An Evaluation of a Local Approach for VPC Capacity Management

Sven-Olof Larsson[1] and Åke Arvidsson[2]

Dept. of Telecommunications and Mathematics,
University of Karlskrona/Ronneby,
S-371 79 Karlskrona, Sweden

SUMMARY By reserving transmission capacity on a series of links from one node to another, making a virtual path connection (VPC) between these nodes, several benefits are obtained. VPCs will simplify routing at transit nodes, connection admission control, and QoS management by traffic segregation. As telecommunications traffics experience variations in the number of calls per time unit due to office hours, inaccurate forecasting, quick changes in traffic loads, and changes in the types of traffic (as in introduction of new services), there is a need to cope for this by adaptive capacity reallocation between different VPCs. We have developed a type of local VPC capacity management policy that uses an allocation function to determine the needed capacity for the coming updating interval, based on the current number of active connections but independently of the offered traffic. We determine its optimal parameters, and the optimal updating interval for different overhead costs. Our policy is shown to be able to combine benefits from both VP and VC routing by fast capacity reallocations. The method of signaling is easy to implement and our evaluations indicate that the method is robust. This paper is based on our earlier work, described in [19]. The calculations are simplified and the methodology is changed.

1. Introduction

To accept a new call a check must be made to ensure that there is enough capacity left to establish the call through a series of links between the end nodes. When a route is found, the required amount of capacity is reserved for the call. (By capacity we mean equivalent bandwidth [1], [2] or transmission capacity needed for a certain traffic.) The established call uses this logical connection, which is called a virtual channel connection (VCC). A virtual path connection (VPC) groups VCCs together to be handled as an entity. A VPC can be seen as reserved bulk capacity between two nodes. By using VPCs the acceptance of a new call is simplified because the routing and reservation of capacity has already been done.

A VPC network constitutes a higher layer which is logically independent of an underlying physical network. Having several VPC networks each supporting one type of traffic simplifies statistical multiplexing and quality-of-service (QoS) management.

There are always variations in telecommunications traffics. Traditional telephone networks have been dimensioned for the so called busy hour to cope with the maximum traffics. This means that much of the capacity will stay unused for most of the time. By using VPCs, the capacity allocation can be altered dynamically. This allows us to meet traffic variations by reshaping the VPCs in order to match the current demands. This means savings on the amount of capacity required in a network, if we can utilize non-coincidental busy hours to reallocate the capacity. The concept of VPCs and VCCs is supported in the asynchronous transfer mode (ATM) and in the synchronous digital hierarchy (SDH/SONET).

We have grouped VPC capacity management approaches into groups which are fundamentally different. This definition is based on the amount of information used in the calculation of the VPC capacity reallocation. We have found that the calculation can be centralized e.g. [3]-[5], distributed e.g. [6], or local e.g. [7]. (A lot of papers describing distributed approaches fall into our definition of a local approach.) The central approach has the ability to make VPC capacity reallocation based on global information. The idea of the local and distributed approaches is to increase the robustness and improve performance compared to a central approach, which is depending on a central computer. By assigning costs for rejected calls and overhead such as control messages, the performance of the various approaches can be evaluated and compared to each other.

Section 2 describes our proposal for a local approach for VPC capacity management. In section

1. e-mail: sveo@itm.hk-r.se, phone: + 46 455 78062, fax: + 46 455 78057
2. e-mail: akear@itm.hk-r.se, phone: + 46 455 78053, fax: + 46 455 78057

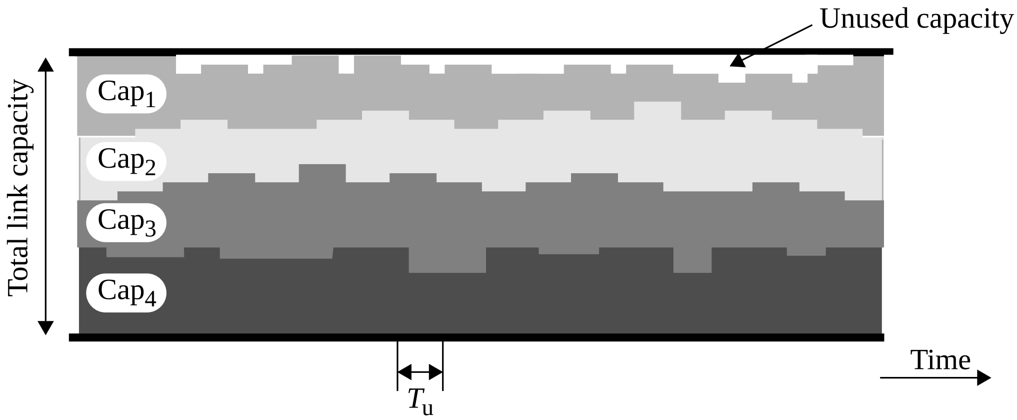

Fig. 1 An example of a reallocation sequence..

