: equal priorities and
ı as bit rates, delay
of the physical level
elop measures for the
ıg process.

ımunications - Differ-
:EE Communications

ı. IEEE Transactions
.

grated services local
27-35.
ınced Techniques for
td., Stevenage (UK),

tching Bus. Proceed-
E Computer Society,

self-similar nature of
/ol. 2, 1994, pp. 1-15.
f Poisson Modeling.
95, pp. 226-244.
ıg Architecture. Pro-
, June 1995, Odense

Packet Switching Ar-
elsinki University of

oceedings of ICT'98,

ıented Switching Ar-
ence, Prague (Czech
11.
Queues: Definitions,
/ACM Transactions

#12

ITC 16 / P. Key and D. Smith (Editors)
© 1999 Elsevier Science B.V. All rights reserved.

An Adaptive Local Method for VPC Capacity Management

Sven Olof Larsson[a] and Åke Arvidsson[b]

[a]Dept. of Telecommunications and Signal Processing, University of
Karlskrona/Ronneby,
S-371 79 Karlskrona, Sweden

[b]Ericsson Utvecklings AB, Soft Center (Etapp III),
S-372 25 Ronneby, Sweden

By reserving transmission capacity on a series of links from one node to another within an ATM or SDH/SONET network, 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 telecommunication traffic exhibit variations in the number of calls per time unit due to office hours, 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 present 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. A method for estimating individual VPC congestion states is described together with an adaptive parameter setting of 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 a new call is simplified because the routing and reservation of capacity has already been done. A VPC network constitutes a higher layer that 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 telecommunication traffic. 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 variations to reallocate the resources. 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 method for VPC capacity management. The problem is how to distribute the capacity in a way that maximizes the revenue. The models used to estimate the parameter for the allocation function is described in Sect. 3. In Sect. 4 we describe a method for estimating the VPC congestion state and then suggest an adaptive parameter setting for the allocation formula. In Sect. 5 we describe the call allocation. The results and evaluations are given in Sect. 6. Finally, we conclude the paper in Sect. 7 and discuss further work in Sect. 8.

2. THE LOCAL METHOD

A lot of papers describing distributed approaches to VPC capacity management fall into our definition of a local method. By a local method 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. 1). The decision is based on the actual number of occupied connections on that VPC. Figure 2 shows a sample reallocation sequence with five VPCs on a link. The VPC capacity management is done with help of signaling. The following two control messages are used: "Increment request" (INC) + Answer, and "Decrement" (DEC).

2.1. Control Messages

The method supports an arbitrary number of VPCs between each node pair. However, only the VPC with the smallest number of hops is used since this increases the network utilization [5]. When more capacity is needed, an INC message is sent on the VPC to find out if the requested capacity is available for the whole path. If less than the requested

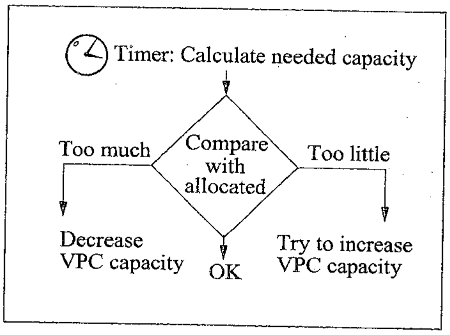


Figure 1. The principle of the local method.

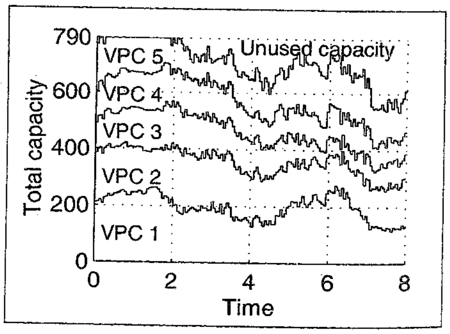


Figure 2. A reallocation sequence.

capacity is available on a particular link, it will get all unreserved capacity. The minimum of these capacities is stored in the INC on successive links. When it reaches the end node, indicating either the requested or available capacity, an answer message is sent back to the originating node and capacity reservations are done in the intermediate nodes for the particular link in question.

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.

