A CAC algorithm for VBR Connections over a VBR Trunk

Jean-Yves Le Boudec, Arnis Ziediņš
Laboratoire de Réseaux de Communication (LRC)
Ecole Polytechnique Fédérale de Lausanne

Technical report N0 97/219

Abstract

We propose an algorithm for connection admission control (CAC) in the case where connections are of the variable bit rate (VBR) type and the trunk carrying the multiplexed VBR connections is itself a VBR connection. The latter assumption occurs for example when a number of ATM connections is multiplexed over one VBR virtual path connection, or when a number of IP/RSVP flows is multiplexed over one VBR virtual channel connection. The algorithm is based on the proportional allocation method. We apply the algorithm to compare on some examples the multiplexing gain obtained by (1) aggregating a number of VBR connections into a VBR trunk (2) into a CBR trunk or (3) without aggregation. These preliminary results show that the use of VBR trunks makes it possible to aggregate a number of connections into one without sacrificing bandwidth efficiency.

Keywords: Call Admission Control (CAC), VBR, Traffic Aggregation.

1 Introduction

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) [13] which is also of the VBR type. Another example is the multiplexing of several IP flows with reservations (using a protocol such as RSVP [14] or ST.II [15]) over one ATM VCC.

We call here trunks the connections that multiplex a number of other connections; this refers to the fact that those connections also have attributes of network internal links, as defined for example with P-NNI [18].Trunks have traditionally been considered as Constant Bit Rate (CBR) connections, but this restriction is not mandatory. In contrast, using other traffic types has obvious benefits. In this paper, we consider trunks of the VBR type. The rationale for using VBR trunks is the following. Integrated services packet networks provide resource reservation; however, they do not allocate its peak rate to every individual connection, but, in contrast, perform some 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 [4]. If only CBR trunks 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 [19]. Further, the CBR trunks 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 trunks, overbooking is thus mainly performed by burst absorption at the multiplexer. In contrast, if

VBR trunks 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 requires specific assumptions about the method used for connection admission control (CAC) and is done for some limited cases in Section 6.

We propose and study here a connection acceptance algorithm for VBR connections over a VBR VT. This problem has received a lot of attention for CBR trunks, but hardly any for VBR trunks [3]. In [6], it is shown that the concept of effective bandwidth extends to VBR trunks in the form of a dual condition bearing on effective bandwidth and effective mean rate, at least for large buffer asymptotic. This supports the idea of resource exchangeability introduced for CBR trunks in [7], and indeed the algorithm we propose here is based on the method of proportional allocation, originally developed for CBR trunks in [7].

We neglect the cell scale phenomenon due to the discrete time nature of ATM cells. We model the flow of data as a fluid and thus ignore the cell delay variation parameter, (which aims at quantifying the impact on buffer sizes of the cell scale). The model is the following:

- Input traffic is sent to a multiplexor with a buffer of size X.
- The output of the multiplexor is a leaky bucket regulated VBR trunk, with peak rate P, sustainable rate M, and burst tolerance B expressed in bits.
- The input connections are leaky bucket regulated VBR connections. There are I types of input connections. An input connection of type i has peak rate p_i, sustainable rate m_i, and burst tolerance b_i expressed in bits.

By analogy with [7], we start by defining in Section 2 a CAC algorithm that is loss-free, at least as far as the fluid model holds. This algorithm accounts for the multiplexing gain obtained from buffering, but not from statistical multiplexing [10]. Then in Section 3 we extend the algorithm to a statistical one, that accepts a specified data loss, under the assumption that input connections are independent and the starting phases of the connections are uniformly distributed. The method uses a large-deviation approximation. The algorithm presented in Section 3 requires some conditions (smoothing conditions) on the input parameters. Removing these restrictions is technically feasible, but it is not presented in the paper. The algorithm is illustrated in Section 4 with several numerical examples.

As in [7], the evaluation of the loss probability relies on the assumption that the worst case input traffic for this CAC is for on-off, leaky-bucket regulated sources. There are indications that on-off sources are not always worst case for some performance indices: in [8, 9], worst cases are obtained for sources [11] that start bursting at the peak rate and then remain at the sustainable rate. We do not have a proof that, in our case, on-off sources are worst case, although this appears to be intuitively true; however, we are able to show on some numerical examples that on-off sources are the worst cases among all sources as in [11].