3 we describe the evaluation of the approach and give the results in section 4. Finally, we conclude the paper in section 5 and discuss further work in section 6.

2. The Local Approach

Each node periodically makes a decision about whether to seize or release capacity on the VPCs originating from that node (Fig. 2). The decision is based on the actual number of occupied connections on each VPC.

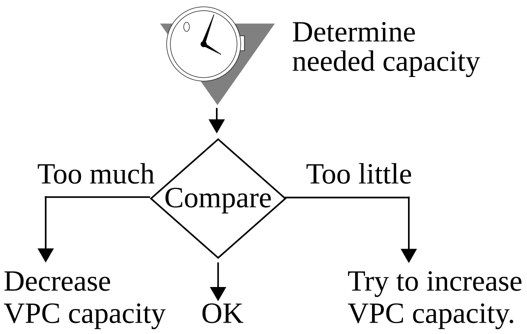

Fig. 2 The principle of the local approach.

Figure 1 shows a reallocation sequence in principle, when having four VPCs on a link. The problem is how to decrease the amount of unused capacity in the best way.

In our study we have used two VPCs between each node pair (though the method supports an arbitrary number of VPCs). The VPC with the smallest number of hops is preferred and is labelled **PVPC**, while the other one is referred to as an optional VPC and labelled **OVPC**.

The VPC capacity management is done with help of signaling. The following four control messages are used:

- Path finding (PTH) + Answer
- VPC Establishment (EST)
- Increment request (INC) + Answer
- Decrement (DEC)

2.1 Control Messages

PTH is used for path identification by broadcasting it from all nodes to all other nodes. The broadcasting can be done from time to time or at command to recover from faulty links [8], [9]. In our evaluation we have only used it once to initiate the management system. When a PTH arrives at the destination, an answer message will be sent back to the originating node, the same way the actual PTH has travelled. This message contains a route to the destination.

The node of origin selects some routes received as answers to PTH and puts them in a table. We have selected node disjoint paths (shortest and link disjoint paths give almost the same performance [10]). The paths are ordered by the number of links and the total physical distances. The VPCs are finally established by sending **EST**, which is a source routed message along the path enabling the intermediate nodes to set up their routing tables.

When the VPCs have been established the periodic management starts in each node. When more capacity is needed an **INC** message is sent on the VPC to find out the capacity allowed for the whole path. This means that a VPC get the minimum allowed capacity on the series of links. The amount of available capacity is stored in the INC on successive links. When it reaches the end node, indicating the available capacity, an answer message is sent back to the originating node.

An increment request message is first sent on the PVPC. If the request cannot be satisfied, the OVPC is tried. When trying to get capacity on the links a temporary reservation must be made. This makes interference from other requests impossible, but can result in deadlock. To avoid deadlocks the following procedure is applied. If a request message reaches a node where the next link is already reserved, a message will be sent back to the node of origin releasing its current reservations. The node of origin tries again after a random delay (sufficiently long).

When an originating node determines that capacity should be released, a **DEC** is sent. The capacity reservation is decreased on each traversed link. No answer message has been used for this message. Capacity on the OVPC is released first.

2.2 Calculation of Needed Capacity

The developed approach for local VPC capacity management is inspired by the one developed by Mocci et al. [7]. This method allocates just enough VPC capacity to meet a predefined limit of call blocking (target blocking) [11].

The idea of this approach is to handle traffic variations in a short time scale, i.e. larger than the mean interarrival time but smaller than the average call holding time. At regular intervals (with length T_u) the needed VPC capacity for the coming interval is determined. This is done by calculating the expected blocking probability over the interval for various capacities given the offered traffic and present occupation state. The capacity that meets the target blocking is the needed one.