2.2. The Allocation Function

The developed method for local VPC capacity management is inspired by the one developed by Mocci et al. [6]. This method allocates just enough VPC capacity to meet a predefined limit of call blocking (target blocking) [7]. 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. Another approach is to apply simplifications along the lines of Virtamo and Aalto [8]. Anyway, it is suggested to use precalculated tables to achieve real time applicability. In [9] an allocation function is developed which does not depend on the actual offered traffic. The idea behind this function is that for specified traffic intensity the mean occupation state is equal to the intensity (if the blocking probability is low). By introducing some constants, it is possible to rewrite the formula in a way that, for a given target blocking probability and T_u, makes N a function of n only [9]:

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

where N is the required capacity and n is the number of currently active calls. K can be seen as a safety factor, which adds extra capacity in units of the standard deviation of the occupation state or just as a variable confidence interval dependent on T_u.

In our study we let the K be independent of n. This is a trade-off between equalized blocking probability among the VPCs and the total amount of handled traffic [10]. 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 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 numbers on links with different capacities poses a very complex problem. In [10] 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 and a larger K. The optimal K for a specific VPC 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 method reallocates the capacity between VPCs 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, and for a long T_u the blocking reaches the same level as for deterministic multiplexing)

3. A METHOD FOR ESTIMATING THE PARAMETER K

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 of a couple of VPCs. In our test networks (see appendix) the mean number of VPCs 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. Traditionally networks are dimensioned by allowing a 1% call blocking probability in the busy hour. 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 traffic each carrying 130 Erlangs, and the link capacity is set to 592. We have evaluated the one-link model in [5] and found that the model is sufficient for determining the optimal K. However, the optimal K changes along with the traffic load and VPC congestion state (Figs. 3 and 4). Using an adaptive K is an attempt to take care of this dependence.

4. ESTIMATING THE VPC CONGESTION STATE

To estimate the congestion state we use a special data field in the INC message. Each intermediate node (including the first one) divides the amount of free capacity with the

'he optimal choice of
traffic (which partly
r should be allocated
utilization and allo-
al T_u determines the
:ly, frequent possibil-
rgins without risking
ɔtal performance, an
d.
fferent combinations
: problem. In [10] we
a link and suggested
. When many VPCs
number of messages
imal K for a specific
; links.
ι deterministic multi-
over a long time, the
:mula for each VPC.
g probability can be
nce the local method
: blocking probability
'or the full multiplex-
same level as for full
the same level as for

K

er the situation on an
traffic consisting of a
ɔer of VPCs on a link
he mean capacity per
' traffic. Traditionally
ity in the busy hour.
/ith this in mind, our
130 Erlangs, and the
in [5] and found that
ıe optimal K changes
1). Using an adaptive

.e INC message. Each
:ree capacity with the

number of VPCs on the actual link, and forwards this value which we have labelled as L. The answer message contains the minimum of L for the actual path. In Figs. 3 and 4, L is related to K for a set of different traffic load situations, for 5 and 15 VPCs respectively. The same one-link model as in [5] has been used. The solid curves shows six different traffic loads (giving 20, 10, 5, 2, 1, and 0.5% call blocking probability, from down-left to upper-right). Each curve has a minimum of lost calls for a specific K and L. The points marked with asterisks show the call blocking which is within 0.05% from the minimum. The dashed line between the minima minimizes the quadratic error of K. The optimal K can be selected in a wider range when the call blocking is low. To simplify the problem of finding a proper K, we use a linear relation between K and L. The straight line in Fig. 4 have been chosen in an attempt to characterize K's dependence on the traffic load. The line is defined by $K = 0.4 + 0.2 \cdot L$. The K is updated according to this formula after each new observation of L. The adaption can be explained by the following example. Let us say that the overall traffic has decreased after a stable period on a single link, having 5 VPCs (see Fig. 3). Accordingly the call blocking decreases, and in this example from, say 5% to 1%. The measured L is thereby increased from ~ 2 to ~ 10, showing that there is more capacity available. The K is increased more than one could expect since the mean K ought to be 1.2, (where the K-L line cuts the 1% blocking line). (For security reasons the K is restricted to be below 1.7, which gives a limit on how much that can be allocated to a single VPC.) However, the effect of having some VPCs having too big a value of K and some having a too small one results in an amount of unreserved capacity which lies somewhere between these extremes of observed Ls. The distribution of L corresponds to the occupancy states but will consequently also on the average correspond to the actual load situation.