Lastly, we use the CAC algorithm to compare the bandwidth utilization gains that can be achieved by the use of VBR trunks. We compare three scenarios: (a) N connections are treated individually by a network node (no connection grouping) (b) the same N connections are grouped into large connections used as CBR trunks and (c) same N connections are grouped into large connections used as VBR trunks. In Section 6 we find on some examples that, if the parameters of the VBR trunks are set appropriately, it is possible to obtain better bandwidth utilization in case (c). Connection grouping is a means to reduce processing costs per connection in a network offering resource reservations, since grouped connections (the trunks in our paper) are seen by intermediate nodes as one single connection [16, 17]. The initial results in this last Section show that, by using VBR trunks, connection grouping is possible without loss of bandwidth utilization.

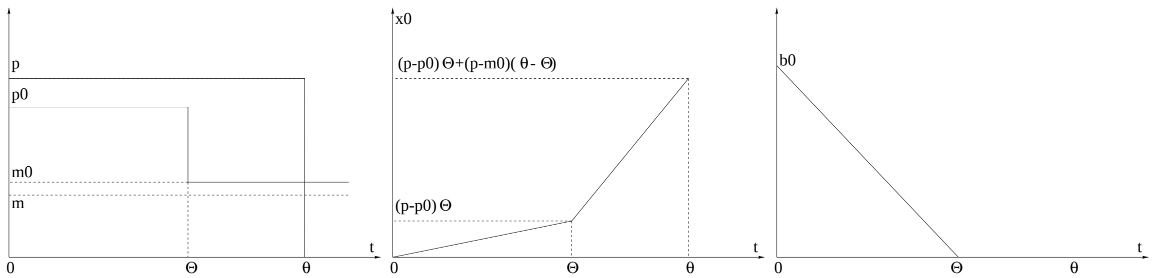
2 Lossless multiplexing

In this Section we define a a lossless CAC for VBR connections over a VBR trunk. The lossless CAC is based on the exchangeability of the network resources. A resource exchangeability principle is introduced in [7] for the CBR output trunk: two resources, bandwidth C and buffer X are available, and every input connection is allocated a fraction of the two resources.

We apply the method to our case. There are four resources to allocate peak rate P, mean rate M, burst tolerance size B and buffer X. The first step in the method is to associate to every connection a *virtual trunk* system, which is a scaled down, virtual copy of the real trunk and buffer system. The virtual trunk system for connection type i has output trunk parameters $\{p_{0i}, m_{0i}, b_{0i}\}$ and virtual buffer size x_{0i}. The proportional allocation method means that the parameters satisfy the three following equations

$$\frac{p_{0i}}{P} = \frac{m_{0i}}{M} = \frac{x_{0i}}{X} = \frac{b_{0i}}{B} \tag{1}$$

Note at this stage that the virtual trunk systems are still underspecified, as there are 3 equations for 4 parameters. The last consists in setting the virtual buffer size x_0 such that there is no data loss in the virtual trunk system. A general formula for the buffer size required for one VBR connection over a VBR, buffered trunk can be found in [11], where the formula is complex and depends on a large number of cases. Here, we can use a simpler formula because we impose additional constraints to the virtual trunk system. Indeed, we impose that the virtual trunk systems makes the (virtual) output connection smoother than the input, by letting the virtual output have three levels P, M and 0 when the input connection has two levels P and 0.



(a) Input and output cell flow

(b) Buffer occupancy

(c) Leaky bucket utilization

Figure 1: Time Diagrams for the virtual trunk system.

The analysis is illustrated on Figure 1, showing the state of the virtual trunk system. The desired effect can be achieved if the virtual trunk parameters are such that $m_{0i} \geq m_i$, $p_{0i} \leq p_i$ and $\theta_i \geq \Theta$, where $\theta_i = b_i/(p_i - m_i)$, $\Theta = b_{0i}/(P_{0i} - M_{0i}) = B/(P-M)$. The last equality is due to Equation 1. If those conditions are fulfilled then the virtual output flow is typically as shown in Fig 1 (a). Note that the output connection is now transmitting with rate m_0 for time duration of $\delta t = \theta - \Theta$ per period. The derivation of the required buffer size in such cases is based on Fig 1 (b) and Fig 1 (c). At time $t = 0$, the virtual buffer is empty, the leaky bucket has depth b_0 and the input source is transmitting with peak rate p. At time Θ, the leaky bucket resources are exhausted, the virtual output connection starts to transmit with mean rate m_0, and the virtual buffer occupancy is $x_0 = (p - p_0)\Theta$. At time θ, the input connection stops transmitting, the buffer content reaches $x_0 = (p - p_0)\Theta + (p - m_0)(\theta - \Theta) = p\theta - p_0\Theta - m_0(\theta - \Theta)$. This last formula is the buffer size required in the virtual trunk system, which gives us the fourth formula:

