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

Abstract

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 (New Year's Eve), 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 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 independent of the offered traffic. In this work we propose and evaluate a method to get an optimal parameter setting of the allocation function based only on average values for a network link. We also discuss the influence of different factors on the allocation function.

1. Introduction

To accept a new call (connection) 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 new calls is simplified because the routing and reservation of capacity have 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 and traffic varia-

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

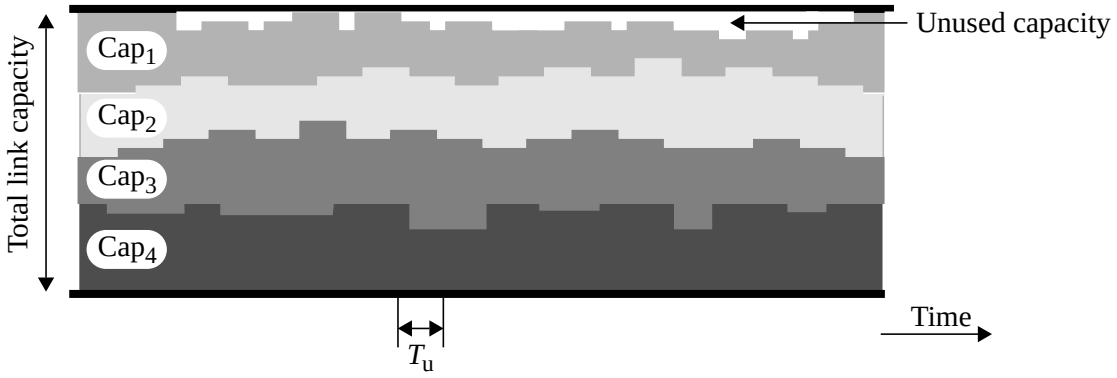

Fig. 1 An example of a reallocation sequence.

tions 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).

Section 2 describes our proposal for a local approach for VPC capacity management. The models used to get the parameter for the allocation function is described in Sect. 3. In Sect. 4 we describe the routing strategy. The results and evaluations are found in Sect. 5. Finally, we conclude the paper in Sect. 6 and discuss further work in Sect. 7.

2. The Local Approach

A lot of papers describing distributed approaches fall into our definition of a local approach. By a local approach we mean that the information needed for the management function is locally available. 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.

Figure 1 shows a reallocation sequence in principle, when having four VPCs on a link. The problem is how to make the VPCs to allocate the capacity so that the total handled traffic is maximized.

The method supports an arbitrary number of VPCs between each node pair. The VPC with the smallest number of hops is favoured and is labelled **FVPC**, while the other ones are referred to as optional VPCs and labelled **OVPCs**.

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)

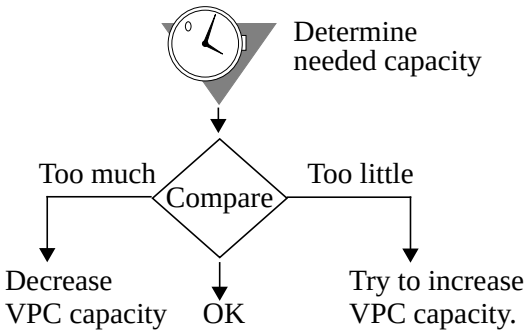

Fig. 2 The principle of the local approach.

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 [3], [4]. 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.) The paths are ordered by the number of links traversed 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 if the wanted capacity exist on every link for the whole path. This means that a VPC gets all of the requested capacity or the minimum available on the series of links. An increment request message is first sent on the FVPC. If the request cannot be satisfied, the OVPCs is tried. 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.

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 then 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 OVPCs is released first.

2.2 The Allocation Function

The developed approach for local VPC capacity management is inspired by the one developed by Mocci et al. [5]. This method allocates just enough VPC capacity to meet a predefined limit of call blocking (target blocking) [6].

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 [7]. In [8] an allocation function is developed which does not depend on the actual offered traffic. It is based on the formula

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

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 occupation state is Poisson distributed (if the blocking probability is low) with a mean equal to the intensity. 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 choose a fixed K so that, in a given range, N is a func-

tion of *n* only [8].

However, although *K*s 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 decrease their number of allocations, others are able to increase 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 determines 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.