The calculation is quite complex, hence it is suggested to use precalculated table to achieve real time applicability. Another approach is to apply simplifications along the line of Virtamo and Aalto [12]. In [13] an allocation function is developed which does not depend on the actual offered traffic. It is based on the formula

$$N(n) = \lceil n + K \cdot \sqrt{n} \rceil \tag{1}$$

where N is the required capacity, n is the number of currently active calls, and K depends on the offered traffic, target blocking probability, and updating interval. The idea behind this function is that for a specified traffic intensity the mean occupation state is equal to the intensity (if the blocking probability is low). K can be seen as a safety factor which adds extra capacity in units of the standard deviation of the occupation state. By introducing some constants, it is possible to rewrite (1) in a way that, in a given range, makes N a function of n only [13].

However, this approach does not take into account the interaction between several VPCs on a physical link of fixed capacity. For example, although Ks computed for 1% target blocking will result in this value (as long as the physical link permits), the result will be under utilization of the link if not all capacity is seized. If, however, a larger K is used, the blocking will actually decrease and the link will be fully utilized. In this case, a higher K will thus better exploit the traffic fluctuations, i.e. when some VPCs temporary increase their number of allocations, others decrease theirs.

Another complication is K's dependence on the updating interval. The optimal choice of updating interval is determined by the trade off between increased traffic (which partly depends on K) and the overhead associated with updating. Capacity should be allocated only when actually needed; allocating too much will lead to under utilization and allocating too little will lead to excessive blocking. The updating interval T_u determine the rate at which new decisions regarding allocations can be made. Clearly, frequent possibilities to change allocations (small T_u) will permit smaller capacity margins without risking excessive blocking during the interval and *vice versa*. To optimize total performance, an optimal allocation strategy in terms of K and T_u must thus be found.

The local approach multiplexes VPCs in a special way. Considering a deterministic multiplexing of the VPCs, i.e. each VPC gets a fair portion of the capacity over a long time, the call blocking probability can be calculated using the Erlang's B-formula for each VPC. When having full statistical multiplexing of all VPCs, the blocking probability can be calculated from the same formula by adding all traffics together. Since the local approach reallocates the capacity on a rather frequent basis, the blocking probability gets lower than for the deterministic multiplexing, but higher than for the full multiplexing. (When using a sufficiently short T_u the blocking reaches the same level as for full multiplexing over one link.)

2.3 The Allocation Function

To obtain an optimal allocation function we first study the effect of K when having many VPCs on the same link. We label the K (for a specific traffic intensity, T_u, and traffic load situation) that mini-

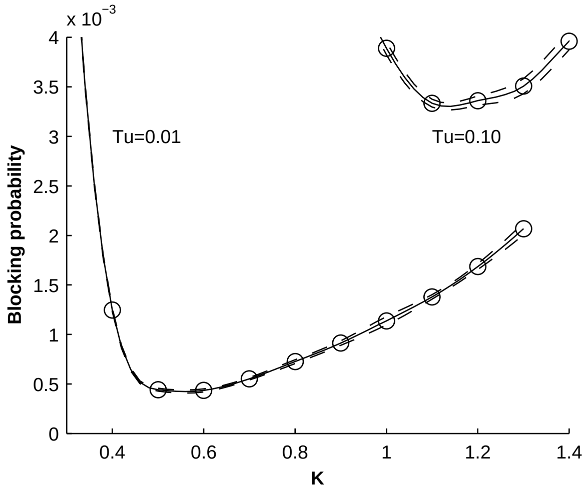

Fig. 3 Multiplexing gain for 5 VPCs on a link.

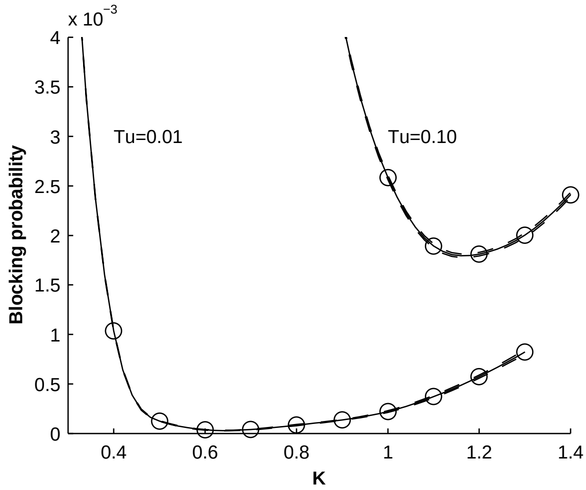

Fig. 4 Multiplexing gain for 10 VPCs on a link.

mizes the total number of blocked calls (for a specific link) as K_{opt} (for that link). Figures 3 and 4 show the impact the number of VPCs have on K_{opt} for T_u = 0.01 and T_u = 0.1. The lines are interpolated between evaluated *K*s (circles). The dashed lines show the 95% confidence intervals.