$$x_{0i} = p_i\theta_i - p_{0i}\Theta - m_{0i}(\theta_i - \Theta) \tag{2}$$

where $\theta_i = b_i/(p_i - m_i)$ and $\Theta = B/(P - M)$.

Solving Equations 1 and 2 gives:

$$x_{0i} = \frac{p_i b_i}{(p_i - mi)(1 + \frac{B}{X} + \frac{M}{X}\frac{b_i}{(p_i - m_i)})}; \tag{3}$$

The values of p_{0i}, m_{0i} and b_{0i} follow directly from Equations 1.

Note that Equation 2 holds only if the limitations we imposed are true, namely, if $\theta_i \geq \Theta$, $p_{0i} \leq p$ and $m_{0i} \geq m$. We call those the *smoothing conditions* as they correspond to the cases where the virtual trunk system produces an output smoother than the input. It is not always possible to satisfy the smoothing conditions. We now examine this point in detail. First, the limitation due to mean rates can be expressed as

$$b_i \geq \frac{X}{M}(1 + \frac{B}{X})m_i; \tag{4}$$

The constraint due to the requirement $\Theta \leq \theta_i$ is equivalent to

$$0 \leq \theta_i - \Theta \leq \frac{X}{P - M}; \tag{5}$$

with $\theta_i = b_i/(p_i - m_i)$, $\Theta = B/(P - M)$.

and the constraint due to the peak rate is equivalent to

$$b_i \leq \frac{X}{P - M}(1 + \frac{B}{X})(p_i - m_i); \tag{6}$$

We require Equation 5 to be true throughout the paper, in oder to to achieve a smoothing effect. Removing this restriction is possible, but is outside the scope of this paper. We can define rules to be followed for the cases when Equation 4 or Equation 6 do not hold .

Case 1 *If (4) is true and if (6) is true then use (1) for resources allocation and (2) for calculating* x_{0i}

Case 2 *If (4) is not true and if (6) is true then*

$$\frac{p_{0i}}{P} = \frac{x_{0i}}{X} = \frac{b_{0i}}{B}; \quad m_{0i} = m \tag{7}$$

and

$$x_{0i} = \frac{b}{1 + \frac{B}{X}} \tag{8}$$

Case 3 *If (4) is not true and if (6) is not true then*

$$\frac{x_{0i}}{X} = \frac{b_{0i}}{B}; \quad p_{0i} = p; \quad m_{0i} = m \tag{9}$$

and

$$x_{0i} = \frac{b}{1 + \frac{B}{X}} \tag{10}$$

Case 4 *If (4) is true and if (6) is not true then*

$$\frac{m_{0i}}{M} = \frac{x_{0i}}{X} = \frac{b_{0i}}{B}; \quad p_{0i} = p \tag{11}$$

and

$$x_{0i} = \frac{pX}{M + \frac{p-m}{b}(X + B)} \tag{12}$$

The second step is to compare the real system with the collection of virtual trunk systems. The basic fact [7] is that, whatever the buffer and trunk scheduling policy in the real system, there are fewer or as much data losses in the real system than in the collection of virtual trunks, provided that the sum of the resources allocated to the virtual trunk systems does not exceed the real resources. In the cases where smoothing conditions hold this latter statement translates into,

$$\sum_{i=1}^{I} K_i p_{0i} \leq P \tag{13}$$

where k_i is the number of connections of type i and p_{0i} is computed as explained above in this Section. In the other cases, it is necessary to verify similar inequalities for all four parameters of the output trunk. Equation 13 expresses our lossless CAC algorithm: connections are accepted as long as it holds.

3 Statistical multiplexing

In this section we define a CAC with statistical multiplexing by allowing a small loss probability L. The basis for the method is as follows. Call $X_{i,j}(t)$ the output rate of the virtual trunk system for the j^{th} connection of type i. If the lossless CAC is applied, then $\sum_{i,j} X_{i,j}(t) \leq P$.