In Figs. 5 and 6 we can see that the selection of K is important especially when there are a lot of VPCs and the blocking is not too high and not too low. (This is the reason why the K-L line has been chosen according to Fig. 4.) It should be noted that high call blocking probability often indicates a high traffic. Since a change in blocking probability has more impact on the number of lost calls for a high traffic than for a low one, the

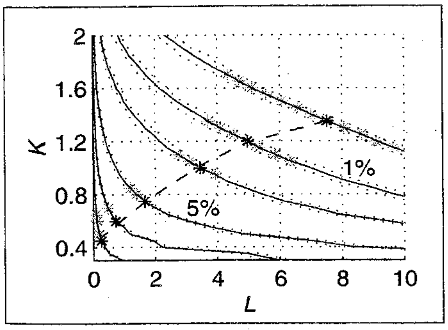


Figure 3. Performance for 5 VPCs.

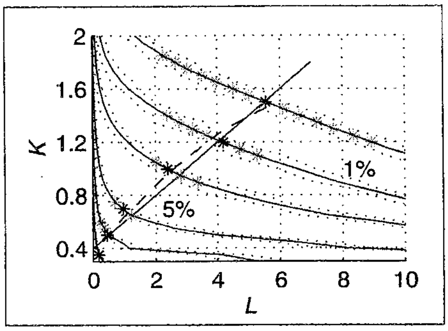


Figure 4. Performance for 15 VPCs.

potential increase in handled calls can not be assessed from the figures directly. Erlang's B-formula was used to calculate the traffics in Figs. 3 - 6, which give the different call blocking probabilities. It is these traffics that should be used when estimating the number of lost calls.

In Figs. 7 and 8 we show the variations of K for a sample VPC. Both figures show the adaptation of K for three full cycles of the traffic demand matrices (see appendix). No alternative routing is used. It is seen that the K is more often reaching the limiting values when the traffic variations are larger. The regulation of K becomes smoother if the change is decreased a little (e.g. using the mean value of the new K and the previous one). However, by doing this the profitability decrease which indicates that even quick traffic fluctuations can be utilized.

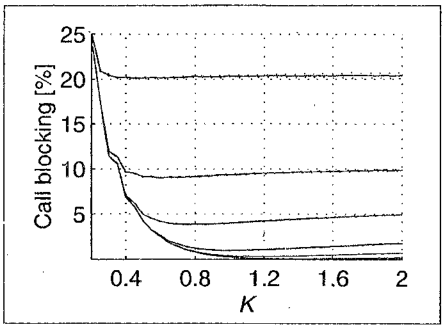


Figure 5. Call blocking, 5 VPCs.

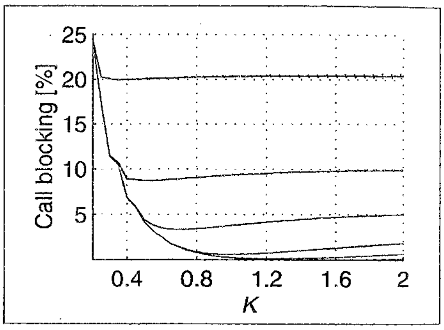


Figure 6. Call blocking, 15 VPCs.

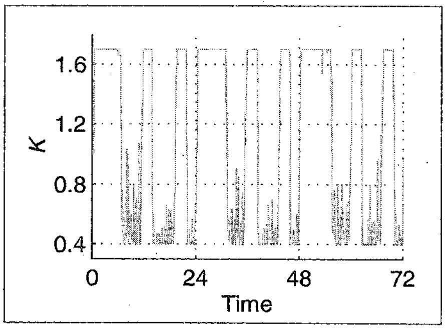


Figure 7. K-trace, high traffic variations.