A real network with traffics of various magnitudes interacting in different combinations and VPCs that traverse different numbers of links with different capacities poses a very complex problem. In [9] we have evaluated the influence of the number of interacting VPCs on a link and suggested a method for selecting the T_u depending on the actual message cost. When many VPCs interact, a larger *K* can be selected. If the message cost is high the number of messages has to be decreased by using a longer T_u thus and a larger *K*. The optimal *K* also depends on the actual traffic load situation on the various links.

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.)

In earlier work [9] we have used a model to get the parameter for the allocation function. In this work we improve the model to cope with the situation when alternative routing is used.

3. A Method for Estimating the Parameter *K*

To simplify the problem of setting the parameter *K* 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 of a couple of FVPCs. In our test networks the mean number of FVPCs on a link is about 4 and each have on the average a traffic of 130 Erlangs. The mean capacity per VPC can be calculated or set to the amount allocated to this size of traffic. The capacity which gives on the average 1% call blocking is 148. With this in mind, our average link consists of 4 VPCs as background traffics each carrying 130 Erlangs, and the link capacity is set to 592.

We have simulated such a link and Figs. 3 and 4 show the comparisons with three examples of networks, when a c.u. can handle 1 and 10 connections respectively. The dashed lines show the 95% confidence interval and the circles show the evaluated *K*s. When using a c.u. of ten (Fig. 4) the optimal *K* is smaller due to the rounding-up of needed capacity into whole capacity units. The optimal *K* depends much more on the actual networks when the capacity is handled in bundles of ten connections, and the result is difficult to estimate. It is seen that the model is sufficient for determining the parameter *K*, particularly when the c.u. = 1 and the blocking

probability is not far from the values in the example networks.

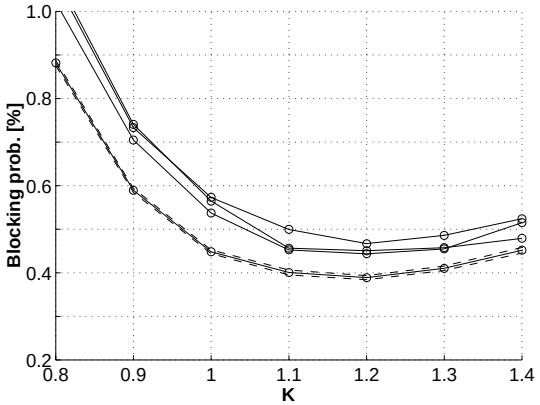

Fig. 3 Comparison with the model, c.u. = 1.

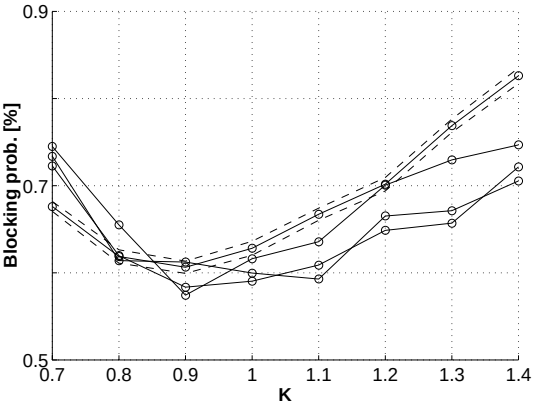

Fig. 4 Comparison with the model, c.u.=10.

When using dynamic alternative routing (DAR) and trunk reservation, (described in Sect. 4), it is important to estimate the overflow traffic in the whole network to be able to get *K*.

For the estimation we again use the one-link model. This time we additionally inject an overflow traffic (B-traffic) to be handled if enough capacity exists for the direct traffic (A-traffic) as determined by the trunk reservation (ten connections). One strategy is to use a sort of Erlang fix point method, i.e the additional B-traffic is iteratively adjusted so it finally equals the resulting rejected traffic. This gives us a value for *K* between 1.1 - 1.3, as seen in Fig. 6. Instead of plotting the profit we show a weighted blocking probability which is explained below.

However, this model does not include the most important aspect of traffic imbalance. When busy centres are used, the links which are over-loaded will cause B-traffic to possibly be alternative routed. A better estimation of the *K* value can be made if over- and under-loaded links are separated. This can be done by studying the measured traffics of a network.