In this section we restrict a study to cases where the smoothing conditions hold. By the proportional allocation rule, if above given condition holds, so do similar conditions for the rest of parameters of the output trunk.

Returning to a statistical multiplexing CAC, we allow those conditions to be violated with a small probability. Specifically, we consider proportional allocations as in Section 2, but do not require that Equation 13 be satisfied. Instead, we require that $\sum_{i,j} X_{i,j}(t) \leq P$ with a probability $1 - P_{loss}$, where P_{loss} is our grade of service requirement. In other words:

$$P_{loss} = \text{Proba}\{\sum_{i,j} X_{i,j}(t) > P\} \tag{14}$$

As in [7], the mapping between such a grade of service requirement and estimations of loss ratios is not obvious and is left for further study.

We further assume that the input connections are statistically independent with phases uniformly distributed in the interval $\Pi_i = \theta_i + \tau_i$ where $\tau_i = b_i/m_i$, where index i is the connection type.

We use a refined large deviation approximation formula, the Bahadur-Rao formula [4], to estimate P_{loss} in Equation 14. Call M_i the moment generating function of $X_{i,j}(t)$, sampled at a random instant t (this function is independent of j since all connections of type i differ only by their phase). From the analysis in the Section 2 we have:

$$M_i(s) = E[\exp(sXi,j(t))] = \alpha_i \exp(sp_{0i}) + \beta_i \exp(sm_{0i}) + \gamma_i \tag{15}$$

We can determine the probabilities α_i, β_i and γ_i for our case:

$$\alpha_i = \frac{\Theta}{\theta_i + \tau_i} \tag{16}$$

$$\beta_i = \frac{\theta_i - \Theta + \frac{x_{0i}}{m_{0i}}}{\theta_i + \tau_i} \tag{17}$$

$$\gamma_i = \frac{\tau_i - \frac{x_{0i}}{m_{0i}}}{\theta_i + \tau_i} = 1 - \alpha_i - \beta_i \tag{18}$$

The Bahadur-Rao estimate gives:

$$P_{loss} \simeq \exp(-\Lambda(s^*))\frac{1}{\sqrt{2\pi\sigma(s^*)s^*}}) \tag{19}$$

where

$$\Lambda(s) = sP - \sum_{i=1}^{I} K_i \ln(M_i(s)) \tag{20}$$

and

$$\Lambda(s^*) = sup_{s>0}(\Lambda(s)) \tag{21}$$

$$\sigma^2(s) = M''_i(s); \tag{22}$$

We can rewrite Equation 20 as

$$\Lambda(s) = sP - \sum_{i=1}^{n} K_i \ln(\alpha_i e^{sp_{0i}} + \beta_i e^{sm_{0i}} + 1 - \alpha_i - \beta_i) \tag{23}$$

and find s^* as positive root of the equation

$$\sum_{i=1}^{n} K_i \frac{\alpha_i p_{0i} e^{sp_{0i}} + \beta_i m_{0i} e^{sm_{0i}}}{\alpha_i e^{sp_{0i}} + \beta_i e^{sm_{0i}} + 1 - \alpha_i - \beta_i} = P \tag{24}$$

The connections are admitted while

$$P_{loss} \leq L \tag{25}$$

where L is a target grade of service requirement specified in advance.

4 Numerical Examples

In this Section we give some numerical examples illustrating the results of Sections 2 and 3.

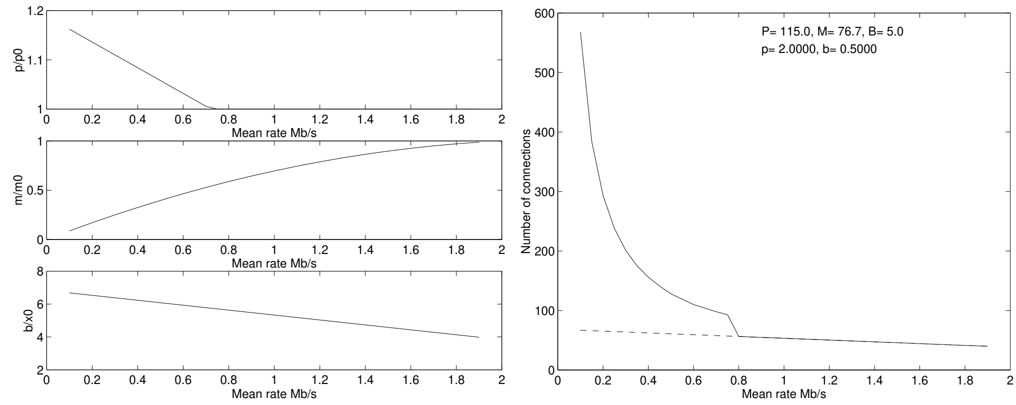
The following values are assumed throughout this section unless specified otherwise: $L = 10^{-9}$, $P = 115Mb/s$, $M = P/1.5$, $B = 5Mb$, $X = 10Mb$.