The curves in Fig. 3 show the mean blocking when having 5 VPCs each carrying 152 Erlangs (a value was chosen in accordance with our test networks, see below) and Fig. 4 refers to 10 VPCs. In both figures, the total bandwidth available is dimensioned so that each VPC can expect 1% loss if deterministic multiplexing was used. It is seen that K_{opt} is more sensitive to the traffic for a short updating interval than for a long one, and that, as expected, a lower blocking is noted for 10 VPCs than for 5.

We also note that the resulting, optimal blocking fall between the full sharing ($1.94 \cdot 10^{-5}$ and $1.8 \cdot 10^{-8}$ for 5 and 10 VPCs respectively) and full partitioning ($1 \cdot 10^{-2}$). It is also interesting to note that the K_{opt} obtained differ from the *K* that meets the fixed target of 1% call blocking [13], which e.g. is 0.74 for T_u = 0.1 and one VPC carrying a traffic of 152 Erlangs. It is concluded that there is a potentially best *K* which appears to depend more on T_u = 0.1 than on the actual traffic.

A real network with traffics of various magnitudes interacting in different combinations and numbers on link of different capacities poses a very complex problem. To simplify the problem for an inhomogeneous network we consider the situation on an average link in the network under study. On this link we put a mean traffic consisting a couple of ‘‘background-VPCs.’’

In our test networks the mean number of PVPCs on a link is about 4 and each have on the average a capacity of 171. Figure 5 shows the average distribution of PVPCs per link for our test networks. Similarly, calculating the average traffic demands over all node pairs results in 152 Erlangs. With this in mind, our average link consists of 4 VPCs as background traffics each carrying 152 Erlangs.

It should be noted that PVPC-traffics in reality interact with OVPCs. The requests for bandwidth on the OVPCs form some kind of overflow process since requests only are made when requests on the PVPC are rejected. It is also difficult to estimate the average behaviour of these VPCs without simulating the complete network. It is for this reason that we have decided to ignore them in our average link.

Table 1 shows the K_{opt}s found for our average link for various updating intervals. The exact values are not critical because the optima are flat as indicated in Figs. 3 and 4.

Table 1 K_{opt}s for the background VPCs.

T_u	0.01	0.04	0.07	0.10	0.13	0.16	0.19
K_{opt}	0.50	0.82	1.03	1.15	1.28	1.31	1.40

2.4 Optimizing the Allocation Function

Having found the K_{opt}s for the background traffic, we now add another VPC and study K_{opt}s for different traffic intensities and different updating intervals on this VPC. (Additional capacity on the link is added so that, as before, deterministic multiplexing would result in 1% loss.)

The result is displayed in Fig. 6 where K_{opt} is given over a range of different traffics and for T_u =

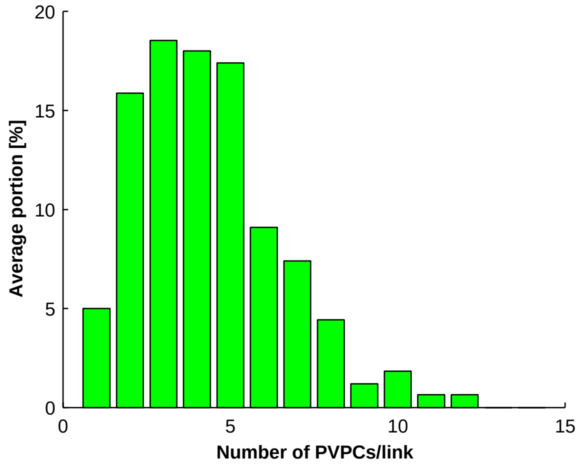

Fig. 5 Distribution of PVPCs per link.

