y, z) \geq 0\}$$

In Section 2 we showed that, under some reasonable assumptions on the VT-CAC and cost functions, the optimization problem can be simplified. We will now try to apply this simplification and to find the solution space when the VT-CAC is requiredBuf.

3.3 Solution Space $\mathcal{S}(z)$

In Section 2 we used two common sense assumptions on $F(y, z)$ to reduce our optimization problem. The assumption of continuity of requiredBuf is proven in [14]. The other assumption, namely that there be a descriptor y_0 in the definition domain of $F(\cdot, z)$ for which $F(y_0, z) \leq 0$, requires some special attention. The domain of connection attributes for which requiredBuf is defined is $R_0 \geq m_0$ and $\tau_0 \geq 0$ by definition, and $m_0 \geq Nm$ for reasons of stability. Depending on the value of B and z it may happen that there is no y_0 for which $F(y_0, z) \leq 0$. Physically, this is the case when the shaping buffer is large enough to absorb all the bursts of the input connections ($B \geq N\tau m$).

Any VT with a sustainable rate larger or equal to Nm is thus sufficient to support the output traffic without loss, $F(y, z) > 0$ for all y of the definition domain of $F(\cdot, z)$ and the reduction is not applicable. However, since all y satisfy the CAC function, y_{opt} is simply the lower bound of the definition domain of requiredBuf. Thus we have that for B larger than $N\tau m$, the optimal VT connection descriptor is a CBR connection with a sustainable rate of Nm :

$$B > N\tau m \Rightarrow y_{opt} = (Nm, 0, Nm) \text{ (CBR)} \quad (3)$$

When $B \leq N\tau m$ we can always find a y_0 for which $F(y_0, z) \leq 0$, for example $(NR - B/t_c, 0, NR - B/t_c)$. The assumptions of Section 2 are thus valid and we can apply our reduction: the solution space \mathcal{S} is a subset of \mathcal{FR} and contains only the non dominated y in \mathcal{FR} for which $F(y, z) = 0$.

From the equality $B - \text{requiredBuf}(y, z) = 0$, we can express one variable in function of the others. We chose to express τ_0 in function of m_0 and R_0 (see [14]):

$$B - \text{requiredBuf}(y, z) = 0 \Rightarrow \tau_0(m_0, z) = \frac{NRt_c - B}{m_0} - t_c$$

We see that τ_0 is independent of R_0 and can be expressed in function of m_0 and the trunk state only. Furthermore we see that for $m_0 > NR - B/t_c$, $\tau_0(m_0, z)$ must be negative and outside the definition domain of requiredBuf for F to be zero. Physically this means that, for a value of m_0 above $NR - B/t_c$, the part of the input bursts which is above m_0 does not fill the shaping buffer. The implication is that for a connection descriptor y , m_0 must be less or equal to $NR - B/t_c$ for y to be in the solution space \mathcal{S} .

$$F(y, z) = 0 \Rightarrow m_0 \leq NR - B/t_c$$

We also find that, given m_0 and $\tau_0(m_0, z)$, $F(y, z)$ is 0 for any value of R_0 larger than $NR - B/t_c$, as demonstrated in [14]. The solution space is made of non dominated elements of $F(y, z) = 0$, thus R_0 must be $NR - B/t_c$, which dominates all larger values of R_0 . Note that this value is always in the definition domain of requiredBuf since m_0 is can not be larger than $NR - B/t_c$.

We can thus express the solution space by the following equation:

$$\mathcal{S} = \{(m_0, \frac{NRt_c - B}{m_0} - t_c, NR - B/t_c)\} \quad (4)$$

$$\text{with } Nm \leq m_0 \leq NR - B/t_c \text{ and } B < N\tau m$$

Note that for m_0 being at its upper bound, the VT becomes a CBR connection.