Figure 2(a) illustrates the allocation of parameters to the virtual trunk systems. Burst tolerance is $b = 0.5Mb$. Mean rate m is changed in the rang from 0 to p.

In Figure 2(a) the smoothing conditions holds for values of the mean rate m in the range from 0 till approximately 0.8. From this point till p the smoothing condition 6 is not true and we use lossless CAC instead.

One can notice that significant buffer resource economy can be achieved when relatively bursty connections are multiplexed (this corresponds to region of relatively small mean rates).

Figure 2(b) illustrates statistical multiplexing. A number of accepted connections is marked by the solid line if statistical multiplexing is applied. By the dashed line is marked the number of accepted connections for lossless case. The statistical multiplexing is performed only in the region where holds smoothing conditions. Outside this region (see Figure 2(a)) we switch to the lossless multiplexing as the statistical multiplexing for these cases is outside of the scope of the paper. In the region where smoothing conditions are true the considerable gain for the number of accepted connections can be achieved if we compare lossless multiplexing (dashed line) and statistical multiplexing (solid line).



(a) Allocation of virtual parameters.

(b) Number of accepted connections for lossless and statistical multiplexing.

Figure 2: Performance of lossless and statistical multiplexing algorithms

5 Worst case traffic

In the derivation of both lossless and statistical multiplexing CAC algorithms, we assumed that the worst case traffic patterns for our cases are on-off sources. There are indications that on-off sources are not always worst case for some performance indices: in [8, 9], worst cases are obtained for sources that start bursting at the peak rate and then remain at the sustainable rate. We do not have a proof that, in our case, on-off sources are worst case, although this appears to be intuitively true; however, as a first step, we verify in this Section that on-off sources are the worst cases among all such sources.

We consider sources that send for a duration time θ_i at the peak rate, then send for a duration T_i at the mean rate, then remain silent for a duration τ_i. If $T_i = 0$ then the source is equal to the on-off source. For the new source, we redo the derivations of Section 3 in order to compute the number of accepted connections. The distribution of the output of the virtual trunk system has the following moment generating function:

$$M_i(s) = E[\exp(sXi, j(t))] = \alpha_i \exp(sp_{0i}) + \beta_i \exp(sm_{0i}) + \beta_i' \exp(sm_i) + \gamma_i \tag{26}$$

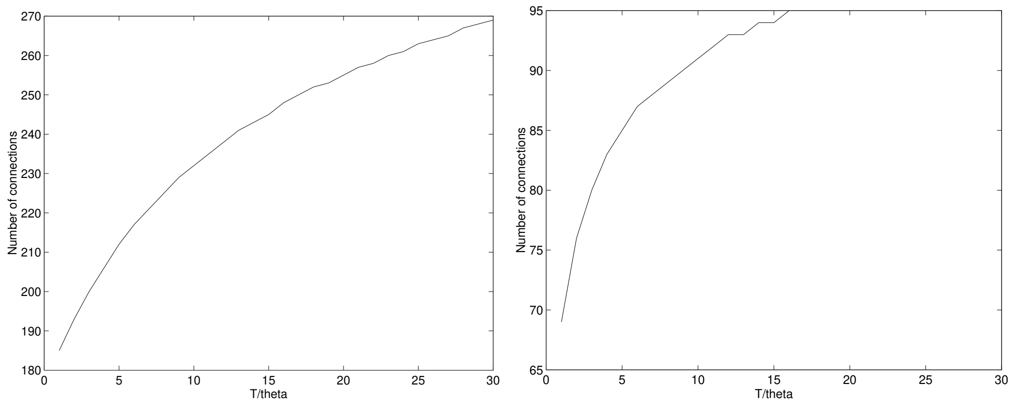
with

$$\alpha_i = \frac{\Theta}{\theta_i + \tau_i + T_i} \tag{27}$$

$$\beta_i = \frac{\theta_i - \Theta + \frac{x_{0i}}{m_{0i}}}{\theta_i + \tau_i + T_i} \tag{28}$$