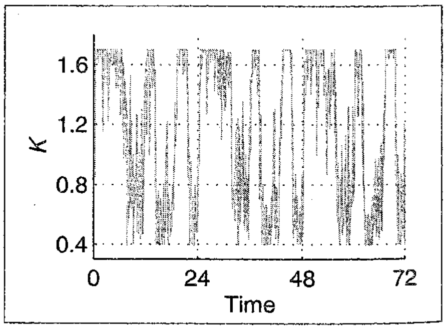
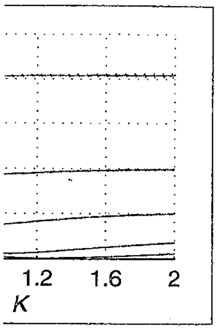


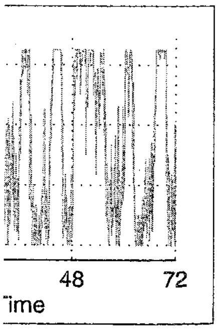
Figure 8. K-trace, moderate traffic variations.

es directly. Erlang's
ive the different call
timating the number

ɔth figures show the
(see appendix). No
eaching the limiting
ɔecomes smoother if
K and the previous
ates that even quick



ıg, 15 VPCs.



ɔderate traffic varia-

5. CALL ALLOCATION

A VCC between two nodes is normally routed over the corresponding direct VPC. When a call arrives it is allocated on this VPC if there is room for it. The one-link VPCs are able to use all unreserved capacity on the link they use. In the evaluation, we have used dynamic alternative routing (DAR) [11] 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, a new transit node is chosen (at random) for the next time that a call needs to be rerouted. Two control messages are used to determine the status of the transit nodes (question + answer). As a complement to the alternative routing, we finally try to allocate the call link by link on the minimum hop path. If this does not succeed the call is rejected.

To ensure stability, we have applied a trunk reservation parameter (*TR*) for direct traffic on all VPCs. This makes the network performance stable and it increases the utilization of network resources at high traffic load situations. The setting of *TR* has already been studied for many types of networks, but we have estimated the optimal setting for our local method. The optimal *TR* gets a value between 2 and 4. One should probably set it to a fixed value and we have set it to 3. This *TR* has also been used for the link by link call allocation.

6. 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 straightforward. 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 aim is now to compare the adaptive allocation formula to the case when a fixed global *K* is used. The profit of handling one call is set to one unit. By assigning costs for rejected calls and overhead such as control messages, the total profitability of the method can be evaluated (2). The profitability is defined as a normalized measure where 100% profitability means that all calls are handled without any overhead costs. (100% is infeasible for high traffic load situations.)

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

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 additional overhead. We have set the message cost to 0.01. The profitability

Table 1
Comparison of different approaches. High traffic imbalance and 95% confidence intervals.

	Number of signals/t.u.	Mean call blocking [%]	Mean max call blocking [%]	Profitability [%]
Fixed	0	8.72 ± 0.03	20.53 ± 0.41	91.28 ± 0.03
Call by call	12890 ± 30	5.01 ± 0.05	16.39 ± 0.22	92.98 ± 0.05
Global K	1417 ± 6	5.74 ± 0.02	17.03 ± 0.85	94.04 ± 0.02
Adaptive K	1710 ± 6	5.39 ± 0.02	18.72 ± 0.54	94.35 ± 0.02
Fixed + DAR	598 ± 1	7.45 ± 0.03	16.75 ± 0.31	92.45 ± 0.03
Global K + DAR	1061 ± 2	4.95 ± 0.04	14.14 ± 0.37	94.88 ± 0.04
Adaptive K + DAR	1154 ± 2	4.50 ± 0.03	14.52 ± 0.36	95.32 ± 0.03

is used to enable a reasonable evaluation of the overall performance by combining gains and costs.

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 to all other nodes. Our test networks have the capacity to handle the mean traffics with 1% call blocking probability when the capacity allocation is fixed. To simulate traffic variations, ten different traffic patterns were generated for each network by randomly selecting a busy center (see appendix). Nodes inside the center increase their traffics above the average and those outside the center decrease theirs. The T_u has been set to 0.1, according to the results in [10].

The results are given in Table 1 when having high traffic imbalance. (Having moderate traffic imbalance gives the same but not so distinct relations.) Compared to having a fixed global and optimal K of 1.2 (as seen in Fig. 3), the adaptive K will increase the profitability only slightly. This can be explained by the fact that most of the links are either under utilized or heavily congested (Figs. 7 and 8), which means that the potential decrease of the call blocking is low as indicated by the discussion about Figs. 5 and 6. The decrease in the number of signals for the local method when using DAR can be explained by the decrease in the number of capacity reallocations. These decrease $\approx$ 6% for the adaptive K and $\approx$ 15% for the global K, i.e. the fluctuations in the number of active connections decrease by accepting alternatively routed connections.