Discussion: In this section we have seen that when the shaping buffer exceeds the size of the input bursts ($N\tau m$) the connection descriptors which minimize the cost of the VT are simply those of a CBR connection with sustainable rate equal to the sum of the sustainable rates of the input (Nm). If the shaping buffer is smaller than the input bursts, we can reduce feasible region of `requiredBuf` from an open three-dimensional space to a limited one-dimensional solution space. The condition that $F(y, z)$ must be zero allows to express one variable (τ_0) in function of one other (m_0). Furthermore, the condition that the solution space be made of non-dominated elements allows to fix the third variable (R_0) to the lowest bound, which is independent of the other two variables. One physical implication of this is that R_0 being at the lowest bound, the duration of the burst at the output of the multiplexer is at the highest bound, which is when the output burst has the same duration as the input burst (see Corollary 1 in [14]). From this property, we deduce a simple equation which is valid for the optimal solution for any cost function that conforms to the assumptions of Section 2. The equation relates the buffer size at the multiplexer, the burst tolerance of the input connections and of the VT, and the sustainable cell rate of the VT, thus for $y=y_{opt}$:

$$B = N\tau m - \tau_0 m_0 - (m_0 - Nm)t_c \quad (5)$$

In particular, when $m_0 = Nm$, we have that the burst of the input traffic is completely absorbed by the buffer and the burst of the VPT.

4 Homogeneous, Loss-less VT-CAC: $c(y)$ equal to Equivalent Capacity

In this section we continue the analysis of the homogeneous case with a specific example as cost function. We consider a system that uses `requiredBuf` for the input connection admission control over VTs, and the *Equivalent Capacity* function [13] for the cost function. We show how the computation is reduced and simplified by applying the results of Section 2.

4.1 Cost Function: Equivalent Capacity

The cost function we use here is the Equivalent Capacity function defined in [13]. It is defined as the rate necessary for achieving a desired buffer overflow probability ϵ , on a given physical link, given a physical link buffer size X . Note that X is *not* related to the buffer size, noted B , at the multiplexer. In the context of this Section, it should be simply interpreted as a parameter of the cost function that influences the cost of a given connection descriptor of the VT. If X is very large, then the cost is mainly influenced by the VT sustainable rate m_0 ; if it is very small, then it is

mainly influenced by the peak rate R_0 . In contrast, B influences the output of the `requiredBuf` function.

The equivalent capacity c_0 , for a VT connection descriptor $y = (m_0, \tau_0, R_0)$ is given by:

$$c_0 = R_0 \frac{Y_0 - X + \sqrt{[Y_0 - X]^2 + 4X\rho_0 Y_0}}{2Y_0} \quad (6)$$

$$\text{where } Y_0 = \ln\left(\frac{1}{\epsilon}\right)\tau_0 m_0 \text{ and } \rho_0 = \frac{m_0}{R_0} \quad (7)$$

We do not prove the monotonicity of Equivalent Capacity. We just argue that, as a typical effective capacity function, *EC* must increase when any one of its parameters increase.

4.2 Application of the Space Reduction

In Section 3.3, we have identified the solution space $\mathcal{S}(z)$. We can now formulate the general solution for in this specific case as an optimization problem depending on one single variable m_0 , as follows:

$$\text{find } m_0 \in [Nm, NR - B/t_c] \text{ that minimizes } g(m_0) \quad (8)$$

where $g(m_0)$ is given by:

$$\frac{\ln\left(\frac{1}{\epsilon}\right)(t_c(NR - m_0) - B) - X + \sqrt{\frac{4Xt_c m_0}{B - NRt_c}(B + t_c(NR - m_0)) + (\ln\left(\frac{1}{\epsilon}\right)(t_c(NR - m_0) + B - X))^2}}{2t_c \ln\left(\frac{1}{\epsilon}\right)(t_c(NR - m_0) + B)/(B - t_c NR)} \quad (9)$$

This function is a non decreasing function, thus, in absence of constraints on m_0 , the solution of the optimization problem can be easily found for the lower bound of m_0 ($m_0 = Nm$), as illustrated in the next section.

4.3 Numerical Example

Here we provide three numerical examples of the optimization problem where the cost function used is equivalent capacity, and also show the complete interaction of the elements of the method defined in this paper.

In the first example, the parameters used for defining the equivalent capacity function are $X = 100$ Mb, and the cell loss probability $\epsilon = 1.0E-05$. The capacity of the shaping buffer at VT1 is $B = 2Mbit$, and defines a feasible region $\mathcal{FR} = \{\text{requiredBuf}(y, z) \leq 2\}$. The current attribute values for the VT are:

$$z = (N, m, \tau, R) = (8, 3, 0.77, 20),$$

$$y = (m_0, \tau_0, R_0) = (87.2162, 0.1529, 145.2941),$$

The equivalent capacity of the VT is thus 89.3416 Mbit/sec.