Especially two aspects are important. When a link has a high traffic load, the number of rejected calls will increase when *K* is increased and when a link has a low traffic load, then the number of rejected calls will decrease when increasing *K*. A trade-off should be made by evaluating the one-link model in two cases, which we denote as Link_O and Link_U. One of them represents over-loaded links, Link_O and the other one represents under-loaded links, Link_U as shown in Fig. 5. We assume that the links are independent and that the B-traffic is Poissonian. The difference compared to the previous method is that we do not use iterations. The estimation of the total B-traffic is the amount of rejected traffic from Link_O, and this is injected to Link_U (Fig. 5).

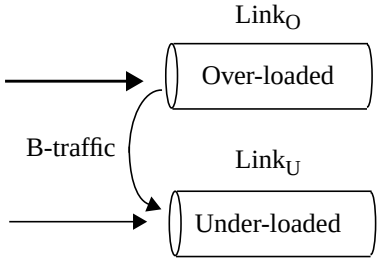

Fig. 5 Two cases of the model.

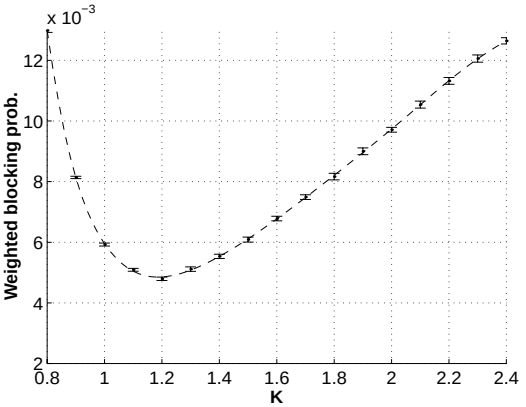

Fig. 6 Traffics are balanced.

When classifying the over- and under-loaded links, we adjust the reference level for the comparison when we evaluate the traffics. We do this to get about the same number of links of both types. The average traffic for the links which has a traffic above the reference level is used for Link_O, and the average traffic for the links which has a traffic below the reference level is used for Link_U. (In our test networks, the number of over- and under-loaded links are almost equal after renormalizing the traffics.)

The profit is a measure of the total number of handled calls in a network. When an overflow call is handled, the profit is less than if it had been allocated on the FVPC since two VPCs are used instead. To account for this we have divided the profit for handling an overflow connection by two. Similarly the cost of rejecting an overflow call is divided by two. To get the weighted blocking probability, the cost of lost calls per VPC is divided by the average traffic.

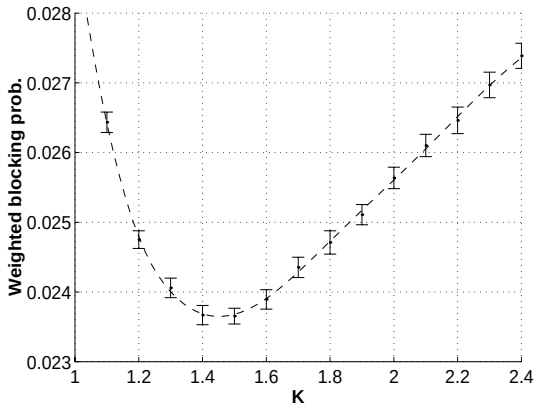

Fig. 7 Moderate traffic imbalance.

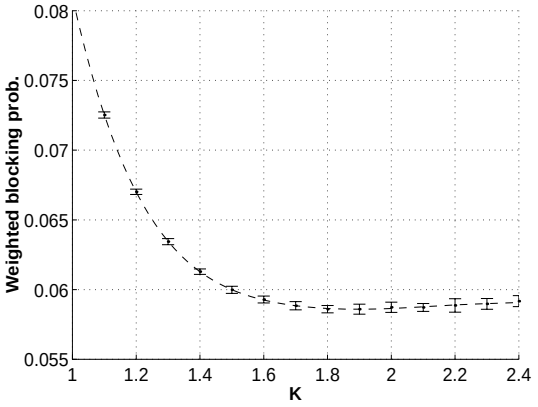

Fig. 8 High traffic imbalance.

Figures 7 and 8 shows the decrease in profit due to lost calls, in a case of moderate and high traffic imbalance respectively (see appendix A).