$$\beta_i' = \frac{T_i}{\theta_i + \tau_i + T_i} \tag{29}$$

$$\gamma_i = \frac{\tau_i - \frac{x_{0i}}{m_{0i}}}{\theta_i + \tau_i + T_i} = 1 - \alpha_i - \beta_i - \beta_i' \tag{30}$$



(a) Mean rate m=0.3

(b) Mean rate m=1

Figure 3: Number of accepted connections for various values of T_i.

Numerical results for the new source models are shown in Figure 3.

For both values of the mean rate given on the figures, the number of accepted connections is increasing with an increasing ratio T_i/θ_i. We can conclude that, for these examples and our grade of service requirements, the on-off sources are indeed the worst case patterns. This is in opposition to the results in [8, 9]. It is probably due to the fact that the grade of service requirements discussed there are mean values, whereas ours is a large deviation probability. A general statement beyond these examples is for further study.

6 Application to Connection Grouping

We now use the CAC algorithm to compare the bandwidth utilization gains that can be achieved by the use of VBR trunks. More specifically, we study the possibility to use VBR trunks as a means to group connections and thus reduce connection processing costs in intermediate networks.

Consider for example Figure 4. It is desirable, from a complexity point of view, that connections routed through Network 3 are seen as one single (time varying) connection [17]. This is essential for providing large scale resource reservations in networks like the Internet. If Network 1 has a number of connections to route though Network 3, then all it needs is to request one VBR connection with descriptors $\{P, M, B\}$, provided that Network 3 advertised the information that it is able to support connection with descriptors $\{P1, M1, B1\}$ such that $P \leq P1, M \leq M1, B \leq B1$. One key issue is then whether it is possible to keep a good level of statistical multiplexing *inside Network 3* given that Network 3 has lost the detailed information about the connections that are crossing Network 3.

For that purpose we compare three scenarios (Figure 5): (a) N connections are treated individually by a network node (no connection grouping) (b) the same N connections are grouped into large connections used as CBR trunks and (c) same N connections are grouped into large connections used as VBR trunks. The larger nodes (labeled "CAC") on the right is at some intermediate points inside Network 3, while the smaller nodes, if any, are at the output of Network 1. The first model corresponds to setting up the connections through Network 3 individually, the second one to setting them up as groups of connections, where one group of connection is seen by Network 3 as one single CBR connection. In the last model, one group of connection is seen by Network 3 as one single VBR connection. The issue is whether the loss of information due to the grouping of connections into larger ones results in loss of

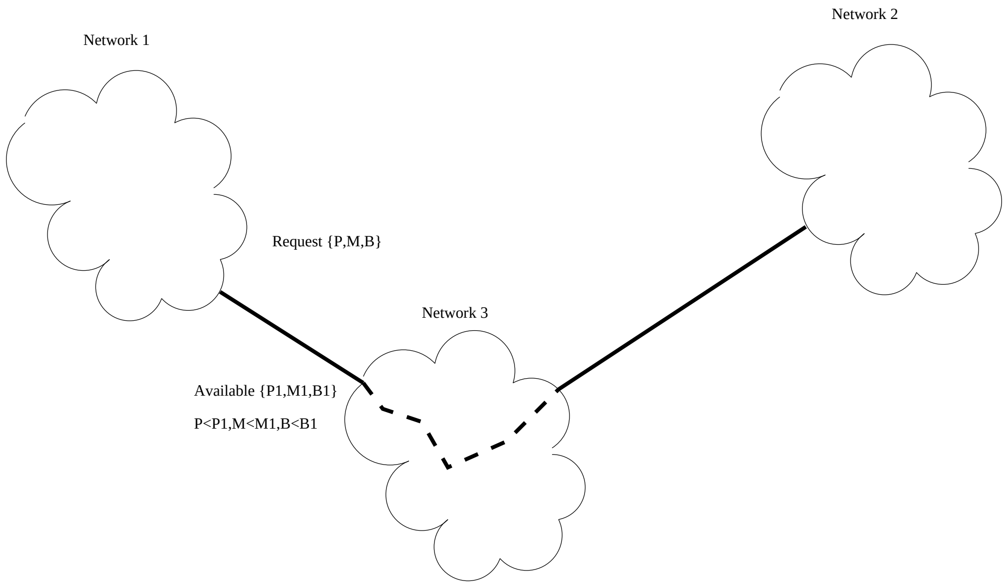


Figure 4: Grouping Connections through one Network.