Assume now that two new input connection requests arrive at the virtual trunk. By accepting the two new connections, the trunk state z would become:

$$z' = (N, m, \tau, R) = (10, 3, 0.77, 20);$$

This would move the trunk attribute out of the feasible region. (requiredBuf(y, z')=7.44 Mb). Thus we want to find a new connection descriptor such that the new connections can be accepted, namely, the new VT connection attributes belong to $\mathcal{FR}(z)$ and the cost function on the links is minimized.

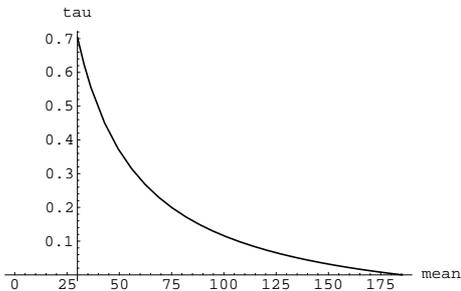


Figure 7: The solution space $\mathcal{S}(z)$ for the numerical example. The third parameter R_0 is equal to $NR - B/t_c = 185.2941$ Mb/s.

From Section 4.2 we know that our solution is the one that minimizes $g(m_0)$. We set R_0 to its lower bound $R_0 = NR - B/t_c = 185.2941$ Mb/s. The resulting solution space is plotted in Figure 7. Minimization of g is found by minimizing m_0 . In this case, we find that the minimum of cost is 127.079 Mb/s; it is obtained for (30, 0.704, 185.2941), as shown on the dashed curve in Figure 8.

In a second example, the traffic input is the same, but we assume a larger value for the parameter X ($X = 500$ Mb) of the equivalent capacity function, which means that the cost of a large burst tolerance is not as high as in the first example. All other parameters are kept unchanged. Thus we expect that the optimal solution will have a smaller cost. The numerical result confirm this expectation: the minimum ($c = 47.0774$) obtained for (30, 0.704, 185.2941) is smaller than before. This case is represented by the dotted curve in figure 8.

In the last case we assume a still larger value for X ($X = 1000$ Mb) with all other parameters kept unchanged. As expected, the cost of the optimal solution (30, 0.704, 185.2941) still decrease, and becomes very close to m_0 ($c = 32.6137$). As expected, in figure 8, the curve relative to this case, the solid curve, is an increasing straight line.

5 Conclusion

We have analyzed in this paper one of the consequences of having VBR trunks in an integrated services network, which we argued is an essential feature for reducing connection handling costs. We have

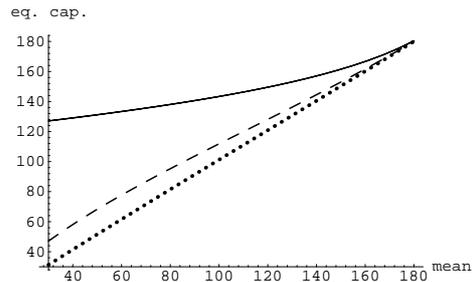


Figure 8: The cost function on the solution set $\mathcal{S}(z)$, for three different values of the cost function parameter X : $X = 100$ (dashed curve), $X = 500$ (dotted curve), $X = 1000$ (solid curve). Small values of X give a high cost to VTs with large burst tolerance. The optimal VT parameter is obtained for the minimum of the sustainable bit rate (“mean” on the figure). If bursts are more expensive (smaller X) then the optimal virtual trunk with the same sustainable cell rate has higher cost. The peak rate optimal value is fixed by the results of Section 3.

formalized the problem of determining optimal VBR trunk connection descriptors, given a CAC method for accepting input connections on the VBR trunk (VT-CAC). We have shown how the optimization problem can be reduced to a simpler problem, and applied the result to the homogeneous case. For the specific case of a cost function equal to the equivalent capacity, we have derived a complete analysis, and we have showed how easily can be find the solution. We have shown that, for the homogeneous case, and for all reasonable cost functions, the optimization problem can be reduced to a one-dimensional problem. we expect this result to hold in all cases, even with lossy VT-CACs, as long as CDV tolerance is neglected. It is this result that makes us expect that simple, real time optimization procedures can be obtained in the general case.