4. The Routing Strategy

In the evaluation, we have used dynamic alternative routing (DAR) [10] 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 (TR) of ten connections for direct traffic on all VPCs. In Figs. 9 and 10 the blocking probability for three test networks is evaluated with two OVPCs (dashed lines) and when only the FVPC is used (solid lines).

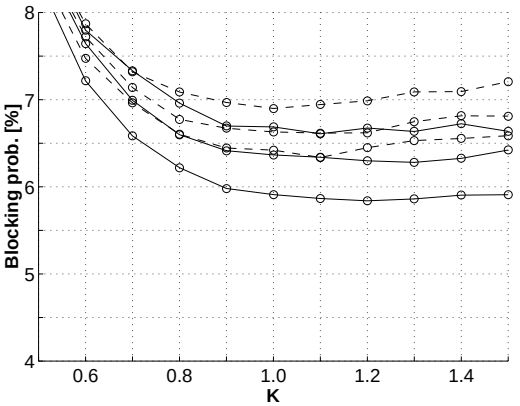

Fig. 9 Result with and without OVPCs, no DAR.

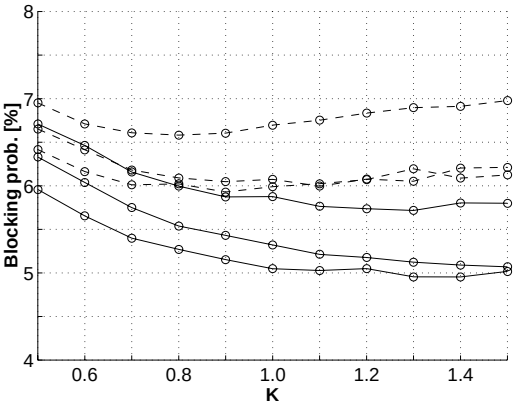

Fig. 10 Result with and without OVPCs, DAR.

For the results in Fig. 9 no alternative routing is used. The test networks are selected to represent all of the test networks and to show the range of different performance (minimum, middle, and maximum). When OVPCs are used, more network capacity is allocated, since the OVPCs have on the average 50% more links to traverse. This results in a higher blocking probability than when not using the OVPCs at all. This indicates that the local VPC management approach handles quick random traffic fluctuations and suggests that no OVPCs should be used.

5. Results

5.1 The Evaluation

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 (FVPCs) to all other nodes. A VCC between two nodes is normally routed over the corresponding FVPC.

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.

Our test networks have the capacity to handle the mean traffics with 1% call blocking probability. To simulate 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. We have used a capacity unit (c.u.) which can handle a certain number of connections. (Having a c.u. = 1, gives the finest distribution of the capacity.) The T_u is set to 0.1 according to [9].

5.2 The Profitability

Our aim is now to compare the results from the models with the test networks for various overhead costs. By assigning costs for rejected calls and overhead such as control messages, the total profitability of the approach can be evaluated. The 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 included 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 $\sim 10^{-4}$. 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.

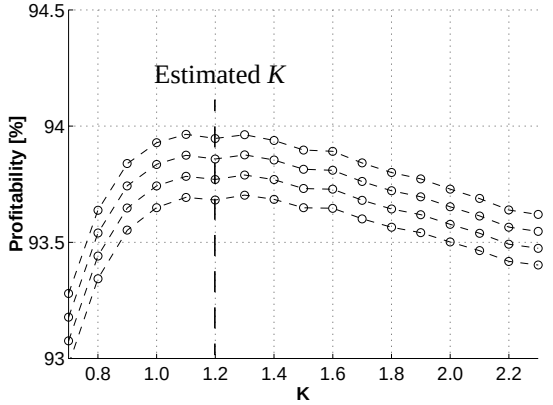

Fig. 11 High traffic imbalance, no DAR.

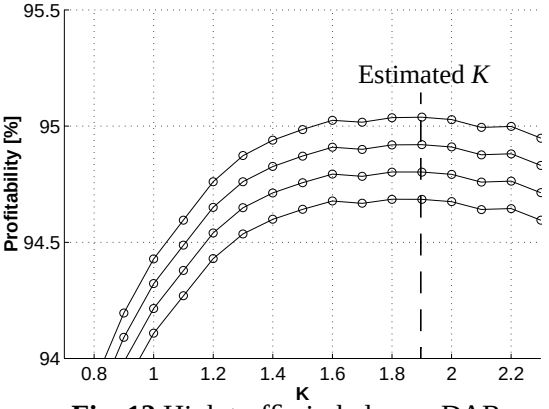

Fig. 12 High traffic imbalance, DAR.