statistical multiplexing. Note that while connection grouping may reduce statistical gain through information loss, the buffering at the multiplexer potentially offers some savings when comparing cases (b) or (c) versus case (a).

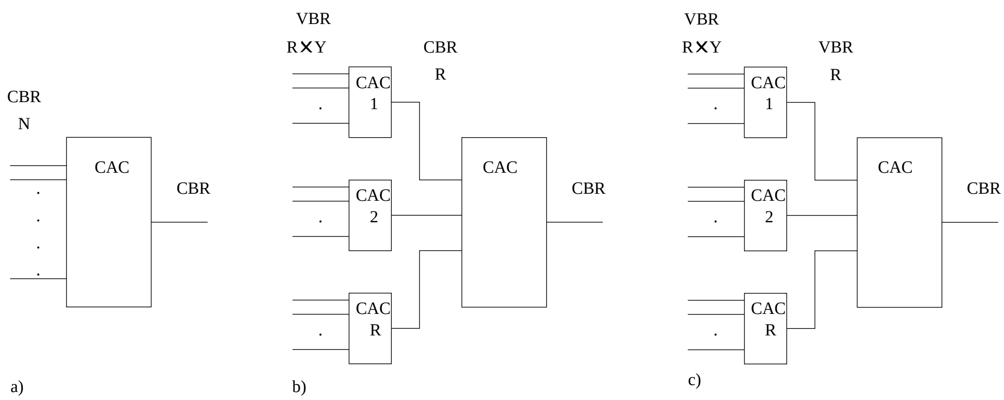
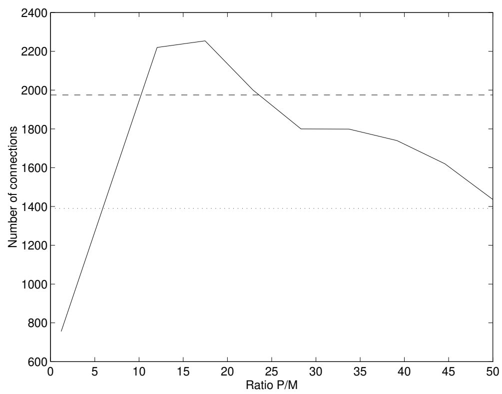


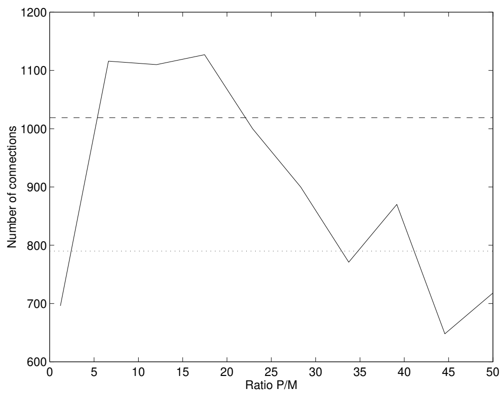
Figure 5: Models for comparing various aggregation strategies: (a) no connection grouping (b) connections grouped as CBR trunks (c) connections grouped as VBR trunks.

We assume the following values for the numerical simulations: for the first model and the second stage of the second and third models, $P = 115Mb/s$ and $X = 10Mb$. Input connections for all three models are identical with $p = 0.2Mb/s$, mean rate have two values $m = 0.05Mb/s$, $m = 0.1Mb/s$, and $b = 0.05$. Smaller nodes in the second model have $P = 11.5Mb/s$ and $X = 1Mb$ and smaller nodes in the third model have $P = 11.5Mb/s$, $B = 0.05Mb$, $X = 1Mb$. M is variable in this case.