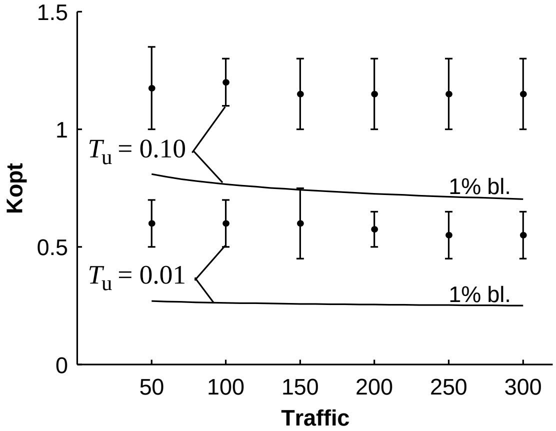

Fig. 6 K_{opt}s and fixed-target K.

0.01 and T_u = 0.1. The values are given as intervals because there is no particular K for which there is a distinct minimum of the total number of lost calls. As a comparison, the K values calculated for only one VPC and a fixed target of 1% call blocking, as described in [13], are also shown (solid lines). The straight lines obtained in the diagrams lead to the conclusion that a fixed K_{opt} can preferably be used for the different traffics, and a comparison between the figures and Table 1 suggests that we may use the values given in the table.

3. The Evaluation

For the sake of simplicity we limit the numerical examples of this study to the case of a single, uniform service class. Multiplexing in the burst scale (e.g. for VBR services) is hidden in the use of equivalent bandwidth [1], [2] hence extensions to bursty traffics is straight forward. Requests for connections arrive according to independent Poisson processes for each node pair. The connection holding time is assumed to be negative exponentially distributed with unit mean.

As indicated before, we have used ten non-hierarchical networks with ten nodes each (which can be seen as core ATM networks). The nodes have both VP and VC routing capabilities and a fully meshed network of VPCs is formed so that all nodes have direct VPCs (PVPCS and possible OVPCs) to all other nodes. A VCC between two nodes is normally routed over the corresponding, direct VPC. However, we have also used dynamic alternative routing (DAR) [14] on the call scale, i.e. if the direct VPC does not have room for an arriving call, rerouting with two VPCs in tandem over a selected transit node is tried. If this does not succeed, the call is rejected and a new transit node is chosen (at random) for the next time a call needs to be rerouted. Two control messages are used to determine the status of the transit nodes (question + answer). To ensure stability, we have applied a trunk reservation parameter of ten connections for direct traffic on all VPCs.

Our test networks have the capacity to handle the mean traffics with 1% call blocking probability. To simulate actual traffic variations, ten different traffic patterns were generated for each network by randomly selecting a busy center. Nodes inside the center increase their traffics above the average and those outside the center decrease theirs. Further details are described in appendix A.

4. Results

Our aim is now to apply the results for an average link summarised in Table 1 to find the optimal T_u for various overhead costs. We also wish to study the actual optimality of the K_{opt} obtained above. The evaluation is based on the reached profitability (2). Profitability is a normalized measure where 100% profitability means that all calls are handled without any overhead costs. (100% is infeasible for high traffic load situations.)

$$\text{Profitability} = \frac{\text{Calls}_{\text{Handled}} - (\text{Messages} \cdot \text{Cost})}{\text{Calls}_{\text{Offered}}} \qquad (2)$$

The profit of handling one call is set to one unit. However, to be able to handle calls, several control messages (by means of signals or RM-cells using some of the bandwidth) have to be used and these affect the total profit. The messages includ-

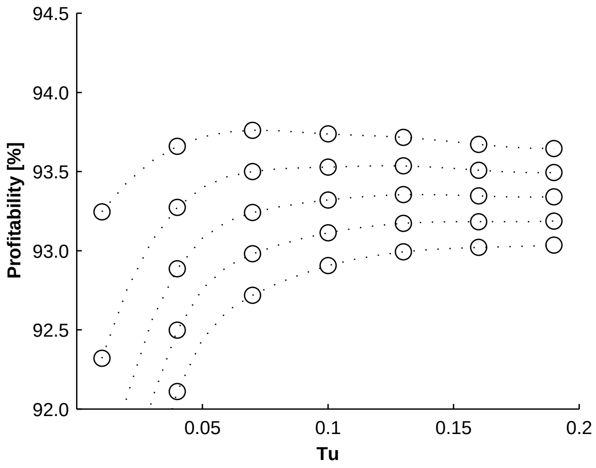

Fig. 7 Profit. for different T_us and message costs.