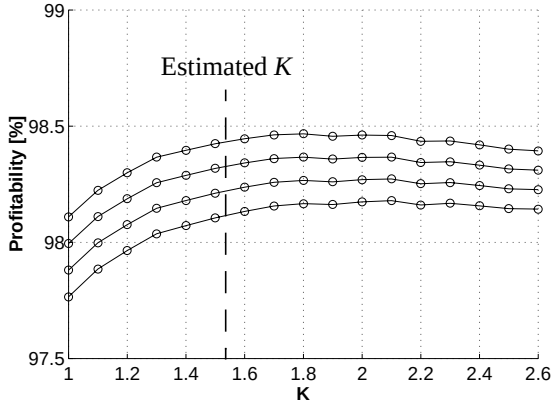

Fig. 13 Moderate traffic imbalance, DAR.

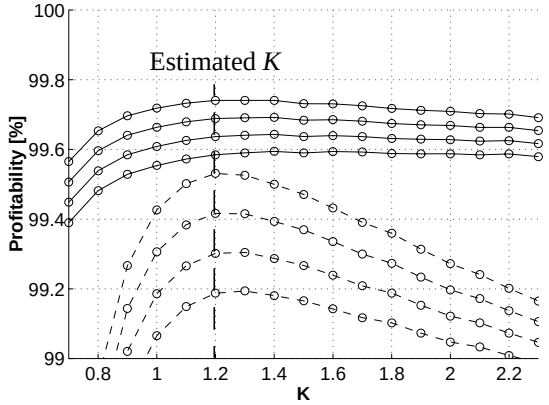

Fig. 14 Traffic balance, with and without DAR.

5.3 Comparisons

Figure 11 shows profitability as a function of K for different message costs without DAR, and Fig. 12 with DAR. Both figures show the result when having a high traffic imbalance. The upper line is the result when having a message cost of 10^{-4}, the lower line when having a message cost of $0.03 + 10^{-4}$, and the step between the message costs is 0.01. The circles show the evaluated Ks. The dashed lines show the results when DAR is not used, and the solid lines when DAR is used.

It is immediately seen that the higher the cost, the lower the profitability. Moreover, it can be seen that the optimum is reached for a higher value of K when DAR is used as indicated in Fig. 12. This depends on the trunk reservation (TR) for the direct traffic (A-traffic). To be able to support alternative routed calls, more capacity has to be reserved, hence a larger K is more profitable to the network. In other words, when using DAR the selection of K depends on the trunk reservations. It is seen that our estimated Ks match the actual values despite the simplifications used.

Figure 14 shows the profitability in a case when traffic intensities are in balance with the link capacities, i.e. no busy center (with same cost settings as in Figs. 11 - 13). The model for overflow traffic using the Erlang fix point method, is designed for a balanced traffic as in Fig. 14. The estimated K is obtained from Fig. 6. The estimated Ks for Figs. 11 and 12 are obtained from Figs. 7 and 8 respectively.

By comparing Fig. 14 with Figs. 11 - 13, we can conclude that it is important to have a larger K when DAR is used, when there is a traffic imbalance, i.e. a mismatch between the link dimensions and the traffics. This is presumably the most probable case in a real network.

The difference in profitability is higher for large traffic imbalance which means that the K

should be selected to cope with high imbalance situations since the negative effect in a balanced situation is minimal. In our example it should be selected from Fig. 12 (if DAR is used, c.u. = 1), which only results in a decrease in profitability by about 0.02% when traffic intensities are in balance.

5.4 Further Evaluation

The estimated values of *K* are in good agreement with the actual values. To validate the method of estimation we have additionally applied it to the networks in two other situations. Firstly, the number of connections reserved for the A-traffic is not optimal. By using smaller values of the trunk reservation, we should actually decrease the optimal *K*. Secondly, we do not normalize the total amount of traffic in the network after the selection of a busy center. This means that the number of over-loaded links becomes less than the number of under-loaded. The method of identifying the two kinds of links and getting the average traffic loads for them, uses the average traffic for the classification.