Figure 6 compares the three models for two different mean rates of the input connections in. Both pictures represent the number of accepted connections versus the ratio P/M. The first model is represented by the dashed line, the second by the dotted line and the third by the solid line. The number of accepted connections does not depend on the ratio P/M for the first and the second models because CBR over CBR multiplexing is involved. The third model performance significantly depends upon the ratio P/M. The results in this case show that the performance of VBR over VBR traffic aggregation can significantly exceed the performance of single CBR CAC if the parameters of the VR virtual trunk are chosen appropriately. One can observe that the best performance is achieved for relatively bursty connections with ratio $P/M \approx 10...20$. Most of the statistical gain is obtained in the second stage, not in the smaller nodes. Thus, by describing a connection aggregate as one bursty connection, it is possible to get a statistical gain in the following nodes. This possibility is lost if we use CBR over CBR traffic aggregation. The gain above model (a) is due to buffering in the smaller node. For the quick formulas for determining optimal VBR trunk parameters, see [11].



(a) Mean rate m=0.05



(b) Mean rate m=0.1

Figure 6: Number of accepted connections for models (a): dashed line, above, (b): dotted line, below, and (c): solid line.

7 Conclusions

We have presented and studied a preliminary CAC algorithm for VBR connections over a VBR trunk. The proposed algorithm is an extension to the algorithm proposed in[7] for the CBR case. We have shown on some examples that supporting VBR trunks makes it possible to aggregate connections in intermediate points without statistical multiplexing loss. We believe that such a feature is essential for scalability of integrated services packet networks. Further work is required in the following directions: demonstrate in more general settings the statements obtained in this paper numerically, map the grade of service requirement to cell loss ratios, and devise an optimal solution for the cases where smoothing conditions do not hold.

References

[1] ATM Forum, User Network Interface 4.0

[2] 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

[3] *VBR over VBR: the Homogeneous, Loss-Free case*, EPFL Technical Report DI 96/199, http://lrcwww.epfl.ch/scone/scone_paper3.ps

[4] P. Varayia and J: Walrand,*High Performance Communication Networks*, Morgan Kaufmann, San Francisco, October 1996

[5] *COST 224: Performance Evaluation and Design of Multiservice Network*, J. Roberts ed., Office for Official Publications of the European Communities, 1992.

[6] Jean-Yves Le Boudec, Gustavo de Veciana and Jean Walrand, *QoS in ATM: Theory and Practice*, CDC Conference, Kobe, 1996

[7] A. Elwalid, D. Mitra, R. H. Wentworth, *A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an ATM node*, IEEE Journal on Selected Areas in Communications, vol.13, N0.6., August 1995.

[8] E. Wallmeier, C.M. Hauber, *Blocking probabilities in ATM pipes controlled by a connection acceptance algorithm based on mean and peak bit rates*, Queuing, performance and control in ATM (ITC-13), Elsevier science publishers, IAC,1991.

[9] N. Yamanaka, Y. Sato, K. Sato, *Performance limitations of leaky bucket algorithm for usage parameter control and bandwidth allocation methods*, IEICE Trans. Commun. vol. E75-B, No. 2 February 1992.

[10] B. T. Doshi, *Deterministic rule based traffic descriptors for broadband ISDN: worst case behavior and connection acceptance control.* ITC 14, 1994 Elsevier science B. V.

[11] S.Giordano, J.-Y. Le Boudec, P. Oechslin, S. Robert, *VBR over VBR: the homogeneous, loss-free case*, LRC-EPFL technical report No 96/199, 1996.

[12] P. Billingsley, *Probability and measure*, John Wiley & Sons, Inc., 1979.

[13] ITU Telecommunication Standardization Sector - Study group 13, *ITU Recommendation I.371, Traffic Control and Congestion Control in B-ISDN*, 1995.

[14] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zapalla, *RSVP: A New Resource ReSerVation Protocol*, IEEE Network, September 1993.

[15] C. Topolcic, *Experimental Internet Stream Protocol, Version 2 (ST-II)*, IEEE Network, September 1993

[16] E. Gauthier, 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

[17] E. Gauthier and J.-Y. Le Boudec,*Scalability Enhancements for Connection-Oriented Networks*, Proceedings of International Zurich Seminar on Digital Communications, 1996.

[18] The ATM Forum,*P-NNI 1.0 Specification* , 1996.

[19] J. Mignault, A. Gravey and C. Rosenberg , *A Survay of Straightforvard Statistical Multiplexing Models for ATM Networks*, First International "ATM Traffic Expert" Symposium, Basel, 1995.