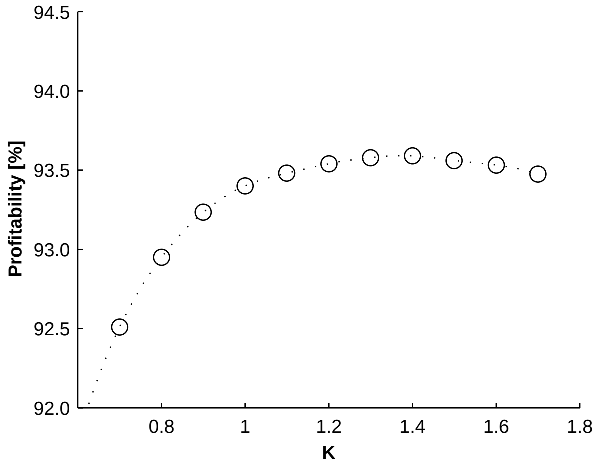

Fig. 8 Profitability for different *Ks*, T_u = 0.10.

ed in our evaluation are INC, DEC, and the status messages for DAR. If a control message is seen as an RM cell, the cost can be related to an average phone call. Suppose that a phone call uses 167 cells/second, then the RM cell could be given a cost of 1/(167·seconds per mean holding time) which is ~10^{-4}. This cost might be too optimistic because there are also costs other than the ones related to bandwidth, e.g. processing. These are difficult to estimate. By using a higher message cost, the messages can be seen as having an overhead. The profitability is used to enable a reasonable evaluation of the overall performance by combining gains and costs.

An interesting aspect is the occasionally occurring capacity violations. These are caused by excess calls on links which have been granted less capacity and that are not disconnected in time before new calls arrive on VPCs which have been granted more capacity as a result of the reallocation. There are three ways to deal with this. One way is to move ongoing surplus connections to a path that can accommodate them. Another way is to wait for the connections to finish until capacity is released. The third way is to use ‘‘guard bands’’ which will not be allocated to any particular VPC. By this one hopes that there will be enough capacity to deal with over allocations. The amount of capacity violations depends not only on the guard band but also on the actual network and traffics. In general, the impact of a capacity violation depends on the degree of violation and the time during which it persists. In this evaluation a guard band is not needed to cope with link violations. Instead, a decrease of capacity for a particular VPC is simply not allowed if some of this capacity is in use.

Figure 7 shows profitability as a function of the updating interval for different message costs. The upper line is the result when having a message cost of 0.01, the lower line when having a message cost of 0.09, and the step between the message costs is 0.02. The dotted lines are interpolated between evaluated *K*s (circles).

It is immediately seen that the higher the cost, the lower the profitability. Moreover, it can be seen that the optimum is further right the higher the updating cost. For example, for a cost of 0.01 (top curve) we find $T_{u,opt} \approx 0.07$ while for a cost of 0.05 (middle curve) we find $T_{u,opt} \approx 0.14$. This observation is in perfect agreement with the assumption that more frequent reallocations are preferable when the message cost is small.

Finally we wish to investigate the actual optimality of our *K*-value. We consider a message cost of 0.03 where, according to Fig. 7, the optimal $T_u \approx 0.1$. The corresponding *K*-value (which was used in Fig. 7) is, according to Table 1, about 1.15.

Figure 8 shows the resulting profitability as a function of *K* averaged on the ten test networks. The figure suggests that the actually optimal *K* lies between 1.3 and 1.4, i.e. somewhat higher than our model predicts. On the other hand, the resulting difference in performance between the two values is small. (It is noted that the alternative to using our model with averages is to conduct full scale simulations of a network over a range of *K* and T_u-values.)

Comparing the results to a fixed allocation scheme where the VPCs are engineered according to the basic traffics, the results are better. (Both schemes use DAR.) For our example with a message cost of 0.03, the profitability for the fixed allocation is 92.2%, while our method reaches 93.5%. This may seem a small difference, but it should be remembered that the span between actual and full profitability represents a potential operator profit increase. The fact that this has dropped from 7.8% with fixed allocation to 6.5% with our method, means a significant step towards profit maximization.

We have earlier studied a central approach [15] which in this case gives higher profitability, but this approach tends to give a lot of link violations which were not accounted for. Furthermore, for high traffic loads, the local approach reaches better profitability than the central one.