When using the mean value of L along the paths (instead of the minimum), the profitability decrease to the same level as for the global K. This means that it is important to set the K according to the most congested link. The use of an additional call allocation on one-hop VPCs increases the profitability in the same order as the use of an adaptive K does. The increase of profitability when using DAR is about the same as the use of both adaptive K and call allocation link by link.

We have not simulated the effect of having a call by call allocation together with DAR. However, it is possible to estimate the greatest possible increase in profitability. If the call by call approach gets a similar increase in profitability as the fixed approach, the estimated profitability will increase to about 94.0% (increased number of signals included).

confidence intervals.

Profitability [%]
91.28 ± 0.03
92.98 ± 0.05
94.04 ± 0.02
94.35 ± 0.02
92.45 ± 0.03
94.88 ± 0.04
95.32 ± 0.03

e by combining gains

which can be seen as
pabilities and a fully
Cs to all other nodes.
vith 1% call blocking
variations, ten differ-
lecting a busy center
the average and those
ding to the results in

e. (Having moderate
ompared to having a
e K will increase the
most of the links are
ns that the potential
ut Figs. 5 and 6. The
AR can be explained
crease ≈ 6% for the
the number of active

imum), the profitabil-
it is important to set
nal call allocation on
use of an adaptive K
me as the use of both

together with DAR.
rofitability. If the call
proach, the estimated
als included).

7. 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. Using a simplified model with average background traffic for the estimation of K - L - *Traffic load* relation works well although the actual distributions of the number of VPCs per link cover a wide range (1-13). Taking this into account and adjusting K according to the estimated experienced VPC congestion state (given by L), we are able to fine tune the setting of it. Although the profitability is not increased very much, this suggests that our method is robust with respect to traffic load situations and the numerous combinations of VPCs. The method takes advantage of the benefits from both VP and VC routing, i.e. enabling fast CAC and using multiplexing of VPCs.

8. FURTHER WORK

It remains to find a relation between the actual traffic load and the optimal T_u, to simplify and automate the setting. The impact of different types of traffics on the performance should be evaluated along with different kinds of network structures and sizes. Functional similarities with RSVP can be studied to see if our method could be used for the Internet. A comparison with different approaches (i.e. centralized and distributed) could give an indication of which one that is to be preferred.

9. APPENDIX: 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 $k, (k = 1, ..., M)$ (see Fig. 9). The time index M 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 M different load situations by the use of "busy center" (Fig. 10). 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%. 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%.

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. S. O. Larsson, Å. Arvidsson, “A Local Approach for VPC Capacity Management,” Proc. 14th Nordic Teletraffic Seminar, pp.41–52, Copenhagen 1998.
6. U. Mocci, P. Pannunzi, and C. Scoglio, “Adaptive capacity management of virtual path networks,” Proc. IEEE Globecom’96, paper no. 19.2, 1996.
7. 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.
8. J. Virtamo, S. Aalto, “Blocking Probabilities in a Transient System,” COST TD257(97)14, The Netherlands, Jan. 1997.
9. J. Virtamo, S. Aalto, “Remarks on the Effectiveness of Dynamic VP Bandwidth Management,” COST TD257(97)15, The Netherlands, Jan. 1997.
10. 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.
11. 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.

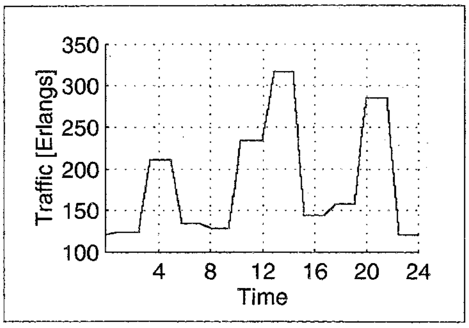


Figure 9. A sample traffic.

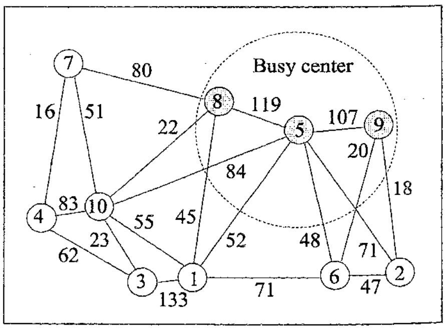


Figure 10. A sample network layout.