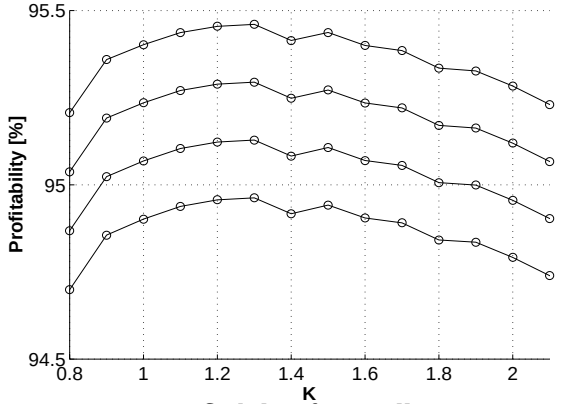

Fig. 15 Profitability for smaller TR.

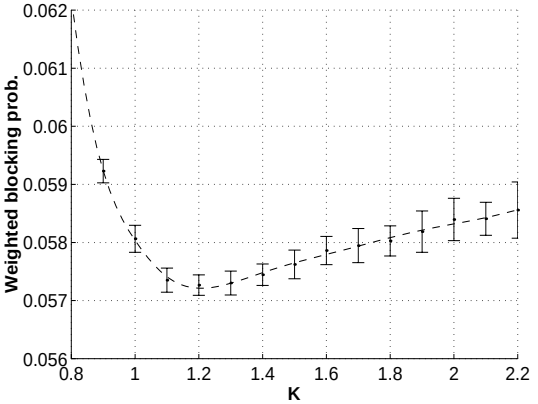

Fig. 16 Cost of lost calls for smaller TR.

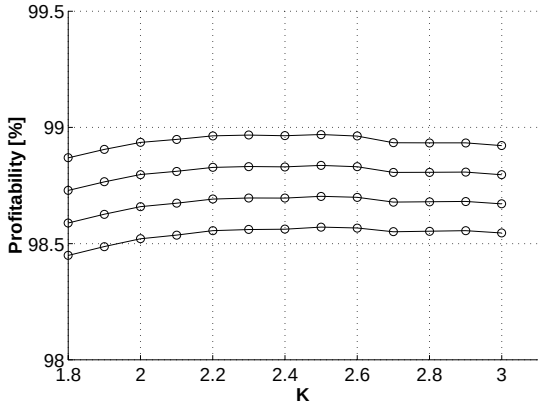

Fig. 17 Profitability for unnormalized traffics.

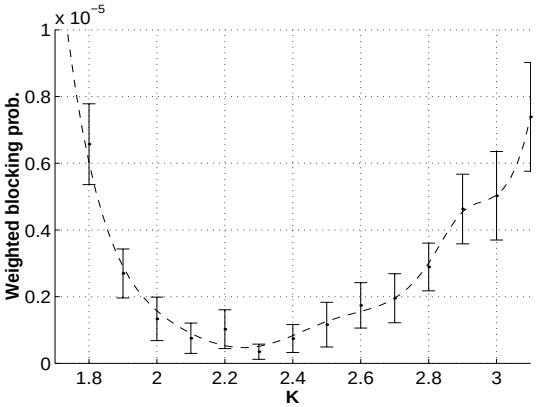

Fig. 18 Cost of lost calls for unnormalized traffics.

In Figs. 15 and 16 we see that the optimal *K* is decreased towards the value when no DAR is used (Fig. 3) and that the actual and estimated *K*s are almost the same. One should also note that the profitability has increased due to a smaller trunk reservation.

In the case of unnormalized traffics we adjust the reference level to 0.83 to get about the same number of over- and under-loaded links. The average load for Link_O gets 1.0, and for Link_U it gets 0.67. The average traffic per VPC is about 110 Erlangs but we use the same capacity as before. Even in this situation we get a good estimate of the optimal value of *K*.

Some general conclusions about the number of messages can be made. Firstly, the number of INC and DEC messages decrease when increasing *K*, since less capacity will be available

and the number of capacity reallocations will decrease. Secondly, when having traffic imbalance, the number of messages and particularly alternative routed calls will increase when increasing *K*. The total message cost is to be related to the profitability of handling more connections. To optimize the profitability a proper setting of K and T_u has to be done. However, in our evaluation the blocking probability has a larger impact on the profitability, and an estimation of the number of messages seems less important when trying to select the proper *K* value.