5. Conclusions

We have proposed a type of local VPC capacity management policy that uses regular updates and a simple allocation function to determine the needed capacity for the coming updating interval. With our proposed, simple procedure based on averages for setting its unknown parameter K, the number of parameters is limited to one, *viz*. the current number of active connections. We have also shown how to determine the optimal interval depending on overhead costs.

Using a simplified model with an average background traffic for the calculation of K works well although the actual distributions of PVPCs and OVPCs covers a wide range as seen in Fig. 5. Moreover, the optimal K values for the allocation function are not critical but optimal over a wider range as shown in Figs. 6, 7, and 8. Both observations suggest that our method is robust with respect to the accuracy of traffic measurements or forecasts.

Applying the allocation scheme, an improvement is achieved compared to a fixed capacity allocation and the approach takes advantage of the benefits from both VP and VC routing, i.e. enabling fast CAC and using multiplexing of VPCs. We also notice that the method of signaling is easy to implement and that link violations can be avoided.

6. Further Work

Sensitivity analysis of resulting blocking probabilities depending on the number of VPCs and traffic load variations might be of interest to further study the robustness of the parameter K. A different direction is to use VPC-dependent Ks rather than a global one. Evaluating the gains of such a finer allocation scheme and relating them to the additional complexity also remains an issue for further study.

Another issue refers to the grouping of VPCs. Using the local approach there might be no need for group VPCs [16] and it could simplify the trade-off between VP and VC routing [17]. Finally, the aspect of deploying the OVPCs as backup-VPCs leads into the issue of self-healing networks [8], [9], [18]. The issues that arise when integrating fault management into the capacity management constitute an interesting area to explore.

References

[1] A. I. Elwalid, D. Mitra, “Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High-speed Networks,” IEEE/ACM Trans. on Networking, vol. 1, no. 3, pp. 329-343, 1991.

[2] R. Guérin, H. Ahmadi, “Equivalent Capacity and its Application to Bandwidth Allocation in High-speed Networks,” IEEE Journal on sel. Areas in Commun., vol. 9, no. 7, pp. 968-981, Sept. 1991.

[3] Å. Arvidsson, “High Level B-ISDN/ATM Traffic Management in Real Time,” *Performance Modelling and Evaluation of ATM Networks*, Vol 1, pp. 177-207, ISBN 0-412-71140-0, Chapman & Hall, 1995.

[4] Å. Arvidsson, “Real Time Management of Virtual Paths,” Proc. IEEE Globecom’94, pp. 1399-1403, 1994.

[5] S. Shioda, “Evaluating the Performance of Virtual Path Bandwidth Control in ATM Networks,” IEICE Trans. Commun., vol. E77-B, no. 10, pp. 1175-87, 1994.

[6] S.-O. Larsson, Å. Arvidsson, “A Study of a Distrubuted Approach for VPC Network Management,” Proc. IEEE Globecom’97, paper no. S50.3, 1997.

[7] U. Mocci, P. Pannunzi, and C. Scoglio, ‘‘Adaptive capacity management of virtual path networks,’’ Proc. IEEE Globecom’96, paper no. 19.2, 1996.

[8] N. D. Lin, A. Zolfaghari, and B. Lusignan, “ATM Virtual Path Self-healing Based on a New Path Restoration Protocol,” Proc. Globecom’94, pp. 794-798, 1994.

[9] R. Kawamura, H. Hadama, and I. Tokizawa, “Implementation of Self-healing Function in ATM Networks Based on Virtual Path Concept,” Proc. IEEE Infocom’95, pp. 303-311, 1995.

[10] S.-O. Larsson, Å. Arvidsson, “Performance Evaluation of Different Local and Distributed Approaches for VP Network Management,” 13’th

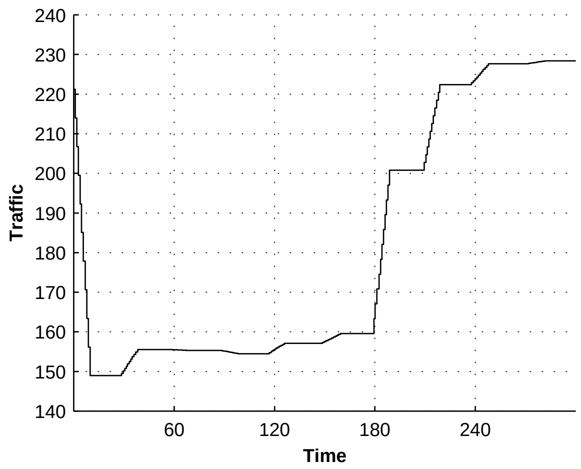

Fig. A1 Sample traffic.