Further work is ongoing to extend the results to the non-homogeneous case, to include the CDV [18], and to support non-loss-free forms of VT-CAC. In the most general case, we believe that non-loss-free VT-CACs will require some form of traffic measurement, and this is the direction that we are pursuing [19]. We are using the direction, and the results described in this paper to form form the basis for such studies.

Related problems that need to be solved for a complete solution include:

- A strategy for renegotiating VBR trunk connection descriptors: once we know what the optimal parameters are for a given input traffic, when is it a good idea to actually setup the VBR trunk with such parameters? What actions would be

taken if the renegotiation fails?

- Relation with routing algorithm: if the control unit performing VT-CAC has access to the routing information of the network supporting virtual trunks, then it may exploit that information to decide whether or not the VBR trunk should be modified; this may also affect the choice of one VBR trunk in cases where several are available at the multiplexer.
- Determination of cases where connection grouping (as assumed in the paper) has significant advantages from a resource usage point of view.

References

- [1] ITU Telecommunication Standardization Sector - Study group 13, *ITU Recommendation I.371, Traffic Control and Congestion Control in B-ISDN*, 1995.
- [2] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zapalla, "RSVP : A New Resource Reservation Protocol," *IEEE Network*, September 1993.
- [3] C. Topolcic, *Experimental Internet Stream Protocol, Version 2, (ST-II)*. IETF, 1990.
- [4] E. Gauthier and S. Giordano and J-Y. Le Boudec, "Reduce Connection Awareness," *High-Speed Networking for Multimedia Applications*, W. Effelsberg, O. Spaniol, A. Danthine, D. Ferrari (eds.), 1995.
- [5] E. Gauthier and J-Y. Le Boudec, "Scalability Enhancements for Connection-Oriented Networks," *International Zurich Seminar on Digital Communications proceedings*, 1996.
- [6] The ATM Forum, *P-NNI 1.0 Specification*, 1996.
- [7] P. Varaiya and J. Walrand, *High-Performance Communication Networks*. Morgan Kaufmann, San Francisco, October 1996.
- [8] J. Mignault, A. Gravey, and C. Rosenberg, "A Survey of Straightforward Statistical Multiplexing Models for ATM Networks," *First International "ATM Traffic Expert" Symposium, Basel*, 1995.
- [9] R. Jain, *Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey*. ATM Forum, invited submission to Computer Networks and ISDN Systems, January 1995.
- [10] The ATM Forum, *ATM User-Network Interface Specification, Version 3.1*, 1994.
- [11] F. Braun, "Managing the Traffic Streams in an Optimal Way," *Com.Tec ATM*, 1995.
- [12] P. Boyer and D. Tranchier, "A Reservation Principle with Applications to the ATM Traffic Control," *Computer Networks and ISDN Systems*, vol. 24, pp. 321-334, 1992.
- [13] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE JSAC*, vol. 9 (7), pp. 968-981, 1991.
- [14] S. Giordano and J-Y. Le Boudec and P. Oechslin and S. Robert, "VBR over VBR: the Homogeneous, Loss-free Case," Technical Report 96/199, DI-EPFL, CH-1015 Lausanne, Switzerland, July 1996.
- [15] E. Aarstad, "A comment on Worst Case Traffic," *COST 242 TD*, 1992.
- [16] D. C. Lee, "Effects of Leaky Bucket Parameters on the Average Queueing Delay: Worst Case Analysis," *IEEE Infocom proceedings*, 1994.
- [17] N. Yamanaka, Y. Sato, and K. Sato, "Performance Limitation of Leaky Bucket Algorithm for Usage Parameter Control and Bandwidth Allocation Methods," *IEICE Trans. Communications*, 1992.
- [18] S. Giordano and J-Y. Le Boudec, "Guaranteed QoS for VBR over VBR," *In preparation*, 1997.
- [19] A. Ziedinsh and J.-Y. Le Boudec, "Adaptive CAC Algorithms," *Immune project report, LRC-EPFL Internal Report*, 1996.