5.5 Impact of Different Factors for the Selection of *K*

Table 1 Factors used when selecting *K*.

Decrease *K* if:	**Increase *K* if:**
Few VPCs per link	Many VPCs per link
High traffic load	Low traffic load
OVPCs are used	Only FVPCs are used
Short T_u	Long T_u
DAR not used	DAR is used
Capacity in large units	Capacity in small units

Table 1 gives different factors which influences the setting of *K*. The more VPCs there are in a link, the better the multiplexing effect will be which enables a larger value of *K*. When the traffic load is low, there will be a lot of free capacity that can be allocated by using a larger *K*. The profitability will decrease if OVPCs are used, because they in general consume more network resources. The updating interval, T_u is selected according to the message cost. If the message cost is high, the T_u is chosen to be longer which in turn decreases the number of control messages, and to compensate for this a larger *K* has to be chosen. Increasing the allocated capacity will enable the handling of alternative routed calls from other VPCs, when DAR is used.

6. 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 measured parameters is limited to one, *viz*. the current number of active connections. In this evaluation, the total call blocking probability is reduced when only the FVPCs are used. This indicates that this approach handles quick random traffic fluctuations.

Using the simple model with an average background traffic to get the parameter *K* works well although the actual distributions of FVPCs covers a wide range. Moreover, the optimal *K* values for the allocation function are not critical but optimal over a wider range as shown in Figs. 11 - 14. 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.

7. 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 *K*s 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.

It should be noted that the averages has been calculated from all the ten test networks, and not one by one. Having a special *K* for each network will probably give even better results.

In our study we have used a reservation of ten connections for direct traffic when DAR is used. However, the allocation formula tells us how much capacity that is wanted for the moment on each VPC, and correspondingly how much capacity that ought to be reserved. Using this amount of reservation would reject almost all alternative routed calls. Therefore, a slight modificaton of (2) could be used.

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] 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.

[4] 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.

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

[6] 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.

[7] J. Virtamo, S. Aalto, “Blocking Probabilities in a Transient System,” COST TD257(97)14, The Netherlands, Jan. 1997.

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

[9] 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

[10] 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.

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 55%. 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 6500 Erlangs.
We also consider a case of more moderate imbalances with limits 10% and 30% (instead of 20% / 60%). The resulting greatest increase is then 43% and greatest decrease 27%.

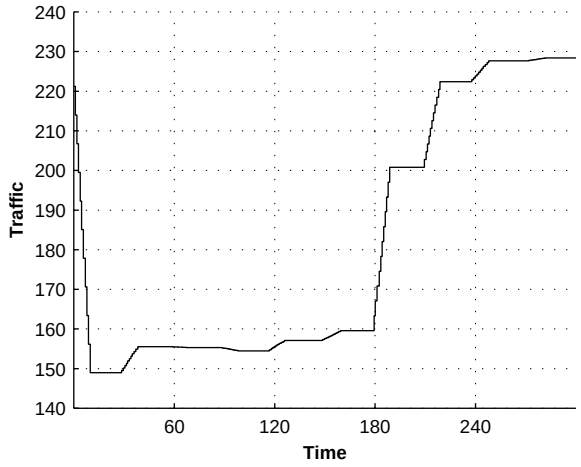

Fig. A1 Sample traffic.

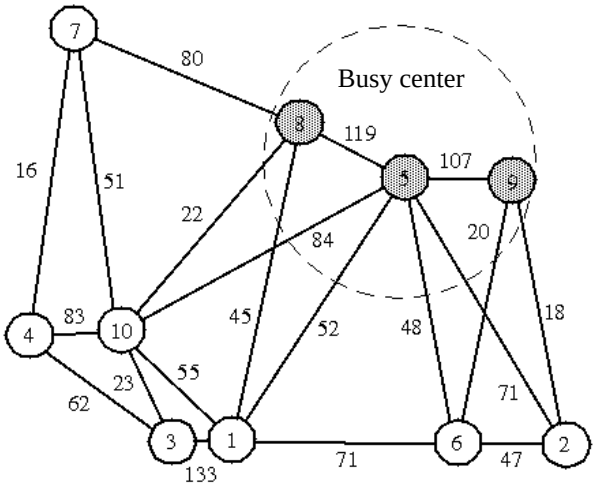

Fig. A2 Sample network layout, 10 nodes.