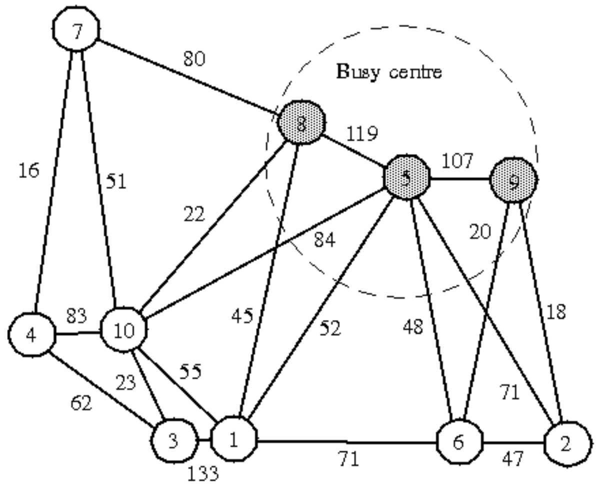

Fig. A2 Sample network layout.

NTS, pp. 98-107, ISBN 82-993980-0-2, NTNU 1996.

[11] C. Bruni, P. D'Andrea, U. Mocci, and C. Scoglio, ''Optimal capacity management of virtual paths in ATM networks,'' Proc. IEEE Globecom'94, pp. 207-211, 1994.

[12] J. Virtamo, S. Aalto, "Blocking Probabilities in a Transient System," COST TD257(97)14, Netherlands, Jan. 1997.

[13] J. Virtamo, S. Aalto, "Remarks on the Effectiveness of Dynamic VP Bandwidth Management," COST TD257(97)15, Netherlands, Jan. 1997.

[14] R. J. Gibbens, F. P. Kelly, and P. B. Key, "Dynamic Alternative Routing - Modelling and Behaviour," Proc. ITC 12, paper no. 3.4A.3, Torino, June 1988.

[15] S.-O. Larsson, Å. Arvidsson, "A Comparison Between Different Approaches for VPC Bandwidth Management," Proc. 5'th IFIP workshop on ATM networks, UK, pp. 52/1-10, July 1997.

[16] M. Omotami, T. Takahashi, "Network Design of B-ISDN Using the Group Virtual Path Scheme," Electronics and Commun. in Japan, Part 1, vol. 79, no.7, pp. 10-22, 1996.

[17] D. Hughes, K. Wajda, "Comparison of Virtual Path Bandwidth Assignment and Routing Methods," Annals of Telecom., vol. 49, no. 1-2, pp. 80-89, 1994.

[18] P. A. Veitch, D. G. Smith, and I. Hawker, "Alternative Routing Strategies for Virtual Path Restoration," IFIP Workshop TC6, IFIP working groups 6.3 and 6.4 participants proc., p. 860, 15/1-10.

[19] S.-O. Larsson, Å. Arvidsson, "Performance Evaluation of a Local Approach for VPC Capacity Management," IEICE Trans. Commun., vol. E81-B, no. 5, pp. 870-876, 1998.

Appendix A: Test Networks

The test networks have been made with a program that generates networks with N nodes. Call holding times are assumed to be negative exponentially distributed with a mean holding time of 1 time unit. User demands are fully characterized by a sequence of known end-to-end traffic demand matrices $A(k)$ (of size $N \cdot N$), where $a_{o,d}(k)$ denotes the traffic from o to d at time k, k : (k = 1,...,K) (see Fig. A1). The time index K indicates intervals such as hour, day of week, or day of year. For each origin-destination pair an offered traffic was assigned to give 1% expected loss for a given transmission capacity. This basic traffic was modified to yield K different load situations by the use of ''busy center'' (Fig. A2). Traffics between busy center nodes were increased randomly between 20 and 60%, traffics between nodes outside the busy region were decreased randomly between 20 and 60%, and the traffic between a busy center node and a node outside the center was modified randomly between -20% and +20%. After the modification the traffics were renormalized to give the same total amount of offered traffic as before. The resulting greatest increase is 97% and greatest decrease 60%. In Fig. A2 the link capacities are given in capacity units, each of which can accommodate ten connections. With N = 10, the total traffic offered to the network at any time is typically about 6800 Erlangs.